

Kapitola 1

Maple

1.1 Úvod

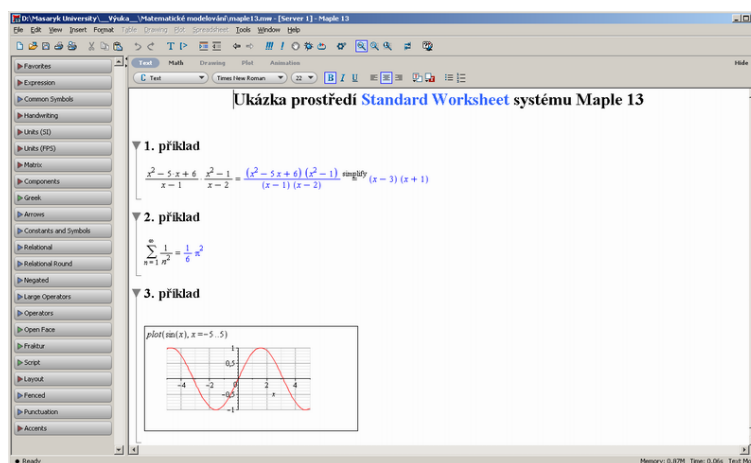
Systém Maple je výkonný program vhodný k řešení komplexních matematických problémů. Patří do skupiny systémů počítačové algebry, umožňuje symbolické i numerické výpočty a slouží především k vytvoření speciálních dokumentů, prezentací, interaktivních výpočetních modulů v prostředí Maple, které rovněž podporuje výuku matematiky.

Výrobcem systému Maple je kanadská společnost Maplesoft Inc., jejíž webové stránky <http://www.maplesoft.com> poskytují základní informace o tomto systému a jeho dalších příbuzných programech jako je MapleSim, MapleNet, Maple T.A. a mnoho dalších Toolboxů a programů. Webové stránky mimo jiné obsahují tzv. *Aplikační centrum* (Application Center), z něž si může každý zaregistrovaný uživatel stáhnout ukázkové programy demonstrující použití systému Maple při řešení mnoha různých matematických i technických problémů. Dále poskytují *Studentské centrum* (Student Center), kde si zaregistrovaný student může stáhnout mnoho studijních materiálů. Dalšími významnými zdroji informací o Maple jsou například diskuzní fórum uživatelů Maple <http://www.mapleprimes.com> a web distributora Maple pro Českou a Slovenskou republiku <http://www.maplesoft.cz>, kde je většina dokumentů v českém jazyce.

Systém Maple je vyvíjen již po tři desetiletí, během nichž prošel mnoha verzemi. Přelomovou se stala desátá verze, od níž je k dispozici grafické uživatelské rozhraní významně usnadňující práci se systémem zejména novým uživatelům. V následujícím textu se budeme zabývat dosud poslední verzí Maple 13. Se systémem je možné pracovat několika různými způsoby, které volíme při spuštění programu Maple 13 ze startovacího menu počítače nebo kliknutím na příslušnou ikonu na ploše.

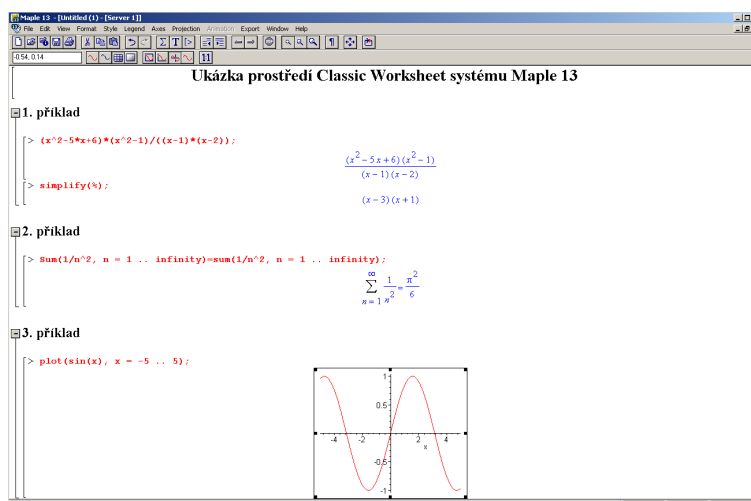
1.1.1 Standardní zápisník (Standard Worksheet)

Grafické uživatelské rozhraní Maple zvané *Standard Worksheet* se spustí ze startovacího menu počítače, kde se vybere (**Programy > Maple 13 > Maple 13**) nebo kliknutím na ikonu Maple 13 na ploše. Toto prostředí poskytuje veškeré možnosti systému Maple a pomáhá vytvářet elektronické dokumenty (zápisníky) zobrazující matematické výpočty, matematické texty a komentáře spolu s propracovanou počítačovou grafikou. Některé výpočty je možné v zápisníku „schovat“ a nechat „odkryté“ jen nejdůležitější pasáže tak, aby

Obrázek 1.1: Maple 13: Prostředí *Standard Worksheet*

dokument poskytoval uživateli potřebné informace. Jelikož vytvořené dokumenty jsou interaktivní, tj. v jistém smyslu „živé“, může si uživatel sám upravovat hodnoty parametrů, vyhodnocovat příkazy a získávat nové výsledky.

Menu zápisníku Maple má tři vodorovné lišty: *hlavní menu* (zcela nahoře), *nástrojovou lištu* (pod hlavním menu) a *kontextovou lištu* (o další „řádek“ níže); a dále *palety* (svislý blok na levé straně) a vlastní pracovní pole – *dokument*, do nějž zadáváme příkazy, texty a grafy. Vlastní pracovní pole je možné zobrazit přes celou obrazovku skrytím palet, nástrojové lišty a kontextové lišty (kliknutím na příslušné položky záložce **View** hlavního menu).

Obrázek 1.2: Maple 13: Prostředí *Classic Worksheet*

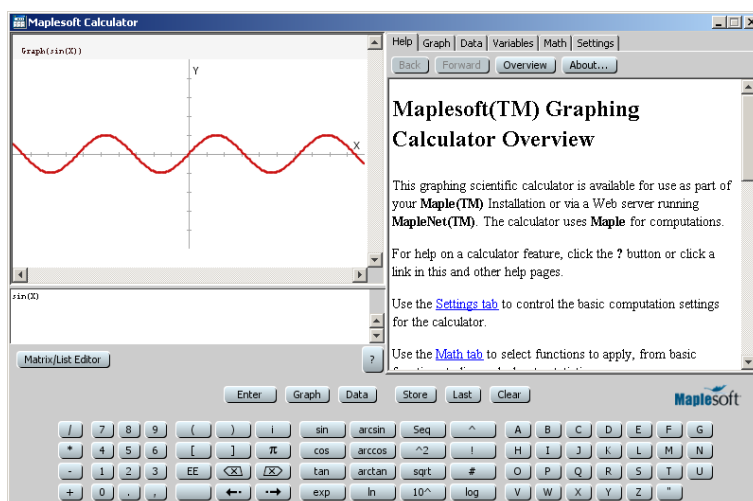
1.1.2 Klasický zápisník (Classic Worksheet)

Classic Worksheet se spustí ze startovacího menu počítače, kde se vybere (**Programy > Maple 13 > Classic Worksheet Maple 13**) nebo kliknutím na ikonu Classic Worksheet na ploše. Tento zápisník Maple je určen především pro méně výkonné počítače s omezenou pamětí. Neposkytuje také všechny funkce, příkazy a možnosti systému Maple jako Standard Worksheet. Před verzí Maple 10 byl tento typ zápisníku jediný možný.

1.1.3 Příkazový řádek a kalkulačka Maple

Se systémem Maple můžeme pracovat i pouze v režimu tzv. příkazového řádku, který se spouští ze startovacího menu počítače, kde se vybere (**Programy > Maple 13 > Command-line Maple 13**) a je určen k řešení rozsáhlých a složitých úloh. K dispozici nejsou žádné grafické prvky.

Dále je možné používat (a vytvářet) tzv. *maplety*, tj. grafická uživatelská rozhraní obsahující okénka, textová pole a další vizuální prvky umožňující pouhým klikáním spouštět výpočty. *Kalkulačka* systému Maple je speciální typ mapletu, který je k dispozici pouze pro operační systémy Windows. Spouští se ze startovacího menu počítače, kde se vybere (**Programy > Maple 13 > Maple Calculator**).

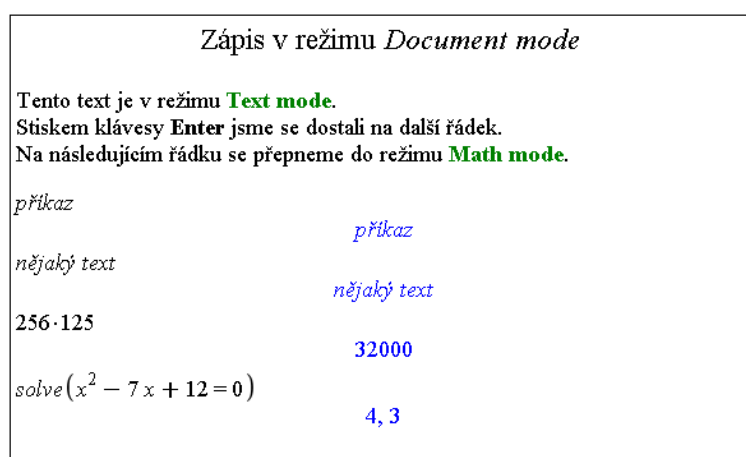


Obrázek 1.3: Maple 13: *Kalkulačka*

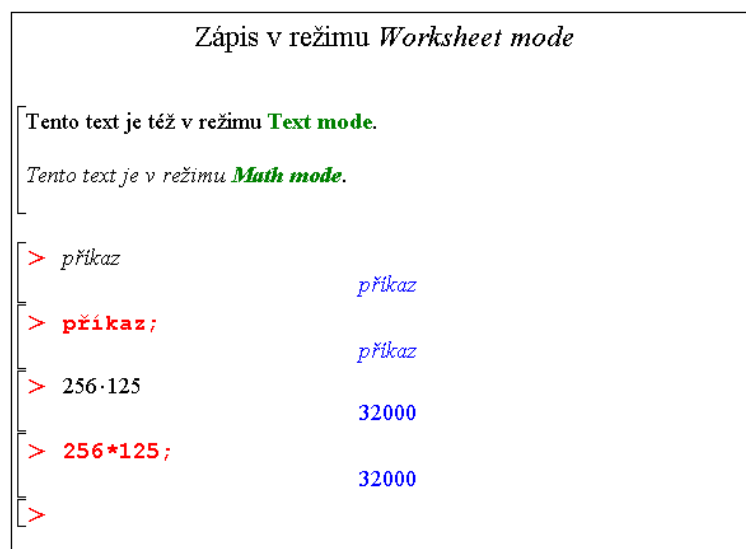
1.1.4 Document mode, Worksheet mode

Ve zbývající části kapitoly se omezíme na prostředí *Standard Worksheet*. V tomto prostředí je možné pracovat ve dvou základních režimech: *Worksheet mode* a *Document mode*. Prvně

jmenovaný odpovídá prostředí *Classic Worksheet*, v němž je každý příkaz Maple uvozen symbolem [$>$] a musí být ukončen dvojtečkou (výsledek se nezobrazí) nebo středníkem (výsledek se zobrazí na dalším řádku uprostřed). Otevírá se v základním menu jako **New**, kde se vybere **Worksheet mode**. Druhý režim poskytuje přehlednější zápis příkazů a matematických vzorců bez „přebytečných“ symbolů. Při otevření nového souboru v základním menu **New** je standardně spuštěn režim *Document mode*.



Obrázek 1.4: Režimy zápisu v *Document mode*



Obrázek 1.5: Režimy zápisu v *Worksheet mode*

Dále máme možnost zapisovat k příkazům komentáře, a to uvedením symbolu mřížky ($\#$) před text, který má být komentářem (viz obrázek 1.6).

```

256*125 # nějaký komentář
                                32000
[ > 256*125; # nějaký komentář
                                32000
[ >

```

Obrázek 1.6: Zápis komentáře v příkazovém režimu

Obvykle je zápisník nastaven do jednoho režimu, který je možné zvolit při otevírání nového souboru v hlavním menu (**File** > **New** > **Worksheet mode** nebo **File** > **New** > **Document mode**). Existuje však i možnost přepínat mezi režimy v rámci jednoho zápisníku, kdy je část vytvořena v jednom režimu, část v druhém. Z *Document mode* se přepneme do *Worksheet mode* kliknutím na ikonku $\left[\right] >$ v nástrojové liště (pod hlavním menu). Naopak z *Worksheet mode* se do režimu *Document mode* přepneme výběrem položky v hlavním menu (**Format** > **Create Document block**).

1.1.5 Math mode, Text mode

Pro rozlišení příkazů a obyčejného textu v režimu *Document mode* slouží kontextová lišta zápisníku, kde máme na výběr *Text mode* a *Math mode*. *Math mode* odpovídá příkazům (po stisku klávesy **Enter** dojde k jeho vyhodnocení), v *Text mode* píšeme texty dokumentu podobně jako např. v textovém editoru Word (po stisku klávesy **Enter** přejdeme na nový řádek bez jakéhokoli vyhodnocení). Volit režim a typ zápisu můžeme buď myší (v kontextové liště nad dokumentem jsou uvedeny názvy představující jednotlivé možnosti) nebo výběrem položky v hlavním menu (**Edit** > **Switch to Text/Math mode**). Totéž lze rychleji provést klávesou **F5**.

V režimu *Worksheet mode* lze pro text i pro příkazy použít oba druhy zápisu. Pro psaní textu je nutné kliknout na ikonku **T** v nástrojové liště nebo zvolit položku v hlavním menu (**Insert** > **Text**). Podobně pro zápis příkazů jazyka Maple je nutné kliknout na ikonku $\left[\right] >$ v nástrojové liště nebo zvolit položku v hlavním menu (**Insert** > **Maple Input**). Při otevření nového souboru je zápisník nastaven na psaní příkazů.

1.2 Základní ovládání

V této části si ukážeme, jak systému Maple zadávat jednoduché příkazy. Budeme proto předpokládat, že zápisník je již nastaven pro psaní příkazů (*Math mode*, viz Úvod).

Zadání zlomku: zadáme číselník, lomítko (/) a jmenovatel. Pro opuštění zápisu jmenovatele stačí stisknout šipku doprava (ve zlomku je též možno pohybovat se šipkami).

Zadání mocniny: zadáme základ, symbol stříška (^) a exponent. Pro opuštění zápisu exponentu je opět možné použít šipku doprava.

Základní operace: pro sčítání používáme symbol plus (+), pro odčítání mínus (−), pro násobení (*), ale pozor, pro dělení musíme používat pouze lomítko (/), dvojtečka (:) má jiný význam (viz dále).

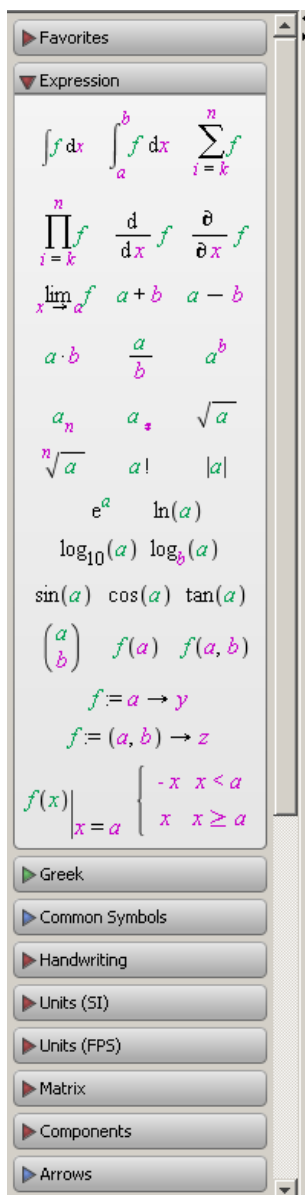
1.2.1 Vyhodnocení příkazů

Příkaz vyhodnotíme stiskem klávesy **Enter**. Výsledek se zobrazí na dalším řádku uprostřed. V dřívějších verzích (méně než 10) systému Maple bylo nutné příkaz ukončovat středníkem, aby se provedl. Tato možnost nadále zůstala (tj. zadáme-li za příkaz středník, „nic nepokazíme“) a v některých situacích je dokonce jediná možná – např. textový režim (*Text mode*) příkazů v *Worksheet mode* nebo při psaní příkazů v prostředí *Classic Worksheet*. Z předešlých verzí Maple se uchovala i funkcionálnota symbolu dvojtečka (:), která po zařazení za příkaz a následného stisku klávesy **Enter** potlačí zobrazení výsledku na dalším řádku (tj. příkaz se provede, ale na obrazovku se nic nevypíše). Proto není možné dvojtečku používat jako operátor dělení. V *Document mode* je navíc možné zapisovat příkaz i s výsledkem na jeden řádek. Po napsání příkazu k tomu stačí místo stisku klávesy **Enter** použít klávesovou zkratku „**Ctrl** + =“.

