

Výrazy a příkazy v Maximě, jejich vkládání a vyhodnocování, vlastnosti, vztahy mezi částmi výpočtu a manipulace s nimi

Cíl této kapitoly

Už jsme napsali, že Maxima je program natolik komplexní, že snažit se podat úplný návod k jeho použití, je pro účely tohoto textu nemožné. Proto od tohoto textu očekávejte jen základní informace, které by mohly pomoci v začátcích práce s programem.

Poznámky se vztahují k chování programu a jeho prostředí „tak, jak bývá nastaven po instalaci“. Může se tedy stát, že někdy bude jeho reakce trochu jiná.

Typy objektů programu Maxima

Jak jsme uvedli v úvodní kapitole textu, **Maxima** je počítačový algebraický systém (tyto programy označujeme CAS – z angličtiny: Computer Algebra System) a její objekty odpovídají tomuto účelu, jsou to tedy číselné, logické a textové proměnné a konstanty a struktury – seznamy, pole a struktury pojmenovaných položek. S nimi uživatel pracuje pomocí *příkazů Maximy*.

Typy objektů v pracovní ploše (listu) wxMaxima

Objekty, se kterými pracujeme v prostředí **wxMaxima**, zahrnují ještě objekty a prostředí nadstavbové, „obslužné“. Při tvorbě didaktických materiálů jistě oceníme možnost vytvářet rovnou materiály (pracovní listy či html stránky), které budou mít vlastní strukturu, budou obsahovat doplňující a vysvětlující texty, grafy, případně animace, a navíc budou mít zřetelnou strukturu. To nám umožní právě prostředí **wxMaxima**. Jednotlivé vložené prvky se v prostředí programu objeví v navazujících pozicích na stránce; tyto pozice, vyhrazená místa, se nazývají „*cells*“, tedy v překladu „buňky“. Do listu tak můžeme vkládat – kromě vstupních výrazů, příkazů a jejich upravených zápisů či vyhodnocených výsledků (*Input Cell*) – doplňující texty, nadpisy, kapitoly a podkapitoly, obrázky, a můžeme zalamovat stránky.

Vkládání výrazů (vstupních polí)

Bez ohledu na to, zda při práci s programem wxMaxima využíváme spíše textový zápis, nebo zda využíváme nástroje usnadňující vkládání výrazů a příkazů, je nezbytné, abychom uměli správně zapsat (vložit) matematický výraz.

Vložení výrazu

V předchozí kapitole věnované prostředí programu jsme napsali, že pro vložení nějakého vstupu stačí začít psát z klávesnice.

Operátory

Pro zápis výrazů platí pravidla, která jsou pro vkládání matematických výrazů asi dosti známá:

Desetinná čísla zapisujeme s desetinnou tečkou.

Důsledně závorkujeme (výhradně pomocí navzájem vnořených (kulatých) závorek) a jako aritmetické operátory používáme znaky +, −, * násobení, / dělení, ^ umocnění (a^b , též **, tj. $a^{**}b$). Nesmíme tedy například napsat $2x + 1$, ale důsledně jedině $2 * x + 1$ (viz obrázek 2).

Další operátory:

Relační (porovnání): <, <=, >, >= a (ne)rovnosti =, # (nerovná se)

Logické: not, or, and

K dispozici jsou i další operátory a navíc má uživatel možnost definovat operátory vlastní. Tuto možnost ale nejspíš nevyužijeme.

