MASARYKOVA UNIVERZITA
Fakulta informatiky

Nekonečné řady s programem Maple

Brno 2001

Petr Kuběna

Prohlašuji, že jsem tuto práci vypracoval samostatně a podle pokynů vedoucího diplomové práce a použil jsem pouze materiály, které jsou uvedeny v seznamu literatury.

Děkuji vedoucímu diplomové práce RNDr. Romanu Plchovi Ph.D. za odborné vedení, cenné rady a připomínky, které mi poskytl při zpracování tohoto tématu.

Obsah

Úvod

V současné době stále více pronikají do našich životů informační technologie. Zejména výpočetní technika expanduje prakticky do všech odvětví lidské činnosti. Znalost práce s počítačem se dnes stává nedílnou součástí všeobecného vzdělání moderního člověka.

Ve své diplomové práci se zabývám možnostmi využití systému pro symbolické výpočty Maple při výuce tématu nekonečné číselné řady. Výhody využití tohoto systému ve výuce matematiky spočívají především v urychlení rutinních výpočtů (studenti ušetří čas, který pak mohou věnovat například rozsáhlejšímu procvičení probírané látky) a v bohatých možnostech vizualizace, která může pomoci lepšímu pochopení probíraného pojmu.

Práce je určena jak učitelům, tak i studentům. Učitelé mohou využít ilustrace a příklady k oživení výuky. Práce může také posloužit jako inspirace jak Maple využít při výuce dalších partií matematiky. Studenti pak mohou práci využít jako doplňková skripta a zároveň sbírku řešených příkladů, týkajících se nekonečných číselných řad.

Často používané postupy jsem automatizoval pomocí Maplovského programovacího jazyka ve formě procedur. Maplovský programovací jazyk je jednoduchý a snadno naučitelný. K jeho ovádnutí stačí základní znalosti algoritmizace. Čtenář si doprogramováním dalších procedur může ověřit nakolik pochopil probírané téma a zároveň si procvičit své programátorské dovednosti.
Diplomová práce je rozdělena do osmi kapitol. V první kapitole popisuji jednotlivé příkazy systému Maple, které budeme používat v dalším textu při řešení příkladů. Předpokládám základní znalost práce se systémem Maple. V práci využívám také vlastní procedury, které se nacházejí na přiložené disketě v souboru `moproc.m`. Kdykoli se v textu zmíním, že nějaká procedura je doprogramovaná, je třeba před jejím použitím provést příkaz `read('moproc.m');`, pokud tak již nebylo učiněno od počátku sezení nebo

od posledního použití příkazu `restart;`.

Další čtyři kapitoly jsou rozděleny vždy na dvě části. V první části je stručně shrnuta teorie. Věty jsou uváděny bez důkazů, které lze nalézt např. v [1]. Kromě teorie jsou v první části uvedeny též ilustrační příklady, které ukazují postupy, jimiž lze řešit příklady týkající se příslušné kapitoly. Druhou část tvoří řešené příklady, které mohou posloužit k procvičení látky probírané v první části kapitoly. Vyjímku tvoří čtvrtá kapitola, která má teoretické části dvě.

Šestá kapitola je poněkud stručnější a obsahuje pouze teoretickou část, neboť téma součin řad se v učitelském studiu zcela vynechává.

Sedmá kapitola obsahuje zdrojové texty doprogramovaných procedur, jež jsou v práci využívány.

Osmá kapitola se věnuje internetovým zdrojům, které se týkají zpracovávaného tématu. V žádném případě se nejedná o úplný seznam, popisuji pouze malou část z nepřeberného množství zdrojů, které lze též využít pro výuku nekonečných číselných řad s využitím systému Maple.

Kapitola 1

Příkazy systému Maple

V této kapitole uvádím seznam příkazů Maplu (včetně jejich parametrů), s nimiž budeme v dalším textu pracovat.

Příkaz abs
Význam: absolutní hodnota výrazu.
Syntaxe:

```
abs(expr)
```

Parametry:
`expr` - výraz, jehož absolutní hodnotu počítáme.

Příkaz limit
Význam: výpočet limity.
Syntaxe:

```
limit(expr,x=a,dir)
```

Parametry:
`expr` - výraz, jehož limitu příkaz počítá,
`x` - jméno proměnné ve výrazu `expr`,
`a` - bod, v němž se limita počítá, (tedy reálné číslo, popř. slovo `infinity`, resp. `-infinity` pro ∞, resp. pro $-\infty$) a
`dir` - nepovinný parametr, který může nabývat hodnot `left`, resp. `right` pro výpočet levostranné, resp. pravostranné limity. Je-li parametr vynechán, počítá se limita oboustranná.

Příkaz simplify
Význam: zjednodušení výrazu.
Syntaxe:

```
simplify(expr)
```

Parametr:
`expr` - výraz, který chceme zjednodušit.

Příkaz ceil
Význam: zaokrouhlení čísla nahoru.
Syntaxe:

```
ceil(x)
```

Parametr:
`x` - číselný výraz, který chceme zaokrouhlit.

Příkaz convert
Význam: převod na daný tvar. My jej budeme používat k rozkladu výrazu na parciální zlomky.
Syntaxe:

```
convert(expr,tvar,x)
```

Parametry:
`expr` - výraz, který převádíme,
`tvar` - tvar, do nějž chceme daný výraz převést. V našem případě budeme používat parametr `‘parfrac‘`, který říká, že chceme výraz rozložit na parciální zlomky. Je lepší uvést tento parametr v pravých apostrofech pro případ, že by v paměti byla načtena proměnná stejného jména,
`x` - jméno proměnné ve výrazu `expr`.

Příkaz subs
Význam: substituce ve výrazu.
Syntaxe:

```
subs(x=a,expr)
```

Parametry:
`x` - jméno proměnné ve výrazu `expr`,
`a` - proměnná, resp. konstanta, kterou substituujeme za proměnnou `x`,
`expr` - výraz, v němž substituci provádíme.

Příkaz evalf
Význam: vyčíslení výrazu v pohyblivé desetinné čárce.
Syntaxe:

```
evalf(expr,n)
```

Parametry:
`expr` - výraz, jehož hodnotu chceme vyčíslit,
`n` - nepovinný parametr, udávající počet platných cifer.

Příkaz sum
Význam: výpočet součtu řady.
Syntaxe:

```
sum(expr,n=a..b)
```

Parametry:
`expr` - výraz a_n zadané řady $\sum a_n$,
`n` - jméno proměnné ve výrazu `expr`,
`a` - dolní mez a
`b` - horní mez dané řady.

Příkaz int
Význam: výpočet integrálu.
Syntaxe:

```
int(expr,x=a..b)
```

Parametry:
`expr` - výraz, který integrujeme,
`x` - jméno proměnné ve výrazu `expr`,
`a` - dolní mez a
`b` - horní mez daného integrálu.
Zapíšeme-li příkaz bez mezí, tedy ve tvaru `int(expr,x)`, Maple spočítá neurčitý intergrál výrazu `expr`.

Příkaz diff
Význam: výpočet derivace.
Syntaxe:

```
diff(expr,x)
```

Parametry:
`expr` - výraz, kterou derivujeme,
`x` - jméno proměnné, podle níž derivujeme.

Příkaz solve
Význam: řešení rovnice, resp. nerovnice, příp. soustavy rovnic, resp. nerovnic.
Syntaxe:

```
solve(rovnice,x)
```

Parametry:
`rovnice` - rovnice, resp. nerovnice, kterou řešíme. Řešíme-li soustavu rovnic, resp. nerovnic, pak tento parametr musíme zapsat jako množinu, tj. uzavřít jej do složených závorek,
`x` - jméno neznámé, příp. neznámých. Tento parametr je nepovinný, Maple sám dokáže rozpoznat jména všech neznámých.

Příkaz fsolve
Význam: stejný jako předchozí příkaz `solve`, pro řešení však používá aritmetiku pohyblivé desetinné čárky.
Syntaxe:

```
solve(rovnice,x,options)
```

Parametry:
`rovnice` - rovnice, resp. nerovnice, kterou řešíme. Řešíme-li soustavu rovnic, resp. nerovnic, pak tento parametr musíme zapsat jako množinu, tj. uzavřít jej do složených závorek,
`x` - jméno neznámé, příp. neznámých. Tento parametr je nepovinný, Maple sám dokáže rozpoznat jména všech neznámých,
`options` - nepovinný parametr, nastavující různé doplňující volby, např. na jakém intervalu se má řešení vyšetřovat (podrobněji viz Help v Maplu).

Příkaz plot
Význam: vykreslení grafu funkce.
Syntaxe:

```
plot(expr,x=a..b,y=c..d,options)
```

Parametry:
`expr` - výraz, jehož graf příkaz vykreslí,
`x=a..b` - `x` udává jméno nezávislé proměnné ve výrazu `expr`; `a, b` jsou čísla, udávající rozsah na ose x,
`y=c..d` - nepovinný parametr, `y` je jméno závislé proměnné a `c, d` čísla, udávající rozsah na ose y. Není-li parametr uveden, je závislá proměnná označena jako `y` a její rozsah dopočítán v závislosti na rozsahu pro `x` a výrazu `expr`,
`options` - nepovinný parametr, udávající různé doplňující volby, např. barvu a sílu křivky, způsob zobrazení os atd. (Podrobnější informace lze nalézt v Helpu systému Maple).

Příkaz display
Význam: zobrazení více grafů do jednoho obrázku.
Syntaxe:

```
display(L,insequence=true,options)
```

Parametry:
`L` - seznam, množina, nebo pole grafových struktur,
`insequence=true` - nepovinný parametr, používá se pro animace, kdy je v každém kroku animace vykreslena jedna položka parametru `L`. Implicitně je parametr `insequence` nastaven na hodnotu `false`, která udává, že se všechny položky z `L` vykreslí najednou do jednoho obrázku,
`options` - nepovinný parametr určující další volby obrázku, jako např. orámování obrázku, legendu apod. (Podrobněji viz Help v Maplu).

Kapitola 2

Nekonečné číselné řady

2.1 Základní pojmy

Definice 2.1
Nechť $\{a_n\}_{n=1}^{\infty}$ je posloupnost reálných čísel. Symbol

$$\sum_{n=1}^{\infty} a_n \text{ nebo } a_1 + a_2 + a_3 + \ldots + a_n + \ldots$$

nazýváme *nekonečnou číselnou řadou.* Posloupnost $\{s_n\}_{n=1}^{\infty}$, kde

$$s_1 = a_1, s_2 = a_1 + a_2, \ldots, s_n = a_1 + a_2 + \ldots + a_n, \ldots,$$

nazýváme *posloupnost částečných součtů* této řady.
Existuje-li vlastní limita $\lim_{n\to\infty} s_n = s$, řekneme, že řada $\sum_{n=1}^{\infty} a_n$ *konverguje* a má součet s. Neexistuje-li vlastní limita $\lim_{n\to\infty} s_n = s$, řekneme, že řada $\sum_{n=1}^{\infty} a_n$ *diverguje.*

Poznámka 2.1
Pro určení součtu řady $\sum a_n$ se používá ve většině případů následující standardní postup: výraz a_n rozložíme na součet parciálních zlomků, určíme n-tý částečný součet s_n dané řady a limitním přechodem $\lim_{n\to\infty} s_n$ určíme její součet.

Věta 2.1
Jestliže řada $\sum_{n=1}^{\infty} a_n$ *konverguje, pak* $\lim_{n\to\infty} a_n = 0$.

Poznámka 2.2
Tato věta je vhodná pro rozhodování, zda řada diverguje a používá se ve tvaru tzv. logické obměny: Je-li $\lim_{n\to\infty} a_n \neq 0$, pak řada $\sum_{n=1}^{\infty} a_n$ diverguje.

Věta 2.2
Nechť $\sum a_n, \sum b_n$ jsou konvergentní řady takové, že $\sum a_n = s, \sum b_n = t$, nechť $k \in \mathrm{R}$. Pak platí:
a) Řada $\sum (a_n + b_n)$ je konvergentní, přičemž $\sum (a_n + b_n) = \sum a_n + \sum b_n = s + t$
b) Řada $k \cdot \sum a_n$ je konvergentní, přičemž $\sum k a_n = k \cdot \sum a_n = k \cdot s$
c) Řada $(a_1 + a_2 + \ldots + a_l) + (a_{l+1} + \ldots + a_m) + \ldots + (a_{p+1} + \ldots + a_q) + \ldots$ je konvergentní a má součet s.

Poznámka 2.3
Opačná implikace platí pouze u tvrzení b) pro $k \neq 0$, u ostatních dvou však obecně ne.

Připomenutí 2.1
Zopakujme si ještě pojem geometrické řady a vzorec pro její součet. *Geometrická řada* je řada tvaru $\sum_{n=1}^{\infty} a q^{n-1}$, kde $a \neq 0, q \neq 0$. Tato řada je pro $|q| < 1$ konvergentní a má součet $s = \sum_{n=1}^{\infty} a q^{n-1} = \frac{a}{1-q}$.

Ilustrace 2.1
Určete součet řady

$$\sum_{n=1}^{\infty} \frac{1}{n(n+3)}.$$

Postupujeme podle Pozn 2.1. Rozložíme výraz $\frac{1}{n(n+3)}$ na součet parciálních zlomků:

```
> rada:=1/(n*(n+3));
```

$$rada := \frac{1}{n\,(n+3)}$$

```
> convert(rada,'parfrac',n);
```

$$\frac{1}{3}\frac{1}{n} - \frac{1}{3}\frac{1}{n+3}$$

Pak tedy

$$\begin{aligned}
s_1 &= \frac{1}{3}(1-\frac{1}{4}) \\
s_2 &= \frac{1}{3}(1-\frac{1}{4}+\frac{1}{2}-\frac{1}{5}) \\
s_3 &= \frac{1}{3}(1-\frac{1}{4}+\frac{1}{2}-\frac{1}{5}+\frac{1}{3}-\frac{1}{6}) \\
s_4 &= \frac{1}{3}(1-\frac{1}{4}+\frac{1}{2}-\frac{1}{5}+\frac{1}{3}-\frac{1}{6}+\frac{1}{4}-\frac{1}{7}) \\
&\vdots \\
s_n &= \frac{1}{3}(1-\frac{1}{4}+\frac{1}{2}-\frac{1}{5}+\frac{1}{3}-\frac{1}{6}+\frac{1}{4}-\frac{1}{7}+\frac{1}{5}-\ldots \\
&\ldots+\frac{1}{n-3}-\frac{1}{n}+\frac{1}{n-2}-\frac{1}{n+1}+\frac{1}{n-1}-\frac{1}{n+2}+\frac{1}{n}-\frac{1}{n+3})
\end{aligned}$$

Vidíme, že většina členů se vyruší a dostaneme

$$s=\lim_{n\to\infty}s_n=\lim_{n\to\infty}\frac{1}{3}(1+\frac{1}{2}+\frac{1}{3}-\frac{1}{n+1}-\frac{1}{n+2}-\frac{1}{n+3})$$

Pro výpočet limity použijeme příkazu `limit`:

```
> s[n]:=1/3*(1+1/2+1/3-1/(n-1)-1/(n-2)-1/(n-3));
```

$$s_n := \frac{11}{18}-\frac{1}{3}\frac{1}{n-1}-\frac{1}{3}\frac{1}{n-2}-\frac{1}{3}\frac{1}{n-3}$$

```
> Limit(s[n],n=infinity)=limit(s[n],n=infinity);
```

$$\lim_{n\to\infty}\frac{11}{18}-\frac{1}{3}\frac{1}{n-1}-\frac{1}{3}\frac{1}{n-2}-\frac{1}{3}\frac{1}{n-3}=\frac{11}{18}$$

Výsledek:

$$\sum_{n=1}^{\infty}\frac{1}{n(n+3)}=\frac{11}{18}.$$

Je zřejmé, že tento způsob výpočtu je poněkud zdlouhavý. Uvedený postup proto automatizujeme pomocí Maplovského programovacího jazyku doprogramovanou procedurou

`poslcass(expr,down,k)`,

kde
`expr` je $n-$tý člen a_n dané řady $\sum a_n$,

`down` je dolní mez řady a
`k` je přirozené číslo, udávající kolik prvních členů posloupnosti částečných součtů si přejeme vypsat.
Jelikož se bavíme o nekonečných řadách, je horní mez implicitně nastavena na ∞. Procedura vypíše prvních `k` členů posloupnosti částečných součtů, její obecný $n-$tý člen a vrátí součet zadané řady:

```
> read(`moproc.m`);
> poslcass(rada,1,3);

s[1] = 1/4

s[2] = 7/20

s[3] = 73/180

s[n] = -1/3*(8+6*n+3*(n+1)^2)/(n+3)/(n+2)/(n+1)+11/18
```

$$\sum_{n=1}^{\infty} \frac{1}{n\,(n+3)} = \frac{11}{18}$$

Pro kontrolu součtu řady $\sum a_n$ můžeme využít Maplovský příkaz `sum`. V našem případě tedy dostaneme:

```
> Sum(rada,n=1..infinity)=sum(rada,n=1..infinity);
```

$$\sum_{n=1}^{\infty} \frac{1}{n\,(n+3)} = \frac{11}{18}$$

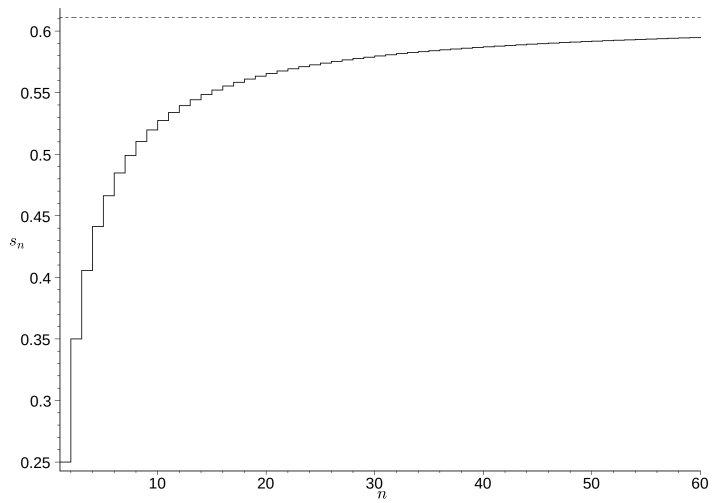
Pro větší názornost si také můžeme nechat vykreslit graf posloupnosti částečných součtů zadané řady. Čárkovaně budeme v tomto a dalších grafech označovat konstantní funkci $s\ = \sum_{n=1}^{\infty} a_n$, tedy limitu posloupnosti částečných součtů (pokud existuje). Z těchto grafů bude také dobře patrná tzv. *rychlost konvergence.* Tento pojem je intuitivně zřejmý - udává jak rychle se posloupnost částečných součtů blíží ke své limitě, tedy k součtu dané řady. Pro zájemce o přesnější vymezení pojmu *rychlost konvergence* odkazuji např. na [4].

Pro vykreslení grafu posloupnosti částečných součtů můžeme využít doprogramovanou proceduru

`sumplots(Sum(rada,n=a..b),c)`,

jejímž parametrem je příkaz `Sum`, psaný s velkým S, neboť pro správnou funkci procedury nesmí být vyhodnocen. Parametr `c` je nepovinný a vyjadřuje konstantu, k níž přičítáme danou řadu (viz Příklad 2.2). Před použitím procedury `sumplots` je třeba načíst knihovnu `plots` příkazem `with(plots)`.

```
> read(‘moproc.m‘);
> with(plots):
> sumplots(Sum(rada,n=1..60));
```



Obr. 2.1. Graf posloupnosti částečných součtů s_n řady $\sum_{n=1}^{\infty} \frac{1}{n(n+3)}$.

Ilustrace 2.2

Vyjádřete ve tvaru zlomku v základním tvaru číslo $-0{,}\overline{12}$.

Rozepišme si číslo $-0{,}\overline{12}$ do nekonečné řady:

$$-0{,}\overline{12} = \frac{-12}{10^2} + \frac{-12}{10^4} + \frac{-12}{10^6} + \ldots + \frac{-12}{10^{2n}} + \ldots .$$

Odtud je zřejmé, že se jedná o geometrickou řadu $\sum_{n=1}^{\infty} aq^{n-1}$, kde $a = \frac{-12}{10^2}$, $q = \frac{1}{10^2}$. Pak podle vzorce pro součet geometrické řady dostáváme:

$$-0{,}\overline{12} = \frac{\frac{-12}{10^2}}{1 - \frac{1}{10^2}}$$

Tento zlomek vyčíslíme pomocí Maplu:

```
> (-12/10^2)/(1-1/10^2);
```

$$\frac{-4}{33}$$

Výsledek:

$$-0{,}\overline{12} = \frac{-4}{33}.$$

K určování kvocientu q a členu a geometrické řady $\sum_{n=1}^{\infty} aq^{n-1}$ v Maplu slouží doprogramovaná procedura

`kvocgoem(expr)`,

kde
`expr` je $n-$tý člen a_n dané řady $\sum a_n$.

