

Dokumentace k programu Wolfram Mathematica

David Janíček



OBSAH

ÚVOD.....	2
1 PROGRAM WOLFRAM MATHEMATICA	3
1.1 PRVNÍ SPUŠTĚNÍ	4
1.2 MENU	5
1.3 ZÁKLADNÍ PRÁCE S PROGRAMEM	8
1.4 NÁPOVĚDA PROGRAMU	10
2 PRÁCE S PROMĚNNÝMI	13
2.1 DEFINICE PROMĚNNÉ.....	13
2.2 DEFINICE FUNKCE	14
2.3 LOKÁLNÍ GLOBÁLNÍ A LOKÁLNÍ PROMĚNNÉ	15
3 ROVNICE	16
4 VEKTORY A MATICE	17
4.1 FUNKCE PRO PŘÍSTUP K ČÁSTEM VEKTORŮ A MATIC	18
4.2 OPERACE S VEKTORY	18
4.3 OPERACE S MATICEMI	19
4.4 FUNKCE TABLE	21
5 RELAČNÍ OPERÁTORY	22
6 2D GRAFY FUNKCÍ	23
6.1 PARAMETRY FUNKCE PLOT	23
SEZNAM POUŽITÉ LITERATURY.....	29
SEZNAM OBRÁZKŮ	30

ÚVOD

Tato bakalářská práce se zabývá jednou ze základních disciplín matematiky a představuje teorii nezbytnou k pochopení problematiky lineární algebry. Ta je následně aplikována v matematickém software. Pro potřeby této práce byl zvolen program Wolfram Mathematica, jelikož jeho licence je dostupná pro všechny studenty fakulty aplikované informatiky Univerzity Tomáše Bati. Tento text by měl pomoci studentům, kteří navštěvují Math Support Centre, nacházející se na Fakultě aplikované informatiky UTB.

Hlavním zdrojem definic a vět použitých v každé kapitole, byly knihy uvedené v doporučené literatuře rozšířené o literaturu uvedenou v seznamu na konci práce. Text je členěn chronologicky a uvedené poučky lze aplikovat i na problémy, které se vyskytují v kapitolách, které následují. U každé definice popisující danou látku je uveden ilustrační řešený příklad, který s ní úzce souvisí a měl by čtenáři osvětlit cestu, jak a proč se dojde k výsledku. Tyto postupy jsou popsány krok po kroku a tak, aby je mohl pochopit i opravdový laik. Pokud je možností jak se dostat ke stejnému a správnému výsledkům více, jsou uvedeny i další tyto postupy.

Toto téma jsem si zvolil z toho důvodu, že pro většinu studentů je nejen lineární algebra ale i matematika, jako taková, problém. Pro studenty je těžký přechod z matematiky, která se vyučuje na středních školách, na matematiku vysokoškolskou. Rozdíly se projevují nejvíce mezi absolventy ze středních odborných škol a absolventy z gymnázií, kteří s matematikou nemají sebemenší problém. Studenti se pak učí látku mechanicky a učí se veškeré postupy nazpaměť. Tempo přednášek jim připadá neúnosné a na případné konzultace se raději nedostaví, aby se v očích ostatních neztrapňovali.

Přesně z tohoto důvodu bylo vytvořeno Math Support Centere v přízemí fakulty aplikované informatiky aby mohli tito studenti přijít a nedostatky dohnat a problematiku pochopit. Svoji práci bych chtěl přispět ke zlepšení kvality výuky v tomto centru a rozšířit obzory navštěvujícím. Za řešenými příklady jsou řešeny i obdobné příklady v software Wolfram Mathematica, které by měli také napomoci k lepší efektivitě řešení a slouží hlavně jako ukázka, že ne všechny problémy, které vyvstávají, je nutné řešit složitě.

V poslední kapitole této práce jsou uvedeny ukázky, jak se teorie lineární algebry aplikuje v praxi s ukázkou na jednoduchých příkladech řešených ve Wolfram Mathematica. Součástí je také dokumentace, sloužící k náhledu pokud by při řešení jakéhokoliv příkladu nastal problém.

1 PROGRAM WOLFRAM MATHEMATICA

Wolfram Mathematica je program, sloužící k provádění číselných i symbolických výpočtů a jejich následnou prezentaci na výstupu. Umí pracovat s čísly s přesností, jakou si uživatel nastaví, vektory, maticemi, rovnicemi a jejich soustavami, nebo například s reálnými a komplexními čísly. Mathematica umožňuje vytvořit písemné, ale i grafické výstupy. Pomocí služeb nacházejících se na internetových stránkách a pomocí speciálního formátu CDF je možné vytvořit takové materiály, které mohou být veřejně prezentovány bez potřeby vlastnitvi samotného programu. Mathematica je neustále hojně využívaným programem ať už pro práci, nebo ke studiu. V současnosti využívá i také přístupu na vlastní cloudovou službu, kde si uživatel může spravovat své soubory. I přes velkou rozšířenost tohoto programu není dostupný ve všech světových jazycích. Veškeré příkazy a nápověda jsou v angličtině a proto je nutné alespoň základní znalost tohoto jazyka. Instalační soubor má velikost kolem 900 MB. Po instalaci kompletní verze zabírá přibližně 3 GB, přičemž polovinu zabírá samotná dokumentace k tomuto programu.

Aplikaci lze rozdělit na 3 části:

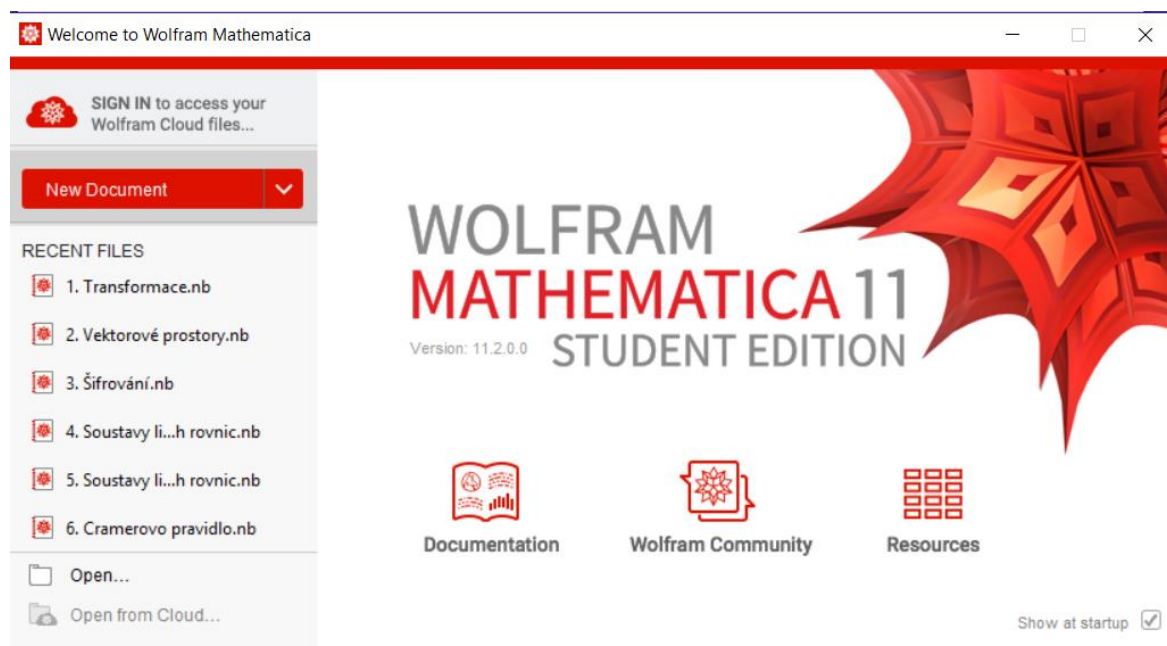
- **Kernel** – Výpočetní jádro programu, které provádí veškeré operace. Při případné chybě stačí znovu pouze spustit výpočetní jádro
- **Front end** – vlastní uživatelské prostředí, které umožňuje komunikaci Kernelu s uživatelem. Základem toho prostředí jsou notebooky, tedy soubory, do kterým zapisujeme své příkazy.
- **Packages** – Doplnující systémové knihovny. Tyto knihovny se načítají pouze v případě potřeby.

Wolfram Mathematica začal vytvářet Stephen Wolfram v roce 1986. Pro tento účel založil firmu Wolfram Research a roku 1988 s kolektivem spolupracovníků vytvořil program Mathematica verze 1. V dnešní době se na vývoji aplikace podílí přes 1000 programátorů.

Program je hojně využíván jak, studenty tak i lidmi, kteří jej potřebují k práci.

