

Diferenciální počet funkcí více proměnných v programu Mathematica

Differential Calculus of Functions of More Variables in
Mathematica

Jiří Grigar

Bakalářská práce
2013



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE (PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: Jiří GRIGAR
Osobní číslo: A10086
Studijní program: B3902 Inženýrská informatika
Studijní obor: Informační a řídicí technologie
Forma studia: prezenční

Téma práce: Diferenciální počet funkcí více proměnných v programu Mathematica

Zásady pro vypracování:

1. Uvedte přehled základních pojmů z teorie diferenciálního počtu funkcí více proměnných.
2. Seznamte se s příkazy programu Mathematica, které se používají při řešení úloh z diferenciálního počtu funkcí více proměnných.
3. Použití jednotlivých příkazů demonstруйте na vybraných interaktivních příkladech. Zaměřte se zejména na aplikace parciálních derivací.
4. Vytvořte notebooky s ukázkovými příklady, které budou sloužit studentům jako podpůrný materiál ke studiu diferenciálního počtu funkcí více proměnných.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. WEIR, Maurice D., Joel HASS, George B. THOMAS a Ross L. FINNEY. Thomas calculus. 11th ed. Boston: Pearson Addison Wesley, 2005. ISBN 0-321-48987-X.
2. OSTRAVSKÝ, Jan. Diferenciální počet funkce více proměnných. Nekonečné číselné řady. Vyd. 4., nezm. Zlín: Univerzita Tomáše Bati ve Zlíně, 2009. ISBN 978-80-7318-856-6.
3. DOŠLÁ, Zuzana a Ondřej DOŠLÝ. Diferenciální počet funkcí více proměnných. 3. vyd. Brno: Masarykova univerzita, 2006. ISBN 80-210-4159-5.
4. REKTORYS, Karel. Přehled užití matematiky I. 6. přeprac. vyd. Praha: Prometheus, 1995. ISBN 80-85849-92-5.
5. CHRAMCOV, Bronislav. Základy práce v prostředí Mathematica. Vyd. 1. Ve Zlíně: Univerzita Tomáše Bati ve Zlíně, 2005. ISBN 8073182688.

Vedoucí bakalářské práce:

Mgr. Jana Řezníčková, Ph.D.
Ústav matematiky

Datum zadání bakalářské práce:

24. února 2013

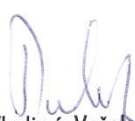
Termín odevzdání bakalářské práce:

14. června 2013

Ve Zlíně dne 24. února 2013


prof. Ing. Vladimír Vašek, CSc.
děkan




prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Bakalářská práce se zabývá užitím softwaru Wolfram Mathematica 8 při řešení vybraných úloh z diferenciálního počtu funkcí více proměnných. V teoretické části jsou k pochopení problematiky vysvětleny základní matematické definice spolu s názornými obrázky. Dále je uveden základní popis použitých příkazů softwaru, které se vyskytují v přiložených animacích. Praktická část se věnuje vytvoření interaktivních příkladů, které mohou sloužit studentům jako pomůcka při studiu této problematiky.

Klíčová slova: parciální derivace, tečná rovina, směrové derivace, absolutní extrémy, Wolfram Mathematica.

ABSTRACT

This thesis deals with the use of software Wolfram Mathematica 8 to solve some selected problems of the differential calculus of functions of more variables. The theoretical part explains the understanding of the basic mathematical definitions along with the illustrative images. There is also given the basic description of the software commands appearing in the attached animations. The practical part is devoted to the creation of interactive examples that can serve students as an aid in the study of this issue.

Keywords: partial derivatives, tangent plane, directional derivatives, absolute extremes, Wolfram Mathematica.

Děkuji Mgr. Janě Řezníčkové Ph.D., vedoucí mé bakalářské práce, za velmi cenné rady a připomínky, které mi v průběhu psaní i během pravidelných konzultací poskytla.

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 FUNKCE VÍCE PROMĚNNÝCH	11
1.1 REÁLNÁ FUNKCE N REÁLNÝCH PROMĚNNÝCH	11
1.2 GRAF FUNKCE	12
2 PARCIÁLNÍ DERIVACE	13
2.1 PARCIÁLNÍ DERIVACE 1. ŘÁDU	13
2.2 SMĚROVÉ DERIVACE	14
2.3 GRADIENT FUNKCE.....	15
3 DIFERENCIÁL FUNKCE	16
3.1 TEČNÁ ROVINA.....	16
3.2 NORMÁLA	16
3.3 TOTÁLNÍ DIFERENCIÁL	17
4 EXTRÉMY FUNKCÍ	18
4.1 STACIONÁRNÍ BOD	18
4.2 ABSOLUTNÍ EXTRÉMY	18
5 WOLFRAM MATHEMATICA	20
5.1 PROGRAMOVACÍ JAZYK	20
5.2 INTERAKTIVNÍ NÁPOVĚDA	21
5.3 PALETY NÁSTROJŮ	21
5.4 GRAFICKÉ FUNKCE	22
5.4.1 Plot3D	22
5.4.2 ParametricPlot3D	23
5.4.3 Graphics3D	23
5.4.4 Show.....	23
5.4.5 Manipulate.....	23
5.5 PODMÍNKY A CYKLY.....	24
5.5.1 Podmínka If.....	24
5.5.2 Switch.....	24
5.5.3 Cyklus For.....	25
5.5.4 Table.....	25
5.6 VYTVÁŘENÍ FUNKCÍ	25
II PRAKTICKÁ ČÁST	27
6 INTERAKTIVNÍ PŘÍKLADY	28
6.1 PARCIÁLNÍ DERIVACE.....	28
6.2 TEČNÁ ROVINA.....	31
6.3 SMĚROVÉ DERIVACE	32
6.4 ABSOLUTNÍ EXTRÉMY	34
6.4.1 Absolutní extrémy na kruhu.....	37
6.4.2 Absolutní extrémy na obdélníku	38
6.4.3 Absolutní extrémy na trojúhelníku.....	39

ZÁVĚR	40
ZÁVĚR V ANGLIČTINĚ.....	41
SEZNAM POUŽITÉ LITERATURY.....	42
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	43
SEZNAM OBRÁZKŮ	44
SEZNAM TABULEK.....	45
SEZNAM PŘÍLOH.....	46

ÚVOD

Bakalářská práce je rozdělena do dvou základních částí, a to teoretické a praktické. V teoretické části jsou vysvětleny základní pojmy z diferenciálního počtu funkcí více proměnných, které jsou doplněny o názorné obrázky. Práce se orientuje především na parciální derivace a jejich aplikace. V jedné z kapitol jsou také uvedeny základy programu Wolfram Mathematica 8, který byl při práci použit. Především jsou zde vysvětleny funkce a základní příkazy, které se vyskytují v přiložených interaktivních příkladech. Snahou bylo, aby všechna teorie korespondovala s praktickou částí. Práce se orientuje na čtenáře, kteří mají již základní znalosti z probírané problematiky, ale nemusejí znát programování v daném softwaru.

V rámci praktické části je pak vytvořeno 6 interaktivních příkladů (animací) z diferenciálního počtu funkcí více proměnných, které mohou také sloužit studentům k pochopení probírané látky. Při vytváření animací byl kladen důraz především na názornost, přehlednost a jednoduché ovládání. Proto mají téměř všechny stejná rozhraní, aby se v nich uživatel mohl lépe orientovat. Obrázek každé animace spolu s důkladným popisem se také nachází v druhé části této práce. Dále se můžeme setkat s popisem částí zdrojového kódu, který slouží pro lepší orientaci, pokud by si chtěl uživatel danou animaci např. rozšířit či změnit.

I. TEORETICKÁ ČÁST

1 FUNKCE VÍCE PROMĚNNÝCH

Při studování reálných jevů se obvykle setkáváme se závislostí na dvou nebo více proměnných. Proto je třeba si rozšířit základní představy o funkci jedné proměnné, abychom pochopili funkce více proměnných. I když zůstávají výpočetní pravidla v podstatě stejná, samotný výpočet je mnohem bohatší. Například s jejich diferenciálním počtem se můžeme setkat v různých aplikacích, jako jsou lineární programování, ekonomika (maximalizace zisku) apod. [7, s. 949]

1.1 Reálná funkce n reálných proměnných

Reálná funkce jedné reálné proměnné je definována jako předpis, podle kterého je každému reálnému číslu $x \in I, I \subseteq \mathbb{R}$, přiřazeno právě jedno reálné číslo $y \in \mathbb{R}$. Zobecněním předešlého je zobrazení z \mathbb{R}^n ($n \geq 2$) do \mathbb{R} , které nazýváme funkcí více proměnných. [2, s. 1]

Definice 1.1. Necht' $M \subseteq \mathbb{R}^n, n \geq 1, M \neq \emptyset$. Zobrazení

$$f: M \rightarrow \mathbb{R} \quad (1)$$

se nazývá **reálná funkce n reálných proměnných** a množina M se nazývá **definiční obor** této funkce a značí se D_f .

Z předchozí definice vyplývá, že každému bodu $x = (x_1, x_2, \dots, x_N) \in M$ je přiřazeno právě jedno reálné číslo $y \in \mathbb{R}$, kde ovšem záleží také na pořadí. Označení symbolem y budeme nazývat **závislou proměnnou** a x_1, x_2, \dots, x_N budou pak **nezávislé proměnné** nebo také argumenty funkce f . Množinu $f(M)$ hodnot y , které jsou přiřazeny jednotlivým bodům $x \in M$, nazýváme **obor hodnot funkce $f(x)$** a značíme H_f . [5, s. 20] Funkci více proměnných můžeme vidět například při výpočtu objemu rotačního válce

$$V = \pi r^2 h, \quad (2)$$

kde r je poloměr podstavy a h je výška válce. Poloměr i výška válce jsou nezávislé proměnné a objem je závislá proměnná. Formální zápis je

$$V = f(r, h). \quad (3)$$

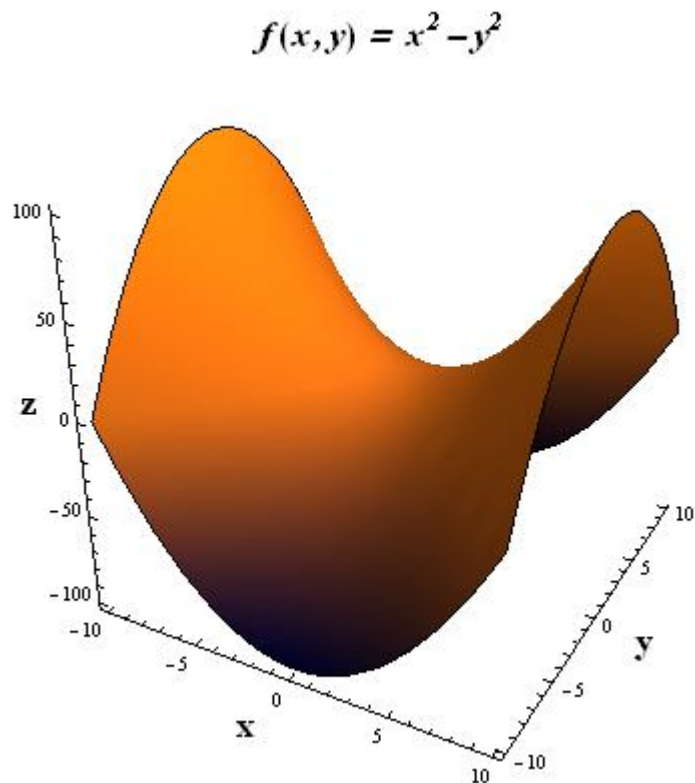
1.2 Graf funkce

Definice 1.2. Necht' f je funkce n proměnných definována na množině $M \subseteq \mathbb{R}^n, n \geq 1$.