Pro určení součtu geometrické řady slouží procedura

`geom(kvoc,a)`,

kde parametry `kvoc` a `a` jsou globální proměnné, nastavené předchozí procedurou `kvocgeom`, avšak mohou za ně být samozřejmě dosazeny i konkrétní hodnoty.
V našem případě tedy:

```
> rada:=-12/10^(2*n);
```

$$rada := -12\,\frac{1}{10^{(2\,n)}}$$

```
> kvocgeom(rada);
```

```
kvoc = 1/100

a = -3/25
```

```
> geom(kvoc,a);
```

$$\frac{-4}{33}$$

Pro kontrolu opět můžeme použít příkaz `sum`:

```
> Sum(rada,n=1..infinity)=sum(rada,n=1..infinity);
```

$$\sum_{n=1}^{\infty} \left(-12\,\frac{1}{10^{(2\,n)}}\right) = \frac{-4}{33}$$

Ilustrace 2.3
Určete součet řady

$$\sum_{n=1}^{\infty}(\sqrt{n+2}-2\sqrt{n+1}+\sqrt{n}) \tag{2.1}$$

Vyjádřeme si několik členů:

$$\begin{aligned}
a_1 &= \sqrt{3}-2\sqrt{2}+1\\
a_2 &= \sqrt{4}-2\sqrt{3}+\sqrt{2}\\
a_3 &= \sqrt{5}-2\sqrt{4}+\sqrt{3}\\
&\vdots\\
a_{n-2} &= \sqrt{n}-2\sqrt{n-1}+\sqrt{n-2}\\
a_{n-1} &= \sqrt{n+1}-2\sqrt{n}+\sqrt{n-1}\\
a_n &= \sqrt{n+2}-2\sqrt{n+1}+\sqrt{n}
\end{aligned}$$

Dostáváme pak

$$\begin{aligned}
s_1 &= \sqrt{3}-2\sqrt{2}+1\\
s_2 &= \sqrt{4}-\sqrt{3}-\sqrt{2}+1\\
s_3 &= \sqrt{5}-\sqrt{4}-\sqrt{2}+1\\
&\vdots\\
s_n &= \sqrt{n+2}-\sqrt{n+1}-\sqrt{2}+1
\end{aligned}$$

a tedy

$$s = \lim_{n\to\infty} s_n = \lim_{n\to\infty}(1-\sqrt{2}-\sqrt{n+1}+\sqrt{n+2})$$

K výpočtu této limity můžeme využít systému Maple:

```
> s[n]:=sqrt(n+2)-sqrt(n+1)-sqrt(2)+1;
```

$$s_n := \sqrt{n+2}-\sqrt{n+1}-\sqrt{2}+1$$

```
> Limit(s[n],n=infinity)=limit(s[n],n=infinity);
```

$$\lim_{n\to\infty}\sqrt{n+2}-\sqrt{n+1}-\sqrt{2}+1 = -\sqrt{2}+1$$

Podívejme se, jak si s řadou (2.1) poradí příkaz `sum`:

```
> rada:=sqrt(n+2)-2*sqrt(n+1)+sqrt(n);
```

$$rada := \sqrt{n+2} - 2\sqrt{n+1} + \sqrt{n}$$

```
> Sum(rada,n=1..infinity)=sum(rada,n=1..infinity);
```

$$\sum_{n=1}^{\infty} (\sqrt{n+2} - 2\sqrt{n+1} + \sqrt{n}) = \sum_{n=1}^{\infty} (\sqrt{n+2} - 2\sqrt{n+1} + \sqrt{n})$$

Vidíme, že příkaz `sum` vrátil pouze zadání řady. To je způsobeno tím, že pro určení součtu řady (2.1) byly použity nestandardní úpravy. Obecně určení součtu řady není vždy možné (podrobněji viz Kap. 3, str. 35). Použití procedury `poslcass` by dopadlo stejně, neboť implementace procedury v sobě obsahuje i příkaz `sum`. V případech, kdy součet řady nelze určit příkazem `sum`, může někdy pomoci numerické vyhodnocení příkazem `evalf`, jímž určíme přibližný součet řady. Pro porovnání pak můžeme příkazem `evalf` vyjádřit přibližnou hodnotu limity posloupnosti částečných součtů s_n:

```
> Sum(rada,n=1..infinity)=evalf(sum(rada,n=1..infinity));
```

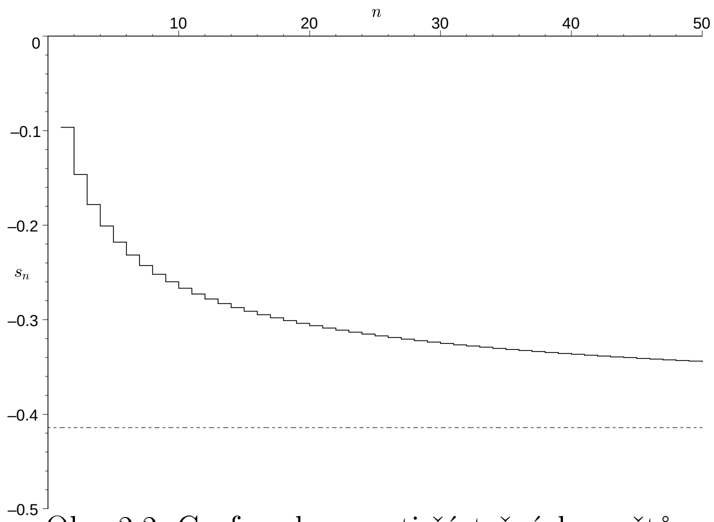
$$\sum_{n=1}^{\infty} (\sqrt{n+2} - 2\sqrt{n+1} + \sqrt{n}) = -.4142135624$$

```
> evalf(limit(s[n],n=infinity));
```

$$-.414213562$$

Pro názornost si opět můžeme vykreslit graf:

```
> sumplots(Sum(rada,n=1..50));
```



Obr. 2.2. Graf posloupnosti částečných součtů s_n řady $\sum_{n=1}^{\infty}(\sqrt{n+2}-2\sqrt{n+1}+\sqrt{n})$.

2.2 Příklady

PŘÍKLAD 2.1
Určete součet řady:

$$\text{a)}\ \sum_{n=1}^{\infty}\frac{1}{4n^2-1}$$

$$\text{b)}\ \sum_{n=1}^{\infty}\frac{3^n+2^n}{6^n}.$$

Řešení:
Postupujeme standardním způsobem.
a) Rozklad na parciální zlomky:

```
> rada := 1/(4*n^2-1):
> rada = convert(rada,'parfrac',n);
```

$$\frac{1}{4\,n^2-1}=\frac{1}{2}\,\frac{1}{2\,n-1}-\frac{1}{2}\,\frac{1}{2\,n+1}$$

Tedy

$$s_1 = \frac{1}{2}(1-\frac{1}{3})$$

$$\begin{aligned}
s_2 &= \frac{1}{2}(1-\frac{1}{3}+\frac{1}{3}-\frac{1}{5}) \\
s_3 &= \frac{1}{2}(1-\frac{1}{3}+\frac{1}{3}-\frac{1}{5}+\frac{1}{5}-\frac{1}{7}) \\
&\vdots \\
s_n &= \frac{1}{2}(1-\frac{1}{3}+\frac{1}{3}-\frac{1}{5}+\frac{1}{5}-\ldots-\frac{1}{2n-1}+\frac{1}{2n-1}-\frac{1}{2n+1})
\end{aligned}$$

Odtud dostáváme

$$s = \lim_{n\to\infty} s_n = \lim_{n\to\infty} \frac{1}{2}(1-\frac{1}{2n+1})$$

a výpočet limity můžeme opět svěřit Maplu:

```
> s[n]:=1/2*(1-1/(2*n+1));
```

$$s_n := \frac{1}{2} - \frac{1}{2}\,\frac{1}{2\,n+1}$$

```
> Limit(s[n],n=infinity)=limit(s[n],n=infinity);
```

$$\lim_{n\to\infty} \frac{1}{2} - \frac{1}{2}\,\frac{1}{2\,n+1} = \frac{1}{2}$$

Výsledek:

$$\sum_{n=1}^{\infty} \frac{1}{4n^2-1} = \frac{1}{2}.$$

V Maplu:

```
> poslcass(rada,1,3);

s[1] = 1/3

s[2] = 2/5

s[3] = 3/7

s[n] = -1/2/(2*n+1)+1/2
```

$$\sum_{n=1}^{\infty} \frac{1}{4\,n^2-1} = \frac{1}{2}$$

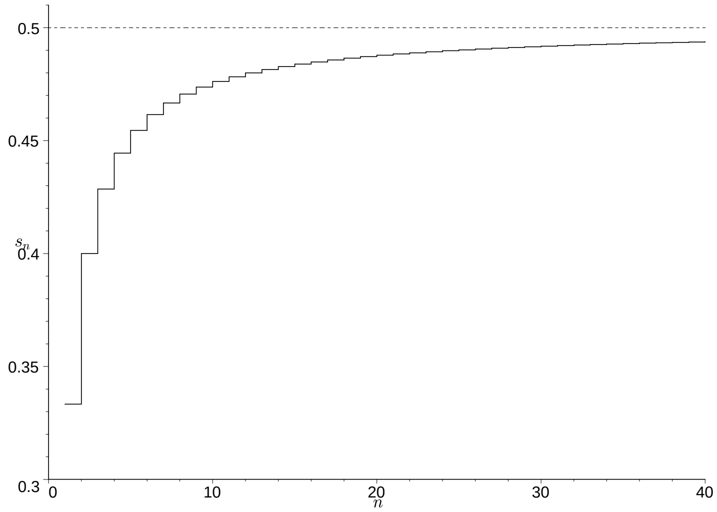
Kontrola:

```
> Sum(rada,n=1..infinity)=sum(rada,n=1..infinity);
```

$$\sum_{n=1}^{\infty} \frac{1}{4\,n^2-1} = \frac{1}{2}$$

Vizualizace:

```
> sumplots(Sum(rada,n=1..40));
```



Obr. 2.3. Graf posloupnosti částečných součtů s_n řady $\sum_{n=1}^{\infty} \frac{1}{4n^2-1}$

b)
Je vidět, že se jedná o součet dvou geometrických řad, z nichž každá má kvocient v absolutní hodnotě menší než jedna. Konkrétně $q_1 = \frac{1}{2}, a_1 = \frac{1}{2}, q_2 = \frac{1}{3}, a_2 = \frac{1}{3}$. K vyřešení tohoto příkladu tedy využijeme Větu 2.2b) a vzorec pro součet geometrické řady (viz Připomenutí 2.1).
Dostáváme

$$\sum_{n=1}^{\infty} \frac{3^n + 2^n}{6^n} = \sum_{n=1}^{\infty} (\frac{3^n}{6^n} + \frac{2^n}{6^n}) = \sum_{n=1}^{\infty} \frac{3^n}{6^n} + \sum_{n=1}^{\infty} \frac{2^n}{6^n} = \frac{\frac{1}{2}}{1-\frac{1}{2}} + \frac{\frac{1}{3}}{1-\frac{1}{3}}$$

Vyčíslení posledního výrazu již přenecháme systému Maple:

```
> 1/2/(1-1/2)+1/3/(1-1/3);
```

$$\frac{3}{2}$$

Výsledek:

$$\sum_{n=1}^{\infty} \frac{3^n + 2^n}{6^n} = \frac{3}{2}.$$

V Maplu:

K automatizaci určení kvocientů a členů a využijeme proceduru `kvocgeom` a pro součet geometrické řady proceduru `geom`:

```
> rada := (3^n+2^n)/6^n;
```

$$rada := \frac{3^n + 2^n}{6^n}$$

```
> kvocgeom(rada);
kvoc[1] = 1/2
a[1] = 1/2
kvoc[2] = 1/3
a[2] = 1/3
> geom(kvoc,a);
```

$$\frac{3}{2}$$

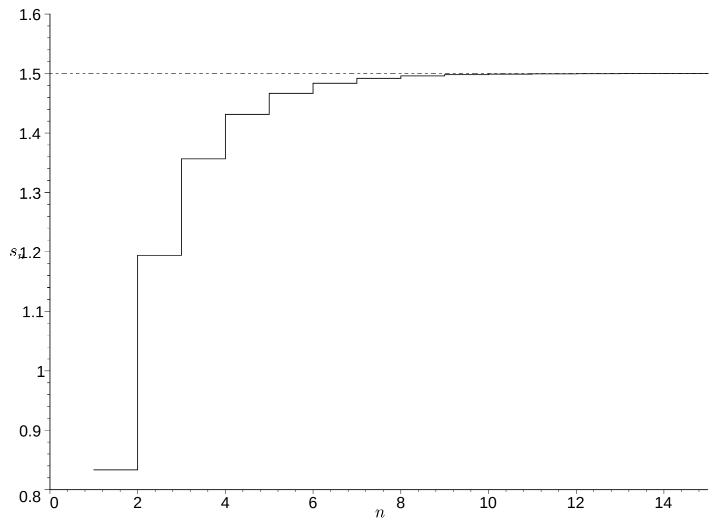
Kontrola:

```
> Sum(rada,n=1..infinity)=sum(rada,n=1..infinity);
```

$$\sum_{n=1}^{\infty} \frac{3^n + 2^n}{6^n} = \frac{3}{2}$$

Vizualizace:

```
> sumplots(Sum(rada,n=1..15));
```



Obr. 2.4. Graf posloupnosti částečných součtů s_n řady $\sum_{n=1}^{\infty} \frac{3^n+2^n}{6^n}$

Z grafu je dobře vidět, že rychlost konvergence dané řady ke svému součtu je velmi rychlá. Graf v takových případech trošku ztrácí na své názornosti v tom smyslu, že by mohl čtenáře zmást. Již pro $n = 11$ se totiž zdá, že částečný součet s_{11} je roven součtu řady $\sum_{n=1}^{\infty} \frac{3^n+2^n}{6^n} = s$. To je však omyl, neboť platí tvrzení:

Nechť $\sum a_n$ je řada, s_n posloupnost jejích částečných součtů a s součet řady $\sum a_n$. Pak pro libovolné přirozené n konečné platí:

$$s - s_n \neq 0.$$

Toto tvrzení platí pro všechny řady s vyjímkou těch, v nichž od jistého indexu počínaje jsou všechny členy nulové.

Vrátíme-li se tedy k našemu příkladu, je zřejmé, že pro $n \geq 11$ je rozdíl $s - s_n$ sice tak malý, že Maple jej v daném měřítku v grafu již zanedbává, v žádném případě však není nulový.

PŘÍKLAD 2.2
Zapište ve tvaru zlomku v základním tvaru $0{,}53\overline{9}$.

Řešení:

$$0{,}53\overline{9} = \frac{53}{10^2} + \frac{9}{10^3} + \frac{9}{10^4} + \ldots = \frac{53}{10^2} + \sum_{n=1}^{\infty} \frac{9}{10^{n+2}}.$$

Daná suma je geometrickou řadou, použijeme tedy opět procedury `kvocgeom` a `geom`.

```
> kvocgeom(9/10^(n+2));
```

```
kvoc = 1/10

a = 9/1000
```

```
> geom(kvoc,a);
```

$$\frac{1}{100}$$

Tedy $a = \frac{9}{10^3}, q = \frac{1}{10}$ a $\sum_{n=1}^{\infty} \frac{9}{10^{n+2}} = \frac{1}{100}$ a celkově dostáváme:

```
> 53/100+1/100;
```

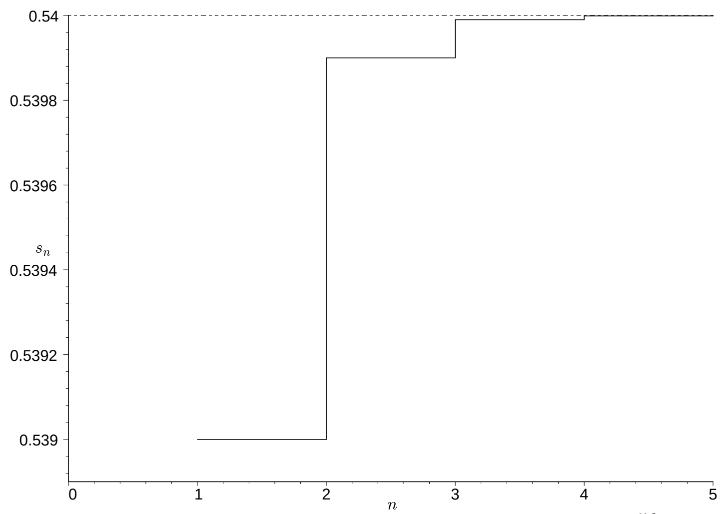
$$\frac{27}{50}$$

Výsledek:

$$0{,}53\overline{9} = \frac{27}{50}.$$

Vizualizace:

```
> sumplots(Sum(rada,n=1..5),53/100);
```



Obr. 2.5. Graf posloupnosti částečných součtů s_n řady $\frac{53}{100} + \sum_{n=1}^{\infty} \frac{9}{10^{n+2}}$

PŘÍKLAD 2.3

Rozhodněte, zda konverguje řada

$$\sum_{n=1}^{\infty} \frac{1}{\arctan n}.$$

Řešení:

Pro řešení tohoto příkladu využijeme Pozn. 2.2. Potřebujeme tedy spočítat

$$\lim_{n\to\infty} \frac{1}{\arctan n}.$$

S využitím Maplu dostáváme:

```
> rada:=1/arctan(n);
```

$$rada := \frac{1}{\arctan(n)}$$

```
> limit(rada, n=infinity);
```

$$2\,\frac{1}{\pi}$$

Vidíme, že $\lim_{n\to\infty} a_n \neq 0$, takže daná řada diverguje.

Výsledek:
Řada $\sum_{n=1}^{\infty} \frac{1}{\arctan n}$ diverguje.

Pozor! Podmínka $\lim_{n\to\infty} a_n = 0$, je pouze nutná pro konvergenci řady $\sum a_n$, tedy pokud by byla splněna, neznamenalo by to ještě, že daná řada konverguje.

Pokusme se ještě určit součet řady pomocí příkazu `sum`, případně jeho numerickým vyhodnocením:

```
> sum(rada,n=1..infinity);
```

$$\sum_{n=1}^{\infty} \frac{1}{\arctan(n)}$$

```
> evalf(%);
```

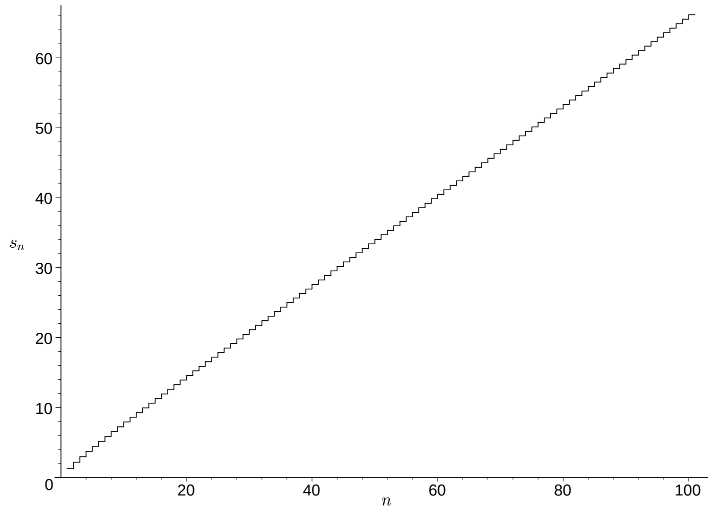
$$\sum_{n=1}^{\infty} \frac{1}{\arctan(n)}$$

Vizualizace:
V případech, kdy nelze součet řady určit příkazem `sum` ani numericky pomocí příkazu `evalf`, musíme pro vykreslení grafu posloupnosti částečných součtů použít doprogramovanou proceduru

`sumplot(Sum(rada,n=a..b),c)`,

která je stejná jako procedura `sumplots` s jediným rozdílem, že nevykresluje limitu posloupnosti částečných součtů.