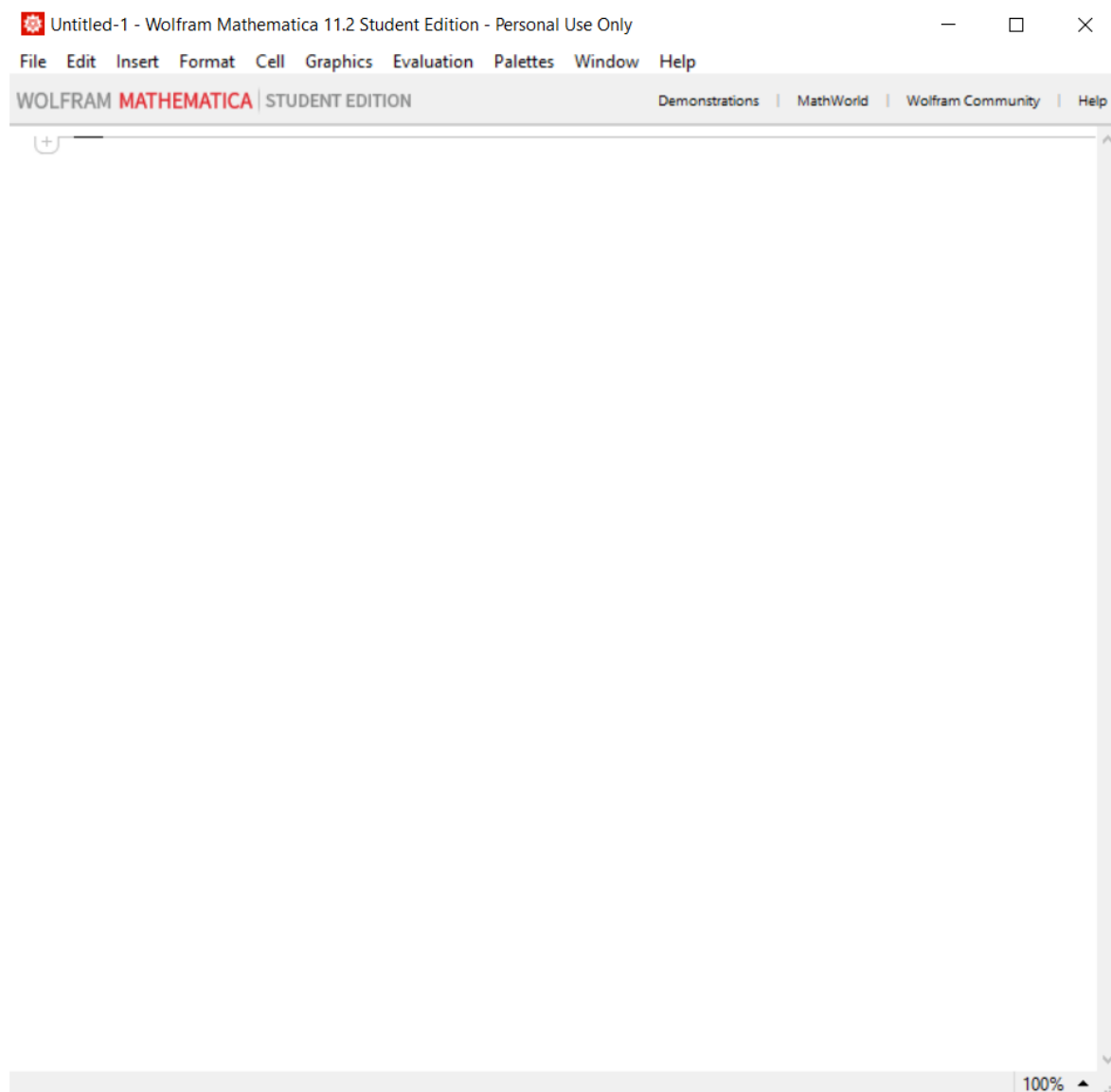
1.1 První spuštění

Při prvním spuštění se zobrazí informační okno programu.



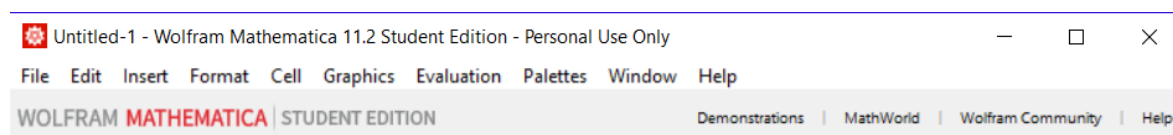
Obrázek 1: Informační okno

V tomto informačním okně se můžeme přihlásit ke svému účtu pomocí přihlašovacího formuláře anebo si vybrat z nabídky posledních otevřených souborů. V levé části si můžeme vybrat, jaký typ souboru chceme otevřít. Pokud bychom si nepřáli zobrazování tohoto okna při dalším spuštění, odškrtneme v pravém dolním rohu nabídku Show at startup. Při otevření nového notebooku se objeví se prázdný dokument. V okamžiku zápisu prvního textu začne vytvářet jednotlivé textové buňky, do kterých se ukládá celý obsah dokumentu. Jednotlivé buňky jsou označeny na pravé straně hranatými závorkami. Podle toho, co se v buňce nachází, se určuje i její obsah (vstup, výstup, nadpis). Styl buňky si můžeme zvolit i z menu nacházejícího se v horní části okna. Pokud bychom chtěli vytvořit novou buňku, je třeba umístit kurzor na pozici, kdy se změní na vodorovnou značku. Pravým tlačítkem myši si pak vybereme styl buňky.

*Obrázek 2: Prázdný dokument*

1.2 Menu

Samotná aplikace je pouze hlavní nabídka. Ostatní okna se mohou libovolně zavírat, ale je třeba si dát pozor, že při zavření menu se program ukončí.

*Obrázek 3: Menu programu*

V nabídce **File** lze nalézt všechny možnosti související s tvorbou a úpravou nových dokumentů, jejich ukládáním, tištěním dokumentů, přehledem posledních souborů a ukončením aplikace.

Nabídka **Edit** obsahuje položky pro práci se schránkou a vyhledáváním. Poslední možností této nabídky jsou Preferences. Zde se dá nastavit vše kolem aplikace samotné.

Insert je nabídka určená pro vkládání objektů a složitějších matematických výrazů. Vedle těchto výrazů se nachází také jejich klávesové zkratky pro rychlejší vkládání. Ty se využívají především při textovém zápisu výrazů.

Položka **Format** se zaměřuje na nastavení formátu jednotlivých buněk, formátu písma, zarovnání textu a stylu sešitu samotného.

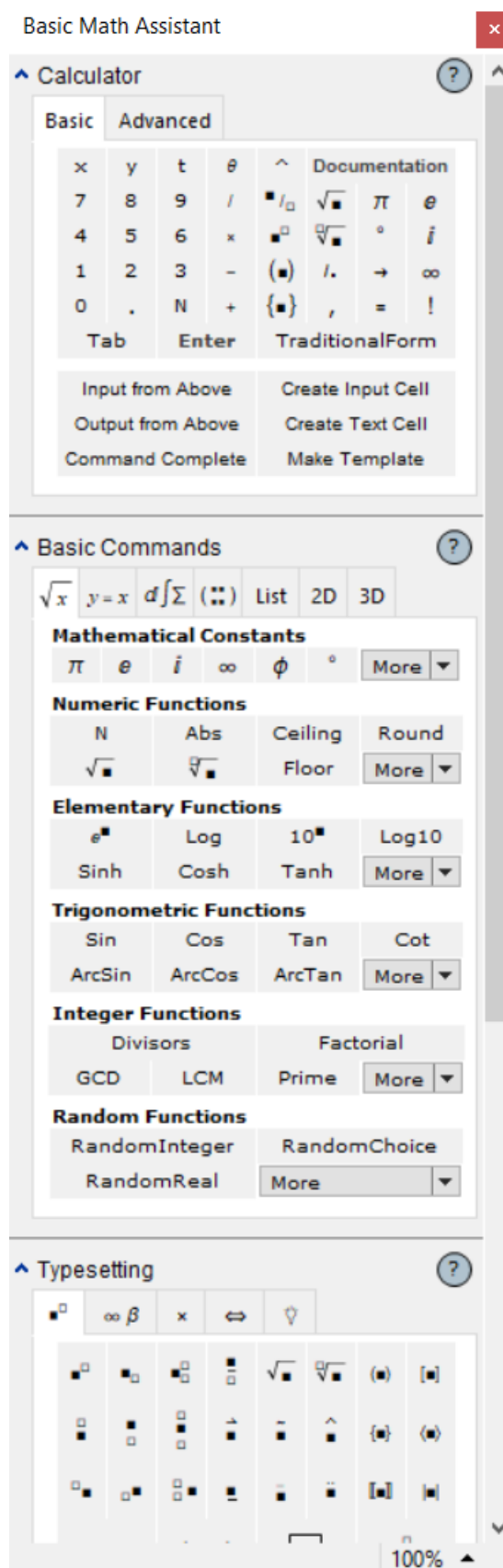
Cell je nabídka činností pro libovolné zacházení s buňkami. Ty můžeme seskupovat, slučovat, rozdělovat, pojmenovávat, nebo odstranit.