Grafem G funkce f nazýváme množinu bodů

$$G(f) = \{[x, y] \in \mathbb{R}^{n+1}: x = (x_1, x_2, \dots, x_N) \in M, y = f(x)\}. \quad (4)$$

Pokud je $n = 2$ (2 nezávislé proměnné), jedná se o funkci dvou proměnných a grafem této funkce je množina bodů v trojrozměrném prostoru (3D graf). Při $n > 2$ ztrácí graf svou geometrickou názornost, jelikož zde naše představivost většinou končí. Příklad 3D grafu je vidět na Obr. 1.



Obr. 1. Vykreslení grafu zadané funkce $f(x, y) = x^2 - y^2$ v trojrozměrném prostoru.

2 PARCIÁLNÍ DERIVACE

Derivace je důležitým pojmem diferenciálního počtu. Opět vycházíme z funkce jedné proměnné $f: \mathbb{R} \rightarrow \mathbb{R}$, kde její derivace f' v bodě x_0 je definována jako limita

$$f'(x_0) = \lim_{x \rightarrow x_0} \frac{f(x) - f(x_0)}{x - x_0}. \quad (5)$$

Derivace funkce v bodě nám udává směrnici tečny ke křivce $y = f(x)$ v bodě $[x_0, f(x_0)]$, a také rychlost změny této funkce. Jestliže má funkce derivaci v bodě x_0 , je v tomto bodě spojitá, a tudíž zde existuje také limita funkce. [2, s. 24]

2.1 Parciální derivace 1. řádu

Limita funkce dvou proměnných je komplikovanější než v případě funkce jedné proměnné. Rychlost, s jakou se mění funkční hodnoty, závisí na směru, ve kterém se k danému bodu blížíme. K bodu se můžeme blížit mnoha způsoby, nicméně se zde budeme napřed zabývat situací, kdy se blížíme k danému bodu ve směru souřadných os x a y . Tím se dostáváme k zavedení pojmu **parciální derivace**. Při parciální derivaci se na jednu nezávislou proměnnou díváme jako na konstantu a derivujeme klasicky podle zbylé proměnné. Pro funkci n proměnných je situace analogická. [2, s. 24]

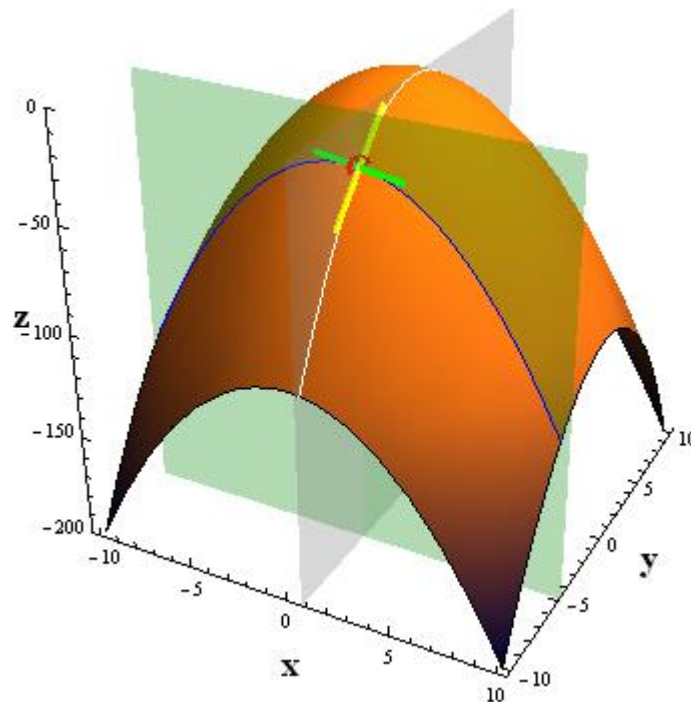
Definice 2.1. Necht' funkce $f: \mathbb{R}^2 \rightarrow \mathbb{R}$ je definována v bodě $[x_0, y_0]$ a nějakém jeho okolí. Položme $\varphi(x) = f(x, y_0)$. Má-li funkce $\varphi(x)$ derivaci v bodě x_0 , nazýváme tuto derivaci **parciální derivací** funkce f **podle proměnné x** v bodě $[x_0, y_0]$ a označujeme $f'_x(x_0, y_0)$, nebo také $\frac{\partial f}{\partial x}(x_0, y_0)$. Je definována jako limita

$$\frac{\partial f}{\partial x}(x_0, y_0) = \lim_{x \rightarrow x_0} \frac{\varphi(x) - \varphi(x_0)}{x - x_0} = \lim_{x \rightarrow x_0} \frac{f(x, y_0) - f(x_0, y_0)}{x - x_0}. \quad (6)$$

Položme $\psi(y) = f(x_0, y)$. Má-li funkce $\psi(y)$ derivaci v bodě y_0 , nazýváme tuto derivaci **parciální derivací** funkce f **podle proměnné y** v bodě $[x_0, y_0]$ a označujeme $f'_y(x_0, y_0)$, nebo také $\frac{\partial f}{\partial y}(x_0, y_0)$. Je definována jako limita

$$\frac{\partial f}{\partial y}(x_0, y_0) = \lim_{y \rightarrow y_0} \frac{\psi(y) - \psi(y_0)}{y - y_0} = \lim_{y \rightarrow y_0} \frac{f(x_0, y) - f(x_0, y_0)}{y - y_0}. \quad (7)$$

Geometrický význam parciální derivace funkce dvou proměnných x a y lze vidět na Obr. 2. Parciální derivace podle proměnné x v bodě $[x_0, y_0]$ je definována jako směrnice tečny ke křivce $\varphi(x)$ v bodě y_0 a je rovna tangensu úhlu, který svírá tato tečna s kladným směrem osy x . V tomto případě je y_0 zafixované a mění se pouze x . Analogicky je tomu u parciální derivace podle proměnné y , která je rovna směrnici tečny ke křivce $\psi(y)$ v bodě x_0 a odpovídá tangensu úhlu, který svírá tato tečna s kladným směrem osy y .



Obr. 2. Parciální derivace funkce v daném bodě podle nezávislých proměnných x a y .

2.2 Směrové derivace

Směrové derivace jsou zobecněním parciálních derivací a popisujeme jimi změnu dané funkce v bodě $[x_0, y_0]$ podél libovolné přímky určené směrovým vektorem \vec{u} . Ten je obvykle jednotkový, pokud tomu tak není, je možné daný vektor normovat. [2, s. 31]

Definice 2.2. Nechť f je funkce n proměnných, A je vnitřní bod D_f , $\vec{u} \in \mathbb{V}^n$, $g: \mathbb{R} \rightarrow \mathbb{R}$. Pro každé $t \in \mathbb{R}$ položme $g(t) := f(A + t\vec{u})$. Pak

$$f'_{\vec{u}}(A) := g'(0) = \lim_{t \rightarrow 0} \frac{g(t) - g(0)}{t} = \lim_{t \rightarrow 0} \frac{f(A + t\vec{u}) - f(A)}{t} \quad (8)$$

se nazývá **derivace funkce f v bodě A ve směru vektoru \vec{u}** .

Symbol \mathbb{V}^n označuje zaměření n rozměrného eukleidovského prostoru. V následující větě je uveden návod pro výpočet derivace funkce v bodě A ve směru vektoru \vec{u} .

Věta 2.1. Necht' $f(x, y)$ má v nějakém okolí bodu A všechny první parciální derivace, které jsou navíc v bodě A spojité. Pak platí

$$\frac{\partial f}{\partial \vec{u}}(A) = \frac{\partial f}{\partial x_1}(A) \cdot u_1 + \frac{\partial f}{\partial x_2}(A) \cdot u_2 + \dots + \frac{\partial f}{\partial x_n}(A) \cdot u_n, \quad (9)$$

kde $\vec{u} = (u_1, u_2, \dots, u_n)$.

2.3 Gradient funkce

Definice 2.3. Vektor všech parciálních derivací

$$\left(\frac{\partial f}{\partial x_1}(A), \frac{\partial f}{\partial x_2}(A), \dots, \frac{\partial f}{\partial x_n}(A) \right) \quad (10)$$

se nazývá **gradient** funkce f v bodě A a značí se $\text{grad } f(A)$ nebo také $\nabla f(A)$.

Pomocí gradientu můžeme vzorec pro výpočet derivace funkce v bodě A ve směru vektoru \vec{u} přepsat do stručnějšího tvaru:

$$\frac{\partial f}{\partial \vec{u}}(A) = \text{grad } f(A) \cdot \vec{u} = \nabla f(A) \cdot \vec{u}. \quad (11)$$

Z předchozího vzorce vyplývá, že se jedná o skalární součin dvou vektorů, tj.

$$\text{grad } f(A) \cdot \vec{u} = |\text{grad } f(A)| \cdot |\vec{u}| \cdot \cos \alpha, \quad (12)$$

kde α je úhel, který svírají vektory $\text{grad } f(A)$ a \vec{u} . Odtud plyne, že derivace je maximální, pokud $\alpha = 0^\circ$. Vektor $\text{grad } f(A)$ tedy určuje směr, ve kterém funkce f nejrychleji **roste**. Analogicky vektor $-\text{grad } f(A)$ určuje směr, ve kterém funkce f nejrychleji **klesá**.
[7, s. 993]

3 DIFERENCIÁL FUNKCE

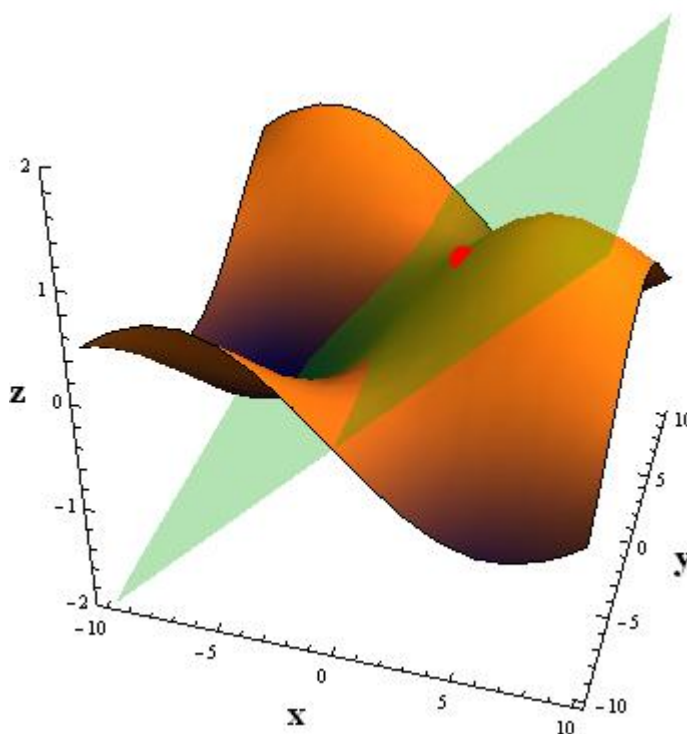
U funkce f jedné proměnné v bodě x_0 rozumíme diferencíalem **přírůstek funkce na tečně** vedené ke grafu funkce v bodě $[x_0, f(x_0)]$. U funkce n proměnných je totální diferenciál definován analogicky. Jedná se o přírůstek funkce na tečné rovině vedené ke grafu funkce bodem $x_0 \in \mathbb{R}^n$. [2, s. 37]

3.1 Tečná rovina

Věta 3.1. Necht' funkce f má definované a spojitě parciální derivace v okolí bodu $[x_0, y_0]$. Pak platí, že funkce f je v bodě $[x_0, y_0]$ spojitá a rovina o rovnici

$$z = f(x_0, y_0) + f'_x(x_0, y_0) \cdot (x - x_0) + f'_y(x_0, y_0) \cdot (y - y_0) \quad (13)$$

je **tečná rovina** ke grafu funkce f v bodě $[x_0, y_0, f(x_0, y_0)]$.