Jak bylo zmíněno dříve, interaktivní dokumenty v Maple jsou „živé“. Tím máme na mysli skutečnost, že i v dříve vytvořeném programu otevřeném po libovolně dlouhé době můžeme kterýkoli výraz upravit, znovu vyhodnotit (stisknout **Enter** nebo „**Ctrl** + =“) a dostaneme nový (správný) výsledek. Označení několika (libovolně mnoha) příkazů a stisknutí ikonky **!** (vykřičník) z nástrojové lišty způsobí postupné vyhodnocení všech označených příkazů. K vyhodnocení všech příkazů v dokumentu slouží ikonka **!!!** (tři vykřičníky).

Maple obsahuje více než tisíc symbolů, pomocí nichž můžeme tvořit matematické výrazy. Patří mezi ně písmena anglické abecedy a číslice, pomocí nichž můžeme vytvářet jména (posloupnost znaků začínající písmenem, za kterým může následovat kombinace písmen, čísel a vybraných symbolů), reálná čísla (celá, racionální, s desetinou tečkou nebo v notaci pohyblivé řádové čárky), komplexní čísla (reálná a imaginární část), aritmetické, booleovské operátory a další (+, −, !, /, *, \int , lim, ...), konstanty (π , e , ...), imaginární jednotka, nekonečno, matematické funkce ($\cos(x)$, $\sin(\frac{\pi}{3})$, ...) a proměnné (pojmenované jménem ...). Velkou předností systému Maple je jeho schopnost symbolických matematických výpočtů (tj. práce s výrazy). Některé z matematických symbolů, které můžeme použít, není na klávesnici, a tak se zadává buď z palety na levé straně zápisníku nebo pomocí svých názvů.

1.2.2 Palety



Obrázek 1.7: Palety

Palety jsou pojmenované „obdélníčky“ při levém okraji zápisníku. Každá paleta obsahuje symboly příslušné skupiny. Například paleta s názvem *Expression* nabízí některé základní matematické výrazy, paleta *Greek* písmena řecké abecedy atd. Standardně zůstává několik palet nezobrazených. V hlavním menu (**View > Palettes**) můžeme seznam zobrazených palet upravit tím, že některé přidáme, odebereme, ale třeba i jinak seřadíme. Totéž lze provést jen za pomoci myši. Přidržením levého tlačítka vybranou paletu přesuneme na jiné místo, stisknutím pravého tlačítka vyvoláme stejnou nabídku, jako bychom postupovali přes hlavní menu. Kliknutím (levého tlačítka myši) na některou paletu zobrazíme (příp. skryjeme) symboly, které nabízí. Vložit z palety symbol do zápisníku pak stačí pouhým kliknutím, případně „přetáhnutím“ s pomocí levého tlačítka myši. Některé symboly („barevné“) je možné dále specifikovat (upravovat).

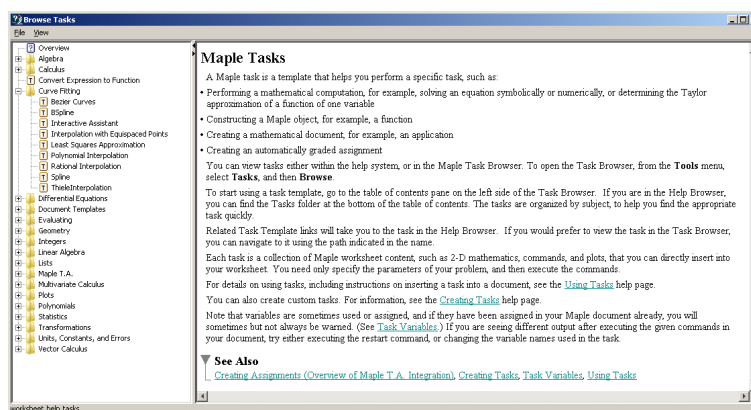
Nyní ukážeme, jak vložit například výraz $\sum_{i=1}^{10} 2^i$. Pro jeho zapsání potřebujeme celkem dva různé symboly: *sumu* a *mocninu*. Sumační symbol nalezneme v paletě **Expression**. Kliknutím na tuto paletu ji otevřeme (na obrázku 1.7 je jako jediná otevřená) a vybereme z ní přítomný sumační symbol. Po jeho vložení do zápisníku pak jednoduše přepíšeme obecné symboly (k , n a f) požadovanými hodnotami. Pohybovat se ve vzorci můžeme pomocí šipek na klávesnici, a přitom k jeho úpravě používat veškeré jiné klávesy jako např. **Delete**, **Backspace**, **mezerník**, ... Znak f smažeme a místo něj zapíšeme mocninný výraz. To můžeme provést jednak vložení dalšího symbolu z palety **Expression** (symbol a^b) a následnou úpravou (specifikací hodnot a , b) nebo užitím již známé klávesy pro tvorbu mocnin, stříšky (^).

1.2.3 Názvy symbolů

Mimo palet můžeme k zápisu symbolů užívat jejich názvů. Například symbol π vložíme zapsáním jeho názvu **Pi**, pro odmocninu je vyhrazen název **sqrt**, takže \sqrt{x} vložíme napsáním **sqrt(x)**. Při vkládání symbolů pomocí názvů může přijít vhod funkce „dokončování“. Ta se vyvolá stisknutím klávesy **Esc**, nebo kláves „**Ctrl + mezerník**“. Pro zadání symbolu pak stačí napsat jeho úvodní písmeno (písmena) a pomocí klávesové zkratky následně z vyskakovacího okénka zvolit požadovanou hodnotu.

1.2.4 Pomocníci, instruktoři a řešené úlohy

Systém Maple poskytuje „pomocné nástroje“ pro řešení úloh. Jsou to tzv. *Pomocníci (Assistants)*, *Instruktoři (Tutors)* a *Úlohy (Tasks)*, které vyvoláme z hlavního menu (**Tools** > **Assistants** nebo **Tools** > **Tutors** anebo **Tools** > **Tasks**). *Pomocníci (Assistants)* mají např. nástroje pro hledání funkční závislosti v datech, optimalizaci funkcí, řešení diferenciálních rovnic a další. Pro daný typ úlohy mají implementováno několik často používaných algoritmů. Po vyvolání provedou uživatele nastavením a specifikací parametrů úlohy a zvolenou metodou úlohu vyřeší. *Instruktoři (Tutors)* provedou uživatele řešenou problematikou pomocí jednoduchých názorných příkladů. *Úlohy (Tasks)* zobrazují na příkladech, jak řešit různé úlohy. Zobrazí se vyvoláním z menu (**Tools** > **Tasks** > **Browse**), viz obrázek 1.8.



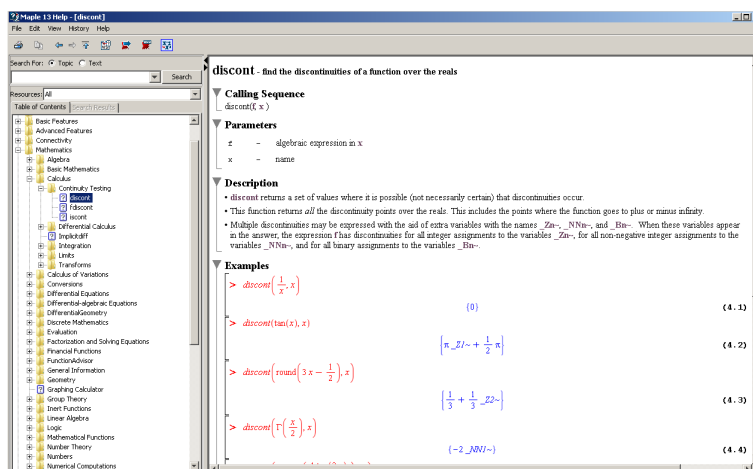
Obrázek 1.8: Zobrazení úloh systému Maple.

1.3 Náповěda

Náповěda je významnou součástí systému Maple a pomáhá jeho uživateli velmi rychle pochopit a využívat prostředí Maple. K dispozici je několik různých typů náповědy, které najdeme v položce **Help** hlavního menu.

1.3.1 Maple Help

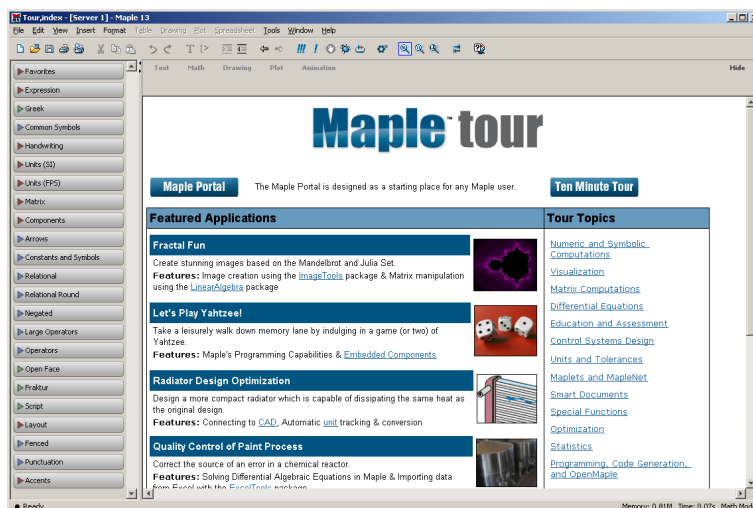
Vyvoláním **Maple Help** se zobrazí základní stránky náповědy se všemi dostupnými informacemi. Je v nich možné jednak vyhledávat požadované téma, ale také procházet jednotlivá témata jako v manuálu. Kromě zvolení položky v menu můžeme tuto náповědu otevřít i v zápisníku zapsáním příkazu **?** (otazník), případně klávesovou zkratkou „**Ctrl** + **F1**“. Na obrázku 1.9 je okno náповědy otevřené na informacích k příkazu **discont**.



Obrázek 1.9: Hlavní nápověda systému Maple.

1.3.2 Tour of Maple

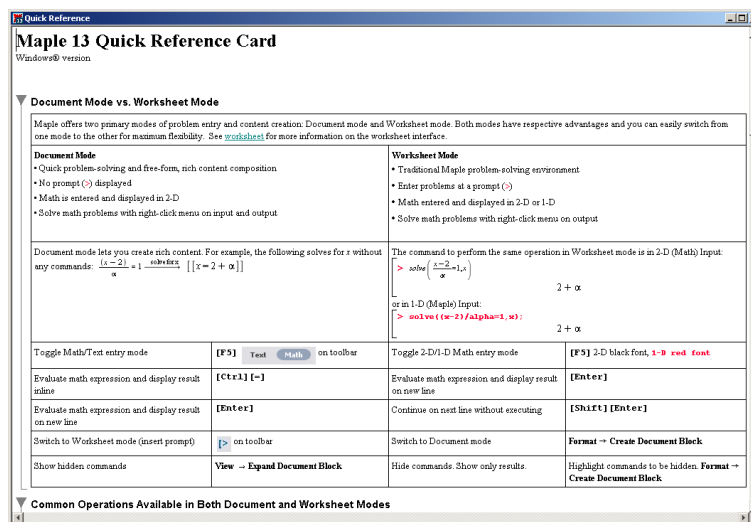
Položka **Take a Tour of Maple** poskytuje interaktivní přehled systému (jeho nejdůležitějších prvků).



Obrázek 1.10: Tour of Maple.

1.3.3 Quick Reference

Kliknutím na **Quick Reference** zobrazíme tabulku informací o ovládní systému Maple, zejména pro nové uživatele. Jedná se o základní informace s odkazy do nápovědy **Maple Help** pro jejich případné doplnění.



Obrázek 1.11: Maple Quick Reference.

1.3.4 Quick Help

Položka **Quick Help** nabízí ještě stručnější tabulku než předchozí nápověda. Standardně se objevuje v každém novém zápisníku při pravé straně v podobě černého okénka (pokud toto nastavení nezrušíme). Po zavření je možné ji vyvolat stiskem klávesy **F1**, či jako položku v hlavním menu.

1.3.5 What's New

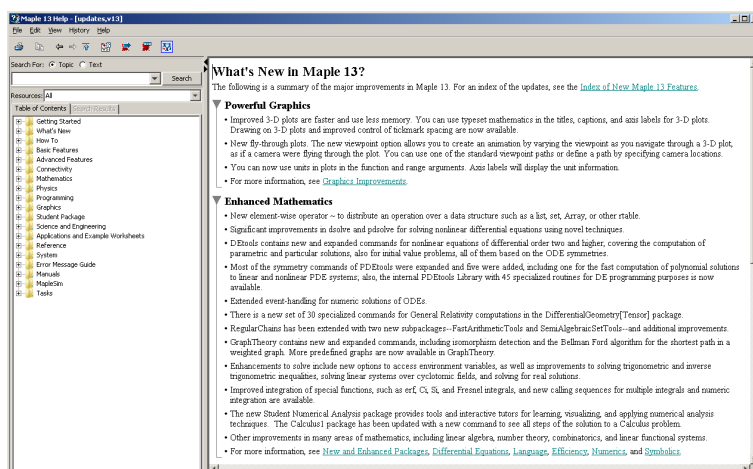
K dispozici je též přehled rozšíření stávající verze Maple oproti předcházející verzi. Doplně přes **Help** > **What's New**.

1.3.6 Startup Dialog

Nápověda **Startup Dialog** obsahuje tipy pro práci se systémem Maple. Zobrazuje se vždy po spuštění systému (pokud toto nastavení nezrušíme).

1.3.7 Manuals, Resources, and more

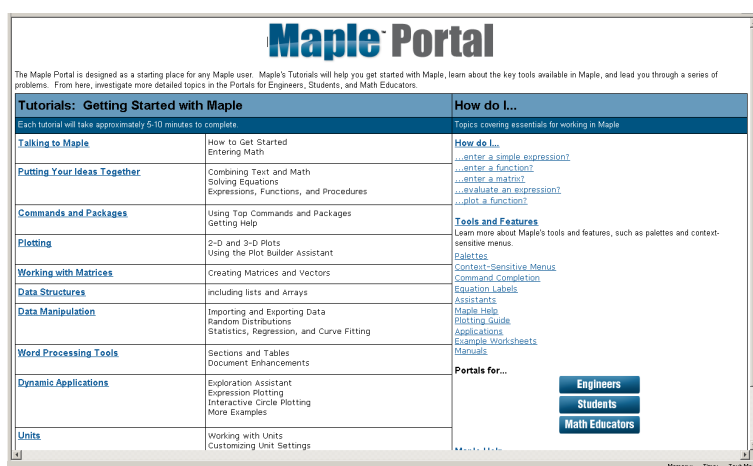
Vyvoláním **Manuals, Resources, and more** přejdeme do další oblasti nápovědy, z níž popíšeme tři nejdůležitější části.



Obrázek 1.12: Přehled novinek ve stávající verzi systému.