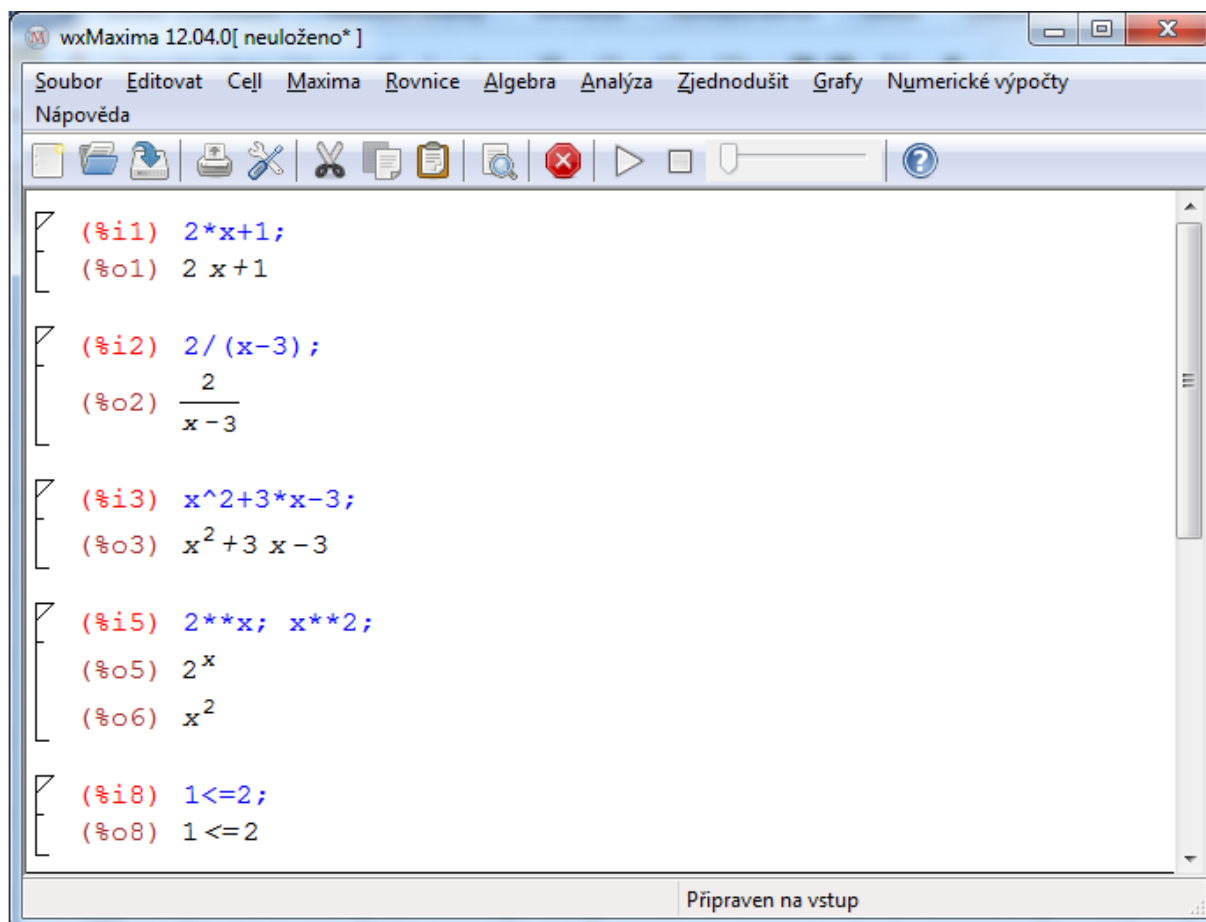
Vstupy a výstupy

Při vkládání výrazu se objeví na začátku řádky (vstupní buňky) symbol \longrightarrow , který se po vyhodnocení změní na označení číslem. Je to pořadové číslo vstupu a program ho generuje a vkládá automaticky.

Zároveň vzniká proměnná s tímto jménem, na niž se můžeme později odvolat.

Označení vstupu má vždy tvar: dvojznak *%i* následovaný pořadovým číslem (např. první vstup: *%i1*).

Podobně každý výstup je označen %O a pořadové číslo (např. čtvrtý výstup: %O4). Výstup (viz dále) je důsledkem vyhodnocení. Značení ukazuje obrázek 1.



Obr. 1

Vyhodnocení vstupu

Po zadání vstupního výrazu nebo příkazu stiskneme **Ctrl+Enter** (pokud není nastaveno jinak). Pokud není ve výrazu chyba (viz obrázek 2), systém vygeneruje a zapamatuje si nový, další výstup.

```
(%i9) 2x;
incorrect syntax: X is not an infix operator
```

Obr. 2

Proměnné

Výše uvedené značení znamená „vstupní proměnná...“ a „výstupní proměnná...“. Jde o systémové proměnné (případně konstanty), které se uvozují znakem % (následovaným *i* pro input – vstup, *O* pro output – výstup a číslem).