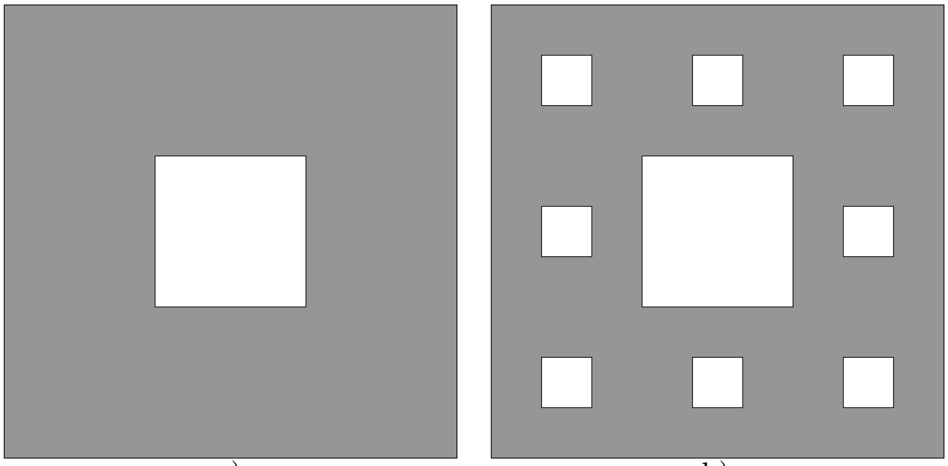
```
> sumplot(Sum(rada,n=1..100));
```



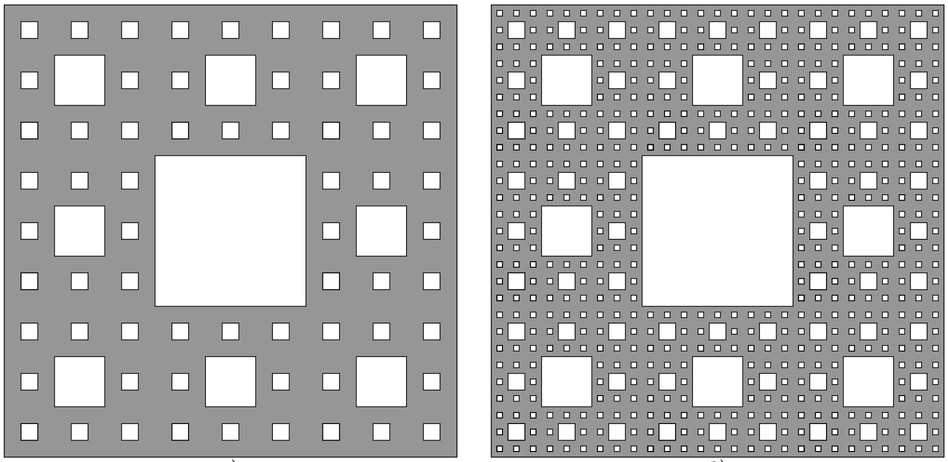
Obr. 2.6. Graf posloupnosti částečných součtů s_n řady $\sum_{n=1}^{\infty} \frac{1}{\arctan n}$

PŘÍKLAD 2.4

Vypočtěte obsah *Sierpinského koberce*, který se konstruuje takto: jednotkový čtverec rozdělíme na devět shodných čtverců a odstraníme vnitřek prostředního čtverce. Každý ze zbývajících osmi čtverců rozdělíme opět na devět shodných čtverců a znovu odstraníme vnitřek prostředního čtverce. Když tuto operaci prodloužíme do nekonečna, dostaneme útvar, který se nazývá Sierpinského koberec. Na Obr. 2.7.a)-d) jsou vykresleny první čtyři iterace Sierpinského koberce.



a) b)



c) d)

Obr. 2.7. První čtyři iterace Sierpinského koberce.

Tyto obrazce i další iterace Sierpinského koberce si lze vykreslit v Maplu pomocí doprogramované procedury

`sierpkob(n)`,

kde parametr `n` udává, kolikátá iterace se má vykreslit.

Řešení:
Obsah jednotkového čtverce je roven jedné a od tohoto obsahu budeme odečítat obsah odstraněných čtverců. Označme a_n obsah odstraněných čtverců v $n-$té iteraci a P hledaný obsah Sierpinského koberce.

V $n-$té iteraci odstraňujeme čtverce o straně $(\frac{1}{3})^n$, tedy obsah takového čtverce je $\left(\frac{1}{9}\right)^n$ a jejich počet v $n-$té iteraci je 8^{n-1}. Pak tedy

$$a_n = 8^{n-1} \cdot \left(\frac{1}{9}\right)^n = \frac{8^{n-1}}{9^n}$$

Celkový obsah Sierpinského koberce pak spočteme jako:

$$P = 1 - \sum_{n=1}^{\infty} a_n = 1 - \sum_{n=1}^{\infty} \frac{8^{n-1}}{9^n}$$

Řada $\sum_{n=1}^{\infty} a_n$ je geometrickou řadou, proto pro určení jejího součtu použijeme procedury `kvocgeom a geom`:

```
> rada:=8^(n-1)/9^n;
```

$$rada := \frac{8^{(n-1)}}{9^n}$$

```
> kvocgeom(rada);

kvoc = 8/9

a = 1/9
> geom(kvoc,a);
```

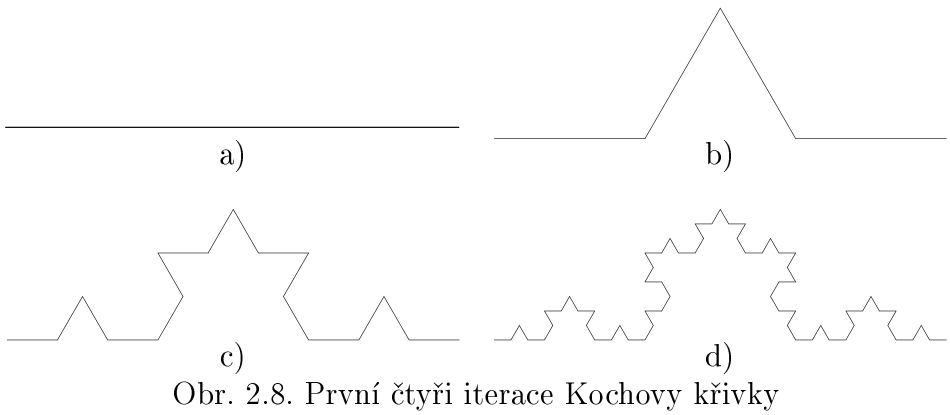
$$1$$

Dostali jsme, že $\sum_{n=1}^{\infty} \frac{8^{n-1}}{9^n} = 1$ a tedy

$$P = 1 - 1 = 0$$

Výsledek:
Obsah Sierpinského koberce je roven nule.

PŘÍKLAD 2.5
Vypočtěte délku tzv. *Kochovy křivky.* Kochova křivka vznikne tímto postupem:
V první iteraci si nakreslíme úsečku jednotkové délky. Ve druhé iteraci si úsečku rozdělíme na tři shodné díly, k prostřednímu dílu dokreslíme rovnostranný trojúhelník tak, že tento díl bude jednou z jeho stran a pak uvedený prostřední díl vymažeme. Dostaneme tak lomenou čáru skládající se ze čtyř shodných úseček. Ve třetí iteraci aplikujeme uvedený postup na všechny čtyři úsečky z předchozí iterace. Prodloužíme-li tento postup do nekonečna, dostaneme útvar, který se nazývá Kochova křivka. Postup konstrukce Kochovy křivky je zcela zřejmý z Obr. 2.8.a)-d), kde jsou znázorněny její první čtyři iterace.



Obr. 2.8. První čtyři iterace Kochovy křivky

Tyto křivky i další iterace Kochovy křivky si lze v Maplu vykreslit pomocí doprogramované procedury `kochkriv(n)`,

kde parametr `n` udává, kolikátá iterace se má vykreslit. Navíc si můžeme Kochovu křivku znázornit v animaci pomocí doprogramované procedury

`kochkranim(n)`,

kde parametr `n` udává, kolik kroků bude animace mít a zároveň na kolikáté iteraci se zastaví.

Řešení:
Označíme L délku Kochovy křivky. Dále lze postupovat dvěma způsoby.

1.způsob:

Označíme
a_n přírůstek délky Kochovy křivky v $n-$té iteraci oproti délce v $(n-1)$ní iteraci,
p_n rozdíl počtu úseček v $n-$té a $(n-1)$ní iteraci a
d_n délku úseček v $n-$té iteraci (jak víme, v libovolné pevně zvolené iteraci jsou si délky všech úseček rovny). Pak platí:

$$p_1 = 1 \qquad p_n = 3 \cdot 4^{n-2} \quad \text{pro } n = 2, 3, \dots$$

$$d_n = \left(\frac{1}{3}\right)^{n-1}$$

a dohromady

$$a_n = p_n \cdot d_n$$

Dostáváme pak:

$$a_1 = 1 \cdot \left(\frac{1}{3}\right)^{1-1} = 1$$

$$a_n = 3 \cdot 4^{n-2} \cdot \left(\frac{1}{3}\right)^{n-1} \qquad \text{pro } n = 2, 3, \dots$$

Tedy

$$L = \sum_{n=1}^{\infty} a_n$$

Výpočet délky L můžeme provést např. pomocí příkazu `sum`, 1. člen a_1 však musíme přičíst zvlášť:

```
> a[1]:=1;
```

$$a_1 := 1$$

```
> a[n]:=3*4^(n-2)/3^(n-1);
```

$$a_n := 3\,\frac{4^{(n-2)}}{3^{(n-1)}}$$

```
> L:=1+sum(a[n],n=2..infinity);
```

$$L := \infty$$

2.způsob:

Označme
s_n délku Kochovy křivky v $n-$té iteraci a
r_n počet úseček v $n-$té iteraci. Pak délka Kochovy křivky

$$L = \lim_{n\to\infty} s_n,$$

kde s_n je součinem délky úseček v $n-$té iteraci d_n a počtu úseček r_n. Platí:

$$r_n = 4^{n-1} \qquad \text{a} \qquad d_n = \left(\frac{1}{3}\right)^{n-1}$$

Tedy

$$s_n = \left(\frac{4}{3}\right)^{n-1}$$

a limitní přechod již můžeme provést pomocí Maplu:

```
> s[n]:=(4/3)^(n-1);
```

$$s_n := (\frac{4}{3})^{(n-1)}$$

```
> L:=limit(s[n],n=infinity);
```

$$L := \infty$$

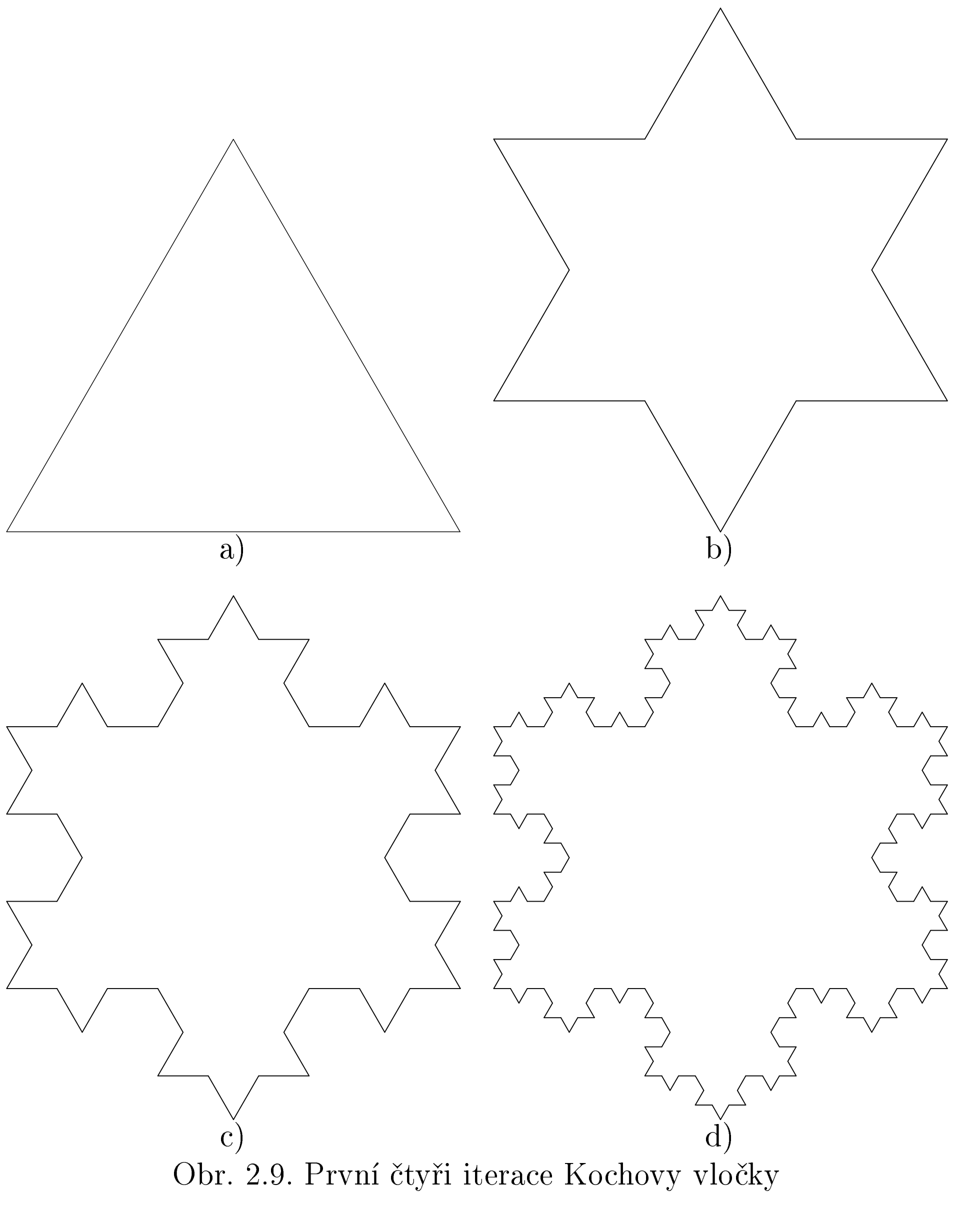
Výsledek:
Délka Kochovy křivky

$$L = \infty.$$

Všimněme si ještě vztahu mezi 1. a 2. způsobem výpočtu. V 1. způsobu jsme sečítali řadu $\sum a_n$, jejíž členy vyjadřovali přírůstky délky křivky v příslušných iteracích, ve 2. způsobu jsme pak hledali limitu posloupnosti s_n, jejíž členy vyjadřovali délku křivky v příslušné iteraci a posloupnost s_n tedy byla vlastně posloupností částečných součtů řady $\sum a_n$. Provedením 2. způsobu jsme tedy určili součet řady $\sum a_n$ přesně podle Definice 2.1.

PŘÍKLAD 2.6

Vypočtěte obvod a obsah tzv. *Kochovy vločky*. Kochova vločka se konstruuje stejným způsobem jako Kochova křivka s tím rozdílem, že první iterace není úsečka jednotkové délky, nýbrž rovnostranný trojúhelník o straně opět délky jedna. Postup konstrukce je zřejmý z Obr. 2.9.a)-d), kde jsou znázorněny opět první čtyři iterace Kochovy vločky.



Obr. 2.9. První čtyři iterace Kochovy vločky

Tyto obrazce, stejně jako další iterace Kochovy vločky lze v Maplu dostat pomocí doprogramované procedury `kochvloc(n)`,

kde parametr `n` udává, kolikátá iterace se má vykreslit. Stejně jako u Kochovy křivky si také můžeme prohlédnout animaci Kochovy vločky. Animaci zajistí doprogramovaná procedura

`kochvlocanim(n)`,

kde parametr `n` má stejný význam jako v proceduře `kochkranim(n)`, tj. udává, kolik kroků bude animace mít a zároveň na kolikáté iteraci se zastaví.

Řešení:
Označme O obvod Kochovy vločky, S její obsah.

Pro obvod Kochovy vločky platí: Každá ze stran rovnostranného trojúhelníka, který tvoří první iteraci, je zároveň první iterací Kochovy křivky. Odtud tedy plyne, že obvod Kochovy vločky bude trojnásobkem délky Kochovy křivky, tedy

$$O = 3 \cdot L,$$

kde L značí délku Kochovy křivky z předchozího příkladu, odkud máme $L = \infty$. Pak tedy

$$O = 3 \cdot \infty = \infty$$

Podívejme se nyní na obsah Kochovy vločky S. Označíme
a_n obsah trojúhelníků, které přikreslíme v $n-$té iteraci,
p_n počet trojúhelníků, které přikreslíme v $n-$té iteraci a
s_n obsah jednoho trojúhelníku přikresleného v $n-$té iteraci (je zřejmé, že obsahy všech přikreslených trojúhelníků v libovolné pevně zvolené iteraci si budou rovny). Pak platí:

$$S = \sum_{n=1}^{\infty} a_n,$$

kde

$$a_n = p_n \cdot s_n,$$

přičemž

$$p_1 = 1, \qquad p_n = 3 \cdot 4^{n-2} \quad \text{pro } n = 2, 3, \ldots$$

a

$$s_n = \frac{\sqrt{3}}{4} \cdot \frac{1}{9^{n-1}},$$

neboť obsah první iterace $s_1 = \frac{\sqrt{3}}{4}$ a z podobnosti trojúhelníků a vztahů pro obsah plyne, že s_n se bude pro každou iteraci devětkrát zmenšovat oproti s_{n-1}. Dostáváme tedy:

$$a_1 = \frac{\sqrt{3}}{4}, \qquad a_n = 3 \cdot 4^{n-2} \cdot \frac{\sqrt{3}}{4} \cdot \frac{1}{9^{n-1}} \quad \text{pro } n = 2, 3, \ldots$$

Pak

```
> a[1]:=sqrt(3)/4;
```

$$a_1 := \frac{1}{4}\sqrt{3}$$

```
> a[n]:=a[1]/9^(n-1)*3*4^(n-2);
```

$$a_n := \frac{3}{4}\,\frac{\sqrt{3}\,4^{(n-2)}}{9^{(n-1)}}$$

```
> S:=a[1]+sum(a[n],n=2..infinity);
```

$$S := \frac{2}{5}\sqrt{3}$$

Výsledek:

$$L = \infty, \qquad S = \frac{2\sqrt{3}}{5}$$

Opět jsme při výpočtu obsahu mohli postupovat i druhým způsobem, tedy počítat přímo obsah $n-$té iterace. Postup je však zcela analogický jako v předchozím příkladu, proto jej již přenechám čtenáři k samostatnému provedení.

Z výsledku plyne zajímavý paradox. Kochova vločka je obrazec, který má nekonečný obvod, ale konečný obsah. Tento a podobné paradoxy (např. nekonečný obvod a nulový obsah, atd.) jsou jednou z vlastností obrazců, které se nazývají fraktály. Mezi fraktály patří mimo jiné i Sierpinského koberec, Kochova křivka a Kochova vločka. Bližším zkoumáním fraktálů se zabývá velice dynamicky se rozvíjející tzv. *Teorie chaosu.*

Kapitola 3

Číselné řady s nezápornými členy

3.1 Základní pojmy

V této kapitole se budeme zabývat číselnými řadami s nezápornými členy, tj. řadami typu $\sum a_n$, kde $a_n \geq 0$ pro $\forall n \in \mathrm{N}$. U těchto řad se budeme snažit naučit se používat vhodná kriteria konvergence číselných řad. Určení součtu řady je totiž v drtivé většině případů velmi obtížný úkol, proto se často spokojíme se zjištěním, zda daná řada konverguje či diverguje.
Maple sám o sobě nemá implementována žádná kriteria konvergence, proto budeme používat doprogramované procedury. Syntaxi jednotlivých procedur si popíšeme vždy u příslušného kriteria, zde pouze uveďme, že všechny procedury mají stejný první parametr `rada`, jehož popis již tedy u jednotlivých procedur vynecháme a za který se dosazuje $n-$tý člen a_n vyšetřované řady $\sum a_n$.

Poznámka 3.1
Číselné řady s nezápornými členy mají jednu důležitou vlastnost. Posloupnosti jejich částečných součtů jsou vždy neklesající a tedy tyto řady buď konvergují, nebo určitě divergují k ∞.

Pro rozhodování o konvergenci, resp. divergenci řady je důležitý výběr správného kriteria. Jediným univerzálním kriteriem konvergence je totiž Cauchy-Bolzanovo (viz [1, str. 8]), které je však k praktickým výpočtům zcela nevhodné. Nyní si tedy uvedeme jednotlivá kriteria a několik zásad, kdy které kriterium použít.

3.1.1 Srovnávací kriterium

Věta 3.1
Nechť $\sum a_n, \sum b_n$ *jsou řady s nezápornými členy a* $a_n \leq b_n$ *pro skoro všechna*[1] $n \in \mathrm{N}$. *Pak platí:*
konverguje-li řada $\sum b_n$, *konverguje i řada* $\sum a_n$;
diverguje-li řada $\sum a_n$, *diverguje i řada* $\sum b_n$.

Toto kriterium má nevýhodu v tom, že potřebujeme dopředu znát jednu konvergentní a jednu divergentní řadu, s nimiž budeme zkoumanou řadu porovnávat. Najdeme-li tedy vhodnou porovnávací řadu, lze toto kriterium použít bez dalšího omezení.

Syntaxe:

```
srovk(rada,por,konv)
```

Parametr `por` je nepovinný a udává s jakou řadou budeme porovnávat vyšetřovanou řadu $\sum a_n$. Porovnáváme-li ji tedy s řadou $\sum b_n$, dosadíme za parametr `por` člen b_n. Použijeme-li parametr `por`, musíme také použít parametr `konv`, který určuje, zda parametr `por` reprezentuje konvergentní, či divergentní řadu. Za parametr `konv` se dosazuje `d` pro divergentní, nebo `k` pro konvergentní řadu. V případě, že je v paměti načtena proměnná stejného jména, je tento parametr nutno uzavřít do pravých apostrofů. Jestliže použijeme proceduru ve tvaru bez nepovinných parametrů, porovnává se implicitně s divergentní řadou $\sum \frac{1}{n}$ a konvergentní řadou $\sum \frac{1}{n^2}$.