Graphics je výběrem činností s grafickými objekty. Především jejich umístění a zarovnání.

Položka označená **Evaluation** slouží k nastavení výpočtů našich buněk. Zde lze krokovat výpočty a přepočítat vybrané buňky.

Nabídka **Palettes** slouží k obsluze palet. Ty jsou zde proto, aby usnadnily zapisování matematických formulí, výrazů nebo objektů.

Standardní nabídkou je položka **Window** a **Help**, ty obsahují funkce uspořádání oken aplikace a podpůrné prostředky obsahující dokumentaci programu a internetových podpůrných zdrojů. Pokud by Mathematica nebyl připojena k internetu je možné, že přestane správně pracovat, jelikož dohledává údaje z externích databází.



Obrázek 4: Základní paleta

1.3 Základní práce s programem

Program Mathematica rozlišuje malá a velká písmena. Všechny funkce a dané příkazy začínají velkým písmenem, ostatní písmena jsou malá. Pokud je název funkce složený z více slov, je vždy první písmeno jednotlivých slov velké. Parametry příkazů a funkcí se vkládají do hranatých závorek. Složené závorky se používají výhradně pro seznamy a tabulky. Kulaté závorky jsou klasické závorky v matematických výrazech.

Základní aritmetické operace:

$x+y$ – sčítání

```
In[32] := 2+2
```

```
Out[32]= 4
```

$x-y$ – odčítání

```
In[32] := 5-2
```

```
Out[32]= 3
```

$x*y$, xy – násobení

```
In[32] := 8*7
```

```
Out[32]= 56
```

x/y – dělení

```
In[32] := 35/7
```

```
Out[32]= 5
```

x^y – dělení

```
In[32] := 2^3
```

```
Out[32]= 8
```

Priorita těchto operací lze snadno upravovat pomocí závorek.

Samozřejmostí je, že i tento program má celou řadu matematických funkcí, z nichž ty nejvýznamnější jsou například:

Sqrt[x] – odmocňování

```
In[32] := Sqrt[4]
```

```
Out[32]= 2
```

Exp[x] – exponenciální funkce e^x

In[32]:= Exp[3]
Out[32]= e^3 **Log[x]** – přirozený logaritmus

In[32]:= N[Log[15]]

Out[32]= 2.70805

Log[z,x] – logaritmus x o základu n

In[32]:= Log[10, 1000]

Out[32]= 3

n! – faktoriál

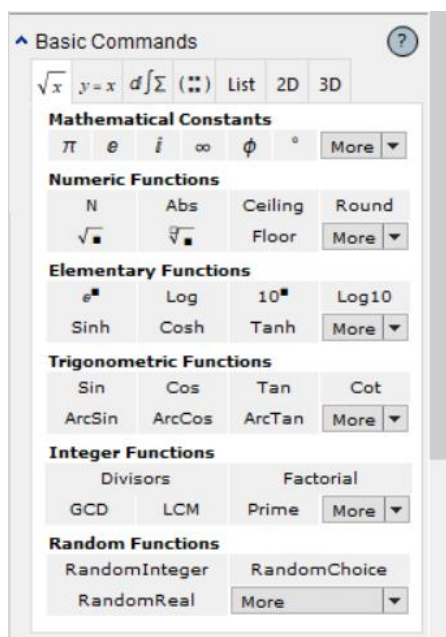
In[32]:= 4!

Out[32]= 24

Sin[], Cos[], Tg[], Cotg[] – trigonometrické funkce

Argumentem trigonometrických funkcí je úhel v radiánech.

Pokud bychom chtěli místo hodnoty úhlu zadat hodnotu $\text{Pi}(\pi)$, můžeme k tomuto účelu využít nástrojovou paletu Basic Math Assistant.



Obrázek 5: Paleta pro vkládání speciálních znaků

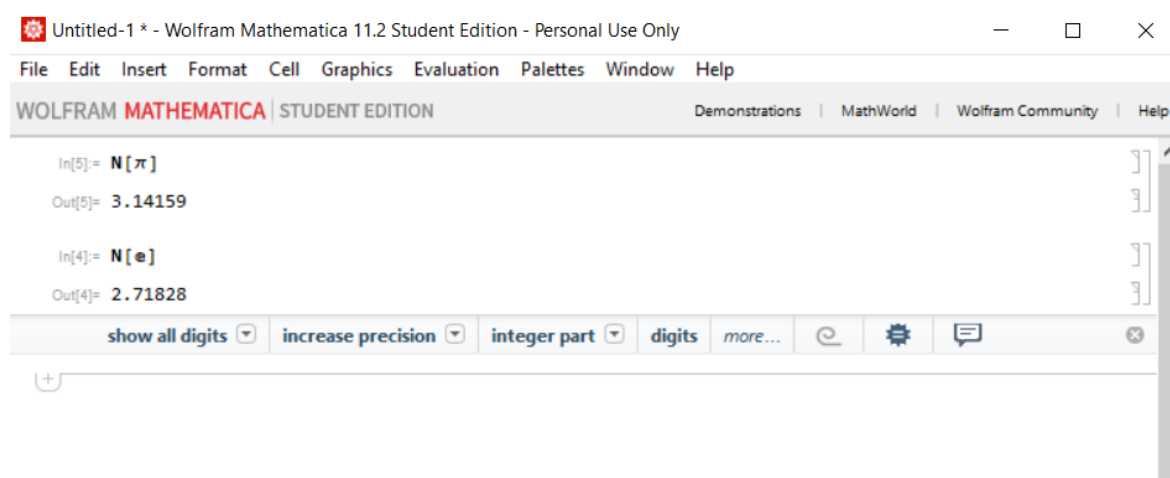
Pokud bychom chtěli jako argument zadat přesnou hodnotu úhlu, je třeba číselnou hodnotu doplnit slovem Degree.

```
In[32]:= Sin[90Degree]
```

```
Out[32]= 1
```

```
In[33]:= Sin[ $\pi$ ]
```

```
In[33]:= 0
```