Další (vybrané) systémové proměnné (konstanty):

%e Eulerova konstanta (základ přirozených logaritmů e)

%i Komplexní jednotka (i)

%pi Ludolfovo číslo (π)

V ukázce na obrázku 1 jsme nejprve zrušili dříve zadaný čtvrtý výraz (vstup), takže další vstup má již číslo 5. V něm jsme zadali k vyhodnocení dva výrazy, vznikly tudíž dva výstupy (s čísly 5 a 6), další vstup (měl číslo 7, aby nevznikl posun v číslování vstupů a výstupů) jsme opět smazali a další vstup i výstup mají číslo 8.

Další (uživatelské) proměnné označujeme názvem, který už znakem % nezačíná.

Zápis funkcí při jejich vkládání do vstupního pole

Standardní matematické funkce vkládáme obvyklým způsobem: za název funkce uvedeme do kulatých závorek její argument (případně argumenty). Kromě obvyklých matematických a goniometrických funkcí je k dispozici pochopitelně mnoho funkcí dalších, ve školské matematice využijeme pouze nemnoho z nich. Některé jsou dostupné až po vyvolání příslušného balíčku funkcí.

Standardně jsou dostupné například funkce *abs* (absolutní hodnota), *sqrt* (odmocnina), *round*, *floor*, *ceiling* (vše zaokrouhlovací funkce), *min*, *max*, *signum*, *mod* (zbytek po dělení – funkce má dva argumenty), nebo funkce z teorie čísel (*next_prime*, *prev_prime*, ...), goniometrické funkce (*sin*, *cos*, *tan*, *cot* – sinus, kosinus, tangens, kotangens, ...) a mnoho dalších.

Před prvním voláním některé z dalších funkcí – například kombinatorické funkce *combination* (*combination*(*n*, *k*) – kombinační číslo) či *permutation* (*permutation*(*n*, *r*), což je ve skutečnosti počet *r*-členných variací z *n* prvků) – je třeba nejprve volat příkaz *load(funcs)*. K takovým funkcím dále patří třeba funkce *arithmetic* (*a*, *d*, *n*), jejímž výstupem je *n*-tý člen aritmetické posloupnosti. Mnohé funkce si ukážeme dále v příkladech. Zápis několika funkcí a výsledky vyhodnocení výrazů, které je obsahují, ilustruje obrázek 3.

```

wxMaxima 12.04.0[ neuloženo* ]
Soubor Editovat Cell Maxima Rovnice Algebra Analýza Zjednodušit Grafy Numerické výpočty Nápověda

[ (%i22) abs(x+2);
  (%o22) |x+2|

[ (%i23) abs(2-%i);
  (%o23) √5

[ (%i24) sqrt(1/x);
  (%o24) 1/√x

[ (%i25) sin(%pi/4);
  (%o25) 1/√2

[ (%i26) round(2.6);
  (%o26) 3

[ (%i27) random(1000);
  (%o27) 734

[ (%i28) log(%e); log(1/x);
  (%o28) 1
  (%o29) -log(x)

[ (%i30) load(funcs);
  (%o30) C:/PROGRA~1/MAXIMA~1.0-2/share/maxima/5.28.0-2/share/simplification/funcs.mac

[ (%i31) combination(5, 2);
  (%o31) 10

[ (%i32) arithmetic(1, 3, 5);
  (%o32) 13

Připraven na vstup

```

Obr. 3

Vkládání příkazů

Uživatelé, kteří jsou dobře seznámeni s (velmi širokou) škálou příkazů programu (a s jejich parametry), budou dávat přednost přímému zápisu příkazů, podobně, jako zapisují příkazy programovacího jazyka.

My budeme využívat prostředí programu usnadňující tvorbu dokumentu a přinejmenším v začátku budeme příkazy vkládat pomocí připravených dialogů a položek uspořádaných do tematických skupin *Hlavního menu* a ve formě tlačítek v *panelech nástrojů*.