Maple Portal

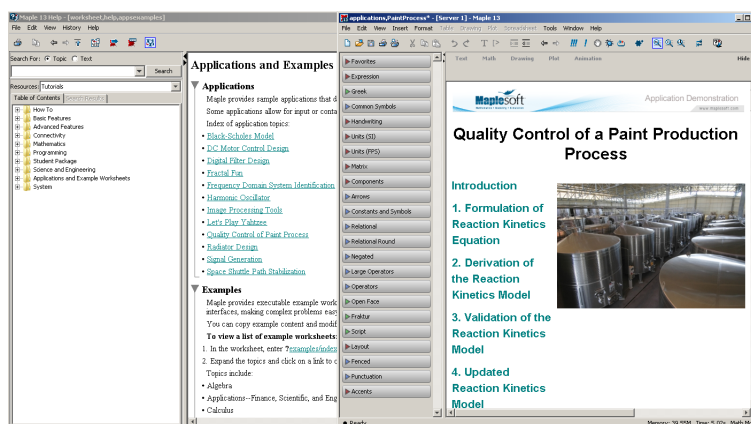
Od nejnovější verze (tedy 13) je k dispozici tzv. *Maple Portal*. Spustit jej můžeme samostatně (*Maple Portal* má vlastní ikonu na ploše) nebo přes nápovědu v hlavním menu (**Help** > **Manuals, Resources, and more** > **Maple Portal**). *Maple Portal* slouží jako pomocník všem novým (a nejen těm) uživatelům systému Maple. Je v něm možné rychle najít detailní popis práce se systémem Maple od řešení nejjednodušších problémů až po velmi složité úlohy.



Obrázek 1.13: Maple Portal.

Applications and Examples

Z nápovědy je možno vyvolat i spustitelné soubory (tj. již vytvořené programy) demonstrující možnosti systému Maple. Vyvoláme je přes nápovědu v hlavním menu (**Help > Manuals, Resources, and more > Applications and Examples**) a pak kliknutím na zvolený příklad, viz obr. 1.14, kde je zvolen příklad *Quality Control of Paint Production Process*.



Obrázek 1.14: Maple Applications and Examples.

Manuals

Z nápovědy je možno vyvolat i anglické manuály **User manual**, **Introductory Programming Guide**, **Advanced Programming Guide** a **Getting Started with Maple Toolboxes** podrobně popisující možnosti systému Maple. Vyvoláme je přes nápovědu v hlavním menu (**Help > Manuals, Resources, and more > Manuals**) a pak kliknutím na zvolený manuál.

1.4 Provádění výpočtů

Maple provádí přesně numerické výpočty s celými a racionálními čísly. Každý zadaný matematický výraz se snaží co nejvíce zjednodušit (např. zlomek zkrátit a převést na základní tvar, upravit algebraický výraz, ...), ale ne za cenu ztráty přesnosti. To znamená, že například racionální čísla (zlomky) udržuje stále v jejich základním tvaru. Podobně s konstantami π , e a dalšími, odmocninami a jinými výrazy pracuje jako se symboly. Tímto je zaručena absolutní přesnost numerických výpočtů i v případě, kdy nepracujeme pouze s celými a racionálními čísly.

Jsou však situace, kdy potřebujeme znát přibližnou hodnotu reálného nebo racionálního čísla v pohyblivé řádové čárce. K tomu slouží příkaz **evalf**, jenž vrátí zaokrouhlenou

hodnotu svého argumentu na počet platných cifer mantisy specifikovaný systémovou proměnnou **Digits**. Ta je standardně nastavena na hodnotu 10. Všechny výpočty, při nichž je nutné zaokrouhlovat čísla, provádí proto Maple s přesností na 10 platných míst. Proměnnou **Digits** můžeme nastavit na takřka libovolné přirozené číslo. Omezení, jak vysoké toto číslo může být, zjistíme příkazem **kernelopts(maxdigits)**. Pro představu uvedme, že pro Maple 13 je toto číslo 268 435 448, tedy více než 268 milionů platných cifer, s kterými dokáže systém počítat.

Aniž bychom měnili nastavení proměnné Digits, můžeme zobrazit libovolný výraz s požadovanou přesností pouze pomocí příkazu **evalf**. Příkaz je možné použít s jedním nebo dvěma parametry. Jediný zadaný parametr znamená, že tento zadaný výraz bude vyhodnocen na počet platných míst specifikovaných v proměnné **Digits**. Přidaný druhý parametr řekne funkci **evalf**, na kolik platných míst má výraz vyhodnotit, viz příklady na obrázku 1.23.

| | |
|---------------|-----------------------|
| Pi | π |
| Digits | 10 |
| evalf(Pi) | 3.141592654 |
| evalf(Pi, 5) | 3.1416 |
| evalf(Pi, 20) | 3.1415926535897932385 |

Obrázek 1.15: Příkaz **evalf**.

S dříve zmíněnou nápovědou souvisí symbol ? (otazník). Jak již víme, samotný otazník (jako příkaz) otevře hlavní nápovědu systému (*Maple Help*). Otazník spolu s názvem příkazu otevře nápovědu rovnou na stránce týkající se zadaného příkazu. Tedy např. příkaz **?evalf** otevře hlavní nápovědu systému na stránce popisující syntaxi a sémantiku příkazu **evalf** spolu s příklady jeho použití. Zapsáním dvou otazníků na začátek příkazu otevřeme tutéž stránku nápovědy ve „sbaleném“ tvaru osnovy, v níž je možné otevřít (odkryt) libovolné části. Nakonec je tu ještě možnost zadání tří otazníků před příkaz, což otevře nápovědu na příkladech použití tohoto příkazu.

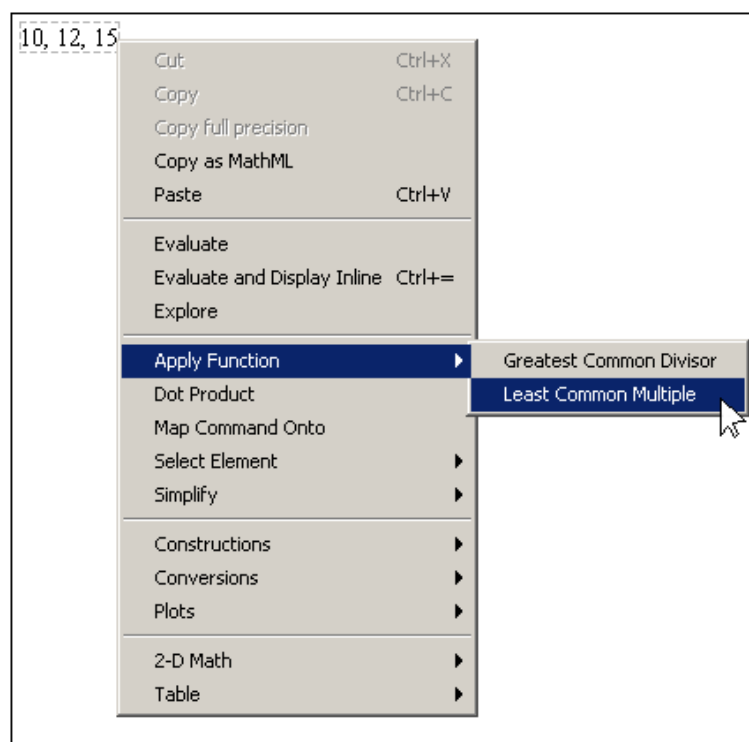
Maple rozeznává přesná čísla (mezi něž patří i zmíněné symboly π a e , zlomky atp.) a čísla typu *Floating-Point*, nebo-li čísla v pohyblivé řádové čárce. Jestliže systému zadáme výraz, v němž některý z jeho podvýrazů bude typu *Floating-Point*, může Maple na celý výraz pohlížet jako by byl tohoto typu a bude výsledky výpočtů zaokrouhlovat. To nejlépe uvidíme na dalších příkladech na obrázku 1.16.

| | | |
|---|-----|--|
| $\frac{2}{3} + \frac{3}{2} = \frac{13}{6}$, | ale | $\frac{2}{3} + 1.5 = 2.166666667$ |
| $\sqrt{2} + \sqrt{3} = \sqrt{2} + \sqrt{3}$, | ale | $\sqrt{2.0} + \sqrt{3} = 1.414213562 + \sqrt{3}$ |

Obrázek 1.16: Přesná čísla a čísla typu **Floating-Point**.

1.4.1 Příkazy

Pro provedení výpočtu máme zpravidla více možností. Tou základní, která je k dispozici ve všech verzích systému, jsou příkazy jazyka Maple. Chceme-li například vypočítat odmocninu z čísla 2.5, zapíšeme v systému Maple příkaz `sqrt(2.5)`. Stejného výsledku dosáhneme použitím symbolu pro odmocninu z palety **Expression**. Pokud chceme určit nejmenší společný násobek čísel 10, 12 a 15, můžeme využít příkazu `lcm`, nebo zapsat čísla na řádek za sebe (oddělená čárkami) a přes pravé tlačítko myši zvolit z kontextové nabídky **Apply Function > Least Common Multiple**, viz obrázky 1.17, 1.18.



Obrázek 1.17: Provedení výpočtu pomocí kontextové nabídky.

The image shows a rectangular box containing four lines of text, each representing a different way to perform a calculation in Maple:

- `sqrt(2.5) = 1.581138830`
- $\sqrt{2.5} = 1.581138830$
- `lcm(10, 12, 15) = 60`
- `10, 12, 15` followed by an arrow labeled "integer lcm" pointing to `60`.

Obrázek 1.18: Různé možnosti v provedení výpočtu.

1.4.2 Označení výsledků

Každému zobrazenému výsledku se v zápisníku přiřazuje číselné označení, které se zapisuje zcela vpravo na řádek s odpovídajícím výsledkem. Označení je možné potlačit (tj. nezobrazovat), znovu vyvolat, případně upravit jeho formát v hlavním menu (**Format** > **Labels** > ...). Díky označení se můžeme na předešlé výsledky odvolávat a používat je při tvorbě dalších příkazů. V ukázkách vytvořených dokumentů (prezentovaných v tomto textu) je označení výsledků vždy potlačeno. Použití označení ilustruje obrázek 1.19.

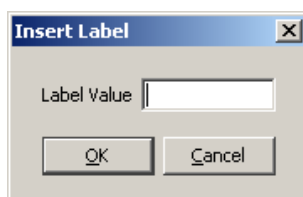
Pokud chceme například přičíst číslo 10 k výsledku s označením (2), pak napíšeme „10 + “ a přes klávesovou zkratku „**Ctrl** + **L**“ vložíme požadované označení (tedy do „vyskakujícího okénka“ zadáme číslo 2 a potvrdíme (**OK**)). Místo klávesové zkratky „**Ctrl** + **L**“ je možné použít horní menu (**Insert** > **Label...**). Upozorníme, že zápis (2) vytvořený (pouze) na klávesnici při tvorbě příkazu Maple nepochopí, pro vložení označení do příkazu je třeba důsledně používat předešlý postup s „vyskakujícím okénkem“ zobrazeným na obrázku 1.20.

The image shows a rectangular box containing four lines of mathematical expressions, each followed by a label in parentheses on the right:

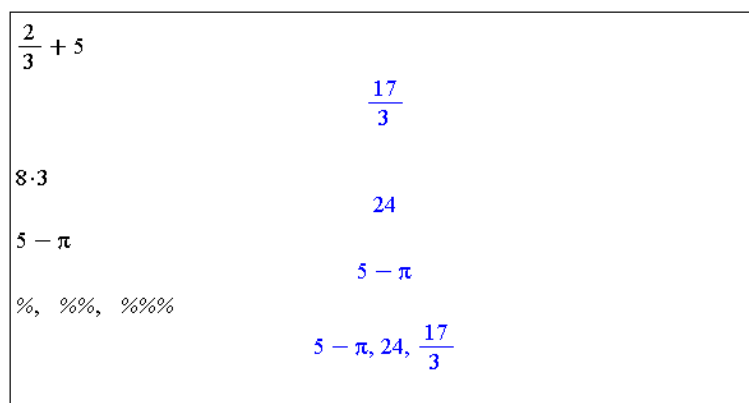
- $\frac{2}{3} - 6$ (1)
- $-\frac{16}{3}$ (2)
- $8 \cdot 3$ (2) (Note: the label (2) is blue, indicating it refers to the result of the second line)
- 24 (2)
- $5 - \pi$ (3)
- $5 - \pi$ (3)
- $10 + (2)$ (4)
- 34 (4)

Obrázek 1.19: Označení výsledků.

Maple dále nabízí možnost odkazovat se na poslední tři výsledky (v tomto případě je jedno, zda byly zobrazeny či nikoliv, a zda mají nějaké označení) pomocí symbolu % (procento). Jedno procento představuje poslední výsledek, dvě procenta předposlední a tři procenta před-předposlední.



Obrázek 1.20: „Vyskakující okénko“ pro zadání označení.



Obrázek 1.21: Využití procent při odkazování se na předchozí výsledky.

1.4.3 Přiřazení hodnot do proměnných

Odkazovat se na nějaké výrazy můžeme také po jejich přiřazení k nějaké proměnné. Operátorem přiřazení je (dvoj)symbol `:=` (dvojtečka + rovnítko).

Místo (dvoj)symbolu `:=` můžeme k přiřazení použít příkaz **assign**. Tak, jak můžeme výrazy do proměnných přiřazovat, můžeme též přiřazení zrušit (tj. odebrat proměnné uloženou hodnotu). Zmíněné provedeme příkazem **unassign** nebo přiřazením názvu proměnné v apostrofech (obrázek 1.22).

Přiřazovat hodnoty můžeme i do tzv. systémových proměnných. Již jsme se setkali s proměnnou **Digits**, která vyjadřuje počet platných míst, s nimiž Maple počítá. Ilustraci na obrázku 1.23 můžeme srovnat s obrázkem 1.15.

Odstranit uloženou hodnotu v systémové proměnné nelze. Do systémových proměnných můžeme hodnoty pouze přiřazovat, nebo současně vrátit příkazem **restart** nastavení všech systémových proměnných na jejich původní hodnoty. Provedení příkazu odstraní všechny uložené hodnoty v paměti (tedy i námi definované proměnné, načtené balíky atd.). Příkaz **restart** se proto používá zpravidla na počátku řešení nové úlohy, zejména pak na začátku každé práce se zápisníkem (aby se předešlo tomu, že budeme používat proměnnou, v níž je z dřívějšíka uložena pro nás nesprávná hodnota).

| | | | |
|--------------------------|----------|-------------------------------|----------|
| <i>a</i> | <i>a</i> | <i>c</i> | <i>c</i> |
| <i>a</i> := 2 | 2 | <i>assign</i> (<i>c</i> , 2) | |
| <i>a</i> | 2 | <i>c</i> | 2 |
| <i>b</i> | <i>b</i> | <i>unassign</i> ('c') | |
| <i>b</i> := 3 · <i>a</i> | 6 | <i>c</i> | <i>c</i> |
| <i>b</i> | 6 | <i>a</i> | 2 |
| | | <i>a</i> := 'a' | <i>a</i> |
| | | <i>a</i> | <i>a</i> |

Obrázek 1.22: Přiřazení hodnot do proměnných a odstranění uložené hodnoty.

| | |
|---------------------|-----------------------|
| Pi | π |
| <i>Digits</i> | 10 |
| <i>evalf</i> (Pi) | 3.141592654 |
| <i>Digits</i> := 5 | 5 |
| <i>evalf</i> (Pi) | 3.1416 |
| <i>Digits</i> := 20 | 20 |
| <i>evalf</i> (Pi) | 3.1415926535897932385 |

Obrázek 1.23: Proměnná **Digits** a příkaz **evalf**.

1.4.4 Řešení rovnic

Pro řešení rovnic má systém Maple příkaz **solve** a několik příkazů k němu příbuzných závislých na typech rovnic, viz tabulka 1.

Tabulka 1: Příkazy pro řešení rovnic

| Typ rovnice | Příkaz pro řešení |
|--|----------------------------|
| Rovnice a nerovnice | solve, fsolve |
| Obyčejné diferenciální rovnice | dsolve |
| Parciální diferenciální rovnice | pdsolve |
| Rovnice v oboru celých čísel | isolve |
| Rovnice v oboru celých čísel v konečném tělese | msolve |
| Lineární integrální rovnice | intsolve |
| Systémy lineárních rovnic | LinearAlgebra[LinearSolve] |
| Rekurentní rovnice | rsolve |

Díky interaktivnímu prostředí *Standard worksheet* můžeme řešit rovnice též pomocí kontextové nabídky. Zapišeme rovnici a přes pravé tlačítko myši zvolíme požadovaný příkaz. Obrázek 1.24 ilustruje některé příklady řešení rovnic.

Příkazy na řešení rovnic nemusí vždy zobrazit všechna řešení. Pokud je chceme zobrazit, přidáme příkazu **solve** nepovinný parametr **AllSolutions**, viz obrázek 1.25.