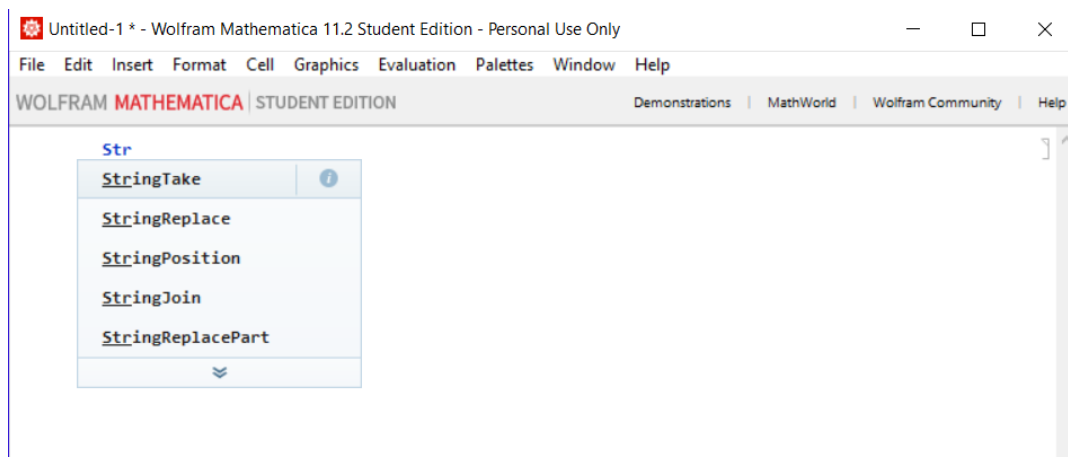


Obrázek 6: Konstanty ve Wolfram Mathematica

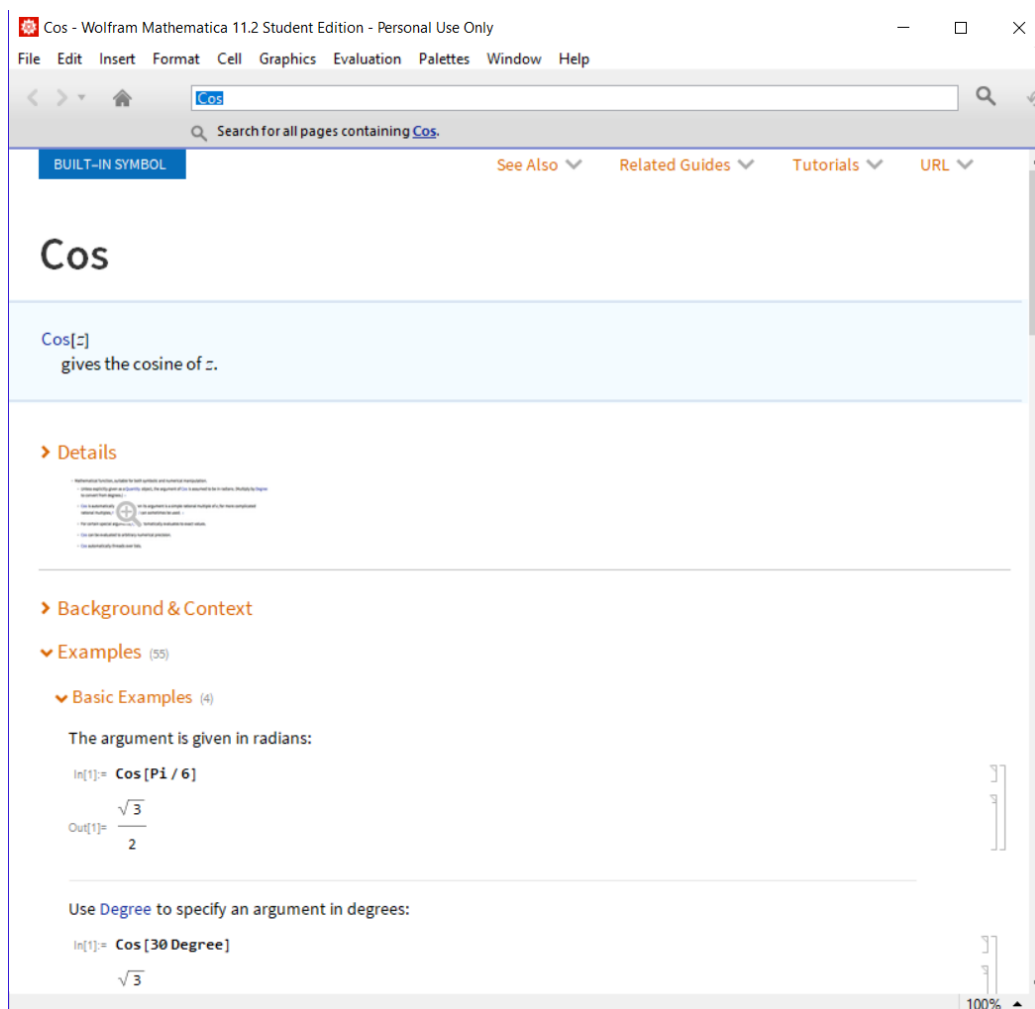
Jelikož si jádro programu indexuje výsledky, můžeme se na předchozí výsledek odkázat znakem %, nebo na vybraný výpočet pomocí indexu symbolem % X, kde X představuje číslo výpočetní buňky.

1.4 Náповěda programu

Program Wolfram Mathematica má rozsáhlý systém aktivní nápovědy. Celá nápověda je tvořena pomocí aktivních notebooků. Ukázkové příklady si můžeme kopírovat do našich projektů a upravovat podle toho jak to zrovna potřebujeme. Nápovědu k příkazu vyvoláme stisknutím klávesy F1. Pokud neznáme celý název funkce, stačí napsat několik úvodních písmen z názvu funkce a z vyskakovací nabídky vybrat tu požadovanou.

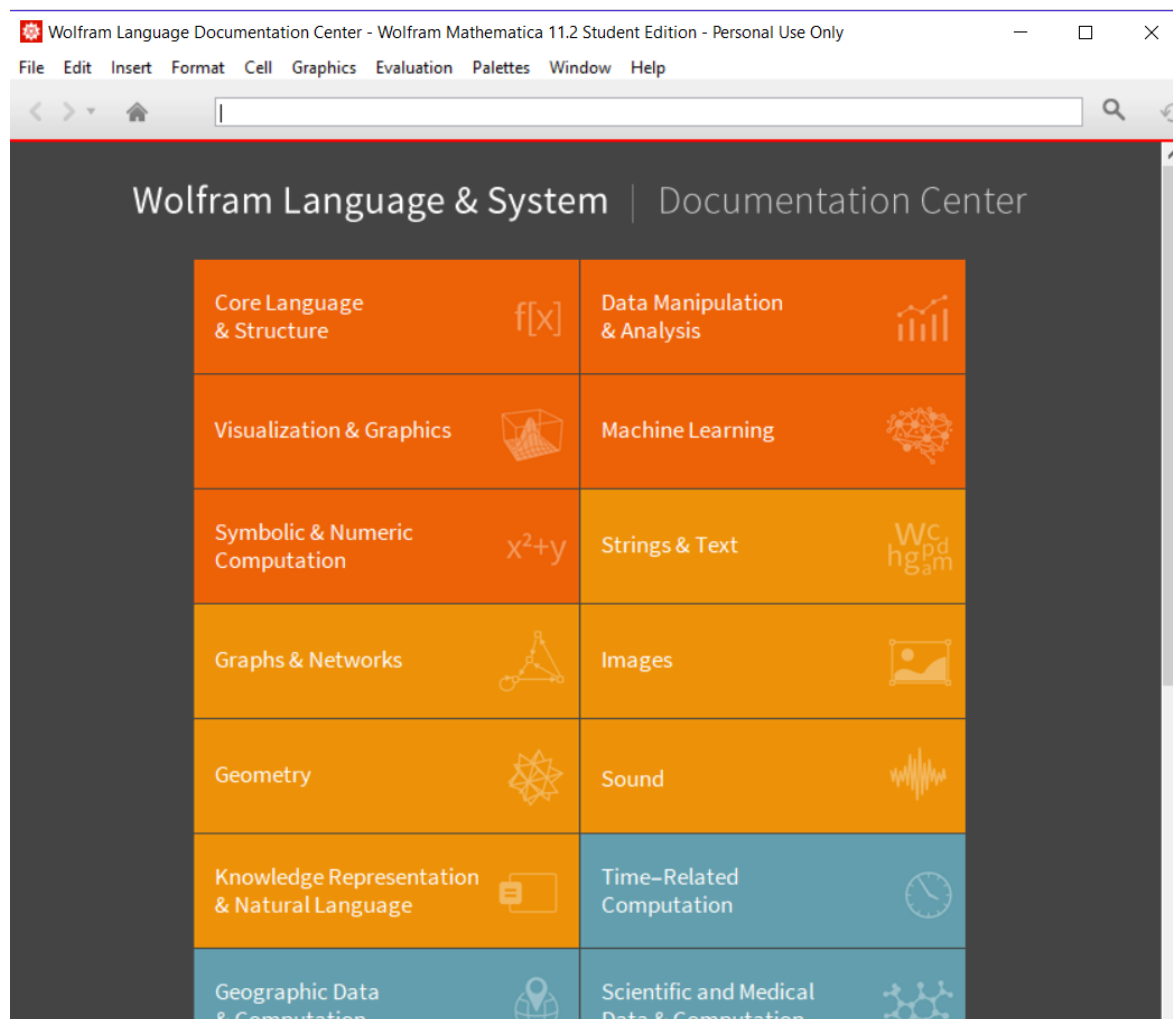


Obrázek 7: Vyskakovací okno nápovědy



Obrázek 8: Nápověda pro funkci cosinus

Kromě dynamické nápovědy nabízí tento program ucelenou elektronickou dokumentaci (klávesa F1).



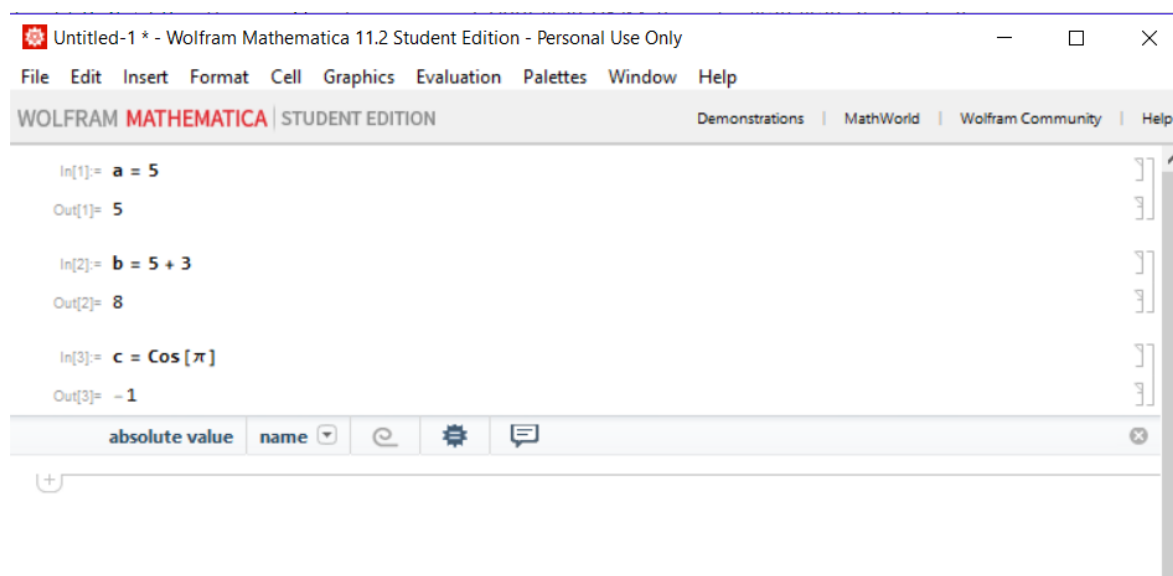
Obrázek 9: Oficiální dokumentace program

Wolfram Mathematica také spolupracuje s webovým serverem <http://www.wolfram-alpha.com/>, kde si je možné vyzkoušet veškeré příkazy tohoto programu bez nutnosti instalace.