Ilustrace 3.1
Rozhodněte o konvergenci, resp. divergenci řady

$$\sum_{n=1}^{\infty} \frac{1}{n(n+1)(n+2)}. \tag{3.1}$$

Danou řadu budeme porovnávat s konvergentní řadou $\sum_{n=1}^{\infty} \frac{1}{n^3}$. Je ihned vidět, že pro $\forall n \in \mathrm{N}$ platí

$$\frac{1}{n^3} > \frac{1}{n(n+1)(n+2)} \tag{3.2}$$

[1] *skoro všechna* znamená všechna s vyjímkou konečného počtu, tj. od jistého indexu počínaje.

a tedy řada (3.1) konverguje.

Využijeme-li procedury `srovk`, musíme v ní uvést volitelné parametry, neboť implicitně má procedura nastavenu konvergentní porovnávací řadu $\sum_{n=1}^{\infty} \frac{1}{n^2}$. Výpočet bude tedy vypadat takto:

```
> rada:=1/(n*(n+1)*(n+2));
```

$$rada := \frac{1}{n\,(n+1)\,(n+2)}$$

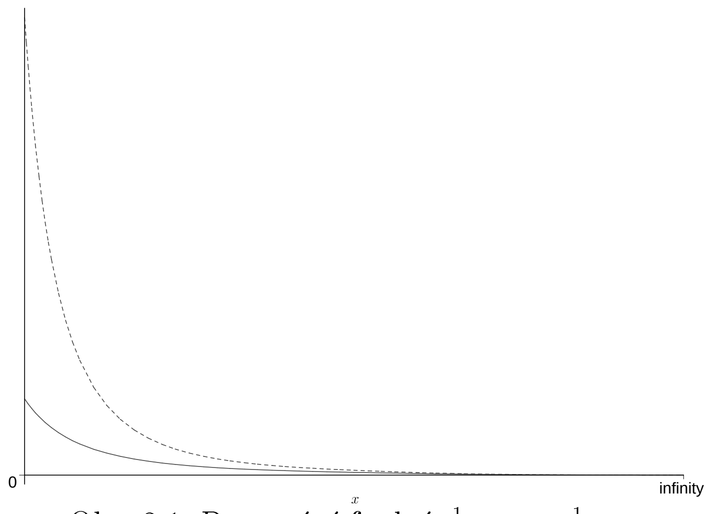
```
> srovk(rada,1/n^3,'k');
```

$$\sum_{n=1}^{\infty} \frac{1}{n\,(n+1)\,(n+2)}$$

$$konverguje$$

Platnost vztahu (3.2) si můžeme ověřit například vykreslením grafů příslušných funkcí:

```
> p1:=plot(1/x^3,x=1..infinity,linestyle=2):
> p2:=plot(subs(n=x,rada),x=1..infinity):
> display({p1,p2});
```



Obr. 3.1. Porovnání funkcí $\frac{1}{x^3}$ a $\frac{1}{x(x+1)(x+2)}$

Pro kontrolu můžeme použít univerzální proceduru

`kritkonverg(rada)`

Kontrola se provádí naprosto stejně u všech příkladů, v nichž rozhodujeme o konvergenci, resp. divergenci řady. Proto kontrolu provedeme pouze zde, v ostatních příkladech již přenechám čtenáři rozhodnutí, zda si kontrolu provede, či nikoliv.

Kontrola:

```
> kritkonverg(rada);
```

$$\sum_{n=1}^{\infty} \frac{1}{n\,(n+1)\,(n+2)}$$

$$\textit{konverguje}$$

3.1.2 Limitní srovnávací kriterium

Věta 3.2
Nechť $\sum a_n, \sum b_n$ *jsou řady s nezápornými členy a nechť existuje*

$$\lim_{n\to\infty} \frac{a_n}{b_n} = L.$$

Je-li $L < \infty$ *a konverguje-li řada* $\sum b_n$*, pak konverguje i řada* $\sum a_n$*. Je-li* $L > 0$ *a diverguje-li řada* $\sum b_n$*, pak diverguje i řada* $\sum a_n$*.*

Zde platí stejná zásada jako pro předchozí kriterium, tedy potřeba znalosti vhodné konvergentní a divergentní řady. V porovnání se Srovnávacím kriteriem je však Limitní srovnávací pro praktické výpočty vhodnější.

Syntaxe:

```
limsrovk(rada,por,konv)
```

Význam parametrů, stejně jako implicitní nastavení porovnávacích řad, je naprosto shodný jako u procedury `srovk`.

Ilustrace 3.2
Rozhodněte o konvergenci, resp. divergenci řady

$$\sum_{n=1}^{\infty} \frac{n^2+n+1}{n^3+1} \tag{3.3}$$

Zadanou řadu budeme porovnávat s divergentní řadou $\sum_{n=1}^{\infty} \frac{1}{n}$. Potřebujeme tedy vypočítat

$$L = \lim_{n\to\infty} \frac{\frac{n^2+n+1}{n^3+1}}{\frac{1}{n}}$$

K tomu můžeme využít Maple:

```
> rada:=(n^2+n+1)/(n^3+1);
```

$$rada := \frac{n^2+n+1}{n^3+1}$$

```
> Limit(rada/(1/n),n=infinity)= limit(rada/(1/n),n=infinity);
```

$$\lim_{n\to\infty} \frac{(n^2+n+1)\,n}{n^3+1} = 1$$

Odtud tedy dostáváme

$$L > 0,$$

což podle Věty 3.2 znamená, že řada (3.3) diverguje.
Nyní si ukážeme použití procedury `limsrovk`. Protože danou řadu porovnáváme s řadou, která je implicitně nastavena, nemusíme uvádět nepovinné parametry:

```
> limsrovk(rada);
```

$$\sum_{n=1}^{\infty} \frac{n^2+n+1}{n^3+1}$$

diverguje

3.1.3 Odmocninové kriterium

Věta 3.3
Nechť $\sum a_n$ je řada s nezápornými členy.
(i) Platí-li pro $\forall n \in \mathrm{N}$ nerovnost $\sqrt[n]{a_n} \le q < 1$, pak řada konverguje. Platí-li pro nekonečně mnoho $n \in \mathrm{N}$ nerovnost $\sqrt[n]{a_n} \ge 1$, řada diverguje.
(ii) Existuje-li

$$\lim_{n\to\infty} \sqrt[n]{a_n} = q, \qquad kde\ q \in \mathrm{R}^*,$$

pak v případě $q < 1$ řada $\sum a_n$ konverguje a v případě $q > 1$ řada $\sum a_n$ diverguje.

Poznámka 3.2
Jestliže $q = 1$, nelze tímto kriteriem rozhodnout. To platí také pro následující Podílové a Limitní Raabeovo kriterium.

Odmocninové kriterium je nejvhodnější pro řady, v jejichž členu a_n se vyskytuje n-tá mocnina.

Syntaxe:

```
odmock(rada)
```

Ilustrace 3.3
Vyšetřete konvergenci řady

$$\sum_{n=1}^{\infty} \arctan^n \frac{1}{n} \tag{3.4}$$

Při ručním výpočtu musíme spočítat

$$q = \lim_{n\to\infty} \sqrt[n]{\arctan^n \frac{1}{n}}$$

Tento úkol je opět vhodné svěřit Maplu:

```
> rada:=(arctan(1/n))^n;
```

$$rada := \arctan(\frac{1}{n})^n$$

```
> Q:=rada^(1/n);
```

$$Q := (\arctan(\frac{1}{n})^n)^{(\frac{1}{n})}$$

```
> Limit(Q,n=infinity)=limit(Q,n=infinity);
```

$$\lim_{n\to\infty} (\arctan(\frac{1}{n})^n)^{(\frac{1}{n})} = 0$$

Dostali jsme, že

$$q < 1$$

a tedy podle Věty 3.3 řada (3.4) konverguje.

Pomocí procedury `odmock` musí samozřejmě výsledek dopadnout stejně:

```
> odmock(rada);
```

$$\sum_{n=1}^{\infty} \arctan(\frac{1}{n})^n$$

$$\textit{konverguje}$$

3.1.4 Podílové kriterium

Věta 3.4
Nechť $\sum a_n$ je řada s kladnými členy.
(i) Platí-li pro $\forall n \in \mathrm{N}$ nerovnost $\frac{a_{n+1}}{a_n} \leq q < 1$, pak řada konverguje. Platí-li pro $\forall n \in \mathrm{N}$ nerovnost $\frac{a_{n+1}}{a_n} \geq 1$, řada diverguje.
(ii) Existuje-li

$$\lim_{n\to\infty} \frac{a_{n+1}}{a_n} = q, \qquad kde\ q \in \mathrm{R}^*,$$

pak v případě $q < 1$ řada $\sum a_n$ konverguje a v případě $q > 1$ řada $\sum a_n$ diverguje.

Toto kriterium je vhodné pro řady, u nichž lze očekávat krácení v podílu $\frac{a_{n+1}}{a_n}$. Takové krácení lze předpokládat především u výrazů s faktoriály.

Syntaxe:

```
podilk(rada)
```

Ilustrace 3.4
Rozhodněte o konvergenci řady

$$\sum_{n=1}^{\infty} \frac{n^2}{n!} \tag{3.5}$$

Ve výrazu se vyskytuje faktoriál, použijeme tedy Podílového kriteria. Chceme tedy spočítat

$$q = \lim_{n\to\infty} \frac{\frac{(n+1)^2}{(n+1)!}}{\frac{n^2}{n!}}$$

Opět přichází ke slovu Maple:

```
> rada:=n^2/n!;
```

$$rada := \frac{n^2}{n!}$$

```
> podil:=subs(n=n+1,rada)/rada;
```

$$podil := \frac{(n+1)^2\, n!}{(n+1)!\, n^2}$$

```
> Limit(podil,n=infinity)=limit(podil,n=infinity);
```

$$\lim_{n\to\infty} \frac{(n+1)^2\, n!}{(n+1)!\, n^2} = 0$$

Odtud

$$q < 1,$$

což podle Věty 3.4 značí, že řada (3.5) konverguje.

Výpočet pomocí procedury `podilk`:

```
> podilk(rada);
```

$$\sum_{n=1}^{\infty} \frac{n^2}{n!}$$

$$konverguje$$

3.1.5 Limitní Raabeovo kriterium

Věta 3.5
Nechť $\sum a_n$ je řada s kladnými členy a nechť existuje

$$\lim_{n\to\infty} n\left(1 - \frac{a_{n+1}}{a_n}\right) = q, \qquad kde\ q \in \mathrm{R}^*.$$

Je-li $q > 1$, pak řada $\sum a_n$ konverguje, je-li $q < 1$, pak řada $\sum a_n$ diverguje.

Toto kriterium nám může pomoci z nesnází v případech, kdy nelze rozhodnout Podílovým a Odmocninovým kriteriem. Je univerzálnější než Podílové a Odmocninové, ale také výpočetně náročnější, proto se užívá pouze v případech, kdy uvedená dvě kriteria nelze použít.

Syntaxe:

```
limraabk(rada)
```

Ilustrace 3.5
Vyšetřete konvergenci řady

$$\sum_{n=1}^{\infty} \frac{n}{n+3} \tag{3.6}$$

Je to případ řady, o níž nelze rozhodnout Podílovým ani Odmocninovým kriteriem. Ručním výpočtem tento závěr ověřovat nebudeme, ale přesvědčíme se např. použitím procedury `podilk`:

```
> rada:=n/(n+3);
```

$$rada := \frac{n}{n+3}$$

```
> podilk(rada);
```

timto kriteriem nelze rozhodnout

Nabízí se tedy Limitní Raabeovo kriterium. Pro jeho použití potřebujeme znát hodnotu

$$q = \lim_{n\to\infty} n\left(1 - \frac{a_{n+1}}{a_n}\right)$$

Pomocí Maplu spočítáme:

```
> podil:=subs(n=n+1,rada)/rada;
```

$$podil := \frac{(n+1)\,(n+3)}{(n+4)\,n}$$

```
> Limit(n*(1-podil),n=infinity)=limit(n*(1-podil),% n=infinity);
```

$$\lim_{n\to\infty} n\,(1 - \frac{(n+1)\,(n+3)}{(n+4)\,n}) = 0$$

Dostáváme tedy

$$q < 1,$$

což podle Věty 3.5 znamená, že řada (3.6) diverguje (všimněte si obrácených nerovností u Limitního Raabova kriteria!)

Výpočet s použitím procedury `limraabk`:

```
> limraabk(rada);
```

$$\sum_{n=1}^{\infty} \frac{n}{n+3}$$

diverguje

3.1.6 Integrální kriterium

Věta 3.6
Nechť f je funkce definovaná na intervalu $[1,\infty)$, která je na tomto intervalu nezáporná a nerostoucí. Nechť $f(n) = a_n$ pro $\forall n \in \mathrm{N}$. Pak řada $\sum_{n=1}^{\infty} a_n$ konverguje právě tehdy, když konverguje nevlastní integrál $\int_1^{\infty} f(x)dx$.

Toto kriterium ukazuje souvislost mezi nekonečnými řadami a nevlastními integrály. Lze jej použít vždy, když umíme spočítat odpovídající nevlastní integrál $\int_1^{\infty} f(x)dx$.

Syntaxe:

```
intk(rada,down)
```

Parametr `down` je nepovinný a udává dolní index dané řady (implicitně nastaven na 1).

Ilustrace 3.6
Vyšetřete konvergenci řady

$$\sum_{n=2}^{\infty} \frac{1}{n \ln n} \tag{3.7}$$

Uvažme tedy příslušnou funkci $f(x) = \frac{1}{x \ln x}$. Musíme ověřit, že funkce f je definovaná, nezáporná a nerostoucí na intervalu $[2,\infty)$ (neboť počáteční index řady (3.7) je roven 2). Že je f na tomto intervalu definovaná je zřejmé na první pohled, zbylé dvě podmínky ověříme pomocí Maplu. Nezápornost ověříme příkazem `solve`, jímž zjistíme na jakém intervalu je f nezáporná. Pro zjištění na jakém intervalu je f nerostoucí využijeme našich znalostí z diferenciálního počtu, odkud víme, že funkce f je nerostoucí právě tehdy, když pro její derivaci platí $f' \leq 0$:

```
> rada:=1/(x*ln(x));
```

$$rada := \frac{1}{x \ln(x)}$$

```
> solve(rada>=0,x);
```

$$\mathrm{RealRange}(\mathrm{Open}(1), \infty)$$

```
> simplify(diff(rada,x));
```

$$-\frac{\ln(x)+1}{x^2 \ln(x)^2}$$

```
> solve(%<=0);
```

$$\text{RealRange}(e^{(-1)}, \text{Open}(1)), \text{RealRange}(\text{Open}(1), \infty)$$

Podmínky Integrálního kriteria jsou tedy splněny a proto tohoto kriteria můžeme použít. Zbývá tedy ještě spočítat nevlastní integrál

$$\int_2^\infty \frac{1}{x \ln x} dx$$

S tímto úkolem nám opět může pomoci systém Maple:

```
> Int(rada,x=2..infinity)=int(rada,x=2..infinity);
```

$$\int_2^\infty \frac{1}{x \ln(x)}\, dx = \infty$$

Vidíme, že nevlastní integrál diverguje, podle Integrálního kriteria tedy diverguje i řada (3.7).
Použití procedury `intk`:

```
> intk(subs(x=n,rada));
```

$$\sum_{n=2}^{\infty} \frac{1}{n \ln(n)}$$

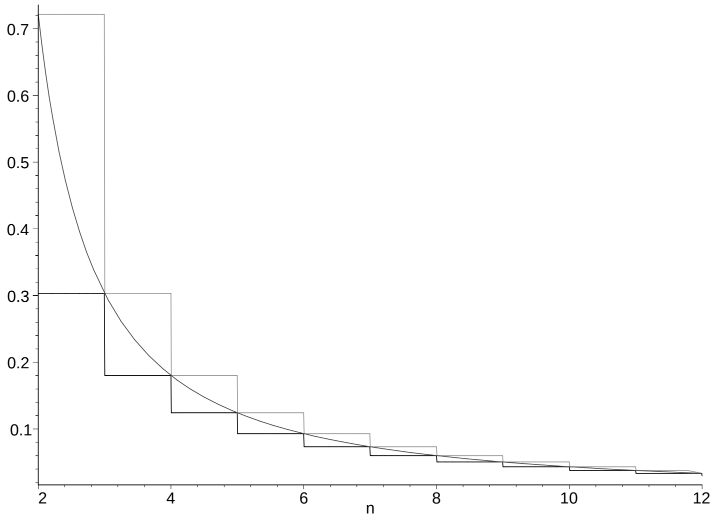
$$\textit{diverguje}$$

Pro názornost si můžeme vykreslit graf. Člen řady nyní zadáme jako Maplovskou funkci:

```
> f:=n->1/(n*ln(n));
```

$$f := n \to \frac{1}{n \ln(n)}$$

```
> plot([f(n),f(floor(n)),f(floor(n+1))], n=2..12,
>     color=[red,green,black]);
```



Obr. 3.2. Ilustrace Integrálního kriteria pro řadu $\sum_{n=2}^{\infty} \frac{1}{n \ln n}$

Obr. 3.2 velice názorně ilustruje vztah mezi nevlastními integrály a nekonečnými řadami a také myšlenku důkazu Integrálního kriteria (důkaz lze najít např v [1]). Větší z obou lomených čar reprezentuje členy řady $a_n = \frac{1}{n \ln n}$, menší reprezentuje členy řady $a_{n+1} = \frac{1}{(n+1) \cdot \ln(n+1)}$ a křivka zobrazuje funkci $f(x) = \frac{1}{x \ln x}$. Je zřejmé, že

$$a_{n+1} = \int_n^{n+1} a_{n+1} dx \leq \int_n^{n+1} f(x) dx \leq \int_n^{n+1} a_n dx = a_n$$

Sečtením nerovností tedy dostáváme

$$\sum_{n=2}^{\infty} a_{n+1} \leq \int_2^{\infty} f(n) dn \leq \sum_{n=2}^{\infty} a_n,$$

přičemž platí:

$$\sum_{n=2}^{\infty} a_{n+1} = \sum_{n=3}^{\infty} a_n$$

Víme, že na konvergenci, resp. divergenci řady nemá vliv to, od kterého indexu řadu sčítáme a odtud již plyne tvrzení Integrálního kriteria.

3.1.7 Kummerovo kriterium

Věta 3.7

Nechť $\{c_1, c_2, \ldots, c_n, \ldots\}$ je posloupnost reálných čísel taková, že řada $\sum_{n=1}^{\infty} \frac{1}{c_n}$

je divergentní. Nechť

$$K_n = c_n \cdot \frac{a_n}{a_{n+1}} - c_{n+1} \qquad \text{a nechť existuje } \lim_{n\to\infty} K_n = K.$$

Je-li $K > 0$, *řada* $\sum a_n$ *konverguje, je-li* $K < 0$, *řada* $\sum a_n$ *diverguje.*

Kummerovo kriterium je zobecněním Limitního Raabeova kriteria. Jeho nevýhodami jsou jednak opět potřeba znalosti jedné divergentní řady a jednak opět náročnější výpočet. Z toho plyne i jeho použití pouze v případech, kdy nelze rozhodnout ani Limitním Raabeovým kriteriem.

Syntaxe:

```
kummerk(rada,c)
```

Parametr `c` je nepovinný a dosazuje se za něj $n-$tý člen posloupnosti $\{c_n\}$ z Kummerova kriteria.

Poznámka 3.3
Další kriteria lze najít např. v [7].

3.2 Příklady

V tomto odstavci si spočítáme několik příkladů. Zadání bude vždy stejné: rozhodněte o konvergenci, resp. divergenci řady. Ruční výpočet ponechám na samostatné iniciativě čtenáře. V každém příkladu tedy odhadneme, která kriteria můžeme použít a vyřešíme úkol pomocí odpovídající doprogramované procedury (půjde-li použít více kriterií, vybereme jedno a jemu příslušnou proceduru).

PŘÍKLAD 3.1

$$\sum_{n=1}^{\infty} \frac{1}{n^n}$$

Řešení:
Pro tuto řadu by se dala použít všechna uvedená kriteria, ale protože se ve výrazu vyskytuje n-tá mocnina, přímo nás to vybízí použít Odmocninové kriterium, které také bude výpočetně nejjednodušší:

```
> rada:=1/n^n;
```

$$rada := \frac{1}{n^n}$$

```
> odmock(rada);
```

$$\sum_{n=1}^{\infty} \frac{1}{n^n}$$

$$konverguje$$

Výsledek:
Řada $\sum_{n=1}^{\infty} \frac{1}{n^n}$ konverguje.