Obr. 3. Příklad tečné roviny ke grafu zadané funkce.

3.2 Normála

Pokud jsou splněny předpoklady předchozí věty, pak říkáme, že funkce f je diferencovatelná v bodě $[x_0, y_0]$. [6, s. 397]

Normála grafu funkce f v bodě $T = [x_0, y_0, f(x_0, y_0)]$ je přímka n kolmá k tečné rovině v bodě dotyku T . Směrový vektor této normály je roven normálovému vektoru tečné roviny. Parametrické vyjádření normály je pak

$$\begin{aligned}x &= x_0 + f'_x(x_0, y_0) \cdot t, \\y &= y_0 + f'_y(x_0, y_0) \cdot t, \\z &= f(x_0, y_0) - t.\end{aligned}\tag{14}$$

3.3 Totální diferenciál

Definice 3.3. Necht' funkce $f(x, y)$ dvou proměnných má spojité parciální derivace v bodě $[x_0, y_0]$. Pak výraz

$$df(x_0, y_0) = f'_x(x_0, y_0) \cdot dx + f'_y(x_0, y_0) \cdot dy\tag{15}$$

nazýváme **totální diferenciál** funkce f v bodě $[x_0, y_0]$. Funkce f se nazývá kmenová funkce tohoto diferenciálu.

Totální diferenciál lze použít např. k přibližným výpočtům funkčních hodnot. Přírůstky dx a dy z rovnice (15) zapisujeme také ve tvaru $dx = x - x_0$, $dy = y - y_0$. [2, s. 39]

4 EXTRÉMY FUNKCÍ

Vyšetřování extrémů funkcí je jednou z důležitých částí diferenciálního počtu. S extrémními úlohami se setkáváme i v běžném životě. Např. snažíme se maximalizovat zisk při co nejmenších nákladech (ekonomické rozhodování). O absolutních extrémech mluvíme tehdy, pokud máme najít na předepsané množině bod, kde funkce nabývá své nejmenší, resp. největší funkční hodnoty. [2, s. 64]

4.1 Stacionární bod

Definice 4.1. Necht' $f: \mathbb{R}^n \rightarrow \mathbb{R}$. Řekneme, že bod $A \in \mathbb{R}^n$ je **stacionární bod** funkce f , jestliže v bodě A existují všechny parciální derivace funkce f a platí

$$\frac{\partial f}{\partial x_i}(A) = 0, \quad i = 1, 2, \dots, n. \quad (16)$$

Funkce $f: \mathbb{R}^n \rightarrow \mathbb{R}$ může mít extrém pouze ve svém stacionárním bodě nebo v bodě, kde alespoň jedna z parciálních derivací neexistuje. [2, s. 65]

4.2 Absolutní extrémy

Definice 4.2. Necht' $f: \mathbb{R}^n \rightarrow \mathbb{R}$, $M \subset D_f$. Řekneme, že bod $A \in \mathbb{R}^n$ je bodem **absolutního maxima (minima)** funkce f na M , jestliže pro každý bod $X \in M$ platí

$$f(X) \leq f(A) \text{ pro absolutní maximum,} \quad (17)$$

$$f(X) \geq f(A) \text{ pro absolutní minimum.}$$

Jsou-li nerovnosti v těchto vztazích ostré, mluvíme o **ostrých absolutních maximech a minimech**. Absolutní maxima a minima nazýváme absolutní (globální) extrémy.

Věta 4.1. Necht' $M \subset \mathbb{R}^n$ je kompaktní množina (tj. uzavřená a ohraničená) a funkce $f: M \rightarrow \mathbb{R}$ je spojitá na M . Pak f nabývá svých absolutních extrémů buď v bodech lokálního extrému ležících uvnitř M nebo v některém hraničním bodě.

Při hledání absolutních extrémů funkce dvou proměnných postupujeme tak, že nejdříve najdeme stacionární body uvnitř dané množiny a poté vyšetříme hranici množiny. Tu rozdělíme na takové části, aby každou z nich bylo možno popsat funkcí jedné proměnné. Tento předpis pak dosadíme do zadané funkce dvou proměnných, čímž dostaneme funkci jedné proměnné. Pro tuto funkci pak derivací najdeme další body

podezřelé z extrému. Poté vypočteme funkční hodnoty ve stacionárních bodech (ležících uvnitř množiny) a v hraničních bodech podezřelých extrémů. Po vypočtení funkčních hodnot těchto bodů nalezneme bod s nejmenší a největší funkční hodnotou. Tyto body pak odpovídají absolutnímu minimu a maximu funkce na dané množině. [1]

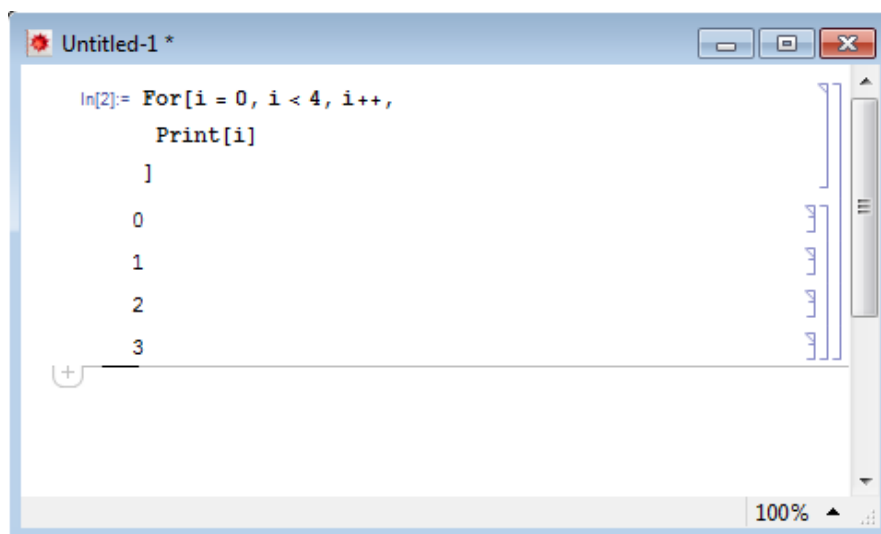
5 WOLFRAM MATHEMATICA

Wolfram Mathematica (dále jen Mathematica) je software vhodný nejen pro matematické operace, ale lze jej využít i v jiných odvětvích jako jsou automatizace, programování aj. Integruje v sobě nástroje pro numerickou a symbolickou matematiku, grafický a dokumentační systém a zajišťuje pokročilé propojení s dalšími aplikacemi. Sjednocuje tak všechny důležité prvky, které byly dříve dostupné jen odděleně. Před vytvořením tohoto softwaru existovaly programy pouze pro specifické úlohy, např. pro tvorbu grafů, matic, řešení rovnic apod. [3, s. 7]

5.1 Programovací jazyk

Programový kód nese název Mathematica a poskytuje silné programové vývojové prostředí. Programování probíhá v textovém dokumentu neboli notebooku (*.nb) a pro zpřehlednění je dále zdrojový kód rozdělen na tzv. „buňky“. Ty lze překládat všechny najednou nebo samostatně, rozdělovat či seskupovat. Překlad probíhá pomocí stisknutí klávesy *ENTER* (u numerického bloku), zkratkou *SHIFT+ENTER* (standardní) nebo klasicky přes menu programu v záložce *Evaluation – Evaluate Notebook*. Při psaní kódu není třeba rozlišovat typy proměnných, dimenze matic nebo řízení paměti. Proto je Mathematica vhodná i pro lidi, kteří se dříve s programováním vůbec neseťkali. Mohou tak začít ihned tvořit program bez programátorských znalostí. [3, s. 8]

Na Obr. 4 vidíme demonstraci programu, kde jako příklad uvádím cyklus *For*, v němž se v každé iteraci provádí tisk počítadla cyklů (proměnné *i*).



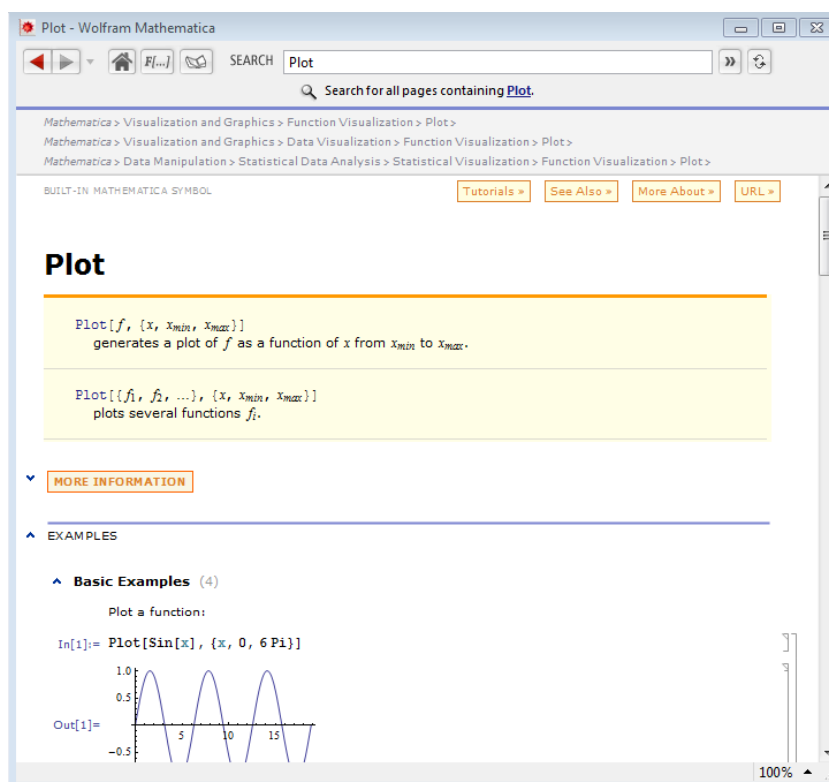
```
In[2]:= For[i = 0, i < 4, i++,  
Print[i]  
]  
0  
1  
2  
3
```

Obr. 4. Ukázka cyklu ve vývojovém prostředí Mathematica spolu s výstupem (tiskem) počítadla cyklů.

5.2 Interaktivní nápověda

Software obsahuje kompletní indexovanou dokumentaci pro všechny funkce (příkazy), které lze použít. Do nápovědy se dostaneme stisknutím klávesy *F1*. V ní lze jednoduše vyhledávat a studovat funkce spolu s interaktivními příklady. Na rozdíl od jiných programů lze tuto nápovědu upravovat a příklady překládat. Změny nejsou trvalé a po uzavření nápovědy dochází k obnově původních dat. [3, s. 8]

Nápověda je celá dostupná online, viz seznam literatury [8]. Příklad vyhledávání vidíme na Obr. 5, kde se zobrazuje nápověda pro vykreslení jednoduchého grafu (2D) spolu s příkladem použití.