2 PRÁCE S PROMĚNNÝMI

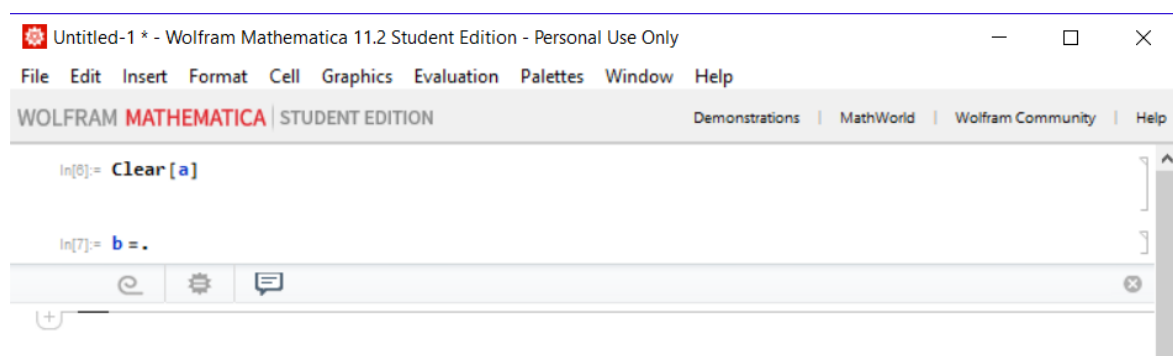
2.1 Definice proměnné

Algebraické výrazy se skládají nejen z čísel, operací a závorek na určení preferencí, ale také z proměnných, které jsou označené písmeny. Názvy proměnných mohou být libovolné, jedinou podmínkou je, aby začínaly písmenem. Jediná provedená definice je platná pro veškeré její další použití.



Obrázek 10: Definice proměnných

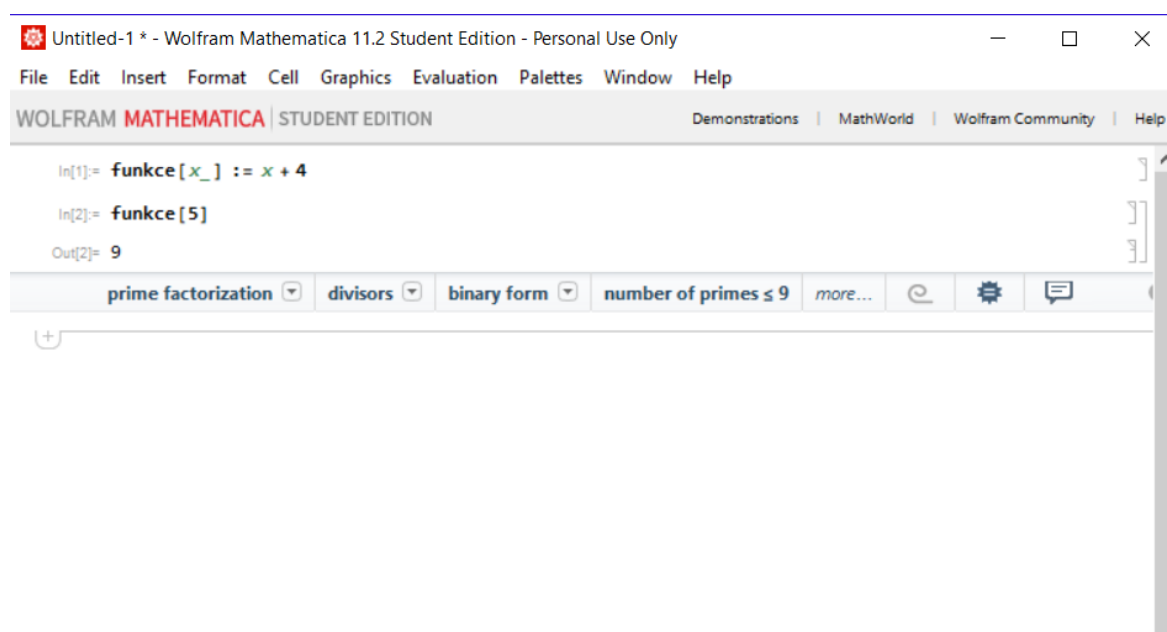
Před dalším zadáním, nebo použitím proměnné je vhodné její uvolnění a zrušení všech předšlých nastavení příkazem `Clear[X]`, nebo přiřazením hodnoty `X = ..`. Odstranění všech hodnot a definic provedeme příkazem `ClearAll[X]`.



Obrázek 11: Uvolnění proměnných

2.2 Definice funkce

Předpis pro vytvoření funkce můžeme zapsat: $f[x_ , y_] := \text{výraz}$; kde f je název funkce; x , y jsou proměnné funkce, aby počítač rozpoznal, kde funkci definujete a kde vypočítáváme její hodnotu, značí se v definici proměnné s podtržítkem. Za proměnnými následuje samotný výraz funkce. Při definici funkce se používá přiřazovacího symbolu $:=$. Pokud bychom chtěli funkci odstranit, použijeme funkci `Clear[]`.



Obrázek 12: Vytvoření jednoduché funkce

Výše uvedená funkce pracuje tak, že se k proměnné bude vždy přičítat číslo 4. Funkci voláme pomocí jejího názvu a po vyplnění argumentu zvoleným číslem, bude aplikován zadaný příkaz. Argumentem nemusí být pouze čísla. Mohou se používat výrazy, textové řetězce ale i symboly. Funkce mohou mít libovolný počet proměnných.

Je pouze na uživateli, jak svou funkci nazve. Je nutné si ale dát pozor aby název nebyl shodný s názvem integrované funkce programového prostředí. Také je důležité si dát pozor, zda nebyl název funkce už použit v notebooku dříve. Došlo by totiž k přepsání.



Obrázek 13: Odstranění definice funkce

2.3 Lokální globální a lokální proměnné

Definice globální proměnné se provádí klasickým způsobem:

```
In[32] := a=10
```

```
Out[32]= 10
```

Pod pojmem lokální proměnná si lze představit proměnnou, která je definována pouze uvnitř nějaké funkce nebo cyklu. Pro definice lokální proměnné se používá funkce `Module[]`. Po její aplikaci bude mít proměnná určitou hodnotu uvnitř funkce `Module`, ale její globální na-definovaná hodnota se nezmění.

```
In[32] := Module[{a}, a=5; a]
```

```
Out[32]= 5
```

```
In[33] := a
```

```
Out[33]= 10
```


3 ROVNICE

Při řešení mnohých úloh je potřeba sestavení rovnic a jejich soustav. Rovnice lze popsat tak, že hledáme řešení pro všechny neznámé takové, aby se levá strana rovnice rovnala straně pravé. Na řešení některých rovnic existují standardní obecné postupy. Pro jednoduché rovnice budeme volit funkci `Solve[]` nebo `Roots[]`. Vstupními parametry jsou rovnice a seznam neznámých, pro které chceme pomocí této funkce najít řešení. Pro numerické výpočty existuje funkce `NSolve[]`. Pro výrazně složitější rovnice se používá funkce `FindRoot[]`. Program `mathematica` hledá vždy řešení v oboru komplexních čísel, ale obor řešení můžeme omezit parametry `Reals`, `Integers` a `Complexes` podle potřeb.