PŘÍKLAD 3.2

$$\sum_{n=1}^{\infty} \frac{1}{n^2}$$

Řešení:
Již víme, že tato řada konverguje, dokažme si to však pomocí Integrálního kriteria. Ověření předpokladů:

```
> rada:=1/x^2;
```

$$rada := \frac{1}{x^2}$$

```
> solve(rada>=0,x);
```

$$x$$

```
> simplify(diff(rada,x));
```

$$-2\,\frac{1}{x^3}$$

```
> solve(%<=0);
```

$$\mathrm{RealRange}(0, \infty)$$

Podmínky jsou tedy splněny, takže můžeme použít Integrálního kriteria:

```
> intk(subs(x=n,rada));
```

$$\sum_{n=1}^{\infty} \frac{1}{n^2}$$

$$konverguje$$

O konvergenci dané řady se též můžeme přesvědčit přímým výpočtem nevlastního integrálu. Dostáváme pak:

```
> Int(rada,x=1..infinity)=int(rada,x=1..infinity);
```

$$\int_1^\infty \frac{1}{x^2}\,dx = 1$$

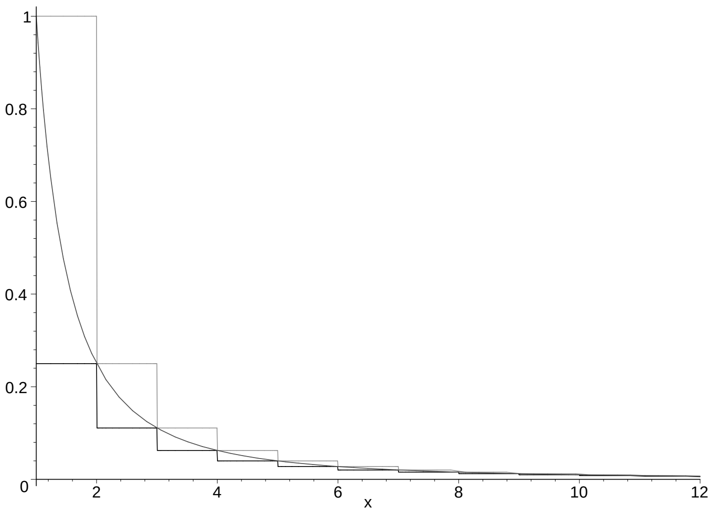
Výsledek:
Řada $\sum_{n=1}^{\infty} \frac{1}{n^2}$ konverguje.

Vizualizace:

```
> f:=n->1/n^2;
```

$$f := n \to \frac{1}{n^2}$$

```
> plot([f(n),f(floor(n)),f(floor(n+1))], n=1..12,
>     color=[red,green,black]);
```



Obr. 3.3. Integrální kriterium u řady $\sum_{n=1}^{\infty} \frac{1}{n^2}$

PŘÍKLAD 3.3

$$\sum_{n=1}^{\infty} \frac{(n!)^2}{(2n)!}$$

Řešení:
Ve výrazu se vyskytují faktoriály, což pro nás signalizuje použít Podílové kriterium. S vyjímkou Srovnávacího by se také dalo použít všeh ostatních,

ovšem výpočet by byl časově náročnější i výpočetně složitější (tab. 3.1 prezentuje čas výpočtu pro jednotlivá kriteria u řady $\sum_{n=1}^{\infty} \frac{(n!)^2}{(2n)!}$). [2]

Procedura	**Čas [s]**
limsrovk	63.542
odmock	41.263
podilk	24.336
limraabk	56.057
intk	690.288
kummerk	60.062

Tabulka 3.1: Časová náročnost

Použijeme tedy Podílového kriteria:

```
> rada:=(n!)^2/(2*n)!;
```

$$rada := \frac{(n!)^2}{(2\,n)!}$$

```
> podilk(rada);
```

$$\sum_{n=1}^{\infty} \frac{(n!)^2}{(2\,n)!}$$

$$konverguje$$

Výsledek:
Řada $\sum_{n=1}^{\infty} \frac{(n!)^2}{(2n)!}$ konverguje.

PŘÍKLAD 3.4

$$\sum_{n=1}^{\infty} \frac{1}{\ln(n+1)}$$

Řešení:
Zde jsou k dispozici kriteria Integrální, Limitní srovnávací a Limitní Raabeovo. My provedeme výpočet pomocí Limitního srovnávacího kriteria. Demonstrujme si na tomto příkladu použití nepovinných parametrů. Nejprve zkusíme použít pro srovnání konvergentní řadu $\sum \frac{1}{n^3}$, neuspějeme-li, zkusíme divergentní řadu $\sum \frac{1}{n}$:

[2]Měřeno na počítači s procesorem 80486DX2 a pamětí 8 MB RAM.

```
> rada:=1/ln(n+1);
```

$$rada := \frac{1}{\ln(n+1)}$$

```
> limsrovk(rada,1/n^3,'k');
```

timto kriteriem nelze rozhodnout

```
> limsrovk(rada,1/n,'d');
```

$$\sum_{n=1}^{\infty} \frac{1}{\ln(n+1)}$$

diverguje

Výsledek:
Řada $\sum_{n=1}^{\infty} \frac{1}{\ln(n+1)}$ diverguje.

PŘÍKLAD 3.5

$$\sum_{n=1}^{\infty} \frac{n^2+1}{n^3}$$

Řešení:
O této řadě lze rozhodnout Srovnávacím, Limitním srovnávacím a Integrálním kriteriem. Zvolme si tedy např. Srovnávací. Danou řadu budeme porovnávat s divergentní řadou $\sum \frac{1}{n}$. Musíme tedy ověřit platnost vztahu

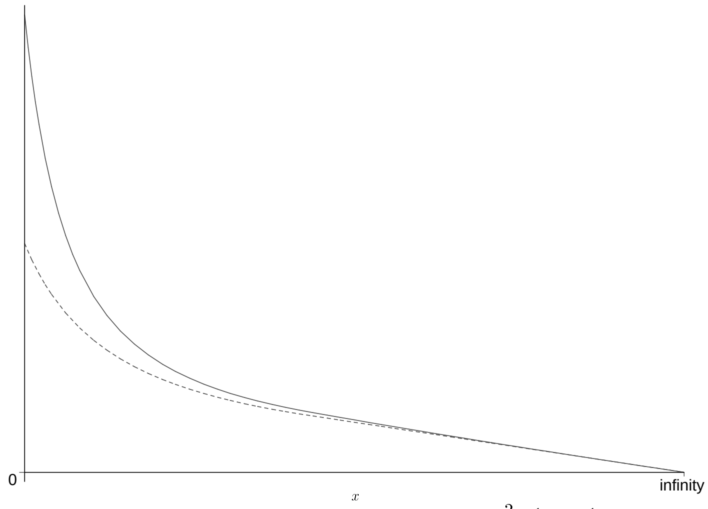
$$\frac{1}{n} \leq \frac{n^2+1}{n^3} \tag{3.8}$$

pro skoro všechna $n \in$ N. Nejnázornější bude ověřit platnost vztahu (3.8) vykreslením grafů příslušných funkcí:

```
> rada:=(n^2+1)/n^3;
```

$$rada := \frac{n^2+1}{n^3}$$

```
> p1:=plot(1/x,x=1..infinity,linestyle=2):
> p2:=plot(subs(n=x,rada),x=1..infinity):
> display({p1,p2});
```



Obr. 3.4. Porovnání funkcí $\frac{x^2+1}{x^3}$ a $\frac{1}{x}$

Vidíme, že vztah (3.8) je splněn na intervalu $(0, \infty)$ a tedy i pro všechna $n \in \mathrm{N}$, takže daná řada podle Srovnávacího kriteria diverguje.
Výpočet pomocí procedury `srovk`:

```
> srovk(rada);
```

$$\sum_{n=1}^{\infty} \frac{n^2+1}{n^3}$$

$$\textit{diverguje}$$

Výsledek:
Řada $\sum_{n=1}^{\infty} \frac{n^2+1}{n^3}$ diverguje.

Poznámka 3.4
Příkladem řady, u níž nelze rozhodnout žádným z uvedených kriterií může být např. řada

$$\sum_{n=1}^{\infty} \frac{1}{n \ln \ln \ln n}.$$

Kapitola 4

Absolutní a neabsolutní konvergence

4.1 Alternující řady

Definice 4.1
Řekneme, že řada $\sum_{n=1}^{\infty} a_n$ je *alternující*, pokud

$$\text{sgn } a_{n+1} = -\text{sgn } a_n \text{ pro } \forall n \in \mathrm{N}.$$

Alternující řadu budeme zapisovat ve tvaru $\sum_{n=1}^{\infty}(-1)^{n-1}a_n$ nebo $\sum_{n=1}^{\infty}(-1)^n a_n$ pro $\forall n \in \mathrm{N}$.

Věta 4.1 (Leibnizovo kriterium)
Nechť a_n je nerostoucí posloupnost kladných čísel. Pak alternující řada $\sum_{n=1}^{\infty}(-1)^{n-1}a_n$ konverguje právě když $\lim_{n\to\infty} a_n = 0$.

Poznámka 4.1
Vidíme tedy, že pro rozhodnutí o konvergenci alternující řady nám stačí ověřit uvedené dvě podmínky. Při ověřování nám může pomoci Maple.

Ilustrace 4.1
Rozhodněte, zda konverguje alternující řada

$$\sum_{n=1}^{\infty}(-1)^{n-1}\frac{1}{n-\ln n} \tag{4.1}$$

Ověření, zda je $\lim_{n\to\infty} a_n = 0$ už zvládneme bez potíží, pro ověření, zda je posloupnost $\{a_n\}$ nerostoucí využijeme obdobný postup jako v Ilustraci 3.6. Uvažme tedy funkci $f(x) = \frac{1}{x-\ln x}$.

Pomocí Maplu tedy řešíme:

```
> f:=1/(x-ln(x));
```

$$f := \frac{1}{x - \ln(x)}$$

```
> limit(f,x=infinity);
```

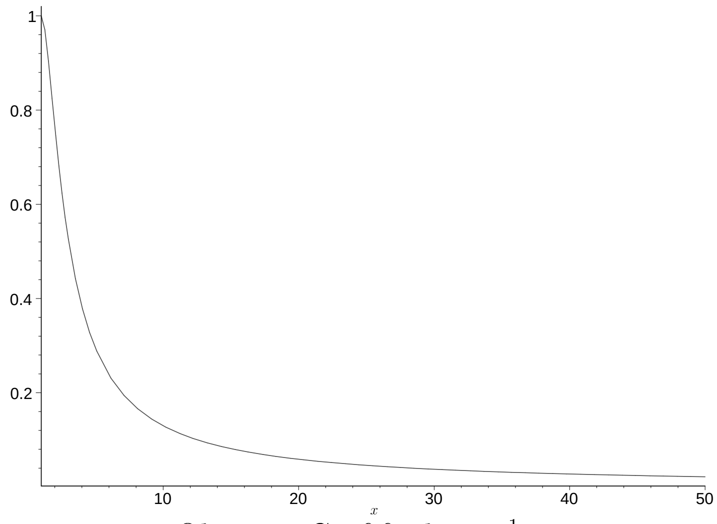
$$0$$

```
> solve(diff(f,x)<=0);
```

$$\mathrm{RealRange}(1, \infty)$$

Funkce f (a tudíž i její příslušná posloupnost) je tedy nerostoucí na intervalu $[1, \infty)$. Že je výraz $\frac{1}{n-\ln n}$ kladný pro $\forall n \in \mathrm{N}$ plyne z platnosti vztahu $n > \ln n$ pro $\forall n \in \mathrm{N}$ a je to zřejmé i z Obr. 4.1, odkud je též zřejmé, že posloupnost $\left\{\frac{1}{n-\ln n}\right\}$ je nerostoucí:

```
> plot(rada,x=1..50);
```



Obr. 4.1. Graf funkce $\frac{1}{x-\ln x}$

Ověřili jsme platnost podmínek Leibnizova kriteria a tedy podle tohoto kriteria řada (4.1) konverguje.

4.2 Absolutní konvergence číselných řad

Definice 4.2
Nechť řada $\sum a_n$ konverguje. Jestliže konverguje i řada $\sum |a_n|$, řekneme, že řada $\sum a_n$ *konverguje absolutně.* Jestliže řada $\sum |a_n|$ diverguje, říkáme, že řada $\sum a_n$ *konverguje neabsolutně.*

Poznámka 4.2
Řada $\sum |a_n|$ je řada s nezápornými členy. Máme-li tedy rozhodnout o absolutní, resp. neabsolutní konvergenci řady $\sum a_n$, převedeme problém na rozhodování o konvergenci řady s nezápornými členy $\sum |a_n|$. Můžeme tudíž používat všechna kriteria uvedená ve třetí kapitole.

Ilustrace 4.2
Rozhodněte o absolutní, resp. neabsolutní konvergenci řady

$$\sum_{n=1}^{\infty}(-1)^{n-1}\frac{1}{\ln^n(n+1)} \tag{4.2}$$

Vidíme, že jde o alternující řadu. Budeme tedy vyšetřovat konvergenci řady

$$\sum_{n=1}^{\infty}\left|(-1)^{n-1}\frac{1}{\ln^n(n+1)}\right| = \sum_{n=1}^{\infty}\frac{1}{\ln^n(n+1)}.$$

Tento úkol již dobře známe. Vidíme, že ve výrazu se vyskytuje n-tá mocnina, což indikuje použití Odmocninového kriteria (procedury `odmock`):

```
> rada:=1/(ln(n+1))^n;
```

$$rada := \frac{1}{\ln(n+1)^n}$$

```
> odmock(rada);
```

$$\sum_{n=1}^{\infty}\frac{1}{\ln(n+1)^n}$$

konverguje

Řada $\sum_{n=1}^{\infty}\left|(-1)^{n-1}\frac{1}{\ln^n(n+1)}\right|$ konverguje, což podle Definice 4.2 znamená, že řada (4.2) konverguje absolutně.

Poznámka 4.3
Na závěr tohoto odstavce si ještě řekněme, jaký význam má vlastnost absolutní konvergence. Význam spočívá v platnosti komutativního zákona. Zatímco pro platnost asociativního a distributivních zákonů stačí, aby daná řada konvergovala (je jedno, jestli absolutně či neabsolutně), komutativní zákon platí pouze pro řady absolutně konvergentní. Pro řady neabsolutně konvergentní platí Riemannova věta (viz [1, str. 30]), která říká, že neabsolutně konvergentní řadu lze přeskládat tak, že může konvergovat k libovolnému předem danému součtu, určitě divergovat nebo i oscilovat. Tento problém přerovnávání řad je velmi zdařile zpracován s pomocí Maplu v [8].

4.3 Příklady

PŘÍKLAD 4.1
Rozhodněte o konvergenci, resp. divergenci řady

$$\sum_{n=1}^{\infty} \frac{(-1)^n n}{5n-2}.$$

Řešení:
Jde o alternující řadu $\sum(-1)^n a_n$, kde $a_n = \frac{n}{5n-2}$. Použijeme tedy Leibnizovo kriterium:

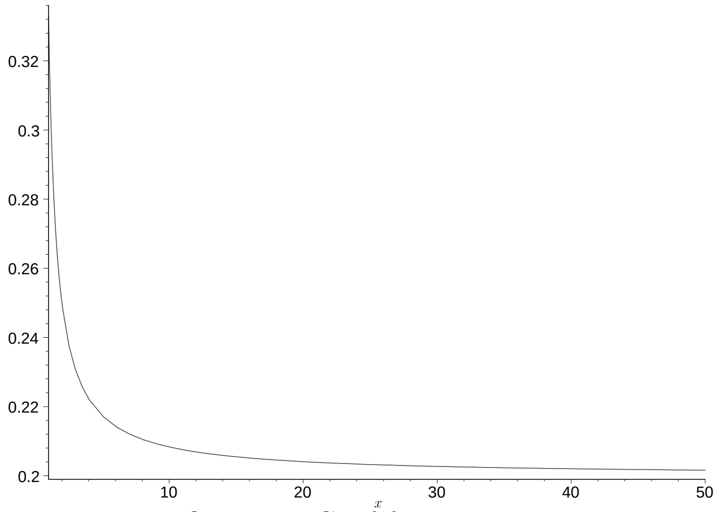
```
> rada:=n/(5*n-2);
```

$$rada := \frac{n}{5\,n-2}$$

```
> Limit(rada,n=infinity)=limit(rada,n=infinity);
```

$$\lim_{n\to\infty} \frac{n}{5\,n-2} = \frac{1}{5}$$

```
> plot(subs(n=x,rada),x=1..50);
```



Obr. 4.2. Graf funkce $\frac{x}{5x-2}$

Z výpočtu limity je vidět, že není splněna podmínka $\lim_{n\to\infty} a_n = 0$ a tudíž podle Leibnizova kriteria daná řada diverguje.

Výsledek:
Řada $\sum_{n=1}^{\infty} \frac{(-1)^n n}{5n-2}$ diverguje.

PŘÍKLAD 4.2
Vyšetřete, zda diverguje, nebo konverguje absolutně či neabsolutně řada

$$\sum_{n=3}^{\infty} (-1)^n \frac{\ln n}{n}.$$

Řešení:
V příkladech, kde máme rozhodnout o divergenci, resp. absolutní, či neabsolutní konvergenci, rozhodujeme nejprve o absolutní, resp. neabsolutní konvergenci. Vyjde-li totiž, že řada $\sum |a_n|$ konverguje (tj. zadaná řada $\sum a_n$ konverguje absolutně), pak odtud již plyne konvergence zadané řady $\sum a_n$ a odpadá tedy rozhodování mezi konvergencí a divergencí řady $\sum a_n$.

Nejprve tedy rozhodneme o konvergenci, resp. divergenci řady

$$\sum_{n=3}^{\infty} \left| (-1)^n \frac{\ln n}{n} \right| = \sum_{n=3}^{\infty} \frac{\ln n}{n}$$

Využijeme Srovnávacího kriteria a budeme porovnávat s divergentní řadou $\sum \frac{1}{n}$. Pro názornost porovnání si opět můžeme vykreslit grafy příslušných funkcí (viz Obr. 4.3.):

```
> rada:=ln(n)/n;
```

$$rada := \frac{\ln(n)}{n}$$

```
> srovk(rada);
```

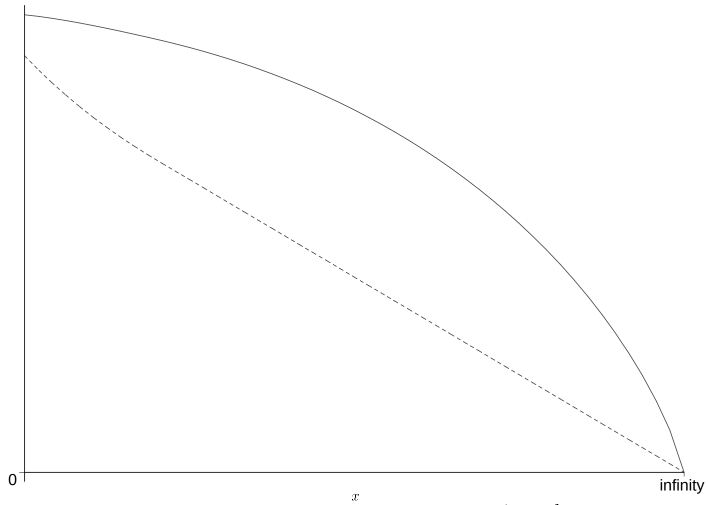
$$\sum_{n=1}^{\infty} \frac{\ln(n)}{n}$$

$$diverguje$$

```
> f:=subs(n=x,rada);
```

$$f := \frac{\ln(x)}{x}$$

```
> p1:=plot(1/x,x=3..infinity,linestyle=2):
> p2:=plot(f,x=3..infinity):
> display({p1,p2});
```



Obr. 4.3. Porovnání funkcí $\frac{1}{x}$ a $\frac{\ln x}{x}$

Dostali jsme, že řada $\sum \frac{\ln n}{n}$ diverguje a zbývá tedy rozhodnout o konvergenci, resp. divergenci řady $\sum (-1)^n \frac{\ln n}{n}$. K tomu využijeme Leibnizova kriteria:

```
> solve(f > 0);
```

$$\text{RealRange}(\text{Open}(1), \infty)$$

```
> diff(f,x);
```

$$\frac{1}{x^2} - \frac{\ln(x)}{x^2}$$

```
> simplify(%);
```

$$-\frac{-1+\ln(x)}{x^2}$$

```
> solve(%<=0);
```

$$\mathrm{RealRange}(e, \infty)$$

```
> Limit(rada,n=infinity)=limit(rada,n=infinity);
```

$$\lim_{n\to\infty} \frac{\ln(n)}{n} = 0$$

Všechny podmínky Leibnizova kriteria jsou tedy splněny a proto daná řada konverguje.