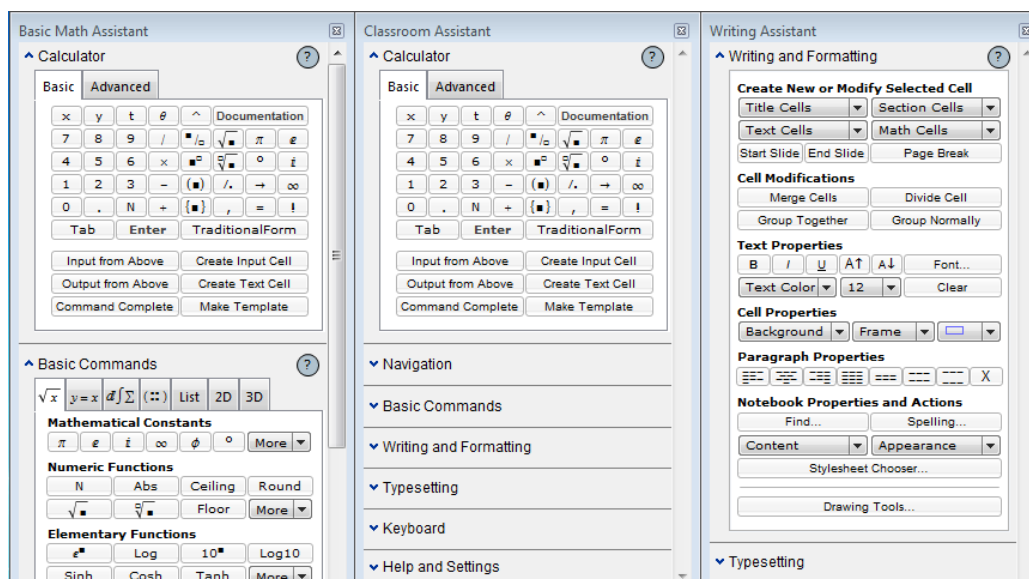
Obr. 5. Ukázka vyhledávání v nápovědě v programu Mathematica.

5.3 Palety nástrojů

Palety nástrojů umožňují jednoduchým způsobem využití jednotlivých funkcí bez znalosti programování či zadávání do programu Mathematica. Najdeme je v menu programu na záložce *Palettes*, kde máme na výběr ze tří možností:

- Basic Math Assistant**, která slouží pro matematické operace.
- Classroom Assistant**, která sjednocuje všechny palety do jedné.
- Writing Assistant**, která slouží k formátování textu.

Nejlepším řešením je použití druhé palety, která v sobě nese všechny možnosti ostatních palet. Zobrazení všech palet je na Obr. 6.



Obr. 6. Výřez palet v programu Mathematica.

5.4 Grafické funkce

5.4.1 Plot3D

Plot3D neboli trojrozměrný graf funkce dvou proměnných lze zadat do programu Mathematica následovně: $\text{Plot3D}[funkce, \{x, x_{min}, x_{max}\}, \{y, y_{min}, y_{max}\}]$. Vykreslení takového grafu vidíme v první kapitole na Obr. 1. U této funkce lze dalšími parametry nastavit způsoby vykreslení zadaného grafu, což uvádí Tab. 1.

Tab. 1 Základní nastavení příkazu Plot3D.

Příkaz	Význam
<i>Axes</i>	povolení vykreslení os
<i>AxesLabel</i>	popisky os
<i>Boxed</i>	ohrazení kolem grafu
<i>BoxRatios</i>	poměry vykreslených os
<i>ColorFunction</i>	barevné vykreslení plochy grafu
<i>Filling</i>	výplň pod plochou
<i>FillingStyle</i>	styl výplně pod plochou
<i>ImageSize</i>	velikost vykresleného grafu
<i>Mesh</i>	vykreslení mřížky
<i>PlotRange</i>	rozsah os
<i>PlotStyle</i>	styl vykreslení grafu

5.4.2 ParametricPlot3D

ParametricPlot3D je dalším zobrazením trojrozměrného grafu s tím rozdílem, že se jedná o parametrické vykreslení grafu. To znamená, že zadáváme souřadnice v osách x , y a z . Příkaz pro vykreslení je pak: `ParametricPlot3D[{fx, fy, fz}, {u, umin, umax}]`. Použití tohoto příkazu lze vidět v druhé kapitole na Obr. 2, kde pomocí něj vykresluji křivky parciálních derivací (žlutá, zelená). Parametry jsou obdobné jako u Plot3D, proto je zde uvádět nebudu (viz. Tab. 1).

5.4.3 Graphics3D

Graphics3D slouží pro vykreslení trojrozměrných objektů a je definován příkazem: `Graphics3D[objekty, nastavení]`. Pomocí něj lze např. v trojrozměrném grafu dokreslovat základní obrazce, jako jsou krychle, válec, koule, ale i šipka nebo jednoduché body. [8] Základní prvky použité v této bakalářské práci jsou v následující tabulce.

Tab. 2 Základní nastavení příkazu Graphics3D.

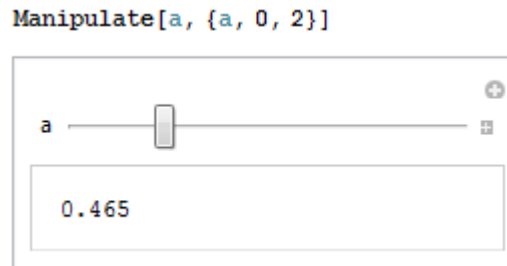
Příkaz	Význam
<i>Arrow</i>	vykreslení šipky
<i>Cylinder</i>	vykreslení válce
<i>Point</i>	vykreslení bodu, příp. bodů
<i>Polygon</i>	vykreslení obrazce
<i>Text</i>	vykreslení textu

5.4.4 Show

Show je příkaz, pomocí kterého lze do jednoho grafu vykreslit více grafických útvarů. Tyto útvary mohou být dvou nebo i tří rozměrné. [9, s. 139] Příkaz je definován: `Show[grafické útvary, nastavení]`. Použit je ve všech vytvořených animacích, kde kombinuji všechny předchozí příkazy.

5.4.5 Manipulate

Manipulate slouží k vykreslení interaktivních výrazů, kde vidíme změnu hodnoty ihned bez dalšího překladu. Integruje v sobě také možnost zadání klasických nebo zatrhávacích tlačítek, tvorbu roletového menu, dynamický výpis proměnných a mnoho dalších zajímavých prvků. [8] Jednoduchá aplikace spolu s definicí příkazu je na Obr. 7. Pohybem posuvníku se mění hodnota proměnné v intervalu od 0 do 2.



Obr. 7. Příklad příkazu Manipulate.

5.5 Podmínky a cykly

5.5.1 Podmínka If

Mezi základní rozhodovací podmínku patří If, která je definovaná následovně: `If[výraz, true, false]`. Ve většině případů je dobré uvádět jak příkazy pro pravdu (true), tak i pro nepravdu (false), jelikož může překladač zahlásit chybu. Pokud se nemá nic provádět, uvádí se obvykle prázdný vektor, který má tvar `{}`.

5.5.2 Switch

Switch neboli přepínač je další rozhodovací funkce, pomocí které lze jednoduše větvit program. Používá se tam, kde by bylo obtížné nebo nepřehledné použití příkazu If. Definice je pak: `Switch[výraz, hodnota1, co se má provést, hodnota2, co se má provést, ...]`. I pro programátory je v nápovědě [8] vysvětlení zadávání toho příkazu poněkud nepřehledné, proto zde uvedu dva menší příklady, které jsou na Obr. 8.

<code>a = 1;</code>	<code>a = 2;</code>
<code>Switch[a,</code>	<code>Switch[a,</code>
<code>1, Print["jedna"],</code>	<code>1, Print["jedna"],</code>
<code>2, Print["dva"]</code>	<code>2, Print["dve"]</code>
<code>]</code>	<code>]</code>
jedna	dve

Obr. 8. Dva příklady příkazu Switch.

Nevýhoda použití tohoto příkazu spočívá v tom, že standardně v něm nemáme možnost „default“, která je programátorům velice dobře známá. Jedná se o to, že pokud v rozhodovací podmínce není zahrnuta možná volba (např. $a=3$), tak příkaz Switch v Mathematice vypíše syntaxi zdrojového kódu (výstupem bude celý příkaz Switch). Proto

je nutné dodatečné ošetření další rozhodovací podmínkou, pokud víme, že tato možnost může nastat.

5.5.3 Cyklus For

K vytvoření jednoduchého cyklu je možné použití příkazu For, který je definován: `FOR[start, test, inkrementace, tělo cyklu]`. Cyklus začíná od hodnoty proměnné *start* a opakuje se až do vyhodnocení testovací podmínky, poté se ukončí. [9, s. 352] Příklad je v kapitole 5.1 na Obr. 4.

5.5.4 Table

Table je funkce obdobná příkazu For s tím rozdílem, že lze pomocí ní také vytvářet vektory, matice apod. Základní konstrukce je: `Table[výraz, {imax}]`. Při vytváření praktické části jsem se setkal s problémem, že v jednom případě nešlo použít příkaz For, ale musel jsem využít této funkce. Pro srovnání jsem vytvořil stejný příklad jako v předchozím případě, který je na Obr. 9. Jedná se o pouhý výpis (tisk) proměnné v závislosti na poslední udávané hodnotě (4).

```
Table[Print[i], {i, 4}];  
1  
2  
3  
4
```

Obr. 9. Příklad příkazu Table.

Pro vytvoření cyklu je možné dále použít příkazy While nebo Do. Mezi další rozhodovací funkce patří také Which. V mé práci jsem si vystačil pouze s příkazy výše uvedenými, proto zde nebudu další rozebírat, pro více informací doporučuji nápovědu programu. [8]

5.6 Vytváření funkcí

Vytváření funkcí lze v programu Mathematica zadávat mnoha způsoby. Uvedu zde pouze jeden z nejběžnějších, se kterým jsem si zcela vystačil. Definice funkce se provádí pomocí názvu funkce (secti), následují v hranatých závorkách vstupní proměnné s podtržítkem (x__y), dále přiřazovací symbol dvojtečka a rovná se (:=), za kterým se

nachází tělo funkce. Je vhodné přidat místo tohoto těla příkaz `Module`, pomocí kterého si lze definovat lokální proměnné a přidávat do něj celé tělo funkce. Zajišťujeme tak, že se nám nebudou přepisovat proměnné v rámci celého programu a bude zachována integrita dat. Jednoduchá funkce pro sečtení dvou čísel s následným zavoláním je na Obr. 10.

```
secti[x_, y_] := Module[{soucet},  
  soucet = x + y;  
  
  Return[soucet]  
]
```

```
secti[10, 2]
```

```
12
```

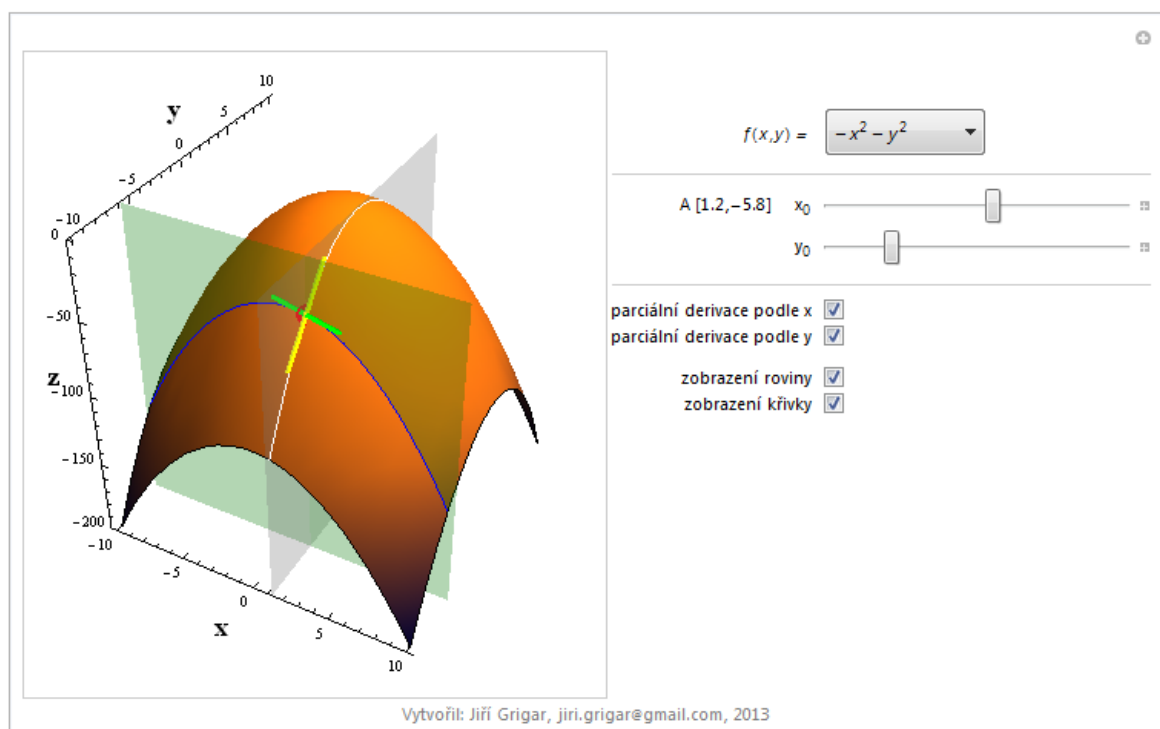
Obr. 10. Příklad funkce pro sečtení dvou čísel spolu s jejím zavoláním.