Symbol „**Z2**~“ představuje libovolnou celočíselnou proměnnou. Že se jedná o celočíselnou proměnnou poznáme podle toho, že se v symbolu vyskytuje písmeno **Z**. Podobně by výskyt například písmena **C** značil proměnnou komplexní. Cifra **2** v symbolu proměnné označuje pořadí, v jakém byla proměnná v zápisníku zavedena. A nakonec znak „~“ vyjadřuje, že proměnná splňuje nějaký předpoklad. Jaké předpoklady proměnná splňuje přitom zjistíme příkazem **about**, případně zápisem proměnné a po kliknutí pravým tlačítkem myši zvolením **What Assumptions** z kontextové nabídky. V zobrazeném příkladu je předpoklad celočíselnosti proměnné přebytečný.

Dále může příkaz **solve** zobrazit výsledek se strukturou **RootOf** vyjadřující kořen (tj. řešení) rovnice v nevyhodnoceném tvaru. Řešení pak vyhodnotíme buď příkazem **allvalues** (pro symbolické vyjádření), nebo příkazem **evalf** (pro numerické vyjádření) – obrázek 1.26. Vedle příkazů můžeme též využít pravého tlačítka myši, zvolit z kontextové nabídky položku **All Values** (pro symbolické vyjádření) a získaný výsledek převést na numerickou hodnotu zvolením **Approximate > 10** (pro 10 platných míst) z kontextové nabídky.

Symbole „**Z**“ ve struktuře **RootOf** nyní nepředstavují celočíselnou proměnnou (neboť za písmenem **Z** nenásleduje číslo), nýbrž proměnnou libovolnou (tj. i komplexní). Struktura **RootOf** z obrázku 1.26 tedy zastupuje kořen rovnice $z^4 - 2 \cdot z^3 + 2 = 0$.

Rovnice
 $\text{solve}(x^2 + 5 \cdot x + 6 = 0)$
 $-2, -3$

$x^2 + 5 \cdot x + 6 = 0 \xrightarrow{\text{solve}} \{x = -2\}, \{x = -3\}$

Nerovnice
 $\text{solve}(x^2 + 5 \cdot x + 6 < 0)$
 $\text{RealRange}(\text{Open}(-3), \text{Open}(-2))$

$x^2 + 5 \cdot x + 6 < 0 \xrightarrow{\text{solve}} \{-3 < x, x < -2\}$

Rekurentní rovnice
 $\text{rsolve}(\{N(t+1) = (1 + a - b) \cdot N(t), N(0) = NO\}, N(t))$
 $NO (1 + a - b)^t$

$N(t+1) = (1 + a - b) \cdot N(t), N(0) = NO \xrightarrow{\text{solve recurrence}} NO (1 + a - b)^t$

Diferenciální rovnice
 $\text{dsolve}\left(\left\{\frac{d}{dt} N(t) = (a - b) \cdot N(t), N(0) = NO\right\}\right)$
 $N(t) = NO e^{(a-b)t}$

$\frac{d}{dt} N(t) = (a - b) \cdot N(t), N(0) = NO \xrightarrow{\text{solve DE}} N(t) = NO e^{(a-b)t}$

Obrázek 1.24: Ukázka řešení různých druhů rovnic vždy pomocí příkazu i pomocí kontextové nabídky.

$\text{solve}(\sin(x) = \cos(x))$
 $\frac{1}{4} \pi$

$\text{solve}(\sin(x) = \cos(x), \text{AllSolutions})$
 $\frac{1}{4} \pi + \pi _Z2\sim$

$\text{about}(_Z2)$
 Originally $_Z2$, renamed $_Z2\sim$:
 is assumed to be: integer

$_Z2 \xrightarrow{\text{list assumptions}} \{_Z2\sim::\text{integer}\}$

Obrázek 1.25: Zobrazení všech řešení.

```

solve(x^4 - 2 x^3 + 2 = 0)
RootOf(_Z^4 - 2 _Z^3 + 2, index = 1), RootOf(_Z^4 - 2 _Z^3 + 2, index = 2),
  RootOf(_Z^4 - 2 _Z^3 + 2, index = 3), RootOf(_Z^4 - 2 _Z^3 + 2, index = 4)

allvalues({%})
{ 1/2 - 1/2 I - 1/2 sqrt(4 + 2 I), 1/2 - 1/2 I + 1/2 sqrt(4 + 2 I), 1/2 + 1/2 I
  - 1/2 sqrt(4 - 2 I), 1/2 + 1/2 I + 1/2 sqrt(4 - 2 I) }

evalf(%)
{-0.5290855140 - 0.7429341359 I, -0.5290855140 + 0.7429341359 I,
  1.529085514 - 0.2570658641 I, 1.529085514 + 0.2570658641 I}

```

Obrázek 1.26: Tvar zobrazení řešení.

```

solve(x^2 < 0)

infolevel[solve] := 1:

solve(x^2 < 0)
solve: Warning: no solutions found

```

Obrázek 1.27: Proměnná **infolevel** a „prázdný výpis“ příkazu **solve**.

System Maple po zadání příkazu vypisuje zpravidla pouze řešení, případně chybová hlášení, či varování. U příkazu **solve** (a nejen u něj) toto chování způsobuje „prázdný výpis“ v případě, že Maple žádné řešení nenašel. Pro výpis podrobnějších informací o průběhu vyhodnocení příkazu a výsledcích slouží proměnná **infolevel**. Můžeme ji nastavit buď pro každý příkaz samostatně, přičemž do hranatých závorek za proměnnou vložíme název příslušného příkazu, nebo ji nastavíme všem příkazům současně na stejnou hodnotu uvedením slova **all** do hranatých závorek. Proměnná může nabývat pěti hodnot (1, ..., 5). Čím vyšší hodnota je přiřazena v proměnné **infolevel**, tím více informací o vyhodnocení příkazu obdržíme. Standardně není proměnná nastavena na žádnou hodnotu, což v podstatě odpovídá nastavení proměnné na hodnotu 0. Použití proměnné **infolevel** dokumentují obrázky 1.27 a 1.28.

1.5 Balíky

Knihovna příkazů jazyka Maple je rozdělena na hlavní knihovnu a tzv. balíky. Příkazy, s nimiž jsme se doposud setkali, patří do hlavní knihovny, a můžeme je tak používat ihned po spuštění systému. Naproti tomu většina příkazů náleží do balíků, které musíme před

```

solve({x+y=2,x-y=0})
                                {x=1,y=1}

infolevel[solve] := 1:
solve({x+y=2,x-y=0})
Linear: # equations 2
                                {x=1,y=1}

infolevel[solve] := 3:
solve({x+y=2,x-y=0})
Linear: # equations 2
Rational: # equations 2
Rational: 2 equations solved, rank: 2
                                {x=1,y=1}

infolevel[solve] := 5:
solve({x+y=2,x-y=0})
Linear: # equations 2
Rational: # equations 2
Rational: # equations 1
Rational: # equations 0
Rational: backsubstitution at: 2
Rational: backsubstitution at: 1
Rational: 2 equations solved, rank: 2
                                {x=1,y=1}

```

Obrázek 1.28: Proměnná `infolevel` a příkaz `solve`.

použitím příslušného příkazu buď načíst do dokumentu pomocí příkazu **with**, nebo zadat příkaz spolu s názvem balíku. Načtení balíku pomocí příkazu **with** umožní používání všech příkazů z příslušného balíku. Naopak zadání příkazu spolu s názvem balíku je nutné provádět při každém použití jakéhokoli příkazu, v případě, že balík nenačteme (příkazem **with**). My budeme (v případě potřeby) vždy načítat celý balík příkazů.

Například příkazy pro práci s vektory a maticemi náleží do balíku **LinearAlgebra**. Jestliže chceme tedy použít příkaz **Eigenvalues** pro nalezení vlastních čísel matice, načteme nejprve balík **LinearAlgebra**, jak dokumentuje obrázek 1.29.

1.5.1 Jednotky

Práci s jednotkami umožňuje balík **Units**. Při výpočtech tak nemusíme pracovat jen s čísly, ale můžeme jim přiřazovat i jednotky. K vložení jednotek do zápisníku využijeme palety **Units**. Obrázek 1.30 ilustruje použití jednotek při výpočtu gravitační síly působící v tíhovém poli Země (tj. gravitační zrychlení je rovno 9.81 ms^{-2}) na těleso o hmotnosti 10 kg . Vidíme, že Maple umí jednotky také zjednodušovat (resp. upravovat na jiný tvar). Ke zjednodušení výrazů přitom slouží příkaz **simplify** (viz dále).

Maple rozpoznává jednotky různých soustav a velikostí, s nimiž umí pracovat a vzájemně je převádět. K převodu jednotek je k dispozici speciální nástroj zvaný **Units Calculator**. Spustit jej můžeme z hlavního menu přes **Tools > Assistants > Units Cal-**

```

Matice :=  $\begin{bmatrix} 1 & 2 & 1 \\ 6 & -1 & 0 \\ -1 & -2 & -1 \end{bmatrix}$  :

with(LinearAlgebra) :

Eigenvalues(Matice)
 $\begin{bmatrix} 0 \\ 3 \\ -4 \end{bmatrix}$ 

```

Obrázek 1.29: Použití balíků.

```

F := 10[[kg]]·9.81  $\frac{[m]}{[s]^2}$ 
 $\frac{98.10 [[kg]] [[m]]}{[s]^2}$ 

simplify(%)
98.10 [[N]]

```

Obrázek 1.30: Použití jednotek.

culator..., ukázkou poskytuje obrázek 1.31.

Pokud chceme použít jednotku, která není v paletě **Units**, můžeme si ji vytvořit sami tak, že přidáme jednotku s názvem **unit** a název přepíšeme. V systému Maple 13 je implementováno celkem 568 jednotek (tzn. v paletě **Units** je pouze několik vybraných).

1.5.2 Vědecké konstanty

Balík **ScientificConstants** poskytuje hodnoty fyzikálních konstant a chemických vlastností látek spolu s jednotkami a neurčitostmi v těchto hodnotách. Příkazem **GetConstant** zobrazíme informace o zadané konstantě, viz obrázek 1.32 pro gravitační zrychlení v tíhovém poli Země a Newtonovu gravitační konstantu.

Příkazem **GetValue** získáme (pouze) numerickou hodnotu konstanty, příkazem **GetUnit** (pouze) jednotku – obrázek 1.33.

MapleSoft
Mathematics • Modeling • Simulation

Units Calculator

Convert between over 500 units of measurement. See [Units help index](#) for details.

First, select a dimension from the drop-down box. Then select the units to convert from and to. Click the "Perform Unit Conversion" button. The "Convert Back" button converts in the opposite direction.

| | |
|--|--|
| Convert: <input type="text" value="10"/> | Result: <input type="text" value="3.048000000"/> |
| From: <input type="text" value="feet (ft)"/> | To: <input type="text" value="meters (m)"/> |
| Dimension: <input type="text" value="length"/> | |

Obrázek 1.31: Units Calculator.

```
with(ScientificConstants):
GetConstant('g')
standard_acceleration_of_gravity, symbol = g, value = 9.80665, uncertainty = 0, units =  $\frac{m}{s^2}$ 
GetConstant('G')
Newtonian_constant_of_gravitation, symbol = G, value = 6.673 10-11, uncertainty = 1.0 10-13,
units =  $\frac{m^3}{kg s^2}$ 
```

Obrázek 1.32: Použití balíku ScientificConstants.

```
g := GetValue(Constant('g')) * GetUnit(Constant('g'))
          9.80665  $\left[ \frac{m}{s^2} \right]$ 
m := 10[[kg]]:
F = m * g
          F = 98.06650 [[kg]]  $\left[ \frac{m}{s^2} \right]$ 
simplify(%)
          F = 98.06650 [[N]]
```

Obrázek 1.33: Práce s vědeckými konstantami.

| S použitím balíku RealDomain | Bez použití balíku RealDomain |
|---|--|
| <i>with(RealDomain) :</i> | |
| <i>simplify</i> ($\sqrt{x^2}$) $ x $ | <i>simplify</i> ($\sqrt{x^2}$) $\text{esgn}(x) x$ |
| <i>solve</i> ($x^2 + 1 = 0$) | <i>solve</i> ($x^2 + 1 = 0$) $I, -I$ |
| $\ln(e^x)$ x | $\ln(e^x)$ $\ln(e^x)$ |
| $\ln(-5)$ <i>undefined</i> | $\ln(-5)$ $\ln(5) + I\pi$ |

Obrázek 1.34: Použití balíku **RealDomain**.

1.5.3 Reálná doména a předpoklady (omezení)

Systém Maple standardně počítá v oboru komplexních čísel. Někdy může být výhodné omezit se pouze na reálná čísla, k čemuž slouží balík **RealDomain**. Rozdíly v chování výpočtů podle toho, zda jsme načítli balík **RealDomain**, dokumentuje obrázek 1.34.

Pokud chceme pracovat v reálném oboru pouze u některých proměnných, případně použít jiné omezení, využijeme příkazu **assuming** – viz příklady na obrázku 1.35.

| | |
|--|-------|
| <i>simplify</i> ($\sqrt{x^2}$) assuming $x :: real$ | $ x $ |
| <i>simplify</i> ($\sqrt{x^2}$) assuming $x :: positive$ | x |
| <i>simplify</i> ($(-1)^x$) assuming $x :: even$ | 1 |
| <i>simplify</i> ($\cos(n * \pi)$) assuming $n :: odd$ | -1 |
| $\ln(e^x)$ assuming $real$ | x |

Obrázek 1.35: Omezení jednotlivých proměnných.

1.6 Základní matematické výpočty

1.6.1 Úpravy výrazů

K úpravám výrazů slouží několik různých příkazů. Pro zjednodušování jsou to především příkazy **simplify**, **normal** a **combine**. **Simplify** je základní zjednodušující příkaz, **normal** je příkaz určený pro úpravu zlomků, **combine** spojuje jednotlivé výrazy.

The image shows a screenshot of a computer algebra system interface with the following content:

| | | |
|-----------------------|---|--------------------------|
| <code>simplify</code> | $\left(\frac{x}{x+2y} + \frac{x-2y}{x^2-4y^2} \right)$ | $\frac{x+1}{x+2y}$ |
| <code>simplify</code> | $(e^{b+\ln(c)})$ | $e^b \cdot c$ |
| <code>normal</code> | $\left(\frac{1}{x} + \frac{x}{x+1} \right)$ | $\frac{x+1+x^2}{x(x+1)}$ |
| <code>normal</code> | $\left(\frac{1}{x} + \frac{x}{x+1}, \text{expanded} \right)$ | $\frac{x+1+x^2}{x^2+x}$ |
| <code>combine</code> | $(e^{\sin(a) \cdot \cos(b)} \cdot e^{\cos(a) \cdot \sin(b)})$ | $e^{\sin(a+b)}$ |
| <code>combine</code> | $(2 \cdot \sin(x) \cdot \cos(x))$ | $\sin(2x)$ |

Obrázek 1.36: Zjednodušení výrazů.

Dále jsou k dispozici například příkazy **collect** (vytýkání), **coef** (koeficient u zvoleného členu polynomu), **sort** (seřazení členů polynomu), **expand** (roznásobení), **factor** (rozklad na kořenové činitele) a **roots** (nalezení kořenů polynomu).