Vložení příkazu prostým zapsáním do vstupní buňky (vstupního pole)

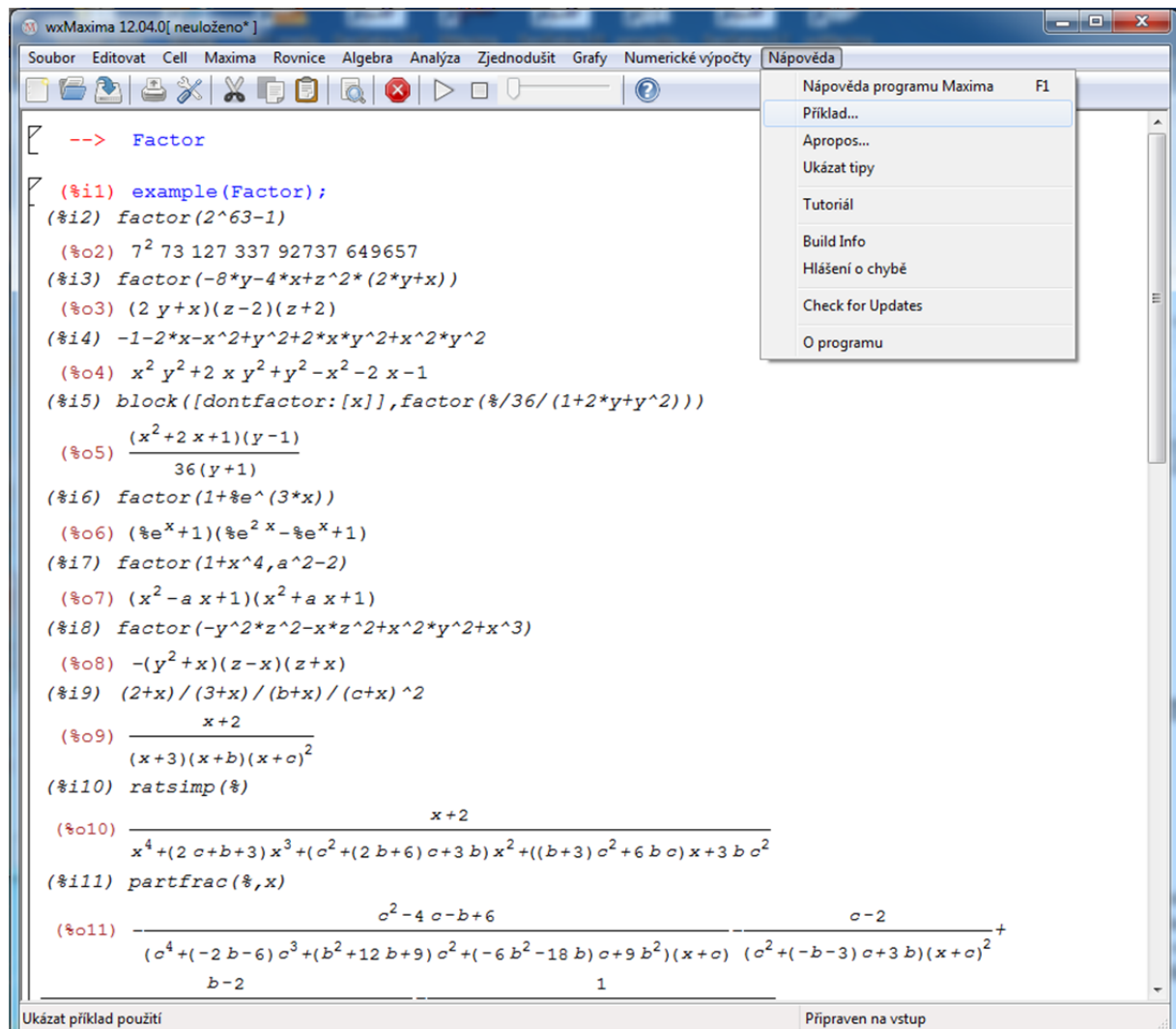
Vložení standardního příkazu nevyžaduje nic více ani méně, než znalost potřebného příkazu a toho, jaké má příkaz parametry. To může uživatel zjistit v *Nápovědě* (v rejstříku nebo v obsahu).