II. PRAKTICKÁ ČÁST

6 INTERAKTIVNÍ PŘÍKLADY

V praktické části bakalářské práce se budu zabývat popisem a objasněním důležitých pojmů při vytváření interaktivní příkladů (animací). Zaměřím se především na důležitá nastavení a popis zdrojového kódu. Všechny vytvořené animace korespondují se zavedením základních pojmů problematiky z teoretické části. Dbal jsem na to, aby téměř všechny uvedené příklady měly stejné rozhraní a uživatel se v nich mohl snáze orientovat. Menší popis každé animace je uveden i v souborech (*.nb) přiložených na CD.

6.1 Parciální derivace



Obr. 11. Animace parciální derivace podle nezávislých proměnných x , y .

První interaktivní notebook zobrazuje parciální derivaci podle proměnných x , y pro 4 vybrané funkce. Na výběr máme možnost zvolit bod derivace (A), parciální derivaci podle proměnné x nebo y , zobrazení roviny a křivky. Graf lze otáčet, posouvat i přibližovat. Změna výše uvedených možností se projeví ihned bez dalšího překladač.

Pokud klikneme na šipku v roletovém menu u zadané funkce, máme na výběr z těchto možností:

$$f(x, y) = x^2 - y^2, \quad (18)$$

$$f(x, y) = -x^2 - y^2, \quad (19)$$

$$f(x, y) = xy, \quad (20)$$

$$f(x, y) = \sin \frac{x}{4} \cdot \cos \frac{y}{4}. \quad (21)$$

Tyto funkce označuji ve zdrojovém kódu jako proměnné *funkce1, ..., funkce4*. Rozhodnutí, která z nich se bude vybírat, provádím pomocí příkazu Switch (přepínač). Pomocí něj také určuji z_{min} a z_{max} , které mi značí rozsah osy ve směru z . Tím je zajištěno, že graf i při jakékoliv změně udržuje stále stejné rozsahy a nijak se nedeformuje. Program Mathematica dokáže spočítat parciální derivace po zadání příslušného výrazu. Já jsem využil jednu z možností, jak tuto derivaci zadat, tedy pomocí palety nástrojů, kde je přímo znak parciální derivace uveden. Proměnné označuji ve zdrojovém kódu *parcx*, *parcy*. Pro zobrazení více grafických prvků kombinuji příkazy Manipulate a Show. Samotné vykreslení grafu zadané funkce není nikterak složité, použil jsem příkaz Plot3D. Pro samotné vykreslení parciální derivace (tečny ke křivce) podle proměnné x je nutné znát odvození rovnice tečny t ke grafu funkce:

$$t: \quad z - z_0 = z'_x(x - x_0) \quad | + z_0 \quad (22)$$

$$z = z_0 + z'_x(x - x_0).$$

Pokud uvažuji

$$z_0 = f(x_0, y_0); \quad t = (x - x_0),$$

pak

$$z = f(x_0, y_0) + z'_x \cdot t. \quad (23)$$

Obdobným způsobem lze odvodit i rovnici tečny pro parciální derivaci podle proměnné y . Parametrické zadání odvozených vztahů do programu je na Obr. 12.

```
(*vykreslení parciální derivace*)
If[derivacex,
  (*zobrazení parciální derivace x*)
  ParametricPlot3D[{t + x0, y0, (f + t*parcx) /. {x -> x0, y -> y0}}, {t, -2, 2}, PlotStyle -> {Green, Thickness[0.01]}],
  {}
],

If[derivacey,
  (*zobrazení parciální derivace y*)
  ParametricPlot3D[{x0, t + y0, (f + t*parcy) /. {x -> x0, y -> y0}}, {t, -2, 2}, PlotStyle -> {Yellow, Thickness[0.01]}],
  {}
],
```

Obr. 12. Příkazy pro vykreslení parciálních derivací (tečen) ke grafu funkce.

Zobrazení křivky na Obr. 11 (bílá a modrá) určují ze znalosti, kdy při parciální derivaci podle proměnné x je y_0 „zafixované“ a mění se hodnota x_0 . Při parciální derivaci podle proměnné y je tomu opačně (viz Obr. 13).

```
(*zobrazení křivky podle parciální derivace x, y0 zafixované*)
If[(derivacex) &&krivka,
  ParametricPlot3D[{a, y0, f /. {x -> a, y -> y0}}, {a, -10, 10}, PlotStyle -> {Blue}],
  {}],
(*zobrazení křivky podle parciální derivace y, x0 zafixované*)
If[(derivacey) &&krivka,
  ParametricPlot3D[{x0, b, f /. {x -> x0, y -> b}}, {b, -10, 10}, PlotStyle -> {White}],
  {}],
```

Obr. 13. Příkazy pro vykreslení křivek parciálních derivací.

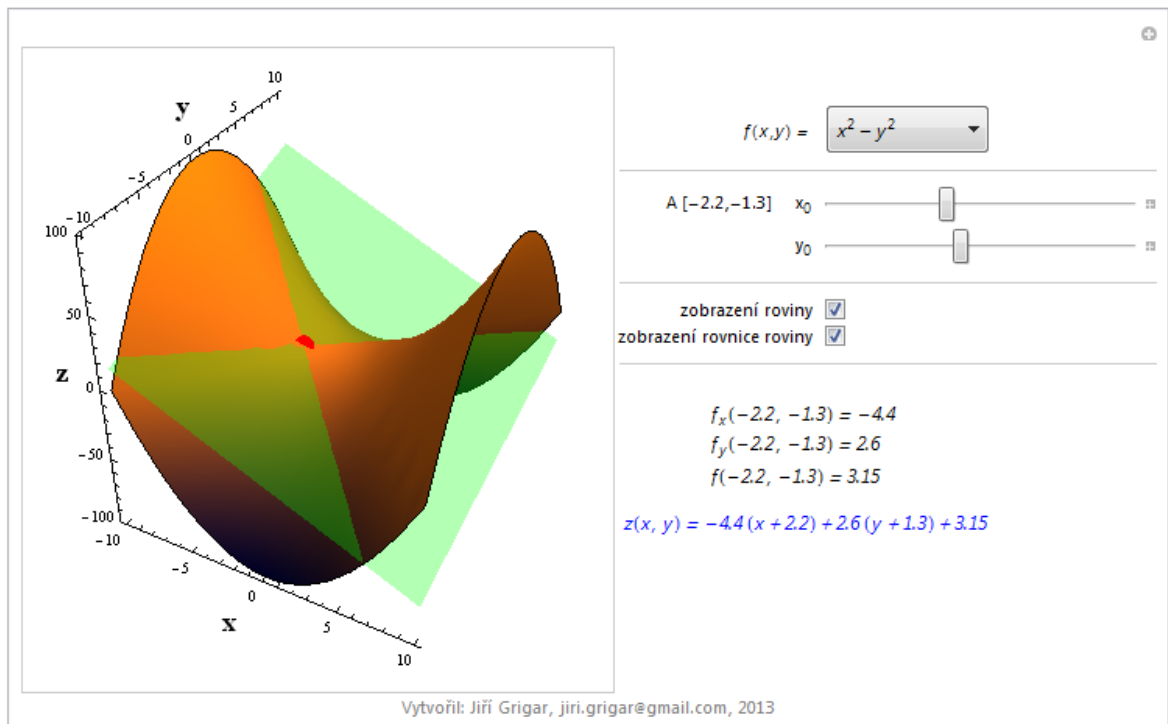
Zobrazení vykreslovaného bodu provádím pomocí příkazu Point tak, že dopočítávám z-ovou souřadnici jako funkční hodnotu v daném bodě, tedy $f(x_0, y_0)$.

K vykreslení rovin používám příkazu Polygon, který mi spojí zadané body plochou. V mém případě tak spojuji 4 body a výsledkem je čtverec. Pokud provádíme parciální derivaci podle proměnné x , tak výsledná rovina je rovnoběžná s rovinou o_{xz} . Analogicky tomu je pro proměnnou y , kdy je rovina rovnoběžná s rovinou o_{yz} . Vytvoření všech ovládacích prvků je na Obr. 14.

```
{{funkce, funkce1, Style["f(x,y) = ", Italic]}, {funkce1 -> TraditionalForm[funkce1],
  funkce2 -> TraditionalForm[funkce2],
  funkce3 -> TraditionalForm[funkce3],
  funkce4 -> TraditionalForm[funkce4]},
  PopupMenu},
Delimiter,
(*body x0 a y0 - posunování*)
{{x0, 0, Dynamic[Row[{"A [" , x0, ", ", y0, " ]      x0"}]}], -10, 10, 0.1},
{{y0, 0, "y0"}, -10, 10, 0.1},
(*výběr z menu*)
Delimiter,
{{derivacex, True, "parciální derivace podle x"}, {True, False}},
{{derivacey, False, "parciální derivace podle y"}, {True, False}},
"",
{{rovina, False, "zobrazení roviny"}, {True, False}},
{{krivka, False, "zobrazení křivky"}, {True, False}},
ControlPlacement -> Right,
FrameLabel -> {Style["Vytvořil: Jiří Grigar, jiri.grigar@gmail.com, 2013", Gray]}
```

Obr. 14. Příkazy pro ovládací prvky animace.

6.2 Tečná rovina



Obr. 15. Animace tečné roviny v daném bodě.

Druhý interaktivní notebook zobrazuje tečnou rovinu pro 4 vybrané funkce. Na výběr máme možnost zvolit bod (A), ve kterém se bude rovina vykreslovat, samotné vykreslení roviny a zobrazení její rovnice. Graf lze otáčet, posouvat i přibližovat. Změna výše uvedených možností se projeví ihned bez dalšího překladu.