1.6.2 Funkce

Definovat funkci můžeme více způsoby. Při zadávání (pouze) na klávesnici napíšeme název funkce, symbol pro přiřazení (`:=`), argument funkce, šipku vpravo a funkční předpis. Šipku přitom vytvoříme jako pomlčku následovanou uzavírající lomenou závorkou (symbol „větší než“). Chceme-li tedy vytvořit funkci $f(x) = x^2$, zapíšeme do zápisníku příkaz `f:=x->x^2`.

```

collect(a*ln(x)-ln(x)*x-x, ln(x))
      (a-x) ln(x) - x
collect(a+a*sqrt(3), a)
      (1+sqrt(3)) a
p1 := x^6 - 8*x^5 + 23*x^4 + 68*x^3 - 244*x^2 - 224*x + 384 :
coeff(p1, x, 4)
      23
coeff(p1, x, 1)
      -224
coeff(p1, x, 0)
      384

sort(1+x^3+3*x^2-4*x^4)
      -4 x^4 + x^3 + 3 x^2 + 1
sort(1+x^3+3*x^2-4*x^4, x, tdeg, ascending)
      1 + 3 x^2 + x^3 - 4 x^4

```

Obrázek 1.37: Vytýkání ve výrazech a práce s polynomy.

```

pol1 := x^6 - 8*x^5 + 23*x^4 + 68*x^3 - 244*x^2 - 224*x + 384
      x^6 - 8 x^5 + 23 x^4 + 68 x^3 - 244 x^2 - 224 x + 384
pol2 := x^4 - 6*x^3 + 22*x^2 - 48*x + 40
      x^4 - 6 x^3 + 22 x^2 - 48 x + 40

s1 := factor(pol1)
      (x-1) (x-3) (x^2-8x+32) (x+2)^2
s2 := factor(pol2)
      (x^2-2x+10) (x-2)^2

expand(s1)
      x^6 - 8 x^5 + 23 x^4 + 68 x^3 - 244 x^2 - 224 x + 384
expand(s2)
      x^4 - 6 x^3 + 22 x^2 - 48 x + 40

roots(pol1)
      [[1, 1], [3, 1], [-2, 2]]
roots(pol2)
      [[2, 2]]

```

Obrázek 1.38: Práce s polynomy.

| | |
|--|---|
| $f := x \rightarrow x^2$ | $x \rightarrow x^2$ |
| $g := (x, y) \rightarrow x \cdot y$ | $(x, y) \rightarrow x y$ |
| $h := (a, b, c, d, e) \rightarrow a \cdot b \cdot c + d \cdot e$ | $(a, b, c, d, e) \rightarrow a b c + d e$ |
| $z(x) := x^2$ | $x \rightarrow x^2$ |
| $f(5)$ | 25 |
| $z(5)$ | 25 |
| $g(2, a + 3)$ | $2 a + 6$ |
| $h(1, 2, 3, 4, 5)$ | 26 |

Obrázek 1.39: Definice funkce.

Pro zjednodušení můžeme využít palet. Jednak je možné při vytváření příkazu použít šipku z palety **Arrows**, nebo můžeme vzít celou šablonu příkazu vytvoření funkce z palety **Expression** a modifikovat v ní požadované symboly. Vytvářet přitom můžeme funkce libovolného počtu proměnných. V prostředí *Standard worksheet* (s nímž celou dobu pracujeme) můžeme funkci vytvořit též zápisem bez šipky: $f(x) := x^2$.

Po odkliknutí příkazu musíme v následně zobrazeném vyskakujícím okénku potvrdit, že se jedná o definici funkce. Funkční hodnotu definované funkce v daném bodě získáme zápisem názvu funkce spolu s hodnotami parametrů v závorce (nemusíme přitom zadávat pouze numerické hodnoty).

Důležité je rozeznávat funkce od výrazů. Jestliže vytvoříme výraz, například x^2 , a přiřadíme jej k nějaké proměnné, např. g , jedná se stále pouze o výraz. Hodnotu g pro $x = 5$ nemůžeme proto určit jako funkční hodnotu v bodě 5, ale musíme použít vyhodnocovacího příkazu **eval**, případně do x přiřadit hodnotu 5, viz obrázek 1.40.

Funkce můžeme též definovat po částech pomocí příkazu **piecewise**. Argumenty v závorce signalizují vždy nejprve interval následovaný funkční hodnotou na tomto intervalu. Poslední interval již zapisovat nemusíme, stačí funkční hodnota, Maple ji doplní ve zbývajících množině zatím nedefinovaných bodů. Je možné též sestrojít funkci, která je definována pouze na libovolné podmnožině reálných čísel. Pokud má funkce definovaná po částech pouze dva různé předpisy, můžeme k jejímu vytvoření využít symbolu otevřené složené závorky z palety **Expression**.

| | |
|--------------------------|---------------------|
| $f := x \rightarrow x^2$ | $x \rightarrow x^2$ |
| $g := x^2$ | x^2 |
| $f(5)$ | 25 |
| $g(5)$ | $x(5)^2$ |
| $eval(g, x = 5)$ | 25 |
| x | x |
| $x := 5$ | 5 |
| g | 25 |
| $f(x)$ | 25 |

Obrázek 1.40: Rozdíl mezi funkcí a výrazem.

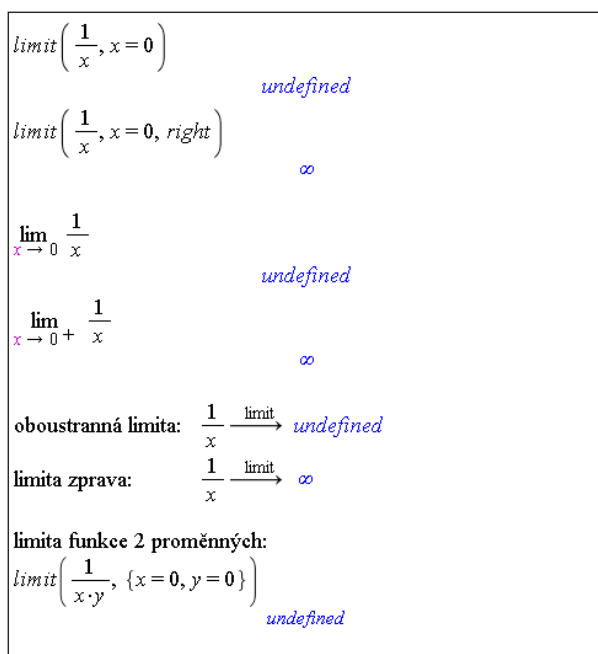
| | |
|---|---|
| $f := x \rightarrow \text{piecewise}(x < 0, -1, x = 0, 0, 1)$ | $x \rightarrow \text{piecewise}(x < 0, -1, x = 0, 0, 1)$ |
| $f(-8)$ | -1 |
| $f(0)$ | 0 |
| $f(153.6)$ | 1 |
| $f2 := x \rightarrow \text{piecewise}(x < 0, -x, \text{undefined})$ | $x \rightarrow \text{piecewise}(x < 0, -x, \text{undefined})$ |
| $f2(-8)$ | 8 |
| $f2(4)$ | undefined |
| $g(x) := \begin{cases} 0 & x < \pi \\ 1 & x \geq \pi \end{cases}$ | $x \rightarrow \text{piecewise}(x < \pi, 0, \pi \leq x, 1)$ |
| $g(0)$ | 0 |
| $g(3)$ | 0 |
| $g(4)$ | 1 |

Obrázek 1.41: Funkce definovaná po částech.

1.6.3 Limity, derivace a integrály

V rámci matematické analýzy si ukážeme, jak v systému Maple určovat limity, derivace a integrály. Na výběr máme znovu několik možností. Ve všech třech případech můžeme využít k tomu určených příkazů, symbolů z palety **Expression**, či kliknutí pravého tlačítka myši a zvolení vhodné položky v kontextové nabídce.

K výpočtu limit slouží příkaz **limit**, který má dva povinné argumenty: funkci a bod, v němž chceme limitu určit. K dispozici je též volitelný parametr pro určení směru limity (zleva, zprava). Místo příkazu můžeme použít odpovídající symbol z palety **Expression**, nebo zapsat do dokumentu funkci, kliknout na ni pravým tlačítkem a z kontextové nabídky zvolit položku **Limit**. Všechny tři postupy ilustruje obrázek 1.42. Určit limitu funkce více proměnných je možné pouze pomocí příkazu **limit**.



Obrázek 1.42: Určování limity.

Derivaci funkce můžeme určit příkazem **diff**, který má dva povinné parametry: funkci a proměnnou, podle níž se má derivovat. Dále můžeme využít symbol pro diferenciální operátor z palety **Expression** (k dispozici jsou dva: „obyčejný“ operátor a operátor partiální derivace), nebo kliknout pravým tlačítkem myši na funkci, již chceme derivovat, a z kontextové nabídky zvolit položku **Differentiate** > ... Ukázkové příklady poskytuje obrázek 1.43.

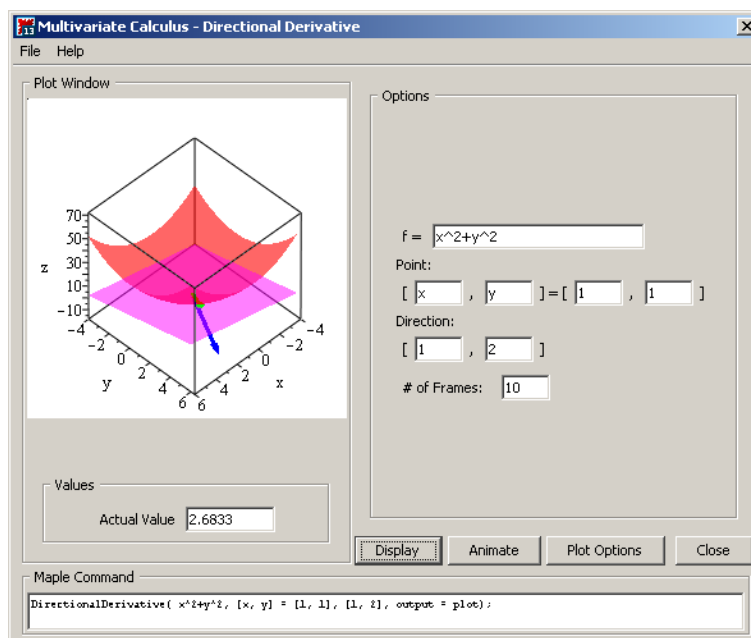
System Maple umí počítat i směrové derivace a pro funkci dvou proměnných má též nástroj na její vizualizaci. Nalezneme jej v hlavním menu kliknutím na **Tools** > **Tutors** > **Calculus - Multi-Variable** > **Directional Derivatives ...** (obrázek 1.44). Výstupem z nástroje je animace ilustrující sekvenci derivací v zadaném bodě v různých směrech.

K symbolickému výpočtu směrové derivace slouží příkaz **DirectionalDerivative** z balíku **Student[MultivariateCalculus]**.

| | |
|--|------------------|
| $\text{diff}\left(\frac{1}{x}, x\right)$ | $-\frac{1}{x^2}$ |
| $\frac{d}{dx} \frac{1}{x}$ | $-\frac{1}{x^2}$ |
| $\frac{\partial}{\partial x} \frac{1}{x}$ | $-\frac{1}{x^2}$ |
| $\frac{1}{x} \xrightarrow{\text{differentiate w.r.t. } x}$ | $-\frac{1}{x^2}$ |

| | |
|---|---------------------|
| $\text{diff}\left(\frac{1}{x}, x, x\right)$ | $\frac{2}{x^3}$ |
| $\text{diff}\left(\frac{1}{x \cdot y}, x, y\right)$ | $\frac{1}{x^2 y^2}$ |
| $\frac{d^2}{dx^2} \frac{1}{x}$ | $\frac{2}{x^3}$ |
| $\frac{d}{dx} \frac{d}{dy} \frac{1}{x \cdot y}$ | $\frac{1}{x^2 y^2}$ |

Obrázek 1.43: Výpočet derivace.



Obrázek 1.44: Zobrazení směrové derivace.

| |
|---|
| <pre>with (Student[MultivariateCalculus]) : DirectionalDerivative(x^2 + y^2, [x, y] = [1, 1], [1, 2]) $\frac{6}{5} \sqrt{5}$</pre> |
|---|

Obrázek 1.45: Výpočet směrové derivace.

Integrál ze zadané funkce můžeme určit buď příkazem **int**, nebo příslušným symbolem z palety **Expression**, případně kliknutím pravým tlačítkem myši na funkci a z kontextové nabídky zvolením položky **Integrate** > ... Použití na jednoduchých příkladech ilustruje obrázek 1.46.

| | | | |
|--|----------|--|-----------------|
| $\text{int}\left(\frac{1}{x}, x\right)$ | $\ln(x)$ | $\text{int}\left(\frac{1}{x}, x, x\right)$ | $x \ln(x) - x$ |
| $\text{int}\left(\frac{1}{x}, x=1..e\right)$ | 1 | $\text{int}\left(\frac{1}{x \cdot y}, x, y\right)$ | $\ln(x) \ln(y)$ |
| $\int \frac{1}{x} dx$ | $\ln(x)$ | $\text{int}\left(\frac{1}{x \cdot y}, x=1..e, y=e..e^3\right)$ | 2 |
| $\int_1^e \frac{1}{x} dx$ | 1 | $\iint \frac{1}{x} dx dx$ | $x \ln(x) - x$ |
| $\frac{1}{x} \xrightarrow{\text{integrate w.r.t. } x}$ | $\ln(x)$ | $\int_1^e \int_e^{e^3} \frac{1}{x \cdot y} dy dx$ | 2 |

Obrázek 1.46: Výpočet integrálu.

1.6.4 Lineární algebra

Jak jsme se již zmínili v části 1.5 (Balíky), pro výpočty lineární algebry slouží balík **LinearAlgebra**. Vytvářet matice, vektory a provádět s nimi základní operace je možné bez načtení tohoto balíku (obrázek 1.47). Jako obvykle máme několik možností, jak vytvořit matici, respektive vektor. Tou první jsou příkazy **Matrix** a **Vector**. Matici též můžeme zadat výčtem svých prvků po řádcích do lomenných závorek. Jednotlivé řádky matice oddělíme svislými čarami. Vektor, jakožto speciální typ matice, můžeme vytvořit obdobně. K dispozici je také paleta **Matrix**.

Balík **LinearAlgebra** obsahuje mnoho příkazů, z nichž zmiňme a na jednoduchých příkladech ukažme některé základní (obrázek 1.48). K určení vlastních čísel a vlastních vektorů zadané matice slouží příkazy **Eigenvalues** a **Eigenvectors**. Příkazem **CharacteristicPolynomial** zobrazíme charakteristický polynom matice, příkazem **JordanForm**

| | |
|--|---|
| $Matrix([[1, 2, 0], [1, 1, 0], [0, 1, 1]])$ $\begin{bmatrix} 1 & 2 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$ | $N := \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$ |
| $\langle 1, 2, 0 1, 1, 0 0, 1, 1 \rangle$ $\begin{bmatrix} 1 & 1 & 0 \\ 2 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$ | $2 \cdot N - 3 \cdot M$ $\begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$ $\begin{bmatrix} 7 & 6 \\ 5 & 4 \end{bmatrix}$ |
| $Vector[row]([1, 0, 1])$ $\begin{bmatrix} 1 & 0 & 1 \end{bmatrix}$ | $M \cdot N$ # nasobeni matic $\begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$ |
| $Vector[column]([1, 0, 1])$ $\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$ | M^{-1} # inverzni matice $\begin{bmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{bmatrix}$ |
| $M := \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ | $M^{%T}$ # transpozice matice $\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$ |

Obrázek 1.47: Práce s maticemi.

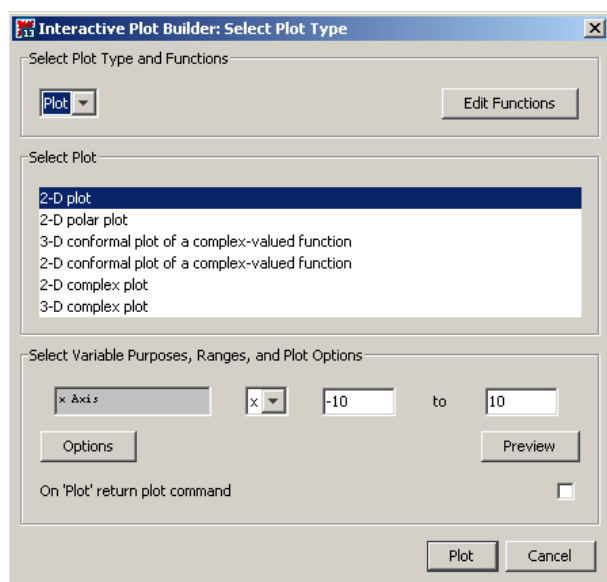
ji převedeme na Jordanův kanonický tvar. K dispozici je též příkaz **GaussElimination** pro převod matice na schodovitý tvar, příkazem **LinearSolve** můžeme řešit systémy lineárních rovnic.