`Roots[eqn,var]` – najde kořeny rovnice *eqn* pro proměnnou *var*

```
In[32]:= Roots[x^2-2 x-1==0,x]
```

```
Out[32]= x==1+√2 | |x==1-√2
```

`Solve[eqn,var]` – vyřeší rovnici *eqn* pro proměnnou *var*

```
In[32]:= Solve[{x1 + 2 x2 + 3 x3 == 14, 3 x1 + 2 x2 + x3 == 10,  
 3 x1 + x2 + 2 x3 == 11}, {x1, x2, x3}]
```

```
Out[32]= {{x1 -> 1, x2 -> 2, x3 -> 3}}
```

`NSolve[eqn,var]` – vyřeší rovnici *eqn* pro proměnnou *var* a převede výsledek na numerický apřibližovaný tvar

```
In[32]:= NSolve[x^5-2 x+3==0,x,Reals]
```

```
Out[32]= {{x->-1.42361}}
```

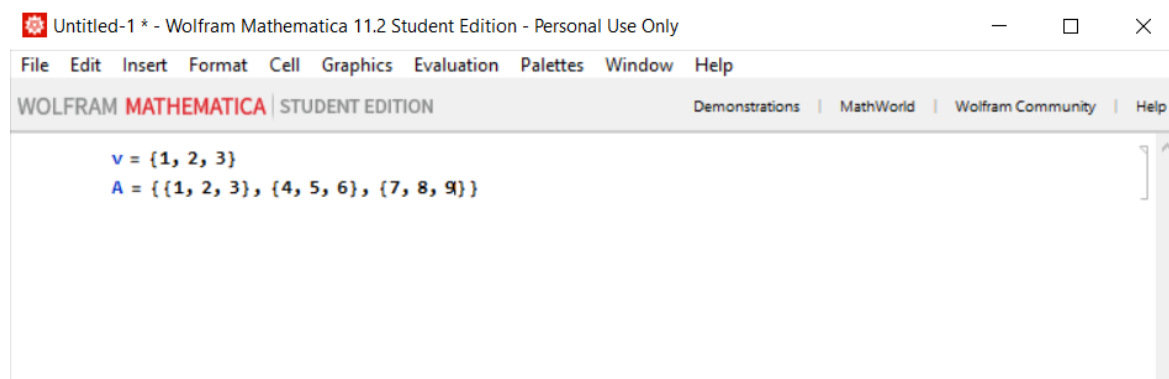
`FindRoot[eqn,x,x0]` – nalezne numerické řešení rovnice *eqn* pro počáteční hodnotu *x0*

```
In[32]:= FindRoot[Cos[x]==x,{x,0}]
```

```
Out[32]= {x->0.739085}
```

4 VEKTORY A MATICE

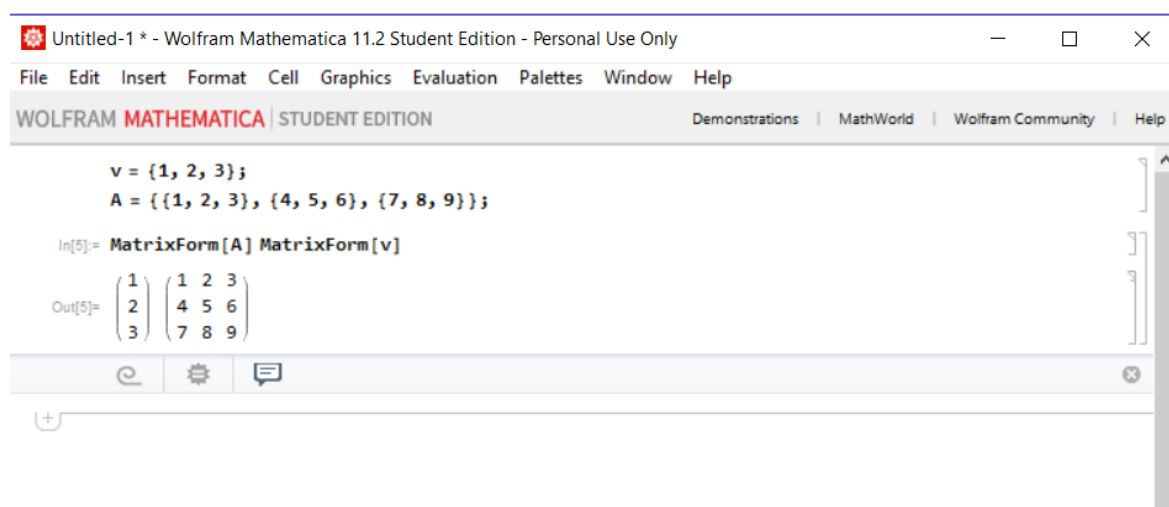
V programu Wolfram Mathematica jsou vektory reprezentovány seznamem prvků a matice jsou reprezentovány seznamem jejich řádků. Těchto seznamů v seznamu je tolik, kolik má matice řádků.



Obrázek 14: Reprezentace matic a vektorů

Vektory a matice lze v prostředí Mathematica vytvářet pomocí složených závorek (jako je uvedeno na obrázku výše), nabídky Input a nebo pomocí nástrojových palet.

Pro lepší znázornění matic se používá integrovaná funkce `MatrixForm[]`, která vypíše matici v tzv. maticovém zápise. Funkce není omezena pouze na matice, ale lze ji použít i pro vektory.



Obrázek 15: Maticový zápis

4.1 Funkce pro přístup k částem vektorů a matic

Definice vektorů a matic:

```
u={1,2,3};
```

```
v={3,2,1};
```

```
A={{1,2},{4,5}};
```

```
B={{5,6},{8,9}};
```

v[[i]] nebo **Part[v,i]** – vypsání *i*-tého členu vektoru *v*

```
In[32]:= u[[2]]
```

```
Out[32]=2
```

m[[i,j]] nebo **Part[m,j,i]** – vypsání prvku *i*-tého řádku a *j*-tého sloupce

```
In[32]:= A[[1,3]]
```

```
Out[32]= 3
```

m[[i]] nebo **Part[m,i]** – vypsání *i*-tého řádku matice

```
In[32]:= A[[1]]
```

```
Out[32]= {1,2,3}
```

m[[All,j]] nebo **Part[m,All,j]** – vypsání *j*-tého sloupce matice

```
In[32]:= A[[All,1]]
```

```
Out[32]= {1,4,7}
```

Tr[m,List] – vypíše diagonální prvky matice *m*

```
In[32]:= Tr[A,List]
```

```
Out[32]= {1,5,9}
```

4.2 Operace s vektory

Range[n] – vytvoří vektor {1,2,...,n}

```
In[32]:= Range[3]
```

```
Out[32]= {1,2,3}
```

Range[n1,n2] – vytvoří vektor v rozmezí *n*₁ až *n*₂

```
In[32] := Range[3, 6]
```

```
Out[32] = {3, 4, 5, 6}
```

Length[v] – vypsání počtu prvků vektoru

```
In[32] := Length[u]
```

```
Out[32] = 3
```

ColumnForm[v] – vypsání vektoru ve formě sloupce

```
In[32] := ColumnForm[u]
```

```
Out[32] = 1
```

```
2
```

```
3
```

v+w – sečte vektory v a w

```
In[32] := u+v
```

```
Out[32] = {4, 4, 4}
```

v-w – odečte vektory v a w

```
In[32] := u-v
```

```
Out[32] = {-2, 0, 2}
```

v*w – vynásobí vektory v a w

```
In[32] := u*v
```

```
Out[32] = {3, 4, 3}
```

v.w – skalární součin v a w

```
In[32] := u.v
```

```
Out[32] = 10
```

4.3 Operace s maticemi

DiagonalMatrix[{1,2,3}] – vytvoří matici s prvky 1,2,3 na hlavní diagonále

```
In[32] := DiagonalMatrix[{1, 2, 3}]
```

```
Out[32] = {{1, 0, 0}, {0, 2, 0}, {0, 0, 3}}
```

Dimensions[A] – vypíše rozměr matice A

```
In[32] := Dimensions[A]
```

```
Out[32]= {3,3}
```

Det[A] – vypsání determinantu matice A

```
In[32]:= Det[A]
```

```
Out[32]= 0
```

Transpose[A] – vypsání transponované matice k matici A

```
In[32]:= Transpose[A]
```

```
Out[32]= {{1,4,7},{2,5,8},{3,6,9}}
```

Inverse[A] – vypsání inverzní matice k matici A

```
In[32]:= Inverse[{{1,7,3},{4,5,6},{7,8,9}}]
```

```
Out[32]={{-(1/10),-(13/10),9/10},{1/5,-(2/5),1/5},{-(1/10),41/30,-(23/30)}}
```

A+B – sečte matice A a B

```
In[32]:= A+B
```

```
Out[32]= {{6,8},{12,14}}
```

A-B – odečte matice A a B

```
In[32]:= A-B
```

```
Out[32]= {{-4,-4},{-4,-4}}
```

c*B – vynásobí matici B skalárem c

```
In[32]:= 2*B
```

```
Out[32]= {{10,12},{16,18}}
```