Výsledek:
Řada $\sum_{n=3}^{\infty}(-1)^n\frac{\ln n}{n}$ konverguje neabsolutně.

PŘÍKLAD 4.3
Rozhodněte, zda diverguje, nebo konverguje absolutně či neabsolutně řada

$$\sum_{n=1}^{\infty}(-1)^n\frac{n^3}{2^n}.$$

Řešení:
Opět musíme nejprve rozhodnout o absolutní, resp. neabsolutní konvergenci řady. Problém tedy převedeme na rozhodnutí o konvergenci, resp. divergenci řady

$$\sum_{n=1}^{\infty}\left|(-1)^n\frac{n^3}{2^n}\right| = \sum_{n=1}^{\infty}\frac{n^3}{2^n}.$$

Pro rozhodnutí použijeme Odmocninového kriteria a s pomocí Maplu tedy dostáváme:

```
> rada:=n^3/2^n;
```

$$\mathit{rada} := \frac{n^3}{2^n}$$

```
> odmock(rada);
```

$$\sum_{n=1}^{\infty} \frac{n^3}{2^n}$$

$$\mathit{konverguje}$$

Dostáváme, že řada $\sum_{n=1}^{\infty} \frac{n^3}{2^n}$ konverguje, což podle Definice 4.2 značí absolutní konvergenci zadané řady. Vidíme, že v tomto případě již nemusíme ověřovat Leibnizovým kriteriem konvergenci zadané řady, neboť ta již plyne z konvergence absolutní.

Výsledek:
Řada $\sum_{n=1}^{\infty}(-1)^n \frac{n^3}{2^n}$ konverguje absolutně.

PŘÍKLAD 4.4
Rozhodněte o divergenci, resp. absolutní či neabsolutní konvergenci řady

$$\sum_{n=1}^{\infty}(-1)^{n-1}\frac{2^{n^2}}{n!}.$$

Řešení:
Jde opět o alternující řadu, tedy začneme znovu rozhodnutím o absolutní, resp. neabsolutní konvergenci řady. Platí:

$$\sum_{n=1}^{\infty}\left|(-1)^{n-1}\frac{2^{n^2}}{n!}\right| = \sum_{n=1}^{\infty}\frac{2^{n^2}}{n!}$$

a pro rozhodnutí o konvergenci, resp. divergenci této řady použijeme podílové kriterium, tedy proceduru `podilk`:

```
> rada:=2^(n^2)/n!;
```

$$rada := \frac{2^{(n^2)}}{n!}$$

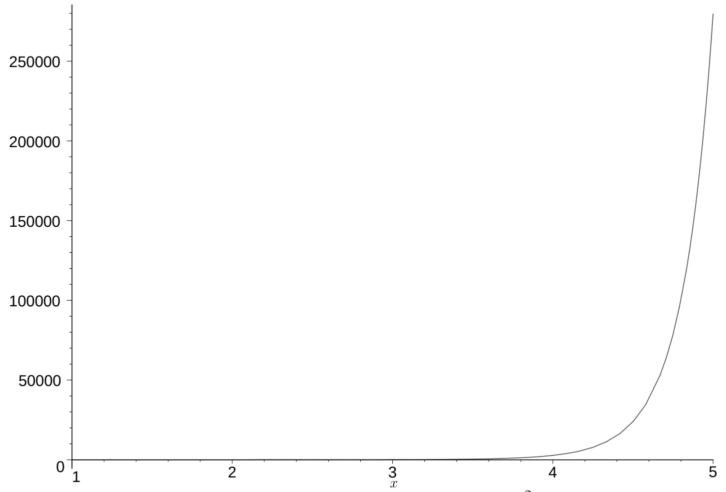
```
> podilk(rada);
```

$$\sum_{n=1}^{\infty} \frac{2^{(n^2)}}{n!}$$

$$\mathit{diverguje}$$

Zbývá tedy ještě vyšetřit konvergenci, resp. divergenci řady $\sum_{n=1}^{\infty}(-1)^{n-1}\frac{2^{n^2}}{n!}$ pomocí Leibnizova kriteria. Musíme tedy ověřit, že posloupnost $\left\{\frac{2^{n^2}}{n!}\right\}$ je nerostoucí posloupností kladných čísel a $\lim_{n\to\infty}\frac{2^{n^2}}{n!} = 0$. V tomto případě ověříme podmínky vykreslením grafu příslušné funkce $f = \frac{2^{x^2}}{x!}$:

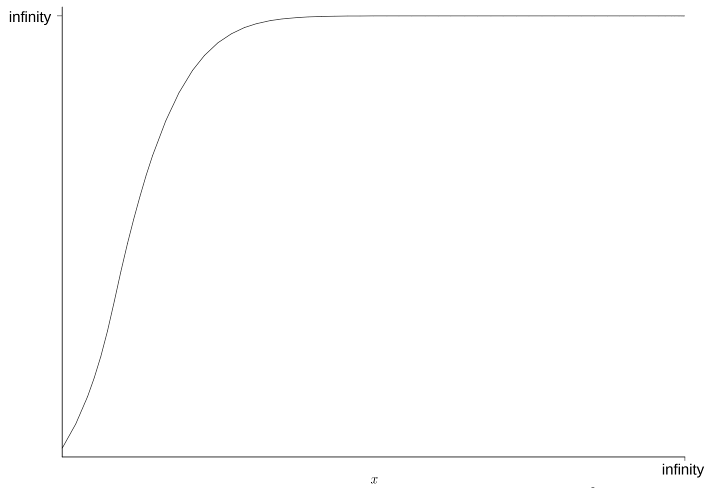
```
>   plot(subs(n=x,rada),x=1..5);
```



Obr. 4.4. Graf funkce $\frac{2^{x^2}}{x!}$

Ještě názornější bude vykreslit si nekonečný graf:

```
>   plot(subs(n=x,rada),x=1..infinity);
```



Obr. 4.5. Nekonečný graf funkce $\frac{2^{x^2}}{x!}$

Z obrázků je zřejmé, že je sice splněna podmínka $\frac{2^{n^2}}{n!} > 0$ pro $\forall n \in \mathrm{N}$ avšak posloupnost $\left\{\frac{2^{n^2}}{n!}\right\}$ je rostoucí a neplatí ani $\lim_{n\to\infty} \frac{2^{n^2}}{n!} = 0$, tedy podle Leibnizova kriteria zadaná řada diverguje. Hodnotu limity si pomocí Maplu můžeme spočítat:

```
> Limit(rada,n=infinity)=limit(rada,n=infinity);
```

$$\lim_{n\to\infty} \frac{2^{(n^2)}}{n!} = \infty$$

Výsledek:
Řada $\sum_{n=1}^{\infty}(-1)^{n-1}\frac{2^{n^2}}{n!}$ diverguje.

Kapitola 5

Numerická sumace

5.1 Základní pojmy

V této kapitole se budeme zabývat konvergentními řadami a především odhady jejich zbytků při numerické sumaci. Součet s každé konvergentní řady $\sum_{n=1}^{\infty} a_n$ můžeme napsat ve tvaru

$$s = s_n + R_n,$$

kde $s_n = a_1 + a_2 + \ldots + a_n$ je n-tý částečný součet řady a

$$R_n = a_{n+1} + a_{n+2} + \ldots$$

se nazývá *zbytek po n-tém členu*. Tento zbytek vyjadřuje velikost chyby, které se dopustíme, jestliže aproximujeme součet řady jejím n-tým částečným součtem.

Věta 5.1
Nechť $\sum a_n$ *je řada a* $r_n = \sum_{k=n+1}^{\infty} a_k$ *je její zbytek po n-tém členu. Nechť* $\sum b_n$ *je konvergentní řada s nezápornými členy taková, že pro* $\forall n \in \mathrm{N}$ *platí* $|a_n| \leq b_n$. *Je-li* $R_n = \sum_{k=n+1}^{\infty} b_k$ *zbytek po n-tém členu řady* $\sum b_n$, *pak platí*

$$|r_n| \leq R_n.$$

Věta 5.2
Nechť $\{a_n\}_{n=0}^{\infty}$ *je nerostoucí posloupnost kladných čísel taková, že* $\lim a_n = 0$. *Pak pro zbytek po n-tém členu* R_n *alternující řady* $\sum (-1)^{n-1} a_n$ *platí*

$$|R_n| < a_{n+1}.$$

Věta 5.3
Nechť $\sum a_n$ je číselná řada, pro kterou platí

$$\left|\frac{a_{n+1}}{a_n}\right| \leq q < 1 \qquad \textit{pro všechna } n \in \mathrm{N}.$$

Pak pro zbytek R_n této řady platí

$$|R_n| \leq |a_n| \frac{q}{1-q}.$$

Věta 5.4
Nechť $\sum a_n$ je konvergentní řada s nezápornými členy. Nechť $a_n = f(n)$, kde f je nezáporná a nerostoucí funkce na intervalu $[1, \infty)$. Pak pro zbytek R_n řady $\sum a_n$ platí

$$R_n \leq \int_n^\infty f(x)dx.$$

Ilustrace 5.1
Kolik členů řady

$$\sum_{n=1}^{\infty} \frac{1}{n^4} \tag{5.1}$$

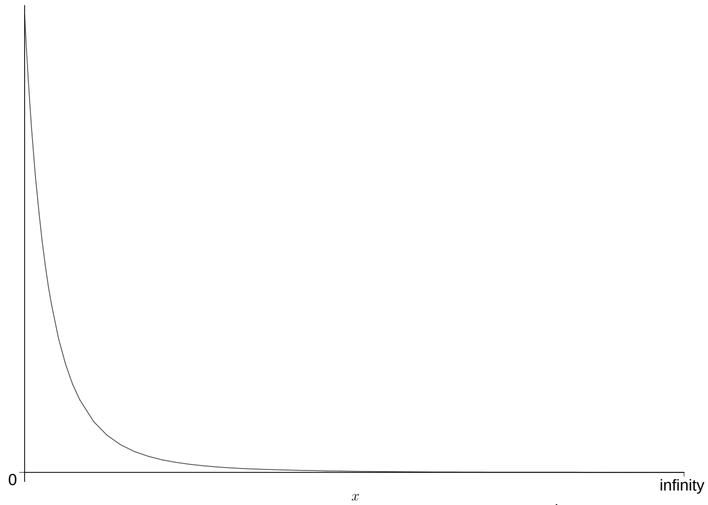
je třeba sečíst, aby byla chyba součtu menší než 0.01?

Musíme tedy nalézt odhad zbytku R_n řady (5.1) a vypočítat nejmenší přirozené n pro něž platí: $R_n < 0.01$. Pro odhad zbytku využijeme Větu 5.4. Musíme tedy nejprve ověřit její předpoklady, tzn. ověřit, že funkce $f(x) = \frac{1}{x^4}$ je nezáporná a nerostoucí na intervalu $[1, \infty)$. Vizuálně můžeme předpoklady ověřit vykreslením grafu:

```
> rada:=1/x^4;
```

$$rada := \frac{1}{x^4}$$

```
> plot(rada,x=1..infinity);
```



Obr. 5.1. Graf funkce $f(x) = \frac{1}{x^4}$.

Nyní ověříme předpoklady výpočtem. Nejprve nezápornost:

```
> solve(rada>0);
```

$$\text{RealRange}(-\infty,\ \text{Open}(0)),\ \text{RealRange}(\text{Open}(0),\ \infty)$$

nezáporná je tedy funkce $f(x)$ i na intervalu $[1, \infty)$. Nyní ještě určíme interval, na němž je $f(x)$ nerostoucí:

```
> diff(rada,x);
```

$$-4\,\frac{1}{x^5}$$

```
> solve(%<=0);
```

$$\text{RealRange}(0,\ \infty)$$

Na intervalu $[1, \infty)$ je tedy $f(x)$ i nerostoucí, jsou tedy splněny předpoklady Věty 5.4 a můžeme tedy odhadnout zbytek dané řady:

$$R_n \leq \int_n^\infty \frac{1}{x^4} dx$$

Výpočet příslušného integrálu provedeme pomocí Maplu:

```
> Int(1/x^4,x=n..infinity)=int(1/x^4,x=n..infinity);
```

$$\int_n^\infty \frac{1}{x^4}\,dx = \frac{1}{3}\,\frac{1}{n^3}$$

Tedy podle Věty 5.4 platí

$$R_n \leq \frac{1}{3n^3}$$

Nyní zbývá nalézt nejmenší $n \in \mathrm{N}$ splňující nerovnost

$$\frac{1}{3n^3} < 0.01$$

Výpočet provedeme vyřešením rovnice $\frac{1}{3n^3} = 0.01$ a zaokrouhlením výsledku na nejbližší vyšší přirozené číslo, resp. přičtením jedničky v případě, že výsledek bude přirozené číslo:

```
> fsolve(1/(3*n^3)=0.01);
```

$$3.218297949$$

```
> ceil(%);
```

$$4$$

Výsledek:
Je třeba sečíst aspoň 4 členy řady $\sum_{n=1}^{\infty} \frac{1}{n^4}$ aby chyba součtu byla menší než 0.01.

Ilustrace 5.2
Najděte odhad zbytku pro Leibnizovu řadu

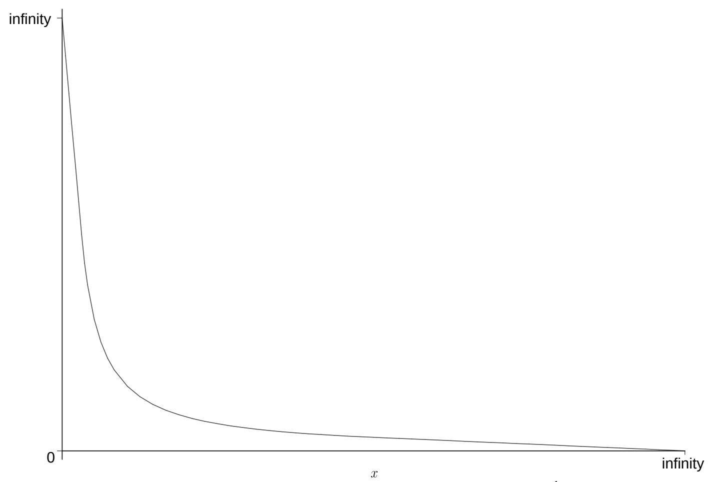
$$\sum_{n=1}^{\infty} (-1)^{n-1} \frac{1}{n} = \ln 2 \tag{5.2}$$

Označme R_n zbytek řady (5.2). Řada (5.2) je alternující řadou, pro řešení proto použijeme Větu 5.2. Musíme tedy nejprve ověřit, zda posloupnost $\left\{\frac{1}{n}\right\}_{n=0}^{\infty}$ je nerostoucí posloupností kladných čísel takovou, že $\lim_{n\to\infty} \frac{1}{n} = 0$. Výpočet budeme opět provádět s příslušnou funkcí, v tomto případě jde tedy o funkci $f(x) = \frac{1}{x}$. Pro vizuální ověření si opět můžeme vykreslit graf:

```
> rada:=1/x;
```

$$rada := \frac{1}{x}$$

```
> plot(rada,x=0..infinity);
```



Obr. 5.2. Graf funkce $f(x) = \frac{1}{x}$.

Výpočetně ověřujeme s pomocí Maplu následujícím způsobem:

```
> solve(rada>0);
```

$$\mathrm{RealRange}(\mathrm{Open}(0), \infty)$$

```
> simplify(diff(rada,x));
```

$$-\frac{1}{x^2}$$

```
> solve(%<=0);
```

$$n$$

```
> limit(rada,x=infinity);
```

$$0$$

Vidíme, že předpoklady Věty 5.2 jsou splněny a podle této věty pak platí

$$|R_n| < \frac{1}{n+1}$$

Z odhadu zbytku je vidět, že rychlost konvergence řady (5.2) je malá a proto tato řada není příliš vhodná pro výpočty logaritmů. Obecně platí, že čím je rychlost konvergence dané řady větší, tím je vhodnější pro aproximaci svého součtu.

5.2 Příklady

PŘÍKLAD 5.1
Kolik členů řady
$$\sum_{n=1}^{\infty} \frac{1}{n(n+1)(n+2)(n+3)}$$
je třeba sečíst, aby byl zbytek menší než 0.0001?

Řešení:
Využijeme Větu 5.1. Označme r_n zbytek řady $\sum_{n=1}^{\infty} \frac{1}{n(n+1)(n+2)(n+3)}$. Za řadu $\sum b_n$ z Věty 5.1 vezmeme v tomto případě řadu (5.1) z Ilustrace 5.1, odkud již také známe její zbytek $R_n = \frac{1}{3n^3}$ a máme ověřeno, že jde o řadu s nezápornými členy. Stačí tedy ověřit, že je konvergentní a pro $\forall n \in \mathrm{N}$ platí:
$$\left| \frac{1}{n(n+1)(n+2)(n+3)} \right| < \frac{1}{n^4} \tag{5.3}$$
K ověření konvergence řady (5.1) použijeme Limitního srovnávacího kriteria, platnost vztahu (5.3) je zřejmá, proto její ověření ponechávám pouze na dobrovolné iniciativě čtenáře:

```
> rada:=1/n^4;
```
$$rada := \frac{1}{n^4}$$
```
> limsrovk(rada);
```
$$\sum_{n=1}^{\infty} \frac{1}{n^4}$$
$$konverguje$$

Podle Věty 5.1 pak platí
$$|r_n| \leq \frac{1}{3n^3}$$
a hledáme tedy nejmenší $n \in \mathrm{N}$, pro něž platí $\frac{1}{3n^3} < 0.0001$. Tuto nerovnost vyřešíme za pomoci Maplu stejně jako v Ilustraci 5.1:

```
> fsolve(1/(3*n^3)=0.0001);
```
$$14.93801582$$
```
> ceil(%);
```
$$15$$

Výsledek:
Aby byl zbytek řady $\sum_{n=1}^{\infty} \frac{1}{n(n+1)(n+2)(n+3)}$ menší než 0.0001, musíme sečíst nejméně 15 členů této řady.

PŘÍKLAD 5.2
Kolik členů řady

$$\sum_{n=2}^{\infty} \frac{1}{n \ln^2 n}$$

je třeba sečíst, aby byl zbytek menší než 0.0001?

Řešení:
Pro odhad zbytku využijeme Větu 5.4. Musíme tedy ověřit, že řada $\sum_{n=2}^{\infty} \frac{1}{n \ln^2 n}$ je konvergentní řadou s nezápornými členy. Konvergenci ověříme Integrálním kriteriem pomocí procedury `intk`:

```
> rada:=1/(n*(ln(n))^2);
```

$$rada := \frac{1}{n \ln(n)^2}$$

```
> intk(rada,2);
```

$$\sum_{n=2}^{\infty} \frac{1}{n \ln(n)^2}$$

$$konverguje$$

```
> solve(subs(n=x,rada)>=0);
```

$$\mathrm{RealRange}(\mathrm{Open}(0), \mathrm{Open}(1)), \mathrm{RealRange}(\mathrm{Open}(1), \infty)$$

Dále musíme ověřit, že funkce $f(x) = \frac{1}{x \ln^2 x}$ je nezáporná a nerostoucí. Nezápornost funkce f je však již ověřena posledním příkazem. Nyní tedy ještě ověříme, že funkce f je nerostoucí. Protože danou řadu sčítáme od indexu $n = 2$, stačí aby f byla nerostoucí na intervalu $[2, \infty)$:

```
> simplify(diff(subs(n=x,rada),x));
```

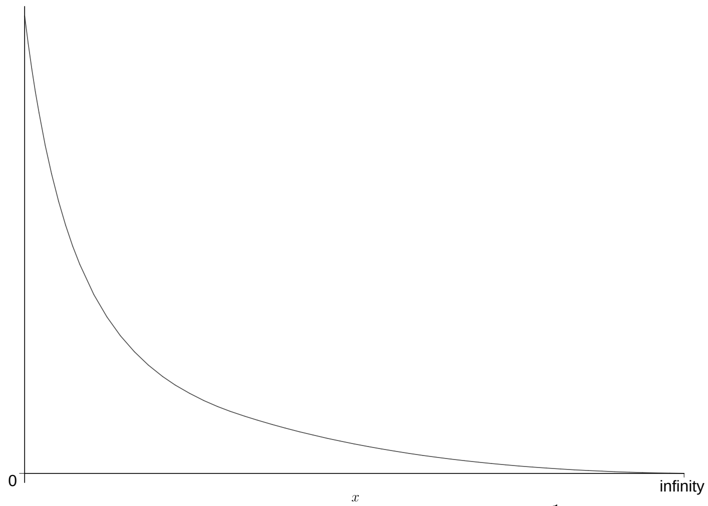
$$-\frac{\ln(x) + 2}{x^2 \ln(x)^3}$$

```
> solve(%<=0);
```

$$\mathrm{RealRange}(\mathrm{Open}(0), e^{(-2)}), \mathrm{RealRange}(\mathrm{Open}(1), \infty)$$

Vizuální ověření vlastností funkce f:

```
> plot(subs(n=x,rada),x=2..infinity);
```



Obr. 5.3. Graf funkce $f(x) = \frac{1}{x \ln x^2}$.