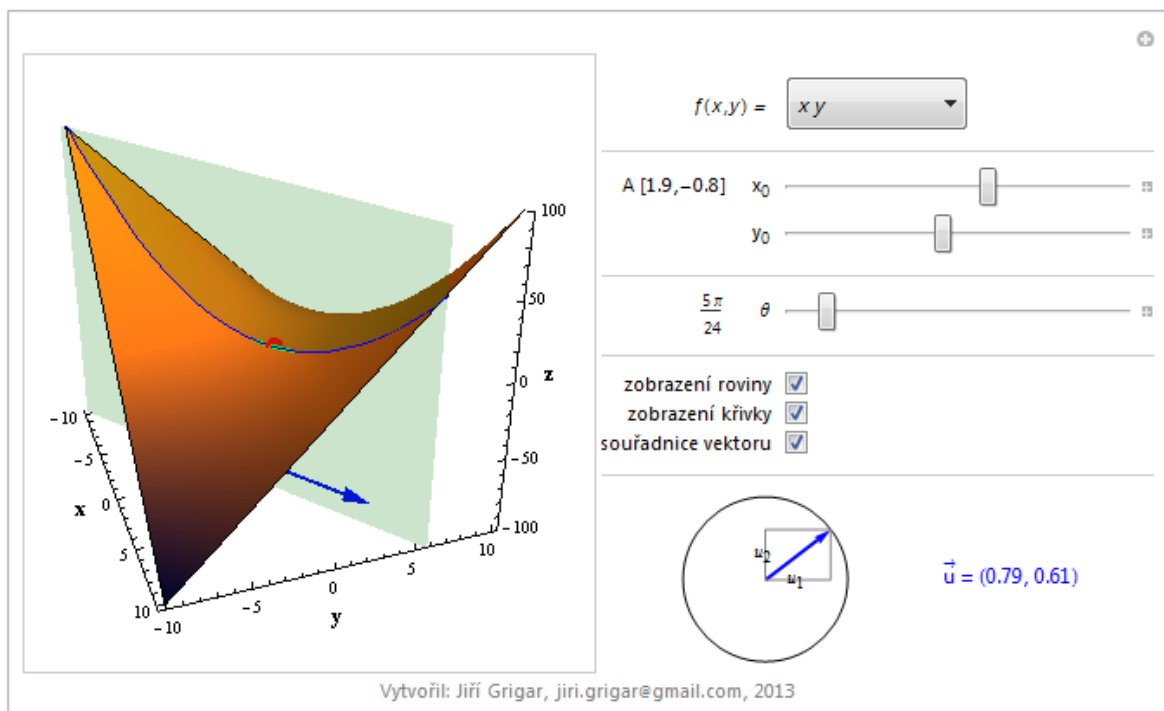
Pokud klikneme na šipku v roletovém menu u zadané funkce, máme na výběr stejné funkce jako u parciální derivace, viz vztahy (18), (19), (20), (21). Základ animace je tedy obdobný jako u předchozí. Důležitou změnou je zadání příkazů tečné roviny, které jsou na Obr. 16.

```
(*výpočet parciálních derivací*)
parcx = ∂x (f);
parcy = ∂y (f);
(*výpočet funkčních hodnot zadané funkce a parciálních derivací*)
funkcniHodnotaF = f /. {x → x0, y → y0};
funkcniHodnotaFx = parcx /. {x → x0, y → y0};
funkcniHodnotaFy = parcy /. {x → x0, y → y0};
(*předpis tečné roviny*)
tecnaRovina = funkcniHodnotaF + funkcniHodnotaFx * (x - x0) + funkcniHodnotaFy * (y - y0);
```

Obr. 16. Příkazy pro výpočet tečné roviny.

K samotnému vykreslení pak používám příkaz Plot3D, kde vstupní proměnné jsou x , y . Základ ovládacích prvků je podobný jako u parciálních derivací (Obr. 14), ale došlo k rozšíření zdrojového kódu a také zvýšení složitosti. Podrobný popis je v notebooku animace v podobě komentáře zdrojového kódu. Základní konstrukce výpisu je uvedena u směrové derivace na Obr. 21.

6.3 Směrové derivace



Obr. 17. Animace směrové derivace v daném bodě.

Třetí interaktivní notebook zobrazuje směrovou derivaci pro 4 vybrané funkce. Na výběr máme možnost zvolit bod derivace (A), úhel theta (θ), zobrazení roviny, křivky a souřadnice vektoru. Graf lze otáčet, posouvat i přibližovat. Změna výše uvedených možností se projeví ihned bez dalšího překladu.

Pokud klikneme na šipku v roletovém menu u zadané funkce, máme na výběr stejné funkce jako u parciální derivace a tečné roviny, viz vztahy (18), (19), (20), (21). Základ animace je tedy obdobný jako u předešlých příkladů. Důležitou změnou je zadávání derivace již v libovolném směru podle vektoru \vec{u} . Tento vektor bude vždy jednotkový (není ho třeba normovat), pokud bude vždy opisovat jednotkovou kružnici v rozsahu od 0 do 2π . Souřadnice tohoto vektoru pak vypočtu ze znalosti určování stran pravoúhlého trojúhelníku

jako \cos a \sin úhlu theta pro souřadnice x a y . Výpočet směrnice je pak dán skalárním součinem gradientu funkce a vektoru \vec{u} . Použité příkazy jsou na Obr. 18.

```
(*výpočet parciálních derivací, funkčních hodnot a směrnice*)
parcx =  $\partial_x(f)$ ;
parcy =  $\partial_y(f)$ ;

funkcniHodnotaFx = parcx /. {x → x0, y → y0};
funkcniHodnotaFy = parcy /. {x → x0, y → y0};

gradF = {funkcniHodnotaFx, funcniHodnotaFy};
smernice = gradF.{a, b};
```

Obr. 18. Výpočet směrnice tečny ke grafu funkce.

Samotné vykreslení tečny pak provádím pomocí příkazu ParametricPlot3D. Na Obr. 19 je zobrazeno zadávání příkazů pro vykreslení křivky i tečny ke grafu funkce (modrá a zelená křivka animace).

```
(*zobrazení křivky*)
If[krivka,
  ParametricPlot3D[{a*t+x0, b*t+y0, f /. {x → (a*t+x0), y → (b*t+y0)}}, {t, -100, 100}, PlotStyle → {Blue}],
  {}],

(*zobrazení tečny*)
ParametricPlot3D[{a*t+x0, b*t+y0, (smernice*t+f) /. {x → x0, y → y0}}, {t, -3, 3}, PlotStyle → {Green, Thickness[0.01]}
```

Obr. 19. Vykreslení křivky a tečny ke grafu funkce.

Základ ovládacích prvků je opět podobný jako u parciální derivace (Obr. 14), ale došlo k rozšíření zdrojového kódu a také zvýšení složitosti, zde však vybrané části popíšu. Pro vykreslení kružnice spolu s vektorem jsem použil příkazy na Obr. 20.

```
(*zobrazení kružnice, vektoru - šipky a křivky*)
"\t",
Graphics[
  {Circle[{0, 0}, 1],
   {Blue, Arrowheads[0.1], Thick, Arrow[{{0, 0}, {a, b}}]},
   {Gray, Line[{{0, 0}, {a, 0}, {a, b}, {0, b}, {0, 0}}]},
   Text["u1", {a*0.5, 0}], Text["u2", {0, b*0.5}]},
  ImageSize → 100],
```

Obr. 20. Vykreslení křivky a tečny ke grafu funkce.

Vykresluji tak kružnici pomocí příkazu Circle s poloměrem 1, šipku (zobrazení vektoru) pomocí příkazu Arrow, kde se mění směr pomocí souřadnic vektoru (hodnoty a , b). Dále

pomocí příkazu `Line` vykresluji úsečky o velikosti vektoru a popisky souřadnic u_1 a u_2 pomocí příkazu `Text`.

Dalším prvkem je jednoduchý výpis souřadnic vektoru \vec{u} , který je na Obr. 21. Podobnými příkazy se dá vypisovat dynamicky text a proměnné. Nesmíme však zapomenout na to, že všechny tyto příkazy musí být ve funkci `Dynamic`. Obdobně jsem řešil i výpisy u tečné roviny, které jsem v práci neuváděl.

```
(*zobrazení souřadnic vektoru s podmínkou*)
If[vektor,
  Style[
    TraditionalForm[Row[{
      "\t", Overscript["u", "->"], " = (", N[a, 2], ", ", N[b, 2], ")"
    }]], 12, Blue]
]
```

Obr. 21. Vykreslení souřadnic vektoru.

6.4 Absolutní extrémy

Pro zobrazení absolutních extrémů jsem vytvořil 3 interaktivní animace pro 3 vybrané funkce na různých množinách. Množiny jsem volil v podobě kruhu, obdélníku a trojúhelníku. Funkce jsem pro názornost vybíral z přednášek a cvičení předmětu Matematika II. Všechny animace jsou založené na stejném základu, ale řešení příkladů je individuální, proto jsem je rozdělil do tří notebooků. Notebooky mají všechny stejné rozhraní i ovládací prvky, proto popis programu provedu přednostně a poté se zaměřím na samotný popis jednotlivých animací. Při vytváření animací absolutních extrémů byla použita literatura [1] a [4].

Jednou ze základních funkcí je testování, zda bod podezřelý z extrému leží v dané množině. V prvním příkladě je množinou kruh s poloměrem 5. Test bodu, zda se nachází v daném kruhu, vidíme na Obr. 22.

```
(*funkce, která testuje zadaný bod, jestli odpovídá omezení*)
testBodu[bod_] := Module[{},
  If[bod[[1]]2 + bod[[2]]2 ≤ 25,
    (*pokud ano, uloží se do vektoru body*)
    body = Append[body, bod],
    {}
  ]
];
```

Obr. 22. Funkce pro test bodu daného omezení.

Pokud bod leží v dané množině, přidá se pomocí příkazu Append do proměnné *body*, která při inicializaci tvoří prázdný vektor.

Pro výpis bodu jsem si vytvořil další funkci, která pouze pomocí příkazu Row spojuje dané proměnné a vypisuje zadaný bod (př. $P1 = [1,2]$).

```
(*funkce pro výpis bodů pod ovládací panely*)
vypisBod[i_] := Module[{},
  Row[{"\tP", i, " = [", body[[i]][[1]], ", ", body[[i]][[2]], "]"
];
```

Obr. 23. Funkce pro výpis bodu.

Pomocí následující funkce vyhledávám body podezřelé z extrému tak, že dosadím funkční předpis pro hranici množiny do zadané funkce, zderivuji ji a najdu stacionární body. Pokud má rovnice řešení, tak provedu zavolání funkce na testování bodu a případně bod uložím jako bod podezřelý z extrému. Konstrukce funkce je pak na Obr. 24.

```
(*funkce pro vyhledání kandidátů*)
ziskejBod[omezeni_] :=
Module[{novaFunkce, derivace, vyres, podle, bod, x1},
  novaFunkce = funkce /. {x -> omezeni[[1]], y -> omezeni[[2]]};
  (*vrací proměnnou*)
  podle = Variables[novaFunkce];

  (*pokud dostanu prázdný vektor, tak rovnice nemá řešení*)
  If[Length[podle] != 0,
    derivace = D[novaFunkce, podle];
    vyres = Solve[derivace == 0, podle];

    x1 = vyres[[1]][[1]][[2]];

    bod = omezeni /. {x -> x1, y -> omezeni[[2]]};
    (*ověříme, zda odpovídá omezení*)
    testBodu[bod],
    {}
  (*nic se neprovede*)
];
```

Obr. 24. Funkce pro vyhledání bodů podezřelých z extrému.

Jako další jsem vytvořil funkci, která ukládá do připraveného prázdného vektoru (proměnná *funkcniHodnoty*) bod a zároveň funkční hodnotu v daném bodě.

```
(*funkce pro získání funkční hodnoty v daném bodě*)
ziskejFunkcniHodnotu[bod_] := Module[{FunkcniHodnota, uloz},
  FunkcniHodnota = funkce /. {x → bod[[1]], y → bod[[2]]};
  uloz = Append[bod, FunkcniHodnota];
  (*výsledek uloží jako bod + funkční hodnotu do vektoru - globální proměnné*)
  funkcniHodnoty = Append[funkcniHodnoty, uloz];
];
```

Obr. 25. Funkce pro získání funkční hodnoty v daném bodě.

Tuto funkci pak volám v cyklu For pro všechny body podezřelé z extrému. Získám tak dvourozměrný vektor, kdy každému prvku odpovídá bod a funkční hodnota. Poté všechny prvky vektoru seřadím podle funkčních hodnot. První prvek pak odpovídá absolutnímu minimu a poslední absolutnímu maximu.