A*B – vynásobí matice A a B

```
In[32]:= A*B
```

```
Out[32]= {{5,12},{32,45}}
```

A.B – maticové násobení A a B

```
In[32]:= A.B
```

```
Out[32]= {{21,24},{60,69}}
```

4.4 Funkce Table

Pro definici nové matice či vektoru lze také použít funkci `Table[]`. Ta usnadňuje práci zejména při tvorbě složitějších struktur

`Table[x, {n}]` – vypíše vektor obsahující n kopií hodnoty x

```
In[32]:= Table[6, {4}]
```

```
Out[32]= {6, 6, 6, 6}
```

`Table[f, {i, max}]` – vytvoří vektor tak, že vyhodnotí funkci pro $i \leq \text{max}$

```
In[32]:= Table[i+1, {i, 3}]
```

```
Out[32]= {2, 3, 4}
```

`Table[f, {i, min, max}]` – vytvoří vektor tak, že vyhodnotí funkci pro $\text{min} \leq i \leq \text{max}$

```
In[32]:= Table[i+1, {i, 2, 5}]
```

```
Out[32]= {3, 4, 5, 6}
```

5 RELAČNÍ OPERÁTORY

Equal[x,y] nebo **x==y** – číselné srovnání, vrací hodnotu *True* pokud se hodnoty rovnají

```
In[32]:= Equal[2,2]
```

```
Out[32]= True
```

SameQ[x,y] nebo **x===y** – vrací hodnotu *True* pokud jsou hodnoty totožné

```
In[32]:= SameQ[6,3.]
```

```
Out[32]= False
```

Unequal[x,y] nebo **x!=y** – číselné srovnání, vrací hodnotu *True* pokud se hodnoty nerovnají

```
In[32]:= Unequal[2,2]
```

```
Out[32]= False
```

UnsameQ[x,y] nebo **x!=y** – vrací hodnotu *True* pokud nejsou hodnoty totožné

```
In[32]:= UnsameQ[6,3.]
```

```
Out[32]= True
```

Less[x,y] nebo **x<y** – vrací hodnotu *True* pro *x* menší než *y*

```
In[32]:= 2<5
```

```
Out[32]= True
```

LessEqual[x,y] nebo **x<=y** – vrací hodnotu *True* pro *x* menší nebo rovno *y*

```
In[32]:= 2<=2
```

```
Out[32]= True
```

Greater[x,y] nebo **x>y** – vrací hodnotu *True* pro *x* větší než *y*

```
In[32]:= 10>11
```

```
Out[32]= False
```

GreaterEqual[x,y] nebo **x>=y** – vrací hodnotu *True* pro *x* větší nebo rovno *y*

```
In[32]:= 10>=7
```

```
Out[32]= True
```

6 2D GRAFY FUNKCÍ

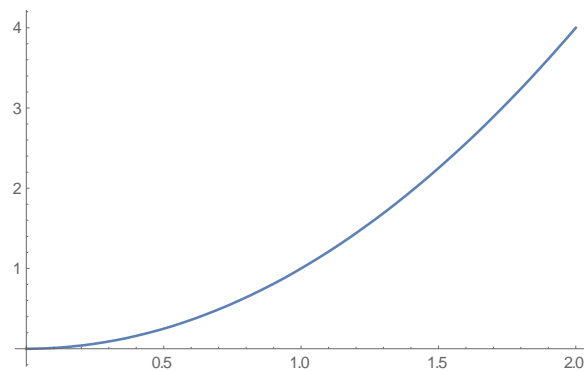
Pro vykreslení grafů v 2D prostoru využívá Mathematica funkci `Plot`.

V argumentech této funkce se zadává vykreslovaná funkce, proměnná pro kterou se funkce vykresluje a omezení proměnné.

`Plot[f, {x, xmin, xmax}]` – vykreslí graf funkce f pro $xmin \leq x \leq xmax$

In[32]:= `Plot[x^2, {x, 0, 2}]`

Out[32]=



Obrázek 16: Graf funkce

6.1 Parametry funkce `Plot`

Funkce **`Plot`** má obrovské množství parametrů, které je možno měnit tak, aby výsledný vzhled grafu odpovídal přesně požadavku uživatele. Pokud bychom žádné parametry nenastavili, graf se vykreslí v základním nastavení, které je viditelné na předchozím obrázku.

Axes – parametry os

Axes-> `True` – obě osy budou na výstupu vykresleny

Axes-> `False` – obě osy nebudou na výstupu vykresleny

Axes-> `{True, False}` – osa x se vykreslí, osa y ne

Axes-> `{False, True}` – osa x se nevykreslí, osa y ano

AxesLabel – parametr popisu os grafu

AxesLabel-> none – osy bez popisku

AxesLabel-> label – obě osy jsou popsány

AxesLabel-> {xlabel,ylabel} – specifické popisky pro jednotlivé osy

AxesOrigin – parametr průsečíků os

AxesOrigin-> {x,y} – průsečík se nachází v bodě {x,y}

AxesOrigin-> Automatic – Mathematica určí průsečík os automaticky

AxesStyle – parametr pro nastavení vzhledu os

AxesStyle-> style – styl „style“ se aplikuje na obě osy

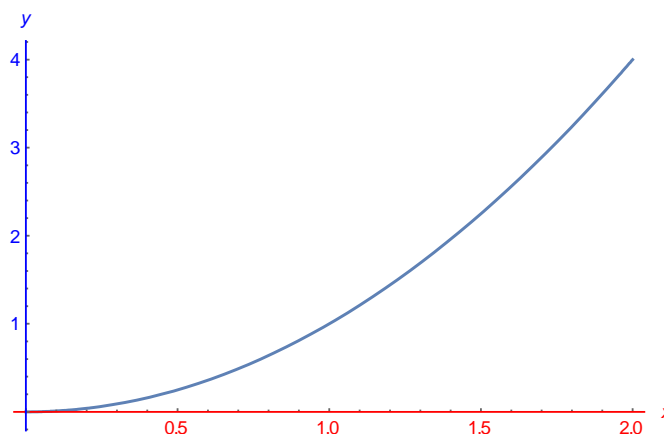
AxesStyle-> {{xstyle},{ystyle}} – rozlišné styly pro obě osy zvlášť

Ticks – parametr pro zobrazení značek os

Ticks-> None – graf nemá značky měřítka na osách

Ticks -> Automatic – značky měřítka jsou automaticky vykresleny

```
In[32]:=Plot[x^2,{x,0,2},AxesLabel->{x,y},  
          AxesStyle->{{Red},{Blue}}]
```



Obrázek 17: Změna stylování os grafu

Frame – parametr orámování

Frame-> True – graf bude orámován

Frame-> None – graf nebude orámován

FrameLabel – parametr pojmenování stran rámu

FrameLabel-> None – obě osy budou na výstupu vykresleny

FrameLabel -> {xlabel, ylabel} – pojmenuje spodní a levou část rámu

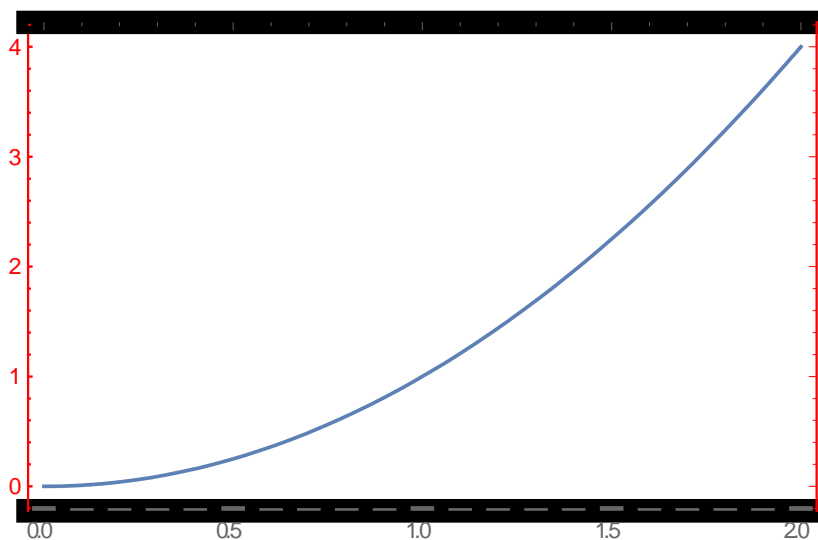
FrameLabel -> {xplabel, yplabel} – pojmenuje horní a pravou část rámu

FrameStyle – parametr definice stylu orámování

FrameStyle-> style – aplikuje jednotný styl pro celý rám

FrameStyle-> {{xmstyle, ymstyle}} – různý styl pro různé části rámu

```
In[32]:=Plot[x^2,{x,0,2},AxesLabel->{x,y},Frame->True,FrameStyle->{Red,Thickness[0.03]}]
```