Pro odhad zbytku určíme hodnotu nevlastního integrálu $\int_n^\infty f(x)dx$. Dostáváme:

```
> 1/(x*(ln(x))^2);
```

$$\frac{1}{x\ln(x)^2}$$

```
> Int(%,x=n..infinity)=int(%,x=n..infinity);
```

$$\int_n^\infty \frac{1}{x\ln(x)^2}\,dx = \frac{1}{\ln(n)}$$

Podle Věty 5.4 tedy platí $R_n \leq \frac{1}{\ln n}$ a hledáme nejmenší $n \in \mathrm{N}$ splňující

$$\frac{1}{\ln n} < \frac{1}{10000}(= 0.0001).$$

Tuto nerovnost vyřešíme v Maplu tak, že z nerovnosti uděláme rovnost (na což již jsme zvyklí), ale ještě ji navíc kvůli názornějšímu výsledku umocníme na -1 a teprve pak ji vyřešíme.
Platí totiž $\frac{1}{\ln n} = \frac{1}{10000}$ právě když $\ln n = 10000$. Vyřešme tedy tuto rovnici:

```
> solve(ln(n)=10000);
```

$$e^{10000}$$

Abychom si byli jisti, že nerovnost $\frac{1}{\ln n} < \frac{1}{10000}$ bude ostrá, museli bychom výsledné číslo ještě zaokrouhlit nahoru, popř. zvětšit o jedničku Toto číslo je však obrovské, proto si pomocí Maplu pouze ukážeme jeho aproximaci v pohyblivé desetinné čárce:

```
> evalf(%);
```

$$.8806818226\, 10^{4343}$$

Výsledek:
Abychom se dopustili chyby menší než 0.0001 u řady $\sum_{n=}^{\infty} \frac{1}{n \ln^2 n}$, museli bychom sečíst alespoň k členů, kde $k > e^{10000}$.

PŘÍKLAD 5.3
Kolik členů řady

$$\sum_{n=1}^{\infty} \frac{1}{n2^n}$$

je třeba sečíst, abychom její součet aproximovali s chybou menší než 0.01?

Řešení:
K určení odhadu zbytku použijeme Větu 5.3. Musíme nejprve určit konstantu $q < 1$ takovou, že pro $\forall n \in \mathrm{N}$ platí

$$\left| \frac{\frac{1}{(n+1)2^{n+1}}}{\frac{1}{n2^n}} \right| \leq q \tag{5.4}$$

Konstantu q určíme s pomocí Maplu:

```
> rada:=1/(n*2^n);
```

$$rada := \frac{1}{n\, 2^n}$$

```
> pod:=simplify(subs(n=n+1,rada)/rada);
```

$$pod := \frac{1}{2} \frac{n}{n+1}$$

```
> q:=limit(pod,n=infinity);
```

$$q := \frac{1}{2}$$

```
> solve(abs(pod)<=q);
```

$$\mathrm{RealRange}(\frac{-1}{2}, \infty)$$

Pro $q = \frac{1}{2}$ je tedy splněn vztah (5.4) a podle Věty 5.3 pak pro zbytek R_n řady $\sum \frac{1}{n2^n}$ platí

$$|R_n| \leq \left|\frac{1}{n2^n}\right| \frac{\frac{1}{2}}{1-\frac{1}{2}}$$

Nyní zbývá určit nejmenší $n \in \mathrm{N}$ takové, že

$$|R_n| \leq 0.01$$

Tento úkol řešíme analogicky jako v Ilustraci 5.1:

```
> R[n]:=abs(rada)*((1/2)/(1-1/2));
```

$$R_n := \left|\frac{1}{n\,2^n}\right|$$

```
> ceil(fsolve(R[n]=0.01));
```

$$5$$

Výsledek:
Abychom aproximovali součet řady $\sum_{n=1}^{\infty} \frac{1}{n2^n}$ s chybou menší než 0.01, musíme sečíst nejméně 5 jejích členů.

Kapitola 6

Součin řad

6.1 Základní pojmy

V této kapitole se budeme zabývat součinem dvou nekonečných řad.

Definice 6.1
Nechť $\sum a_n, \sum b_n$ jsou řady, posloupnost $\{c_n\}$ je permutací množiny $\{a_i b_k; i,\ k \in \mathrm{N}\}$. Řadu $\sum c_n$ nazýváme *součinem řad* $\sum a_n, \sum b_n$.

Poznámka 6.1
Obecně existuje nekonečně mnoho různých součinů řad $\sum a_n, \sum b_n$, přičemž jeden z druhého vznikne přeřazením.

Věta 6.1
Nechť řady $\sum a_n = a$, $\sum b_n = b$ konvergují absolutně. Je-li $\{c_n\}$ libovolná posloupnost, jež je permutací množiny $\{a_i b_k; i, k \in \mathrm{N}\}$, pak řada $\sum c_n$ konverguje absolutně a platí

$$\sum c_n = a \cdot b.$$

Definice 6.2
Dirichletovým součinem řad $\sum a_n, \sum b_n$ rozumíme řadu $\sum c_n$, kde

$$c_n = a_1 b_n + a_2 b_n + \ldots + a_n b_n + a_n b_{n-1} + \ldots + a_n b_1.$$

Definice 6.3
Cauchyovým součinem řad $\sum a_n$, $\sum b_n$ rozumíme řadu $\sum c_n$, kde

$$c_n = a_1 b_n + a_2 b_{n-1} + \ldots + a_{n-1} b_2 + a_n b_1.$$

Věta 6.2
Nechť $\sum a_n = a$, $\sum b_n = b$ jsou konvergentní řady a nechť $\sum c_n$ je jejich Dirichletův součin. Pak $\sum c_n$ je konvergentní a platí

$$\sum c_n = a \cdot b.$$

Věta 6.3 (Mertensova)
Nechť řady $\sum a_n = a$, $\sum b_n = b$ konvergují a alespoň jedna z nich absolutně. Nechť $\sum c_n$ je Cauchyův součin těchto řad. Pak $\sum c_n$ konverguje a platí

$$\sum c_n = a \cdot b.$$

Ilustrace 6.1
Nechť $q \in \mathrm{R}, q > 1$. Určete Cauchyův součin řady $\sum_{n=1}^{\infty} \frac{1}{q^n}$ se sebou samou. Pomocí získaného výsledku určete součet řady

$$\sum_{n=1}^{\infty} \frac{n}{q^n} \tag{6.1}$$

Řešení:
Označme tedy $\sum_{n=1}^{\infty} c_n$ Cauchyův součin řady $\sum_{n=1}^{\infty} \frac{1}{q^n}$ se sebou samou. Pak podle Definice 6.3 dostáváme:

$$\begin{aligned}
c_1 &= \frac{1}{q} \cdot \frac{1}{q} = \frac{1}{q^2} \\
c_2 &= \frac{1}{q^2} \cdot \frac{1}{q} + \frac{1}{q} \cdot \frac{1}{q^2} = \frac{2}{q^3} \\
c_3 &= \frac{1}{q} \cdot \frac{1}{q^3} + \frac{1}{q^2} \cdot \frac{1}{q^2} + \frac{1}{q^3} \cdot \frac{1}{q} = \frac{3}{q^4} \\
&\vdots \\
c_n &= \frac{1}{q} \cdot \frac{1}{q^n} + \frac{1}{q^2} \cdot \frac{1}{q^{n-1}} + \ldots + \frac{1}{q^n} \cdot \frac{1}{q} = \frac{n}{q^{n+1}}
\end{aligned}$$

tedy

$$\sum_{n=1}^{\infty} c_n = \sum_{n=1}^{\infty} \frac{n}{q^{n+1}}$$

a k určení součtu této řady již můžeme využít Maplovský příkaz `sum`:

```
> Sum(n/q^(n+1),n=1..infinity)=sum(n/q^(n+1),n=1..infinity);
```

$$\sum_{n=1}^{\infty} \frac{n}{q^{(n+1)}} = \frac{1}{(-1+q)^2}$$

Pro automatizaci tohoto postupu lze použít doprogramovanou proceduru

`cauchys(an,bn,d,poc)`,

kde
`an` je $n-$tý člen a_n řady $\sum a_n$,
`bn` je $n-$tý člen b_n řady $\sum b_n$ a
`d` je přirozené číslo, udávající kolik prvních členů Cauchyova součinu si přejeme vypsat,
`poc` je nepovinný parametr (implicitně nastavený na 1), který udává počáteční index obou řad (z definice Cauchyova soucinu plyne, že pro obě řady musí být stejný).
Procedura vypíše prvních `d` členů Cauchyova součinu řad $\sum a_n$ a $\sum b_n$, jeho obecný $n-$tý člen a vrátí hodnotu Cauchyova součinu řad $\sum a_n$ a $\sum b_n$:

```
> cauchys(1/q^n,1/q^n,3);

c[1] = 1/q^2

c[2] = 2/q^3

c[3] = 3/q^4

c[n] = q^(-n-1)*n
```

$$\frac{1}{(-1+q)^2}$$

Zbývá tedy ještě určit součet řady $\sum_{n=1}^{\infty} \frac{n}{q^n}$. Tuto řadu můžeme rozepsat jako

$$\sum_{n=1}^{\infty} \frac{n}{q^n} = \sum_{n=1}^{\infty} \frac{q \cdot n}{q^{n+1}} = \sum_{n=1}^{\infty} q \cdot c_n = q \cdot \sum_{n=1}^{\infty} c_n$$

a dosazením výsledku Cauchyova součinu dostáváme:

$$\sum_{n=1}^{\infty} \frac{n}{q^n} = q \cdot \frac{1}{(q-1)^2} = \frac{q}{(q-1)^2}.$$

Všimněme si, že pro určení Cauchyova součinu by se v tomto příkladě též mohla použít Věta 6.3. Řada $\sum_{n=1}^{\infty} \frac{1}{q^n}$ je totiž pro $q > 1$ konvergentní geometrickou řadou (viz Připomenutí 2.1) a protože platí

$$\sum_{n=1}^{\infty} \left| \frac{1}{q^n} \right| = \sum_{n=1}^{\infty} \frac{1}{q^n}$$

je dokonce absolutně konvergentní, jsou tedy splněny předpoklady Věty 6.3 a Cauchyův součin jsme mohli spočítat také takto:

$$\sum_{n=1}^{\infty} c_n = \sum_{n=1}^{\infty} \frac{1}{q^n} \cdot \sum_{n=1}^{\infty} \frac{1}{q^n} = \frac{1}{q-1} \cdot \frac{1}{q-1} = \frac{1}{(q-1)^2}$$

Poznámka 6.2
Analogií procedury `cauchys` pro Dirichletův součin řad $\sum a_n$ a $\sum b_n$ je doprogramovaná procedura

`dirichs(an,bn,d)`,

jejíž parametry jsou shodné s procedurou `cauchys`. Jediný rozdíl spočívá v tom, že procedura `dirichs` vypisuje a vrací Dirichletův součin namísto Cauchyova.

Ilustrace 6.2
Určete Cauchyův součin řad

$$\sum_{n=0}^{\infty} \frac{x^n}{n} \text{ a } \sum_{n=0}^{\infty} \frac{y^n}{n} \tag{6.2}$$

Řešení:
Cauchyův součin těchto dvou řad určíme pomocí procedury `cauchys`. Protože zadané řady mají počáteční index nula, musíme použít nepovinný parametr:

```
> r1:=x^n/n!;
```

$$r1 := \frac{x^n}{n!}$$

```
> r2:=y^n/n!;
```

$$r2 := \frac{y^n}{n!}$$

```
> cauchys(r1,r2,3,0);
c[0] = 1
```

```
c[1] = y+x

c[2] = 1/2*y^2+x*y+1/2*x^2

c[n] = y^n/GAMMA(n+1)*((y+x)/y)^n
```

$$e^{(y+x)}$$

Cauchyův součin řad (6.2) je tedy $e^{(x+y)}$.

Kapitola 7

Zdrojové texty vlastních procedur

V této kapitole uvádím zdrojové texty procedur, které jsem vytvořil s pomocí Maplovského programovacího jazyka. U každé procedury si též připomeneme její význam, avšak jen velmi stručně, neboť podrobně již byla každá popsána v předchozím textu vždy při jejím prvním použití.

Procedura poslcass:
Vypisuje zadaný počet prvních členů posloupnosti částečných součtů, její $n-$tý člen a vrací součet zadané řady.

```
> poslcass:=proc(a,b,d)
> local i,j,s,e;
> s:=0; j:=0;
> for i from b to d+b-1 do
> j:=j+1;
> s:=s+eval(subs(n=i,a));
> lprint(evaln(s[j])=s);
> od; e:=sum(a,n=b..n);
> lprint(evaln(s[n])=e);
> Sum(a,n=b..infinity)=limit(e,n=infinity);
> end;
```

Procedura sumplots:
Vykresluje graf posloupnosti částečných součtů a její limitu v případě, že existuje a je vlastní. Procedura je (stejně jako následující procedura `sumplot`) naprogramována s využitím zdrojového textu z [5].

```
> sumplots:=proc(rada)
> local term,n,a,b,psum,m,points,i,c,sn,p1,p2;
> if nargs=2 then c:=args[2] else c:=0; fi;
> if typematch(rada,'Sum'(term::algebraic,
> n::name=a::integer..b::integer))
> then psum:=evalf@unapply(Sum(term, n=a..(a+m-1)),m);
> points:=[seq([[i,psum(i)+c],[i+1,psum(i)+c]],
> i=1..(b-a+1))];
> points:=map(op,points);
> p1:=PLOT(CURVES(points),AXESLABELS(n,"s[n]"));
> sn:=evalf(sum(term,n=a..infinity));
> else ERROR("expecting a Sum structure as input");
> fi;
> if sn<infinity then p2:=plot(sn,n=a..b,linestyle=4);
> display({p1,p2});
> else p1; fi;
> end;
```

Procedura sumplot:
Vykresluje graf posloupnosti částečných součtů. Používá se v případech, kdy danou řadu pomocí Maplu nelze sečíst a předchozí procedura `sumplots` se tudíž nedá použít.

```
> sumplot:=proc(rada)
> local term,n,a,b,psum,m,points,i,c,sn,p1,p2;
> if nargs=2 then c:=args[2] else c:=0; fi;
> if typematch(rada,'Sum'(term::algebraic,
> n::name=a::integer..b::integer))
> then psum:=evalf@unapply(Sum(term, n=a..(a+m-1)),m);
> points:=[seq([[i,psum(i)+c],[i+1,psum(i)+c]],
> i=1..(b-a+1))];
> points:=map(op,points);
> PLOT(CURVES(points),AXESLABELS(n,"s[n]"));
> else ERROR("expecting a Sum structure as input");
> fi;
> end;
```

Procedura kvocgeom:
Určuje kvocient q a člen a geometrické řady $\sum_{n=1}^{\infty} aq^{n-1}$.

```
> kvocgeom:=proc(rada)
```

```
> local i;
> global operat, kvoc, b, a;
> b:=expand(rada);
> if type(b,`+`) then operat:=nops(b);
> for i from 1 to operat do
> kvoc[i]:=simplify(subs(n=n+1,op(i,b))/op(i,b));
> a[i]:=simplify(op(i,b)/(kvoc[i])^(n-1));
> lprint(evaln(kvoc[i])=kvoc[i]); lprint(evaln(a[i])=a[i]);
> od
> else operat:=1;
> kvoc:=simplify(subs(n=n+1,b)/b);
> a:=simplify(b/kvoc^(n-1));
> lprint('kvoc'=kvoc); lprint('a'=a);
> fi;
> end;
```

Procedura geom:
Počítá součet geometrické řady.

```
> geom:=proc(k,fclen)
> local s,i;
> s:=0;
> if operat>1 then for i from 1 to operat do
> s:=s+fclen[i]/(1-k[i]);
> od else s:=fclen/(1-k);
> fi;
> end;
```

Procedura cauchys:
Vypíše zadaný počet prvních členů Cauchyova součinu dvou zadaných řad, $n-$tý člen a vrátí hodnotu Cauchyova součinu.

```
> cauchys:=proc(a,b,d)
> local j, poc;
> global c;
> if nargs=4 then poc:=args[4] else poc:=1; fi;
> for j from poc to d+poc-1 do
> c[j]:=simplify(sum(subs(n=i,a)*(subs(n=j-i+poc,b)),
>     i=poc..j));
> lprint(evaln(c[j])=c[j]);
> od;
```

```
> c[n]:=simplify(sum(subs(n=i,a)*(subs(n=n-i+poc,b)),
>     i=poc..n));
> lprint(evaln(c[n])=c[n]);
> sum(c[n],n=poc..infinity);
> end;
```

Procedura dirichs:
Vypíše zadaný počet prvních členů Dirichletova součinu dvou zadaných řad, $n-$tý člen a vrátí hodnotu Dirichletova součinu.

```
> dirichs:=proc(a,b,d)
> local j, poc;
> global c;
> if nargs=4 then poc:=args[4] else poc:=1; fi;
> c[poc]:=(subs(n=poc,a))*(subs(n=poc,b));
> lprint(evaln(c[poc])=c[poc]);
> for j from poc+1 to d+poc-1 do
> c[j]:=simplify(sum((subs(n=i,a))*(subs(n=j,b))
>   +(subs(n=j,a))*(subs(n=i,b)),
>   i=poc..j-1)+((subs(n=j,a))*(subs(n=j,b))));
> lprint(evaln(c[j])=c[j]);
> od;
> c[n]:=simplify(sum((subs(n=i,a))*b+a*(subs(n=i,b)),
>   i=poc..n-1)+a*b);
> lprint(evaln(c[n])=c[n]);
> sum(c[n],n=poc..infinity);
> end;
```

Procedura kochkriv:
Vykresluje $n-$tou iteraci Kochovy křivky.

```
> kochkriv:=proc(n)
> local i,j,x,y,dx,dy,uy;
> global kres,s,pomsez;
> uy:=evalhf(sqrt(1/12));
> s:=[x,y],[x+dx/3,y+dy/3],[x+dx/2-uy*dy,
>   y+dy/2+uy*dx],[x+(2/3)*dx,y+(2/3)*dy];
> kres[1]:=[0,0],[1,0];
> pomsez:=[0,0],[1,0];
> for i from 2 to n do dx:=pomsez[2,1]-pomsez[1,1];
> dy:=pomsez[2,2]-pomsez[1,2];
```

```
> x:=pomsez[1,1]; y:=pomsez[1,2];
> kres[i]:=s;
> for j from 2 to nops([pomsez])-1 do
> x:=pomsez[j,1]; y:=pomsez[j,2];
> dx:=pomsez[j+1,1]-x; dy:=pomsez[j+1,2]-y;
> kres[i]:=kres[i],s;
> od;
> pomsez:=kres[i],[1,0]; kres[i]:=pomsez;
> od;
> PLOT(CURVES([kres[n]]),SCALING(CONSTRAINED),
>   AXESSTYLE(NONE));
> end;
```

Procedura kochkranim:
Ukazuje animaci Kochovy křivky od první do $n-$té iterace.

```
> kochkranim:=proc(n)
> local x,y,dx,dy,uy,i,j;
> global kres,pl,pomsez,s,kresl;
> uy:=evalhf(sqrt(1/12));
> s:=[x,y], [x+dx/3,y+dy/3], [x+dx/2-uy*dy,y+dy/2+uy*dx],
>   [x+(2/3)*dx,y+(2/3)*dy];
> kres[1]:=[0,0],[1,0];
> pl[1]:=PLOT(CURVES([kres[1]]));
> pomsez:=[0,0],[1,0];
> for i from 2 to n do
> dx:=pomsez[2,1]-pomsez[1,1];
> dy:=pomsez[2,2]-pomsez[1,2];
> x:=pomsez[1,1]; y:=pomsez[1,2];
> kres[i]:=s;
> for j from 2 to nops([pomsez])-1 do
> x:=pomsez[j,1]; y:=pomsez[j,2];
> dx:=pomsez[j+1,1]-x; dy:=pomsez[j+1,2]-y;
> kres[i]:=kres[i],s;
> od;
> pomsez:=kres[i],[1,0];
> kres[i]:=PLOT(CURVES([pomsez]));
> pl[i]:=kres[i];
> od;
> for i to n do kresl[i]:=plots[display]([pl[i]]) od;
```

```
> i:='i';
> plots[display]([kresl[i]$i=1..n],axes=none,
>   insequence=true, scaling=constrained);
> end;
```