1.7 Kreslení a animace

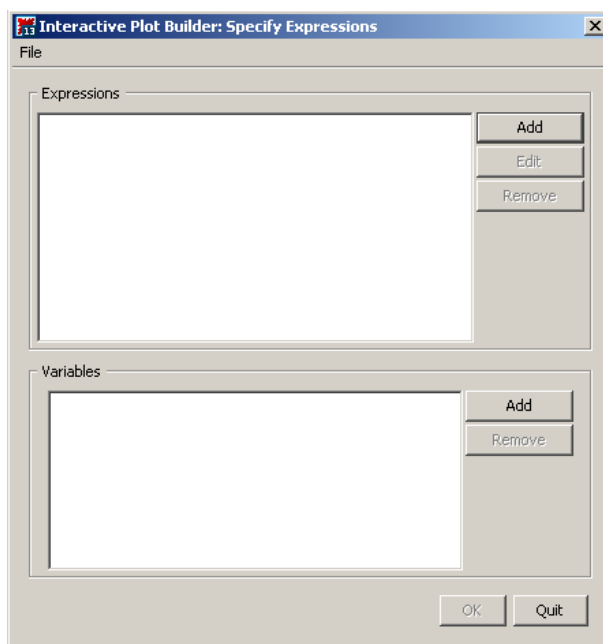
Interaktivní prostředí *Standard worksheet* nabízí hned několik možností k vykreslení grafu funkce. Nejrychlejší a zřejmě nejjednodušší je zadat do zápisníku předpis funkce, kliknout na něj pravým tlačítkem myši a z kontextové nabídky zvolit **Plots > 2-D Plot** (v případě funkcí jedné proměnné). Zvolením **Plots > 3-D Plot** z kontextové nabídky spolu se specifikací proměnných zobrazíme funkce dvou proměnných.

Dále můžeme využít pomocníka **Plot Builder**, a to dvěma způsoby. Buď opět zapíšeme do dokumentu předpis funkce, klikneme pravým tlačítkem myši a zvolíme **Plots > Plot Builder**, nebo zamíříme do hlavního menu a vybereme **Tools > Assistants > Plot Builder...**. V prvním případě se objeví okénko **Interactive Plot Builder** (obrázek 1.49), v němž upřesníme typ vykreslení. Pokud uvažujeme funkci jedné proměnné, obvyklou volbou je **2-D Plot**. Můžeme však volit i jiné jako například vykreslení v polárních souřadnicích (**2-D polar plot**). Obdobně je tomu u trojrozměrného vykreslování funkcí dvou proměnných. Kliknutím na tlačítko **Plot** graf zobrazíme v dokumentu.

| | |
|---|---|
| <pre> B := Matrix([[2, -1, 4], [0, 1, 0], [1, -1, -1]]) </pre> $\begin{bmatrix} 2 & -1 & 4 \\ 0 & 1 & 0 \\ 1 & -1 & -1 \end{bmatrix}$ <pre> with(LinearAlgebra): Eigenvectors(B) # vlastní vektory a k nim prisl. vl. cisla </pre> $\begin{bmatrix} -2 \\ 1 \\ 3 \end{bmatrix}, \begin{bmatrix} -1 & 1 & 4 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$ <pre> CharacteristicPolynomial(B, λ) # charakteristicky polynom </pre> $6 + \lambda^3 - 2\lambda^2 - 5\lambda$ <pre> J := JordanForm(B) # Jordanova matice prislusna k mat. B </pre> $\begin{bmatrix} -2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix}$ | <pre> A := Matrix([[1, 2, 3], [1, 0, 1], [0, 1, 2]]) </pre> $\begin{bmatrix} 1 & 2 & 3 \\ 1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$ <pre> b := <1, 2, 1> </pre> $\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$ <pre> LinearSolve(A, b) # resi rovnici A·x = b </pre> $\begin{bmatrix} \frac{1}{2} \\ -2 \\ \frac{3}{2} \end{bmatrix}$ <pre> GaussianElimination(A) # Gaussova eliminace </pre> $\begin{bmatrix} 1 & 2 & 3 \\ 0 & -2 & -2 \\ 0 & 0 & 1 \end{bmatrix}$ |
|---|---|

Obrázek 1.48: Příkazy balíku **LinearAlgebra**.Obrázek 1.49: Zvolení typu vykreslení v **Plot Builderu**.

V druhém případě, kdy **Plot Builder** vyvoláme z hlavního menu, se nám objeví okénko na obrázku 1.50. Do něj zadáme předpis funkce, kterou chceme zobrazit (zadání nám umožní tlačítka **Add**, resp. **Edit**), a proměnné (pokud výraz obsahuje pouze proměnné, systém je vyplní sám). Kliknutím na tlačítko **OK** přejdeme do již známého okénka pro zvolení typu vykreslení (obrázek 1.49).

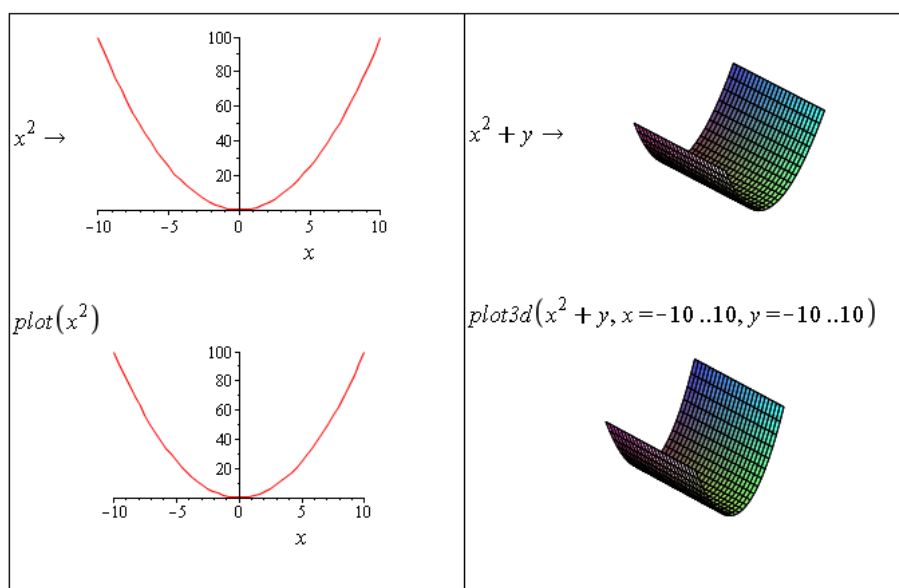


Obrázek 1.50: Okénko pro zadání předpisu funkce a proměnných v **Plot Builderu**.

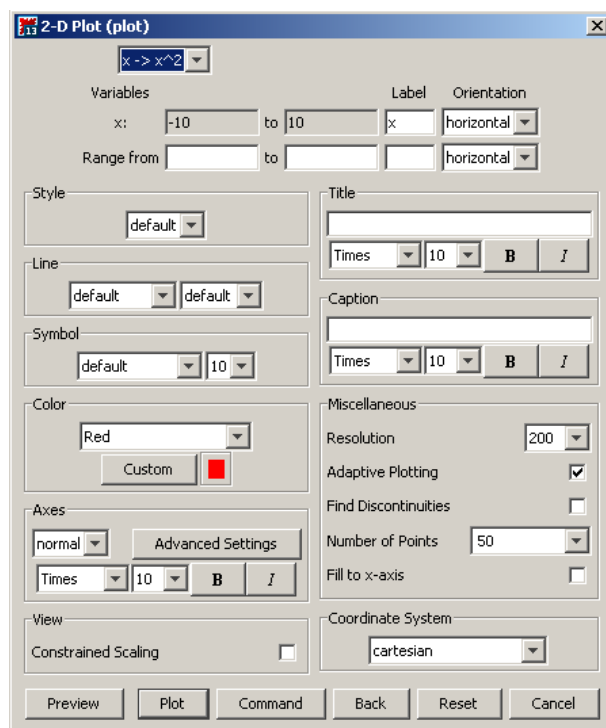
Nakonec máme možnost k vykreslení grafu funkce použít příkaz **plot** v případě funkce jedné proměnné a příkaz **plot3d** pro funkci dvou proměnných. Příkazu **plot3d** musíme zadat i rozsahy hodnot, kterých mohou proměnné nabývat. Při trojrozměrném vykreslování se graf standardně zobrazuje bez souřadných os (obrázek 1.51).

Při vykreslování je k dispozici několik atributů, které mění podobu grafu. Opět je několik možností, jak atributy zadávat. Při použití pomocníka **Plot Builder** se v okénku **Interactive Plot Builder** (obrázek 1.49) objevuje tlačítko **Options**. Kliknutím na toto tlačítko přejdeme na okénko (viz obrázek 1.52) umožňující nastavit parametry vykreslení jako jsou rozsah hodnot závisle i nezávisle proměnné, barva a styl vykreslované křivky, titulek grafu, legendu atd. Užitečné je navíc tlačítko **Preview** umožňující předběžně si prohlédnout současný stav a následně pokračovat v dalším nastavování atributů vykreslení grafu.

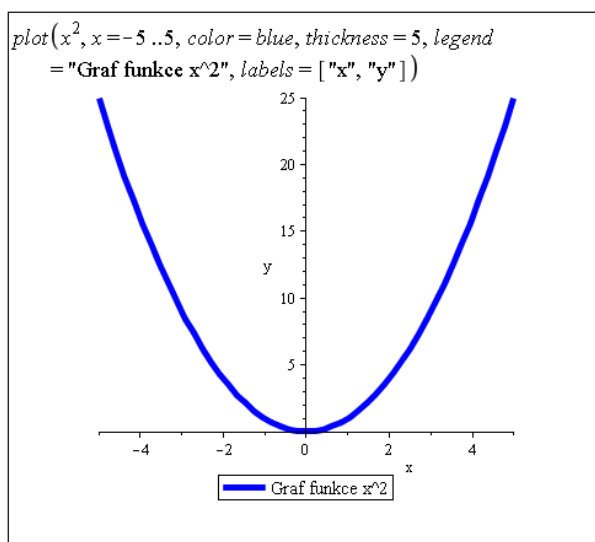
Při použití příkazu **plot** (resp. **plot3d**) můžeme totéž provést specifikací nepovinných parametrů jako jsou **thickness** pro tloušťku křivky, **color** pro její barvu, **discont** pro zobrazení nespojitostí, **labels** pro popisky os, **legend** pro tvar legendy u obrázku, **axes** pro nastavení souřadných os a další. Ukázkou použití příkazu **plot** s nastavením některých nepovinných parametrů nabízí obrázek 1.53.



Obrázek 1.51: Vykreslení grafů pomocí kontextové nabídky a příkazů **plot**, **plot3d**.



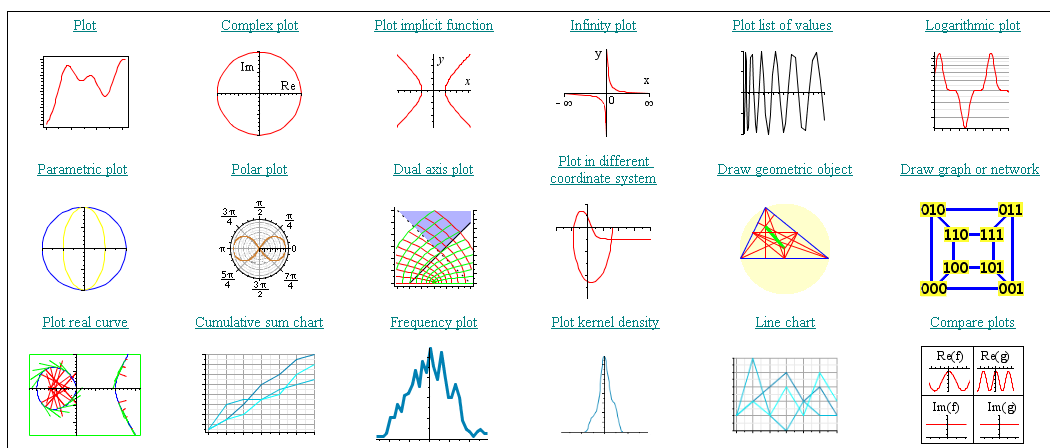
Obrázek 1.52: Okénko **Plot Builderu** pro nastavení parametrů grafu.



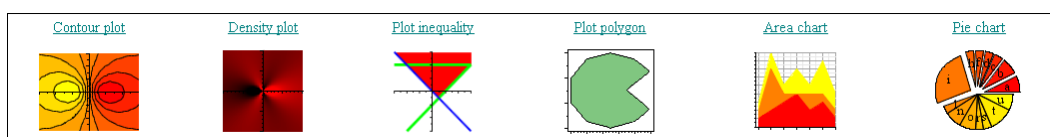
Obrázek 1.53: Vykreslení grafu při specifikaci některých nepovinných parametrů.

I u vytvořeného grafu lze upravovat jeho vzhled. Jednak můžeme na graf kliknout pravým tlačítkem myši a z kontextové nabídky vybírat vlastnosti grafu, které jsme mohli měnit již dříve, nebo můžeme využít kontextové lišty těsně nad dokumentem. Po kliknutí levým tlačítkem myši na graf se ve zmíněné liště zobrazí nástroje skupiny nazvané **Plot**. K dispozici je též skupina s názvem **Drawing**. Nástroje v těchto skupinách umožňují do hotového grafu přidávat text, kreslit, či jinak upravovat.

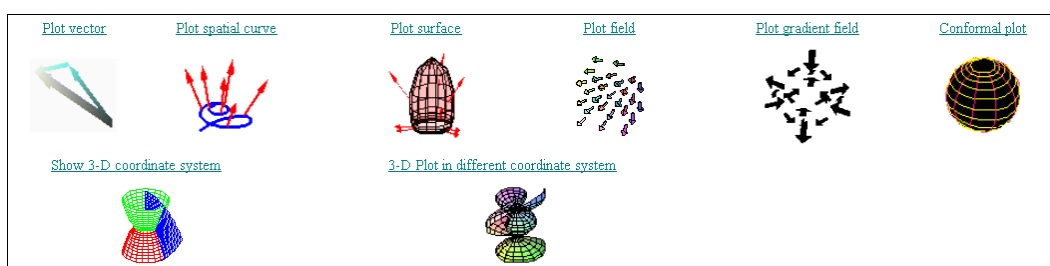
Možnosti systému Maple při kreslení grafů jsou daleko větší. Pro představu, co všechno Maple umí, si ukažme průvodce grafy (*Plotting Guide*), který je součástí nápovědy systému. Jde o uspořádaný přehled grafů všech jeho funkcí. Je to velmi dobrý pomocník umožňující uživateli získat rychlý přehled o vizualizačních možnostech Maple. Ukázky z tohoto průvodce jsou uvedeny na obrázcích 1.54 až 1.57.



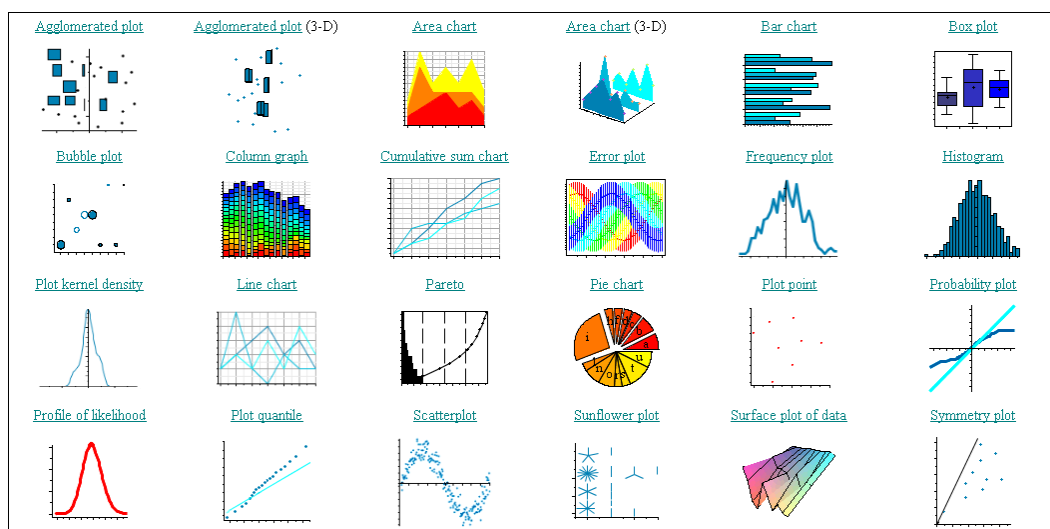
Obrázek 1.54: Dvourozměrné čárové grafy.