Obrázek 18: Změna orámování grafu

PlotLabel – parametr pro pojmenování grafu

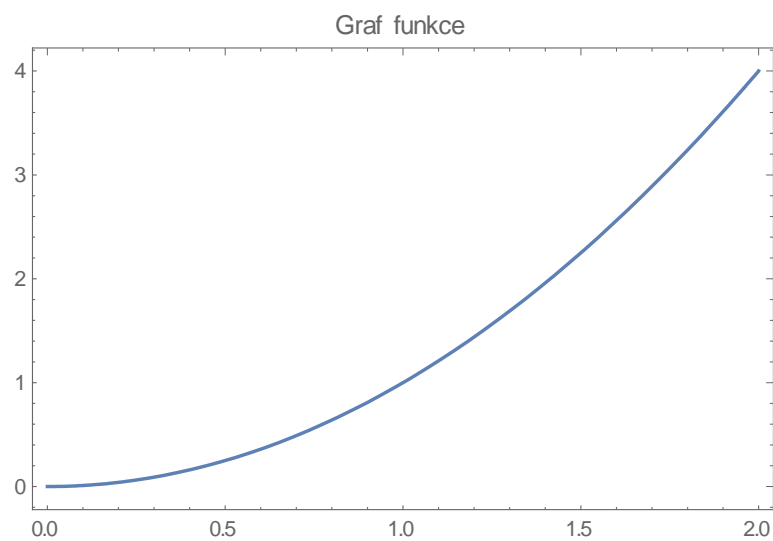
PlotLabel-> none – graf nemá pojmenování

PlotLabel-> label – graf je pojmenován jako „label“

TextStyle – parametr pro nastavení stylu písma grafu

TextStyle-> {"style"} – výběr stylu písma

```
In[32]:=Plot[x^2,{x,0,2},AxesLabel->{x,y},Frame->True,PlotLabel->"Graf funkce"]
```



Obrázek 19: Změna popisku grafu

AspectRatio – parametr poměru výšky a šířky

AspectRatio-> 1/GoldenRatio – výchozí nastavení pro 2D graf

AspectRatio-> Automatic – výchozí nastavení pro 3D graf

AspectRatio-> x/y – nastavení libovolného poměru

GridLines – parametr pro vytvoření mřížky

GridLines-> None – graf bez mřížky

GridLines -> Automatic – mřížka je vykreslena automaticky

Background – parametr pro nastavení pozadí grafu

Background-> None – graf bez pozadí

Background -> Automatic – graf má stejné pozadí jako notebook

Background -> RGBColor[red,green,blue] – graf má nastavené pozadí

PlotRange – parametr definice mezí os

PlotRange-> All – všechny body budou zahrnuty do os

PlotRange-> Automatic – vzdálené body jsou vynechány

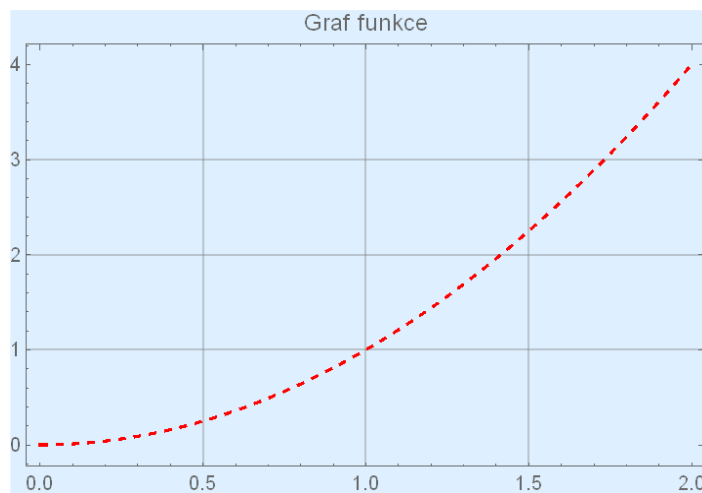
PlotRange -> {min,max} – explicitní definování mezí os

PlotStyle – parametr definice stylu křivky

PlotStyle-> style – nastavení pro všechny křivky tvořící graf

PlotStyle-> {{style1a},{style2b},...} – nastavení stylu více křivek

```
In[32]:=Plot[x^2,{x,0,2},AxesLabel->{x,y},Frame->True,PlotLabel->"Graf funkce",Background->LightBlue,PlotStyle->{Dashed,Red},GridLines->Automatic]
```



Obrázek 20: Změna stylu grafu i pozadí

ZÁVĚR

Cílem této práce bylo seznámit čtenáře se základním oborem matematiky, kterým je lineární algebra. Text bakalářské práce byl vytvořen jako podpůrný text pro studenty navštěvující Math Support Centre. Témata, která jsou probírána se řadí mezi ty nejzákladnější a nejnutnější k pochopení principu fungování lineární algebry.

V první polovině bakalářské práce jsem vysvětlil pojem vektorový prostor, co musí vektorové prostory splňovat a jejich vybrané příklady. Za těmito příklady je uvedena také teorie lineárních kombinací vektorů. Následuje teorie týkající se matic. Ta obsahuje podkapitoly jako je představení různých typů matic, operace s maticemi a elementární úpravy matic.

V druhé polovině práce se zabývám determinanty a teorií s nimi spjatou. Ilustruji výpočty determinantů pro různě velké čtvercové matice, za kterými následuje definice inverzních matic. V následující kapitole je probírána teorie lineárních rovnic a hlavně způsoby, jakými soustavy počítat.

Veškeré příklady uvedené v práci jsou řešeny tak, aby jim porozuměl i naprostý laik. K dispozici jsou slovně popsány postupy a možnosti řešení. Pro ještě lepší představu jsou představeny možnosti řešení v matematickém software Wolfram Mathematica. Tento program je velice vhodný pro účely výuky lineární algebry, jelikož je snadno ovladatelný a srozumitelný.

Lineární algebra je jeden ze stavebních kamenů matematiky a studenti by se s ní neměli setkávat až na vysokých školách. Matematika jako taková se jeví mnohem snazší s poznatky, které vzniknou studiem této disciplíny. Srdečně doufám, že tyto podpůrné materiály najdou uplatnění v již zmíněném centru a pomohou studentům Univerzity Tomáše Bati nejenom z fakulty aplikované informatiky.

SEZNAM POUŽITÉ LITERATURY

- [1] *Mathematica - příručka s příklady pro učitele a studenty* [online]. Praha, 2012 [cit. 2018-05-24]. Dostupné z: <https://www.pslib.cz/jaromir.oscadal/PRG%20-V1/Mathematica%20-%20p%C5%99%C3%ADru%C4%8Dka3.pdf>. Příručka. Střední průmyslová škola strojní a elektrotechnická.
- [2] CHRAMCOV, Bronislav. *Základy práce v prostředí Mathematica* [online]. Zlín, 2005 [cit. 2018-05-24]. Dostupné z: <http://vyuka.fai.utb.cz/course/view.php?id=293>. Skripta. Univerzita Tomáše Bati.

SEZNAM OBRÁZKŮ

<i>Obrázek 1: Informační okno</i>	<i>4</i>
<i>Obrázek 2: Prázdný dokument</i>	<i>5</i>
<i>Obrázek 3: Menu programu</i>	<i>5</i>
<i>Obrázek 4: Základní paleta</i>	<i>7</i>
<i>Obrázek 5: Paleta pro vkládání speciálních znaků</i>	<i>9</i>
<i>Obrázek 6: Konstanty ve Wolfram Mathematica</i>	<i>10</i>
<i>Obrázek 7: Vyskakovací okno nápovědy</i>	<i>11</i>
<i>Obrázek 8: Nápověda pro funkci cosinus</i>	<i>11</i>
<i>Obrázek 9: Oficiální dokumentace programu</i>	<i>12</i>
<i>Obrázek 10: Definice proměnných</i>	<i>13</i>
<i>Obrázek 11: Uvolnění proměnných</i>	<i>13</i>
<i>Obrázek 12: Vytvoření jednoduché funkce</i>	<i>14</i>
<i>Obrázek 13: Odstranění definice funkce</i>	<i>14</i>
<i>Obrázek 14: Reprezentace matic a vektorů</i>	<i>17</i>
<i>Obrázek 15: Maticový zápis</i>	<i>17</i>
<i>Obrázek 16: Graf funkce</i>	<i>23</i>
<i>Obrázek 17: Změna stylování os grafu</i>	<i>24</i>
<i>Obrázek 18: Změna orámování grafu</i>	<i>25</i>
<i>Obrázek 19: Změna popisku grafu</i>	<i>26</i>
<i>Obrázek 20: Změna stylu grafu i pozadí</i>	<i>27</i>