Asi nejdůležitější část zdrojového kódu tvoří zobrazení množiny na dané ploše, které je na Obr. 26. To jsem prováděl pomocí funkce Plot3D s kombinací RegionFunction. Tyto příkazy nám vykreslí na ploše pouze ohraničení, proto je vhodné ručně dokreslit danou množinu do roviny o_{xy} . Dokreslení jsem prováděl pomocí příkazu Sphere, který nám vytvoří rotační válec, nicméně s nulovou (minimální) výškou dostaneme kruh.

```
(*zobrazení omezení*)
If[omezeni,
  Plot3D[funkce, {x, -6, 6}, {y, -6, 6},
    Mesh → False, Boxed → False, ColorFunction → "RustTones",
    RegionFunction → Function[{x, y, z}, x2 + y2 <= 25], BoundaryStyle → Directive[Red, Thick],
    Filling → -200
  ], {}
],
```

Obr. 26. Zobrazení kruhu na dané ploše.

Další část zdrojového kódu tvoří zobrazení všech bodů podezřelých z extrému. To jsem vyřešil pomocí příkazu Table a Point, jelikož mi nešly vykreslit všechny body v cyklu pomocí příkazu For. Zde vidíme, jak je výhodné mít různé alternativy při programování, jelikož s takovým problémem se občas setkáváme. Příkazy použité na vykreslení všech bodů jsou na Obr. 27.

```

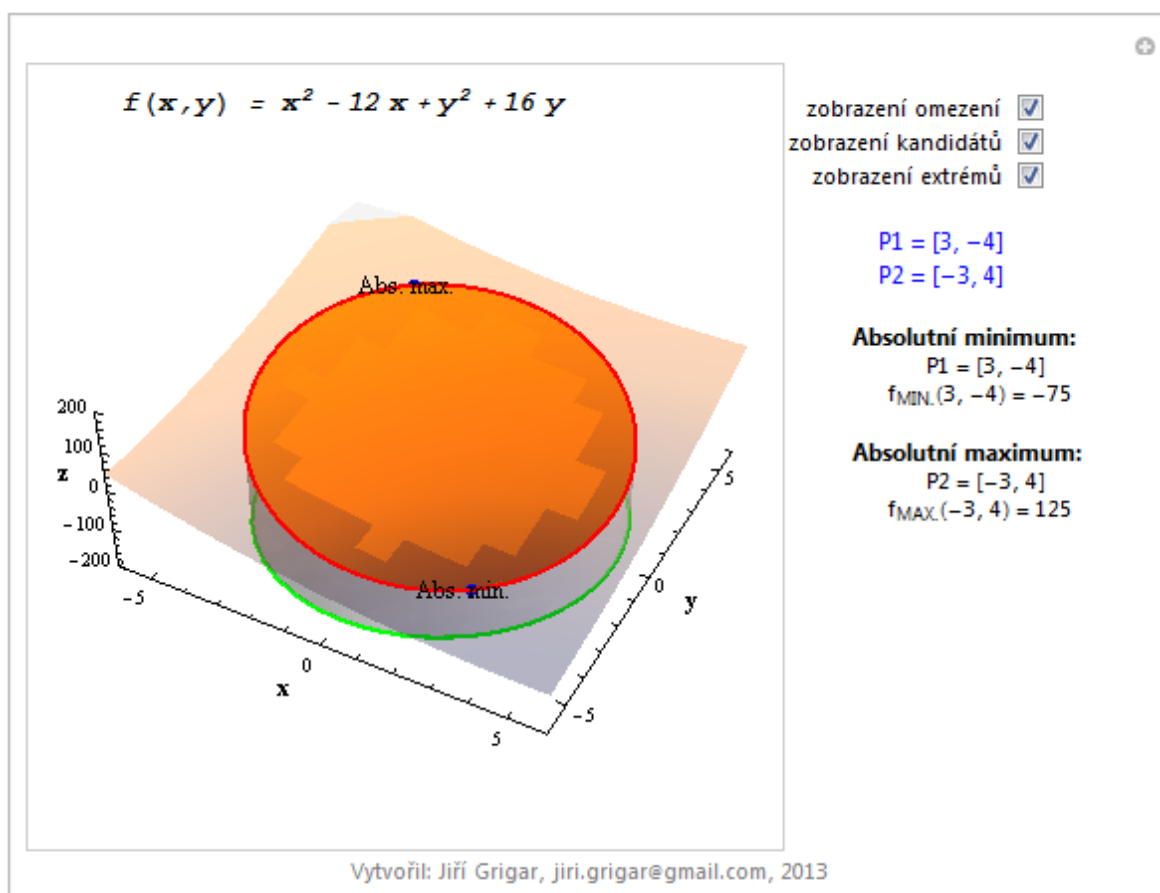
(*vykreslení všech kandidátů na extrém*)
If[kandidati,
  Graphics3D[{PointSize[0.02], Blue,
    Point[
      Table[funkcniHodnoty[[t]], {t, 1, Length[funkcniHodnoty], 1}]
    ]
  ]
], {},
, {}],

```

Obr. 27. Zobrazení bodů podezřelých z extrému.

Všechny animace obsahují stejné nastavení, zda chceme zobrazit omezení na ploše, body podezřelé z extrému a také absolutní maximum a minimum. Při zvolení některé z posledních dvou možností se vykreslují a označují body nejen v grafu, ale také se vypisují pod ovládacími prvky spolu s funkční hodnotou v daném bodě. Konstrukce výpisu je obdobná jako v předchozích animacích, viz Obr. 21.

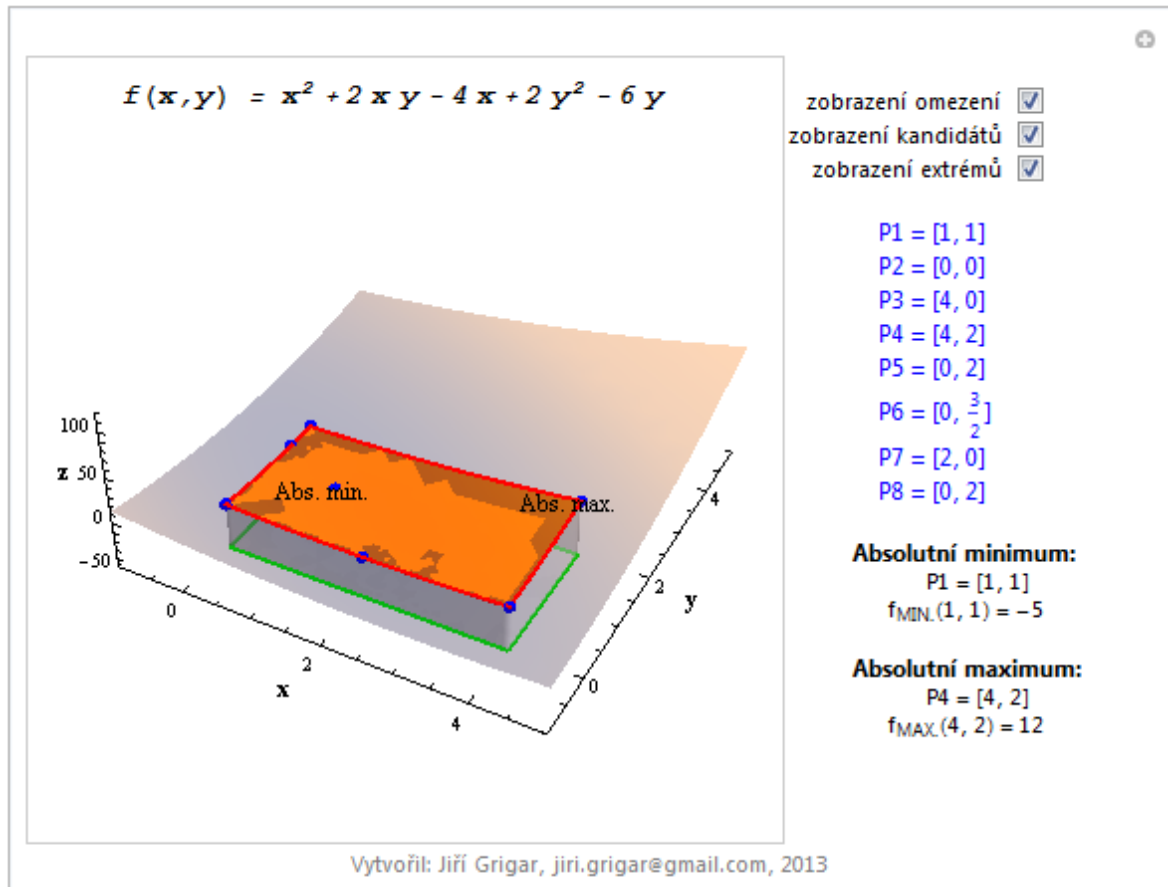
6.4.1 Absolutní extrémy na kruhu



Obr. 28. Absolutní extrémy na kruhu.

Další popis této animace v tomto případě nepovažuj za nutný, protože téměř celý úvod do kapitoly 6.4 se zabývá právě tímto příkladem.

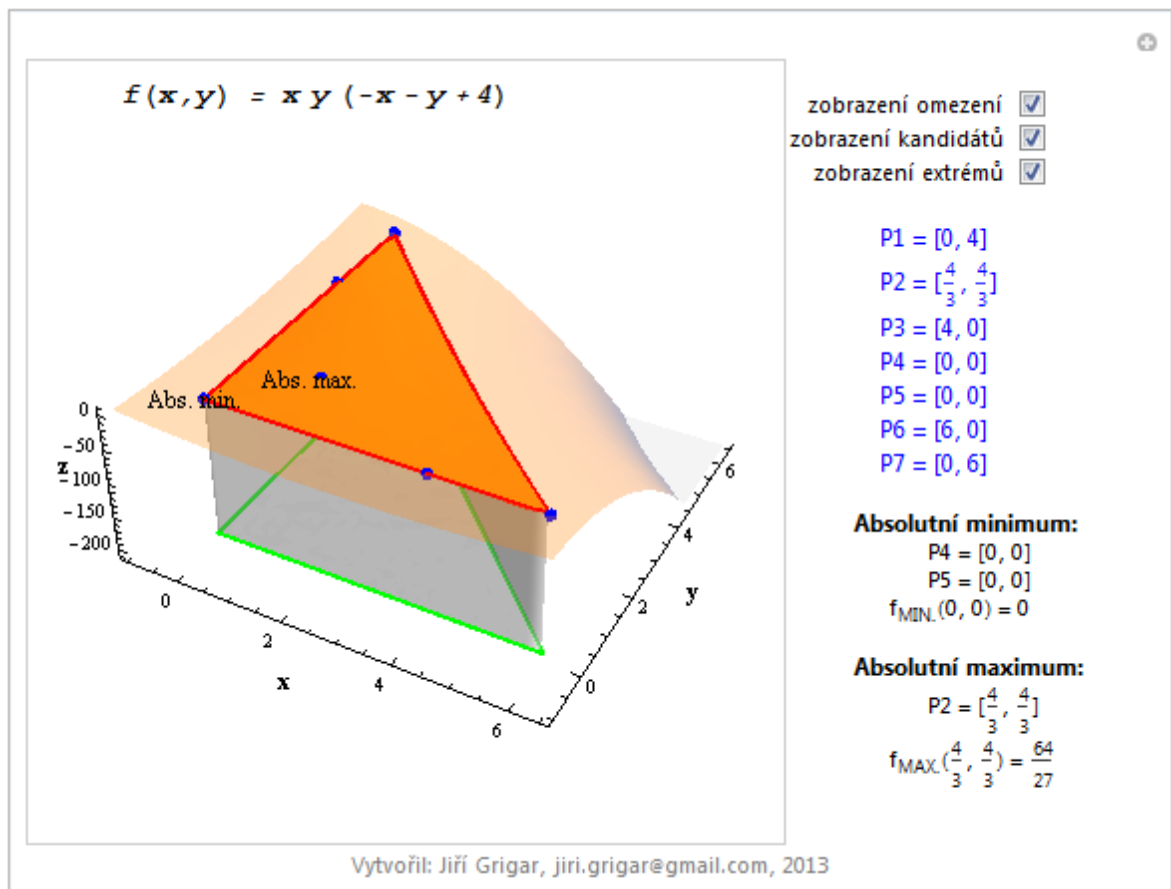
6.4.2 Absolutní extrémy na obdélníku



Obr. 29. Absolutní extrémy na obdélníku.