Obrázek 1.55: Dvourozměrné vyplňované grafy.



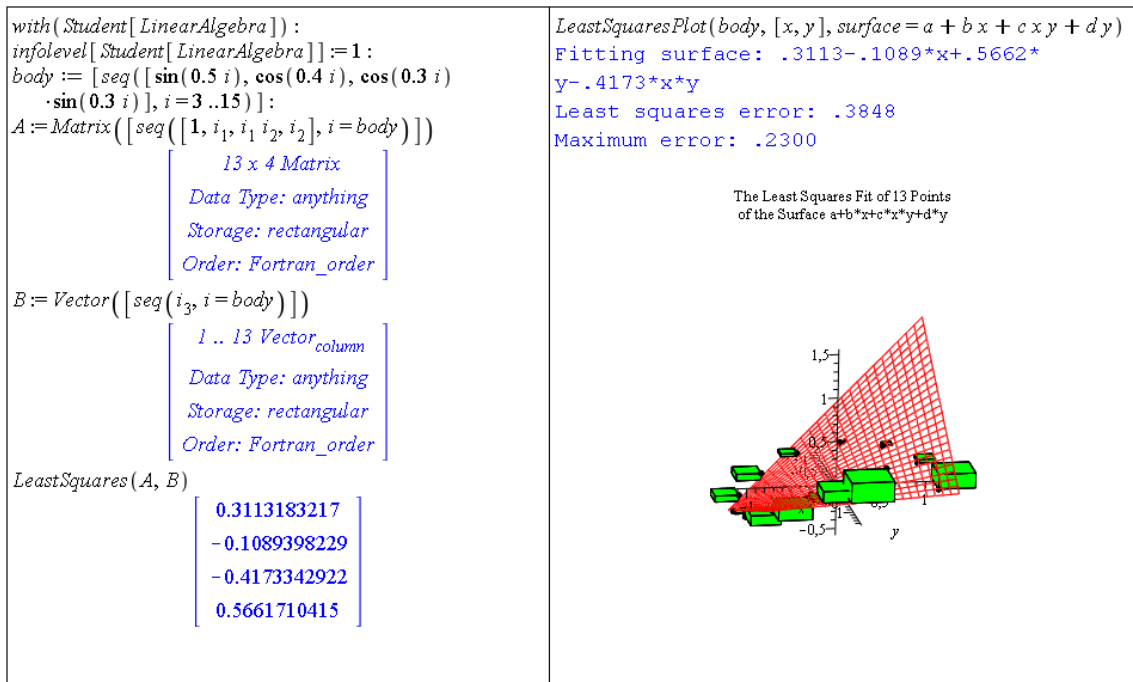
Obrázek 1.56: Trojrozměrné vektorové grafy.



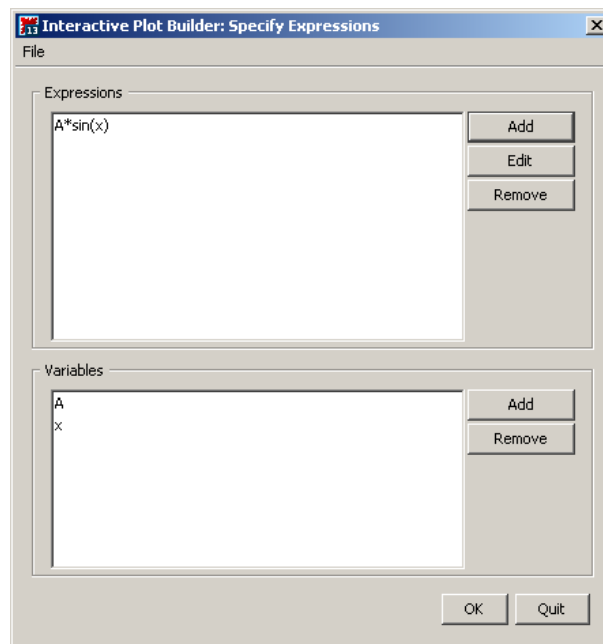
Obrázek 1.57: Statistické grafy.

Kliknutím na příslušný název (příkaz) získáme podrobnou nápovědu k tomuto příkazu. Na konkrétním příkladu si ukažme vizualizaci „proložení“ lineární parametrické funkce naměřenými body v prostoru. Využijeme k tomu graf s názvem **Plot least squares fit** z třídy matematických konceptů.

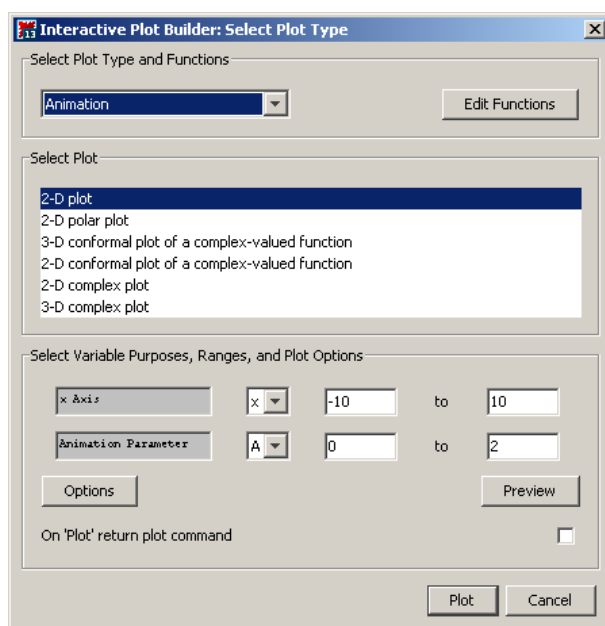
Na obrázku 1.58 je prezentováno řešení příkladu, jak „proložit“ metodou nejmenších čtverců lineární funkci $f(x, y) = a + b \cdot x \cdot y + c \cdot x \cdot y + d \cdot y$ parametry a, b, c, d množinou třinácti bodů $\{[x = \sin(0.5 \cdot t), y = \cos(0.4 \cdot t), z = \cos(0.3 \cdot t) \cdot \sin(0.3 \cdot t)], t = 3, \dots, 15\} \in E_3$.



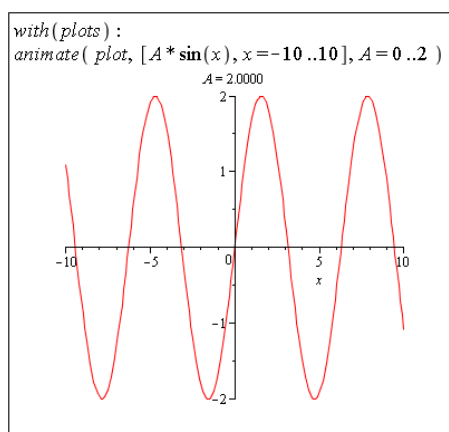
Obrázek 1.58: Vizualizace „proložení“ funkce body metodou nejmenších čtverců.

Obrázek 1.59: Zadání předpisu funkce v **Plot Builderu**.

V systému Maple můžeme též vytvářet animace. Animace se skládá z několika grafů, které jsou po spuštění zobrazené v sekvenci za sebou. Vytvoříme ji buď příkazem **animate** z balíku **plots**, nebo pomocí **Plot Builderu**. Obrázky 1.59 a 1.60 ukazují nastavení **Plot Builderu** pro vytvoření animace, obrázek 1.61 ilustruje tentýž příklad na použití příkazu **animate**.



Obrázek 1.60: Zvolení druhu vykreslení (animace) v **Plot Builderu**.



Obrázek 1.61: Animace vytvořená příkazem **animate**.

Ukončení **Plot Builderu**, resp. provedení příkazu, umístí do dokumentu „prázdný“ graf. Kliknutím na něj zobrazíme skupinu nástrojů v kontextové liště s názvem **Animation**. Pomocí těchto nástrojů můžeme animaci spustit, změnit její rychlost, podívat se

na libovolný snímek animace atd.

Animace můžeme upravovat stejně jako grafy (vytvořené příkazy **plots**, **plots3d**), tj. měnit tloušťku, barvu a druh křivky, souřadné osy, legendu apod. Navíc máme k dispozici několik nepovinných parametrů, díky nimž můžeme například určit počet grafů, z nichž se animace skládá, nebo kolik grafů má zůstat trvale zobrazených (po spuštění animace).

1.8 Práce s neurčitostmi

Pro práci s neurčitostmi nabízí systém Maple několik nástrojů. Jejich použití závisí zpravidla na množství informace, kterou máme.

1.8.1 Intervalová aritmetika

V případě, že známe pouze velikost (intervalu) neurčitosti, využíváme intervalové aritmetiky implementované v balíku **Tolerances**. Proměnné (resp. konstanty) definujeme spolu s intervalem chyby (neurčitosti) pomocí symbolu $\&\pm$ a pracujeme s nimi jako s „obyčejnými“ proměnnými (resp. konstantami), viz obrázek 1.62.

```
with(Tolerances) :
a := 5 &+ - 0.05           5.00 ± 0.0500
b := 2 &+ - 0.03           2.00 ± 0.0300

a + b                       7.00 ± 0.0800
a · b                       10.0 ± 0.250
a2 - 3 · b + 1           20.0 ± 0.590
```

Obrázek 1.62: Intervalová aritmetika s balíkem **Tolerances**.

1.8.2 Scientific Error Analysis

Balík **ScientificErrorAnalysis** umožňuje pracovat s veličinami, které mají nějakou střední hodnotu a přiřazenou neurčitost (resp. chybu). Na rozdíl od intervalové aritmetiky je nyní interval neurčitosti prokládán normálním rozložením hodnot se střední hodnotou ve středu intervalu a směrodatnou odchylkou rovnou délce poloviny intervalu. Navíc je možné přiřazovat veličinám korelaci (vzájemný vztah) mezi sebou. Veličiny definujeme příkazem **Quantity**, propagaci chyb daným výrazem určíme příkazem **combine**, korelaci nastavíme příkazem **SetCorrelation**. Ilustraci poskytuje obrázek 1.63.


```

with(ScientificErrorAnalysis) :

a := Quantity(5, 0.05)
Quantity(5, 0.05)
b := Quantity(2, 0.03)
Quantity(2, 0.03)

combine(a + b, errors)
Quantity(7., 0.05830951895)
combine(a·b, errors)
Quantity(10., 0.1802775638)
combine(a2 - 3·b + 1, errors)
Quantity(20., 0.5080354318)

SetCorrelation(a, b, 1)

combine(a + b, errors)
Quantity(7., 0.08000000000)
combine(a·b, errors)
Quantity(10., 0.2500000000)

SetCorrelation(a, b, -0.5)

combine(a + b, errors)
Quantity(7., 0.04358898944)
combine(a·b, errors)
Quantity(10., 0.1322875656)

```

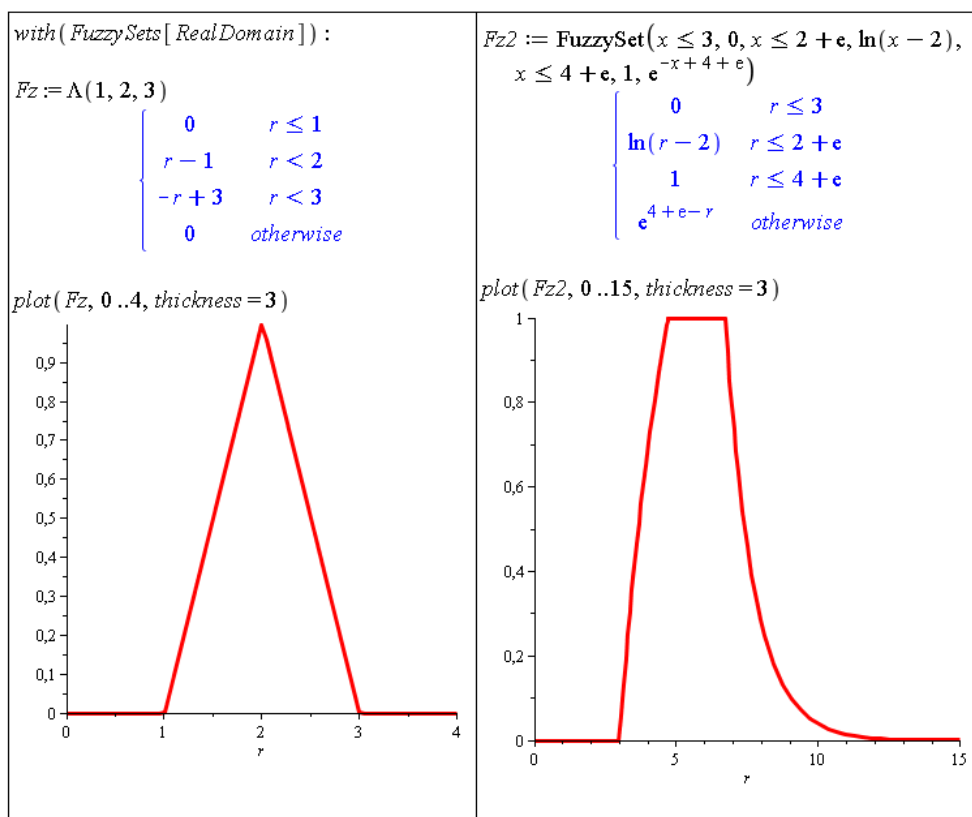
Obrázek 1.63: Práce s balíkem **ScientificErrorAnalysis**.

1.8.3 Fuzzy Sets Toolbox

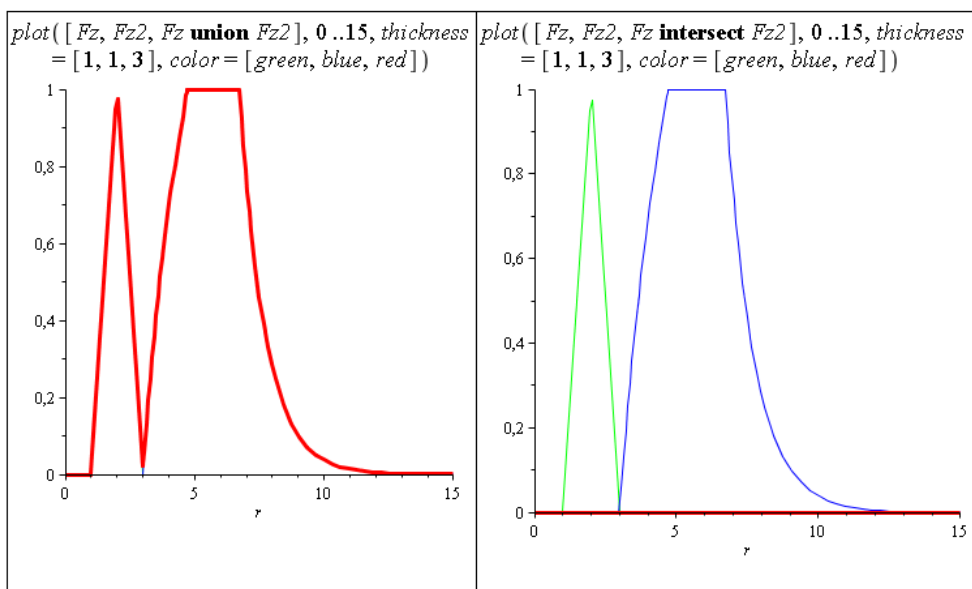
Toolbox **FuzzySets** není standardní součástí systému Maple a je nutné jej dokoupit. Po instalaci funguje jako klasický balík příkazů, který umožňuje využívat fuzzy logiku a konstruovat a používat fuzzy množiny. Fuzzy množina může být definována obecně na libovolné množině objektů U , je však možné (a často vhodné) omezit balík **FuzzySets** pouze na množinu reálných čísel. K dispozici je několik konstruktorů fuzzy množin, přičemž ten nejuniverzálnější je téměř totožný s příkazem **piecewise** pro tvorbu po částech definovaných funkcí.

Obrázek 1.64 ilustruje načtení balíku **FuzzySets** s omezením na množinu reálných čísel a konstrukci dvou různých fuzzy množin s následným vykreslením jejich funkcí příslušnosti. V prvním případě (vlevo) je použit konstruktor Λ vytvářející fuzzy množinu s funkcí příslušnosti, jejíž tvar odpovídá tomuto řeckému písmenu. Napravo je fuzzy množina vytvořena pomocí obecného konstruktora (který je analogický k příkazu **piecewise**).