Procedura kochvloc:
Je naprogramována s využitím zdrojového textu, jehož autorem je Greg Fee ze Simon Fraser University. Procedura vykresluje $n-$tou iteraci Kochovy vlocky. Ve svém těle volá podproceduru next_Koch, jejíž zdrojový text je uveden ihned za procedurou `kochvloc`.

```
> kochvloc := proc(n)
> local k,e,d,ox,oy,i,j;
> global sez,konsez,x,y;
> e:=4; d:=e;
> x:=hfarray(1..e); y:=hfarray(1..e);
> x[1]:=0; x[2]:=.5; x[3]:=1; x[4]:=x[1];
> y[1]:=0; y[2]:=evalhf(sqrt(.75));
> y[3]:=y[1]; y[4]:=y[1];
> sez:=[x[1],y[1]];
> for k~from 2 to d do sez:=sez,[x[k],y[k]] od;
> if n=1 then konsez:=[sez];
> else for i from 2 to n do
> d:=e; ox:=hfarray(1..d); oy:=hfarray(1..d);
> for j to d do ox[j]:=x[j]; oy[j]:=y[j]; od;
> e:=4*d-3; x:=hfarray(1..e); y:=hfarray(1..e);
> evalhf(next_Koch(d,ox,oy,e,x,y));
> sez:=seq([x[j],y[j]],j=1..e);
> od;
> konsez:=[sez];
> fi;
> PLOT(CURVES(konsez),SCALING(CONSTRAINED),
>   AXESSTYLE(NONE));
> end;
> next_Koch := proc(d,ox,oy,e,x,y)
> local j,tr,ux,uy,i,px,py,qx,qy,dx,dy;
> j := 0; tr := evalhf(1/3);
> ux := .5; uy := evalhf(sqrt(.75));
> qx := ox[1]; qy := oy[1];
> for i from 2 to d do
```

```
> px := qx; qx := ox[i]; py := qy; qy := oy[i];
> j := j+1; x[j] := px; y[j] := py; j := j+1;
> dx := qx-px; dy := qy-py; dx := tr*dx;
> dy := tr*dy; x[j] := px+dx; y[j] := py+dy;
> x[j+1] := x[j]+ux*dx-uy*dy; x[j+2] := x[j]+dx;
> y[j+1] := y[j]+ux*dy+uy*dx; y[j+2] := y[j]+dy;
> j := j+2;
> od;
> j := j+1; x[j] := qx; y[j] := qy;
> end;
```

Procedura kochvlanim:
Ukazuje animaci Kochovy vločky od první do $n-$té iterace. Procedura je převzata z CD-ROM, přiloženého k [6], pouze s malými úpravami.

```
> kochvlocanim:=proc(n)
> local i,c,eliste,dliste,schritt,rekplot;
> eliste:=[[0,0],[1,0],[1/2,1/2*sqrt(3)],[0,0]]:
> dliste:=[[[0,0],[1,0],[1/2,1/2*sqrt(3)]]]:
> schritt:=proc(eliste,dliste)
> local ne,i,tp,s,satz,teilpunkte,
>   spitze,neliste,ndliste,nd;
> teilpunkte:=proc(anfangspunkt,endpunkt)
> local t1,t2;
> t1:=[op(1,anfangspunkt)+1/3*(op(1,endpunkt)-
>   op(1,anfangspunkt)), op(2,anfangspunkt)+
>   1/3*(op(2,endpunkt)-op(2,anfangspunkt))];
> t2:=[op(1,anfangspunkt)+2/3*(op(1,endpunkt)-
>   op(1,anfangspunkt)), op(2,anfangspunkt)+
>   2/3*(op(2,endpunkt) -op(2,anfangspunkt))];
> [t1,t2]
> end:
> spitze:=proc(t1,t2)
> local betrag;
> betrag:=sqrt((op(1,t2)-op(1,t1))^2+
>   (op(2,t2)-op(2,t1))^2);
> [(op(1,t1)+1/2*(op(1,t2)-op(1,t1)))+
>   1/2*sqrt(3)*(op(2,t2)-op(2,t1)),
>   (op(2,t1)+1/2*(op(2,t2)-op(2,t1)))-
>   1/2*sqrt(3)*(op(1,t2)-op(1,t1))];
```

```
> end:
> neliste:=[]:
> ndliste:=[]:
> for i from 1 to nops(eliste)-1 do
> tp:=teilpunkte(op(i,eliste),op(i+1,eliste)):
> s:=spitze(op(1,tp),op(2,tp)):
> satz:=[op(i,eliste),op(1,tp),s,op(2,tp)];
> neliste:=[op(neliste),op(satz)];
> nd:=[op(1,tp),s,op(2,tp)];
> ndliste:=[op(ndliste),nd]
> od;
> neliste:=[op(neliste),op(1,neliste)];
> [neliste,ndliste]
> end:
> rekplot:=proc(el,dl,ebene,tiefe)
> option remember;
> global a,b,d;
> if ebene>0 then
> b[tiefe-ebene+1]:=plots[polygonplot]
>   (el,style=PATCH);
> a[tiefe-ebene+1]:=plots[polygonplot]
>   (dl,style=PATCH);
> d[tiefe-ebene+1]:=plots[polygonplot]
>   (dl,style=PATCH);
> rekplot(op(1,schritt(el,dl)),op(2,schritt(el,dl)),
>   ebene-1,tiefe);
> fi
> end:
> rekplot(eliste,dliste,n,n);
> for i from 1 to n do c[i]:=plots[display]([b[i]])
> od:
> i:='i':
> plots[display]([c[i]$i=1..n], insequence=true,
>   axes=none, scaling=constrained):
> end;
```

Procedura sierpkob:
Vykresluje $n-$tou iteraci Sierpinského koberce.

```
> sierpkob:=proc(n)
```

```
> local x,y,d,i,j;
> global s,kr,sez,poms,kre,sq;
> s:=[x,y],[x+d,y],[x+2*d,y],[x,y+d],[x+2*d,y+d],
>   [x,y+2*d],[x+d,y+2*d],[x+2*d,y+2*d];
> kr:=POLYGONS([s[5],s[8],s[7],[x+d,y+d]]);
> x:=0; y:=0; d:=1/3;
> kre:=kr; sez:=s; poms:=sez;
> sq:=POLYGONS([[0,0],[1,0],[1,1],[0,1]],
>   COLOR(RGB,0,1,0));
> for i to n-1 do
> d:=d/3;
> for j to nops([sez]) do
> x:=sez[j,1]; y:=sez[j,2];
> poms:=poms,s; kre:=kre,kr;
> od;
> sez:=poms;
> od;
> PLOT(kre,sq,AXESSTYLE(NONE),SCALING(CONSTRAINED));
> end;
```

Procedura srovk:

Implementuje Srovnávací kriterium konvergence číselných řad s nezápornými členy. Implicitně je nastavena divergentní porovnávací řada na $\sum_{n=1}^{\infty} \frac{1}{n}$ a konvergentní na $\sum_{n=1}^{\infty} \frac{1}{n^2}$, avšak použitím nepovinných parametrů lze zadanou řadu porovnávat s libovolnou řadou.

```
> srovk:=proc(a)
> local b,c,kor,f;
> if nargs=3 then
> if args[3]='k' then b:=args[2];
> f:=simplify(b-a);
> kor:=ceil(max(fsolve(f=0,n),1));
> if evalf(subs(n=kor+1,f))>0 then
> print(Sum(a,n=1..infinity)); print(konverguje)
> else print(`porovnanim s touto
>   radou nelze rozhodnout`);
> fi;
> elif args[3]='d' then c:=args[2];
> f:=simplify(a-c);
> kor:=ceil(max(fsolve(f=0,n),1));
```

```
> if evalf(subs(n=kor+1,f))>0 then
> print(Sum(a,n=1..infinity)); print(diverguje)
> else print(`porovnanim s touto
>   radou nelze rozhodnout`);
> fi;
> else ERROR("treti parametr musi
>   byt 'k' nebo 'd'");
> fi;
> elif nargs<>1 then
> ERROR("chybny pocet parametru");
> else
> b:=1/n^2; c:=1/n; f:=simplify(b-a);
> kor:=ceil(max(fsolve(f=0,n),1));
> if evalf(subs(n=kor+1,f))>0 then
> print(Sum(a,n=1..infinity)); print(konverguje)
> else f:=simplify(a-c);
> kor:=ceil(max(fsolve(f=0,n),1));
> if evalf(subs(n=kor+1,f))>0 then
> print(Sum(a,n=1..infinity)); print(diverguje)
> else
> print(`timto kriteriem nelze rozhodnout`);
> fi; fi; fi;
> end;
```

Procedura limsrovk:
Implementuje Limitní srovnávací kriterium konvergence. Implicitní nastavení i použití nepovinných parametrů je shodné s předchozí procedurou `srovk`.

```
> limsrovk:=proc(a)
> local b,L;
> if nargs=1 then
> b:=1/n^2; L:=limit(a/b,n=infinity);
> if L<infinity then print(Sum(a,n=1..infinity));
> print(konverguje);
> else b:=1/n;
> L:=limit(a/b,n=infinity);
> if L>0 then print(Sum(a,n=1..infinity));
> print(diverguje);
> else print(`timto kriteriem nelze rozhodnout`);
> fi; fi;
```

```
> elif nargs<>3 then
> ERROR("chybny pocet parametru");
> elif args[3]='k' then b:=args[2];
> L:=limit(a/b,n=infinity);
> if L<infinity then print(Sum(a,n=1..infinity));
> print(konverguje);
> else print(`timto kriteriem nelze rozhodnout`);
> fi;
> elif args[3]='d' then b:=args[2];
> L:=limit(a/b,n=infinity);
> if L>0 then print(Sum(a,n=1..infinity));
> print(diverguje);
> else print(`timto kriteriem nelze rozhodnout`);
> fi;
> else ERROR("treti parametr musi
>   byt 'k' nebo 'd'");
> fi;
> end;
```

Procedura odmock:
Implementuje Odmocninové kriterium konvergence.

```
> odmock:=proc(a)
> local q;
> q:=evalf(limit(convert(surd(a,n),power),
>   n=infinity));
> if q<1 then print(Sum(a,n=1..infinity));
> print(konverguje)
> elif q>1 then print(Sum(a,n=1..infinity));
> print(diverguje)
> else print(`timto kriteriem nelze rozhodnout`);
> fi;
> end;
```

Procedura podilk:
Implementuje Podílové kriterium konvergence.

```
> podilk:=proc(a)
> local q;
> q:=evalf(limit(subs(n=n+1,a)/a,n=infinity));
> if q<1 then print(Sum(a,n=1..infinity));
```

```
> print(konverguje)
> elif q>1 then print(Sum(a,n=1..infinity));
> print(diverguje)
> else print(`timto kriteriem nelze rozhodnout`);
> fi;
> end;
```

Procedura limraabk:
Implementuje Limitní Raabeovo kriterium konvergence.

```
> limraabk:=proc(a)
> local q;
> q:=evalf(limit(n*(1-subs(n=n+1,a)/a),n=infinity));
> if q>1 then print(Sum(a,n=1..infinity));
> print(konverguje)
> elif q<1 then print(Sum(a,n=1..infinity));
> print(diverguje)
> else print(`timto kriteriem nelze rozhodnout`);
> fi;
> end;
```

Procedura intk:
Implementuje Integrální kriterium konvergence.

```
> intk:=proc(a)
> local r,b;
> if nargs=2 then b:=args[2];
> else b:=1;
> fi;
> r:=int(a,n=b..infinity);
> if r=infinity then print(Sum(a,n=b..infinity));
> print(diverguje)
> elif evalf(r)<infinity then
> print(Sum(a,n=b..infinity)); print(konverguje)
> else print(`timto kriteriem nelze rozhodnout`);
> fi;
> end;
```

Procedura kummerk:
Implementuje Kummerovo kriterium konvergence.

```
> kummerk:=proc(rada)
```

```
> local a,b,pod,k,kn;
> if nargs=1 then b:=n
> elif nargs<>2 then
> ERROR("chybny pocet parametru");
> else b:=args[2];
> fi;
> a:=rada;
> pod:=simplify(a/subs(n=n+1,a));
> kn:=simplify(b*(pod)-subs(n=n+1,b));
> k:=limit(kn,n=infinity);
> if k>0 then print(Sum(a,n=1..infinity));
> print(konverguje)
> elif k<0 then
> print(Sum(a,n=1..infinity)); print(diverguje)
> else
> print(`timto kriteriem nelze rozhodnout`);
> fi;
> end;
```

Procedura kritkonverg:
Procedura implementuje všechna výše uvedená kriteria konvergence. Je tedy z uvedených procedur implementujících kriteria konvergence nejuniverzálnější, proto je vhodná např. ke kontrole řešení.

```
> kritkonverg:=proc(a)
> local t,b,c,q,r,kor,f,pod,kn,k;
> t:=false;
> b:=1/n^2; c:=1/n;
> q:=limit(a/b,n=infinity);
> if q<infinity then print(Sum(a,n=1..infinity));
> print(konverguje); t:=true
> else q:=limit(a/c,n=infinity);
> if q>0 then print(Sum(a,n=1..infinity));
> print(diverguje); t:=true;
> fi; fi;
> if not t then
> q:=evalf(limit(convert(surd(a,n),power),
> n=infinity));
> if q<1 then print(Sum(a,n=1..infinity));
> print(konverguje); t:=true
```

```
> elif q>1 then print(Sum(a,n=1..infinity));
> print(diverguje); t:=true;
> fi; fi;
> if not t then
> q:=evalf(limit(subs(n=n+1,a)/a,n=infinity));
> if q<1 then print(Sum(a,n=1..infinity));
> print(konverguje); t:=true
> elif q>1 then print(Sum(a,n=1..infinity));
> print(diverguje); t:=true;
> fi; fi;
> if not t then
> q:=evalf(limit(n*(1-subs(n=n+1,a)/a),n=infinity));
> if q>1 then print(Sum(a,n=1..infinity));
> print(konverguje); t:=true
> elif q<1 then print(Sum(a,n=1..infinity));
> print(diverguje); t:=true;
> fi; fi;
> if not t then
> r:=int(a,n=1..infinity);
> if r=infinity then print(Sum(a,n=1..infinity));
> print(diverguje); t:=true
> elif evalf(r)<infinity then
> print(Sum(a,n=1..infinity));
> print(konverguje); t:=true
> fi; fi;
> if not t then f:=simplify(b-a);
> kor:=ceil(max(fsolve(f=0,n),1));
> if evalf(subs(n=kor+1,f))>0 then
> print(Sum(a,n=1..infinity)); print(konverguje);
> t:=true
> else f:=a-c; kor:=ceil(max(fsolve(f=0,n),1));
> if evalf(subs(n=kor+1,f))>0 then
> print(Sum(a,n=1..infinity)); print(diverguje);
> t:=true
> fi; fi;
> if not t then b:=n;
> pod:=simplify(a/subs(n=n+1,a));
> kn:=simplify(b*(pod)-subs(n=n+1,b));
> k:=limit(kn,n=infinity);
```

```
> if k>0 then print(Sum(a,n=1..infinity));
> print(konverguje)
> elif k<0 then
> print(Sum(a,n=1..infinity)); print(diverguje)
> else print(‘ Nelze rozhodnout o konvergenci resp.
>   divergenci teto rady nami pouzivanymi kriterii‘);
> fi; fi; fi;
> end;
```

Kapitola 8

Internetové zdroje

V poslední době se stává obrovským fenoménem světová počítačová síť Internet. Mimo jiné proniká také do výuky v čím dál větším počtu vzdělávacích institucí. Ty prostřednictvím Internetu zveřejňují nejrůznější výukové texty, domácí úkoly, testovací a demostrační programy a spoustu dalších informací. Internet je tedy vhodným prostředkem k doplnění a oživení výuky.

V této kapitole uvádím seznam několika zajímavých zdrojů, které lze využít při výuce nekonečných řad se systémem Maple. Internet je velice dynamický, rychle se rozrůstá, ale některé zdroje také zanikají. Je proto možné, že v době, kdy čtenář studuje tuto práci, některé uvedené zdroje už neexistují. Snažil jsem se však vybrat ty, které vypadaly poměrně stabilně (zpracováno ke dni 15.4.2001).

Domovská stránka systému Maple se nachází na adrese
`http://www.maplesoft.com/`

Bohaté možnosti využití Maplu pro různé partie matematiky demonstruje Aplikační centrum systému Maple, které lze najít na
`http://www.mapleapps.com/`

Domovskou stránku skupiny Symbolic Computation Group z University Waterloo, která Maple vyvinula najdeme na
`http://daisy.uwaterloo.ca/home.html`

Dále existuje spousta stránek, které seznamují se systémem Maple a se základy práce s ním. Uveďme pouze několik z nich:
`http://www.variercollege.qc.ca/Auxiliary/Mathematics/comp/getstart.htm`

http://www.fz-rossendorf.de/FVTK/NUTZESEMINARE/sem73.html
http://www.math.muni.cz/~plch/vyuka/vyuka.html
http://web.mit.edu/maple/www/maple.html

Úvod do numerické analýzy využívající Maple poskytuje adresa
http://www.cs.laurentian.ca/badams/numeric/intro/intro.html

Na Internetu lze také nalézt spoustu matematických serverů a skupin:
http://www.kfa-juelich.de/zam/zam/html - Central Institut for Applied Mathematics,
http://www.joma.org/articles/xiao/xiao4.html - interaktivní matematický server,
http://www.research.att.com/~njas/sequences/eisonline.html - online encyklopedie číselných posloupností
http://www.math.tamu.edu/~harold.boas/
courses.math696/index.html - Mathematical Communication and Technology. Na této stránce lze nalézt také nejrůznější kursy, např. Maple, LaTeX, Internet, Unix atd.

Nyní si uvedeme několik adres, týkajících se speciálně nekonečných řad s využitím systému Maple.

http://academics.smcvt.edu/gashline/111/111lab4.htm - zde lze nalézt cvičení na nekonečné řady,
http://www.mscs.uwstout.edu/~ellisconm/Webpage2.htm - na této stránce se nacházejí výukové texty a příklady na kriteria konvergence, podobně jako na stránce
http://www.uwec.edu/Academic/Math/Calculus/215-Fall2000/labs/
lab10/lab10.htm
http://www.wfu.edu/~ekh/mth112v4.html - nabízí maplovské zápisníky týkající se nekonečných řad a integrálů.

Výukové texty a testy z nekonečných řad, příp. dalších oblastí matematické analýzy lze nalézt např. na těchto stránkách:
http://www.mapps.com/powertools/postsecondary/html/06051.html
http://www.sosmath.com/calculus/calculus.html
http://archives.math.utk.edu/visual/calculus/6/index.html
http://www.simpson.edu/~math.labs/calcii.html
http://www.Ellerbruch.nmu.edu/Maple/Calculus.html
http://www.math.utah.edu/lab/ms/maple/examples.html
http://www.archives.math.utk.edu/utk.calculus/142toc.html

Závěr

V této diplomové práci jsem se snažil ukázat, jakým způsobem lze využít systém Maple při výuce tématu nekonečné řady.

Pro řešení příkladů jsem používal verzi Maple 6. Kromě standardních příkazů, které jsou uvedeny v první kapitole jsem využil i doprogramované procedury, které jsou uloženy v souboru `moproc.m` na přiložené disketě. Při programování jsem vycházel především z [5] a z nápovědy systému Maple.

Práci jsem vysázel v systému LaTeX. Při tvorbě práce jsem využil možnosti exportu maplovského zápisníku do LaTeXu a uložení grafů do PostScriptu, což mi umožnilo pozdější zpracování. Využil jsem také možnosti exportu maplovských zápisníků do formátu HTML.

Součástí této práce je také stručná webovská prezentace, záměrně však neuvádím její adresu, neboť s ukončením mého studia se stránka přesune. S největší pravděpodobností na stránky mého vedoucího diplomové práce RNDr. Romana Plcha Ph.D., nebo na stránky Fakulty informatiky, které se týkají diplomových prací.

Jsem rád, že jsem se díky této práci alespoň částečně naučil pracovat jak se systémem Maple, tak i se systémem LaTeX, což může být velkým přínosem pro mé budoucí uplatnění.

Zároveň jsem také rád, že jsem spolu s [8], [9] a dalšími diplomovými pracemi napomohl rozšíření výpočetní techniky do výuky matematiky.

Literatura

[1] Došlá Z., Novák V.: Nekonečné řady. MU, Brno, 1998.

[2] Fuchsová L.: Matematická analýza I. MU, Brno, 1992.

[3] Rybička J.: LaTeX pro začátečníky. Konvoj, Brno, 1999.

[4] Horová I.: Numerické metody. MU, Brno, 1999.

[5] Monagan M.B., Geddes K.O., Heal K.M., Labahn G., Vorkoetter S.M.: Programming guide Maple V Release 5. Springer Verlag, 1998.

[6] Westermann T., Buhmann W., Diemer L., Endres E., Laule M., Wilke G.: Mathematische Begriffe visualisiert mit Maple V für Lehrer und Dozenter. Springer, 1999.

[7] Fichtengolyc G.M.: Kurs differencialnogo i integralnogo isčislenija II. Nauka, Moskva, 1951.

[8] Kroutil P.: Diplomová práce Absolutní konvergence číselných řad a řady funkcí. MU, Brno, 1998.

[9] Chybová K.: Diplomová práce Diferenciální počet funkcí jedné proměnné s programem Maple V. MU, Brno, 1998.