V tomto příkladu je hranicí množiny obdélník zadaný 4 vrcholy. Tyto 4 body jsou pak logicky vždy body podezřelé z extrému, proto je nutné je zahrnout do výsledného řešení. Rozdíl oproti předchozímu příkladu spočívá v jiném rozdělení množiny. Dále je patrné, že dostáváme více možných řešení, a zdrojový kód se tak liší v některých částech. Zobrazení množiny do roviny o_{xy} jsem prováděl pomocí příkazu Polygon, kdy jsem spojil všechny vrcholy obdélníku.

6.4.3 Absolutní extrémů na trojúhelníku



Obr. 30. Absolutní extrémů na trojúhelníku.

V posledním příkladu tvoří hranici množiny úsečky, které vytváří pravoúhlý trojúhelník. Vrcholy tohoto trojúhelníku jsou pak logicky vždy body podezřelé z extrému, proto je nutné je zahrnout do výsledného řešení. Rozdíl oproti předchozím příkladům spočívá v jiném rozdělení hranice množiny a také omezení přímkou (přeponou trojúhelníka). Zobrazení množiny do roviny o_{xy} jsem prováděl pomocí příkazu Polygon, kdy jsem spojil všechny vrcholy trojúhelníku.

ZÁVĚR

Hlavním cílem bakalářské práce bylo vytvoření interaktivních příkladů, které budou zobrazovat danou problematiku. První část práce se zaměřuje na základní pojmy z diferenciálního počtu funkcí více proměnných. Jsou zde uvedeny základní vztahy a ilustrativní obrázky, které slouží pouze pro zopakování dané problematiky. Pro více informací doporučuji literaturu [2]. V kapitole 5 se také vyskytuje základní popis a programování v programu Wolfram Mathematica 8. Důraz je kladen na vysvětlení příkazů a funkcí, které byly při práci použity. Dle mého názoru je tento program jedním z nejlepších, které lze použít. Je to z toho důvodu, že i když není člověk znalý programování, může ihned intuitivně tvořit jednoduché programy. Software totiž obsahuje rozsáhlou indexovanou nápovědu, která zobrazuje i příklady použití. Ty lze přímo v ní jednoduše modifikovat a výstup vidíme ihned. Při uzavření nápovědy se všechna původní data obnovují.

Popis animací nalezneme v druhé části této práce. Celkem bylo vytvořeno 6 interaktivních příkladů demonstrující především aplikace parciálních derivací. Animace mají pro názornost a jednoduchost téměř shodná rozhraní. Uživatel se v nich tak bude lépe orientovat. Dále zde uvádím části zdrojových kódů, kde jsou k vidění kombinace příkazů z teoretické části. Čtenář má tak ihned představu o náročnosti vytvoření takových animací. Do zdrojových kódů animací jsem také doplnil komentáře, které jsou vhodné pro pochopení daných úseků kódu.

Práci jsem si vybral z toho důvodu, že i já při studiu diferenciálního počtu funkcí více proměnných jsem měl horší představivost. Např. když vypočtu parciální derivaci zadané funkce v určitém bodě, jak pak vypadá v grafu? Téměř v každé literatuře jsou uvedeny teoretické základy, ale v některých chybí i názorný obrázek, který považuji v takovém případě za velmi důležitý. Proto jsem vytvořil interaktivní příklady, v nichž lze např. volit souřadnice bodu, v němž počítáme parciální derivace, měnit směr derivace či zobrazovat body podezřelé z extrému. Animace mohou dále sloužit jako podpůrný materiál k předmětu Matematika II a já pevně doufám, že pomohou studentům při studiu.

ZÁVĚR V ANGLIČTINĚ

The aim of this thesis was to create some interactive examples that will display the issue. The first part of the thesis focuses on the basic concepts of differential calculus of functions of more variables. There are basic relations and visual images that only serve to revise the issue. I can recommend some source [2] for more information. In chapter 5 there is also basic description and programming in the Wolfram Mathematica 8. Emphasis is placed on explaining the commands and functions that have been used in the work. In my opinion this program is one of the ones that you can use. Even if someone is not familiar with programming, he can create simple programs immediately and intuitively. The software includes an extensive indexed help which displays examples of use. You can edit and see the output immediately. If you close the help, the all original data will be restored.

Description of animations can be found in the other part of this work. Six interactive examples were created for demonstrating applications, especially partial derivatives. Animations have almost the same interface for their clarity and simplicity. The user will be better oriented in them. Furthermore, I present parts of the source codes, where you can see a combinations of commands from the theoretical part. The reader can immediately see how difficult it was to create such animations. I added comments in the source codes animations for better understanding the given parts of the codes.

I have chosen as studying differential calculus of functions of more variables I also had worse conception. For example if computing the partial derivative of the given function at a certain point, what does it look like in the graph? Theoretical foundations are given almost in every source, but in some of them a visual image are still missing, which I consider to be a very important part. That is why I created interactive animations in which we can choose the coordinates in order to we compute partial derivatives, change the direction of the derivative or display points suspicious of extreme etc. Animations can also serve as a support material for Mathematics II course. I firmly believe it will help students with their studies.

SEZNAM POUŽITÉ LITERATURY

- [1] Calc III 2013 13.8 pt 2 Absolute Extrema. In: *Youtube* [online]. 26.2.2013 [cit. 2013-05-22]. Dostupné z: <http://www.youtube.com/watch?v=qQ0oCiezsU>. Kanál uživatele BeamerMath.
- [2] DOŠLÁ, Zuzana a Ondřej DOŠLÝ. *Diferenciální počet funkcí více proměnných*. 3. vyd. Brno: Masarykova univerzita, 2006, 144 s. ISBN 80-210-4159-5.
- [3] CHRAMCOV, Bronislav. *Základy práce v prostředí Mathematica*. Vyd. 1. Ve Zlíně: Univerzita Tomáše Bati ve Zlíně, 2005, 122 s. ISBN 8073182688.
- [4] MAŘÍK, Robert. *Absolutní extrémy*. In: 3D obrázky k diferenciálnímu počtu funkcí dvou proměnných [online]. 2012 [cit. 2013-05-30]. Dostupné z: <http://user.mendelu.cz/marik/wiki/3D/absolutni2.html>
- [5] OSTRAVSKÝ, Jan. *Diferenciální počet funkce více proměnných. Nekonečné číselné řady*. Vyd. 4., nezm. Zlín: Univerzita Tomáše Bati ve Zlíně, 2009, 158 s. ISBN 978-80-7318-856-6.
- [6] REKTORYS, Karel. *Přehled užití matematiky I*. 6. přeprac. vyd. Praha: Prometheus, 1995, 720 s. ISBN 80-85849-92-5.
- [7] WEIR, Maurice D., Joel HASS, George B. THOMAS a Ross L. FINNEY. *Thomas' calculus*. 11th ed. Boston: Pearson Addison Wesley, 2005, 1228 s. ISBN 0-321-48987-X.
- [8] *Wolfram Mathematica Documentation* 8. In: Wolfram [online]. 2013 [cit. 2013-5-21]. Dostupné z: <http://reference.wolfram.com/mathematica/guide/Mathematica.html>
- [9] WOLFRAM, Stephen. *The Mathematica book*. 5. ed. Champaign, Ill: Wolfram Media, 2003, 1464 s. ISBN 15-795-5022-3.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

∂	Znak parciální derivace.
\in	Patří do.
*.nb	Soubory s příponou notebook.
et al.	a jiní.
\subseteq	Je podmnožinou.
\emptyset	Označení prázdné množiny.
o_{xy}	Rovina určená souřadnými osami x, y .
\mathbb{R}	Množina všech reálných čísel.
\mathbb{R}^n	n rozměrný reálný prostor.
\vec{u}	Označení vektoru u .
\mathbb{V}^n	Označení pro zaměření n rozměrného reálného prostoru.
∇	Operátor nabra, který označuje gradient funkce.

SEZNAM OBRÁZKŮ

<i>Obr. 1. Vykreslení grafu zadané funkce $f(x,y) = x^2 - y^2$ v trojrozměrném prostoru.</i>	12
<i>Obr. 2. Parciální derivace funkce v daném bodě podle nezávislých proměnných x a y.</i>	14
<i>Obr. 3. Příklad tečné roviny ke grafu zadané funkce.</i>	16
<i>Obr. 4. Ukázka cyklu ve vývojovém prostředí Mathematica spolu s výstupem (tiskem) počítačidla cyklů.</i>	20
<i>Obr. 5. Ukázka vyhledávání v nápovědě v programu Mathematica.</i>	21
<i>Obr. 6. Výřez palet v programu Mathematica.</i>	22
<i>Obr. 7. Příklad příkazu Manipulate.</i>	24
<i>Obr. 8. Dva příklady příkazu Switch.</i>	24
<i>Obr. 9. Příklad příkazu Table.</i>	25
<i>Obr. 10. Příklad funkce pro sečtení dvou čísel spolu s jejím zavoláním.</i>	26
<i>Obr. 11. Animace parciální derivace podle nezávislých proměnných x, y.</i>	28
<i>Obr. 12. Příkazy pro vykreslení parciálních derivací (tečen) ke grafu funkce.</i>	29
<i>Obr. 13. Příkazy pro vykreslení křivek parciálních derivací.</i>	30
<i>Obr. 14. Příkazy pro ovládací prvky animace.</i>	30
<i>Obr. 15. Animace tečné roviny v daném bodě.</i>	31
<i>Obr. 16. Příkazy pro výpočet tečné roviny.</i>	31
<i>Obr. 17. Animace směrové derivace v daném bodě.</i>	32
<i>Obr. 18. Výpočet směrnice tečny ke grafu funkce.</i>	33
<i>Obr. 19. Vykreslení křivky a tečny ke grafu funkce.</i>	33
<i>Obr. 20. Vykreslení křivky a tečny ke grafu funkce.</i>	33
<i>Obr. 21. Vykreslení souřadnic vektoru.</i>	34
<i>Obr. 22. Funkce pro test bodu daného omezení.</i>	34
<i>Obr. 23. Funkce pro výpis bodu.</i>	35
<i>Obr. 24. Funkce pro vyhledání bodů podezřelých z extrému.</i>	35
<i>Obr. 25. Funkce pro získání funkční hodnoty v daném bodě.</i>	36
<i>Obr. 26. Zobrazení kruhu na dané ploše.</i>	36
<i>Obr. 27. Zobrazení bodů podezřelých z extrému.</i>	37
<i>Obr. 28. Absolutní extrémy na kruhu.</i>	37
<i>Obr. 29. Absolutní extrémy na obdélníku.</i>	38
<i>Obr. 30. Absolutní extrémy na trojúhelníku.</i>	39

SEZNAM TABULEK

<i>Tab. 1 Základní nastavení příkazu Plot3D.</i>	<i>22</i>
<i>Tab. 2 Základní nastavení příkazu Graphics3D.</i>	<i>23</i>

SEZNAM PŘÍLOH

Všechny vytvořené interaktivní animace, které jsou popisovány v kapitole 6, jsou přiloženy na CD.