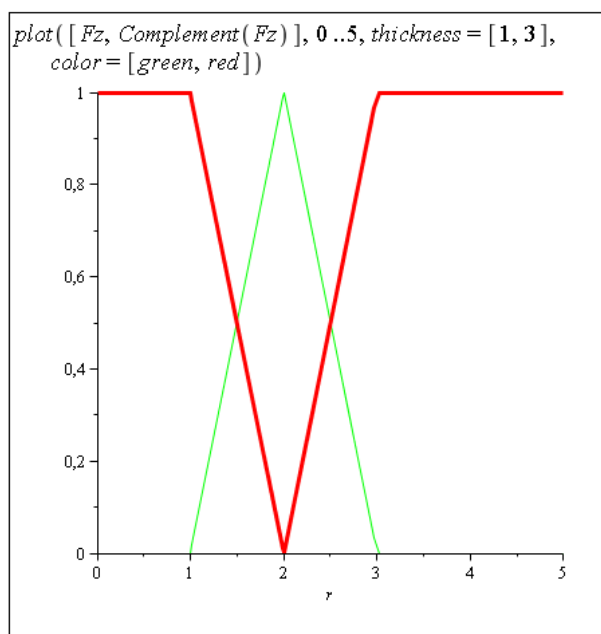
S fuzzy množinami můžeme provádět množinové operace. Na obrázku 1.65 jsou vyznačeny operace sjednocení (vlevo) a průnik (vpravo) právě vytvořených množin. K dispozici přitom máme i další množinové operace jako jsou odečítání množin, komplement množiny a další. Ukázkou komplementu fuzzy množiny představuje obrázek 1.66. Na obrázcích (1.65, 1.66) jsou vždy vykresleny funkce příslušnosti původních množin i těch výsledných (po aplikaci dané operace). Výsledné množiny jsou znázorněny tlustou čarou.



Obrázek 1.64: Konstrukce fuzzy množiny a její zobrazení.



Obrázek 1.65: Zobrazení sjednocení a průniku fuzzy množin.



Obrázek 1.66: Komplement fuzzy množiny.

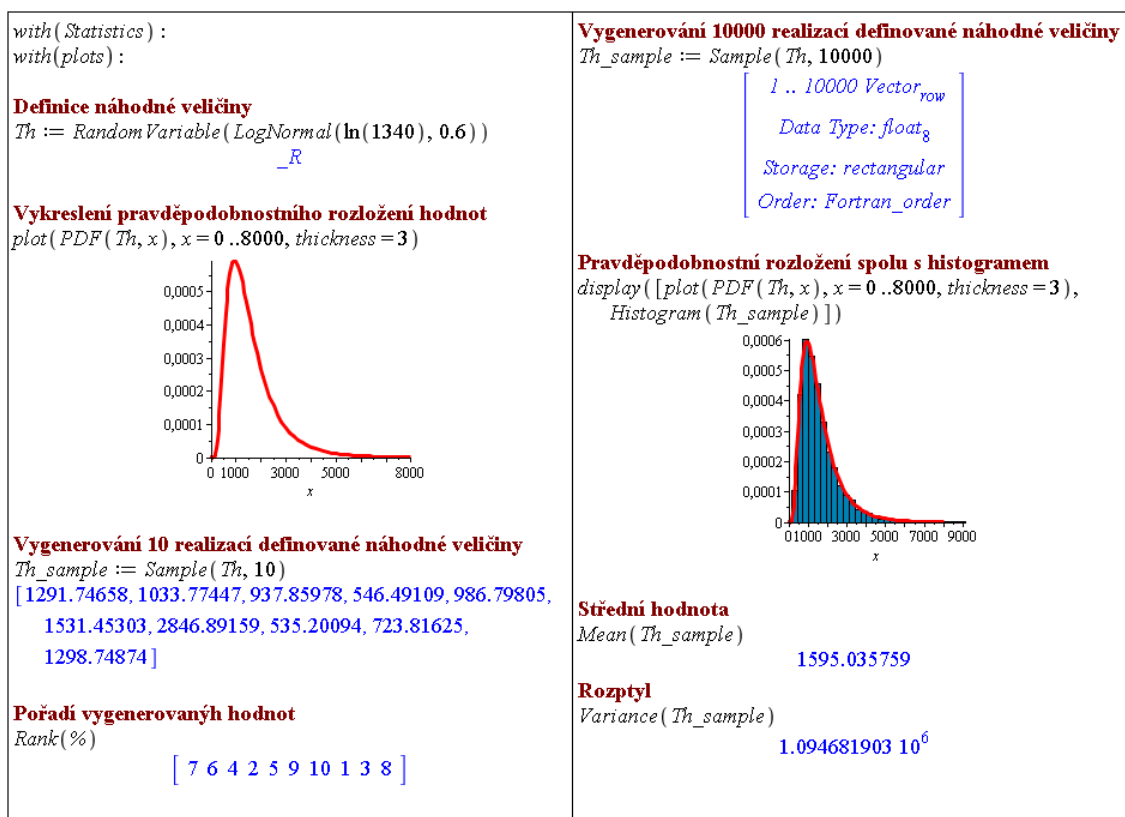
Balík **FuzzySets** poskytuje též prostředky defuzifikace, tedy přiřazení číselné hodnoty dané fuzzy množině. Standardní proces defuzifikace přiřadí fuzzy množině její těžiště (první moment), je však možné použít i jiné typy defuzifikací. Pro omezení na doménu reálných čísel je k dispozici ještě metoda hledající medián funkce příslušnosti (specifikuje se pomocí atributu **method=area**), viz obrázek 1.67.

| | |
|--|---|
| <code>Defuzzify(Fz)</code> | 2 |
| <code>Defuzzify(Fz2)</code> | $\frac{\frac{9}{4} + \frac{1}{4} e^2 + \frac{1}{2} (4+e)^2 - \frac{1}{2} (2+e)^2 + e^{4+e} \left(\frac{5 e^{-4}}{e^e} + \frac{e^{-4} e}{e^e} \right)}{3 + \frac{e^{4+e} e^{-4}}{e^e}}$ |
| <code>evalf(%)</code> | 5.813027376 |
| <code>Defuzzify(Fz, method=area)</code> | 2.000000000 |
| <code>Defuzzify(Fz2, method=area)</code> | 5.718281828 |

Obrázek 1.67: Defuzifikace.

1.8.4 Statistics

Pro práci s nástroji matematické statistiky a analýzy dat je určen balík **Statistics**. Nabízí širokou škálu příkazů, mezi nimiž jsou i příkazy pro vytváření náhodných veličin mnoha různých pravděpodobnostních rozložení a práci s nimi. Z pohledu neurčitosti se zaměříme právě na generování náhodných veličin, které využíváme při Monte Carlo simulacích. Na obrázku 1.68 můžeme vidět načtení balíku **Statistics** a balíku **plots**, který potřebujeme k použití příkazu **display**. Definujeme náhodnou veličinu s lognormálním rozložením pravděpodobnosti (se střední hodnotou 1340 a směrodatnou odchylkou rovnou 0.6) a vykreslíme její hustotu. Následně vygenerujeme 10 realizací této veličiny, kterým přiřadíme příkazem **Rank** pořadí (od nejmenší hodnoty). Poté vygenerujeme 1000 realizací definované náhodné veličiny, opět vykreslíme její hustotu, tentokrát spolu s histogramem vygenerovaných hodnot. Příkazy **Mean** a **Variance** určíme střední hodnotu a rozptyl vygenerovaných hodnot.



Obrázek 1.68: Práce s balíkem **Statistics**.

1.9 Základy programování

Doposud jsme vykonávali příkazy jednotlivě. Systém Maple obsahuje též programovací jazyk, který umožňuje vytváření složitějších programových konstrukcí.

1.9.1 Podmíněný příkaz *if*

Syntax:

```

if      podmínka1  then  příkazy1
elif    podmínka2  then  příkazy2
elif    podmínka3  then  příkazy3
      :
      :
      :
      else  příkazyN
end if

```

Jak můžeme vidět na obrázku 1.69, v podmíněném příkazu nemusí být použity všechny prvky uvedené výše. Podmínku tvoří vždy výraz, u něž je možné rozhodnout, zda je pravdivý, či nikoliv. Příkazů, které se mají provést při splnění podmínky, může být více. Od sebe je oddělíme středníkem, případně dvojtečkou.

Při vytváření příkazu na více řádků použijeme k přechodu mezi řádky současného stisknutí kláves **Shift** a **Enter**. Stisk samotné klávesy **Enter** by způsobil vyhodnocení rozepsaného příkazu.

| | |
|--|---|
| <pre> A := 3 3 if A < 5 then print("A je menší než pět.") end if "A je menší než pět." A := 6 6 if A < 5 then print("A je menší než pět.") else print("A je větší nebo rovno pěti.") end if "A je větší nebo rovno pěti." if A=6 then B := 7; C := 8; E := 156.89 end if 7 8 156.89 </pre> | <pre> A := 3; B := 5 3 5 if (A < 5 and B < 5) then print("A i B jsou menší než pět.") elif (A < 5 and B ≥ 5) then print("A je menší než pět, B je větší nebo rovno pěti.") elif (A ≥ 5 and B < 5) then print("A je menší než pět, B je větší nebo rovno pěti.") else print("A i B jsou větší nebo rovny pěti.") end if "A je menší než pět, B je větší nebo rovno pěti" </pre> |
|--|---|

Obrázek 1.69: Použití podmíněného příkazu **if**.

1.9.2 Cyklus *for*

Syntax 1:

```
for      iterator  from  pocatek  by  prirustek  to  konec  do  prikazy
end do
```

Syntax 2:

```
for promenna in vyraz do prikazy end do
```

Na příkladech (obrázek 1.70) opět vidíme, že nemusíme využít všech prvků obecné syntaxe příkazu. Zejména pokud nevedeme *prirustek*, bude system pracovat s hodnotou *prirustku* rovnou jedné. Příkazů, které se mají v jednom průchodu cyklem provést, může být opět více (a musí být od sebe odděleny středníkem, či dvojtečkou).

| | |
|---|---|
| <pre>for i from 1 to 5 do print(i²) end do 1 4 9 16 25 for i from 1 by 3 to 10 do print(i²) end do 1 16 49 100</pre> | <pre>for i from 10 by -1 to 3 do if isprime(i) then print(i) end if end do 7 5 3 seznam := [22, 97, 222, 397, 622] [22, 97, 222, 397, 622] for i in seznam do print(sqrt(i+3)) end do 5 10 15 20 25</pre> |
|---|---|

Obrázek 1.70: Použití cyklu **for**.

1.9.3 Cyklus *while*

Syntax:

```
while podminka do prikazy end do
```

Na obrázku 1.71 je pomocí cyklu **while** číslo 112 celočíselně dělené dvěma, dokud je zbytek po dělení roven nule (zbytek po dělení určíme příkazem **irem**).

```

x := 112
                                112
while irem(x, 2) = 0 do x := x/2 end do
                                56
                                28
                                14
                                7

```

Obrázek 1.71: Použití cyklu **while**.

1.9.4 Iterativní příkazy

Maple obsahuje několik iterativních příkazů, které jsou optimalizovány pro určité specifické operace.

Tabulka 2: Přehled iterativních příkazů

| Příkaz | Funkce |
|--------------|--|
| seq | vytvoří posloupnost |
| add | vypočítá numerický součet |
| mul | vypočítá numerický součin |
| select | vypíše operandy, které vyhovují podmínce |
| remove | vypíše operandy, které nevyhovují podmínce |
| selectremove | vypíše zvlášť operandy, které vyhovují podmínce, a operandy, které ji nevyhovují |
| map | aplikuje operaci na operandy výrazu |
| zip | aplikuje binární operaci na operandy dvou seznamů či vektorů |

Obrázek 1.72 ilustruje použití několika vybraných iterativních příkazů.

| | |
|--|--|
| $seq(i^2, i = 1 .. 5)$ 1, 4, 9, 16, 25 | $map(x \rightarrow x^2, 2 \cdot x + 3 \cdot y)$ $4x^2 + 9y^2$ |
| $seq(i^2, i \text{ in } [7, 11, 13, 24])$ 49, 121, 169, 576 | $map(x \rightarrow x^2, [7, 11, 13, 24])$ [49, 121, 169, 576] |
| $add(i^2, i = 1 .. 5)$ 55 | $zip((x, y) \rightarrow x + y, [1, 2, 3], [4, 5, 6])$ [5, 7, 9] |
| $add(i^2, i \text{ in } [7, 11, 13, 24])$ 915 | $zip('+', [1, 2, 3], [4, 5, 6])$ [5, 7, 9] |

Obrázek 1.72: Použití iterativních příkazů.

1.9.5 Procedury

V systému Maple můžeme vytvářet i vlastní procedury. Definice procedury začíná vždy slovem **proc** a končí slovy **end proc**. Definici přiřadíme jméno, pomocí něhož následně proceduru voláme (obrázek 1.73).

| | |
|--|---|
| <pre>uvitani :=proc() print("Vítá Vás systém Maple") end proc proc() print("Vítá Vás systém Maple") end proc</pre> | <pre>moje_procedura :=proc(a, b, c) a*b + c end proc proc(a, b, c) a*b + c end proc</pre> |
| <pre>uvitani() "Vítá Vás systém Maple"</pre> | <pre>moje_procedura(1, 2, 3) 5</pre> |
| <pre>uvitani() "Vítá Vás systém Maple"</pre> | <pre>moje_procedura(12, 3, -5) 31</pre> |

Obrázek 1.73: Vytváření procedur.

Obsah

| | | |
|----------|--|----------|
| 1 | Maple | 1 |
| 1.1 | Úvod | 1 |
| 1.1.1 | Standardní zápisník (Standard Worksheet) | 1 |
| 1.1.2 | Klasický zápisník (Classic Worksheet) | 3 |
| 1.1.3 | Příkazový řádek a kalkulačka Maple | 3 |
| 1.1.4 | Document mode, Worksheet mode | 3 |
| 1.1.5 | Math mode, Text mode | 5 |
| 1.2 | Základní ovládání | 5 |
| 1.2.1 | Vyhodnocení příkazů | 6 |
| 1.2.2 | Palety | 7 |
| 1.2.3 | Názvy symbolů | 7 |
| 1.2.4 | Pomocníci, instruktoři a řešené úlohy | 8 |
| 1.3 | Nápověda | 8 |
| 1.3.1 | Maple Help | 8 |
| 1.3.2 | Tour of Maple | 9 |
| 1.3.3 | Quick Reference | 9 |
| 1.3.4 | Quick Help | 10 |
| 1.3.5 | What's New | 10 |
| 1.3.6 | Startup Dialog | 10 |
| 1.3.7 | Manuals, Resources, and more | 10 |
| 1.4 | Provádění výpočtů | 12 |
| 1.4.1 | Příkazy | 14 |
| 1.4.2 | Označení výsledků | 15 |
| 1.4.3 | Přiřazení hodnot do proměnných | 16 |
| 1.4.4 | Řešení rovnic | 18 |
| 1.5 | Balíky | 20 |
| 1.5.1 | Jednotky | 21 |
| 1.5.2 | Vědecké konstanty | 22 |
| 1.5.3 | Reálná doména a předpoklady (omezení) | 24 |
| 1.6 | Základní matematické výpočty | 25 |
| 1.6.1 | Úpravy výrazů | 25 |
| 1.6.2 | Funkce | 25 |
| 1.6.3 | Limity, derivace a integrály | 29 |
| 1.6.4 | Lineární algebra | 31 |
| 1.7 | Kreslení a animace | 32 |

| | | |
|-------|--------------------------------------|----|
| 1.8 | Práce s neurčitostmi | 40 |
| 1.8.1 | Intervalová aritmetika | 40 |
| 1.8.2 | Scientific Error Analysis | 40 |
| 1.8.3 | Fuzzy Sets Toolbox | 41 |
| 1.8.4 | Statistics | 44 |
| 1.9 | Základy programování | 45 |
| 1.9.1 | Podmíněný příkaz <i>if</i> | 45 |
| 1.9.2 | Cyklus <i>for</i> | 46 |
| 1.9.3 | Cyklus <i>while</i> | 46 |
| 1.9.4 | Iterativní příkazy | 47 |
| 1.9.5 | Procedury | 48 |