Na názvy příkazů (stejně jako na názvy funkcí) nemá vliv zvolený jazyk prostředí programu, jsou to tedy vždy slova či zkratky, které vycházejí z (anglické) matematické terminologie.

Pokud víme, jak se příkaz jmenuje, ale nejsme si jisti jeho použitím, můžeme si nechat zobrazit příklady použití příkazu. Část stránky s příklady, které systém nabídl po zadání názvu *Factor* a volbě *Příklad...* z podmenu *Nápověda* (v *Hlavním menu programu*), ukazuje obrázek 4. Mezery v řadě čísel výstupu 2 (%o2) znamenají součin – výraz je prvočíselným rozkladem uvedeného čísla. Některé typy rozkladů v uvedeném příkladu přesahují učivo základní a střední školy.

V nápovědě to nevadilo, ale jinak je v názvech příkazů program citlivý na malá/velká písmena.

Všimněte si, jak je interpretováno postupné dělení ve vstupu 9 – operace stejné priority se provádějí po řadě zleva doprava.



Obr. 4

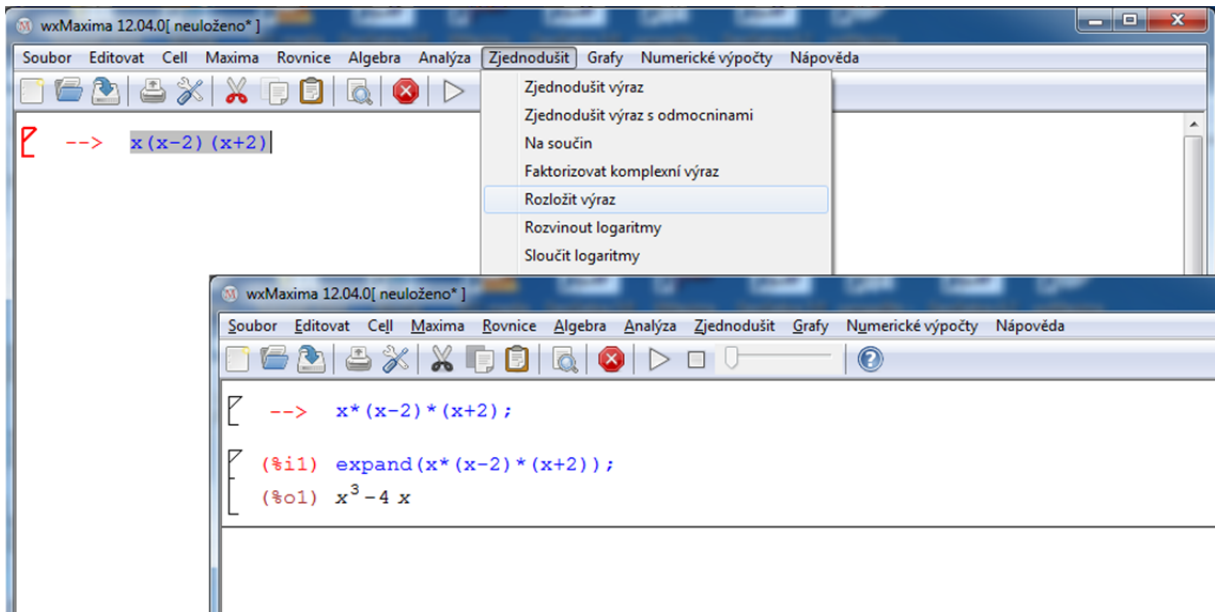
Nyní můžeme zkusit zapsat třeba $factor(1024)$;
Výsledek nás nejspíš nepřekvapí: 2^{10} .

Vložení příkazu pomocí menu

Chceme-li roznásobit výraz $x*(x-2)*(x+2)$, musíme výraz nejprve zapsat do vstupního pole a vzápětí (aniž bychom vstup vyhodnocovali) označit (vybrat pomocí myši) a najít/zvolit potřebný příkaz.

V podmenu *Zjednodušit* (v *Hlavním menu*) najdeme příkaz *Rozložit výraz* (první snímek na obrázku 5).

Klikneme na něho. Tím příkaz vložíme. Výsledek ukazuje vložený snímek na obrázku 5.

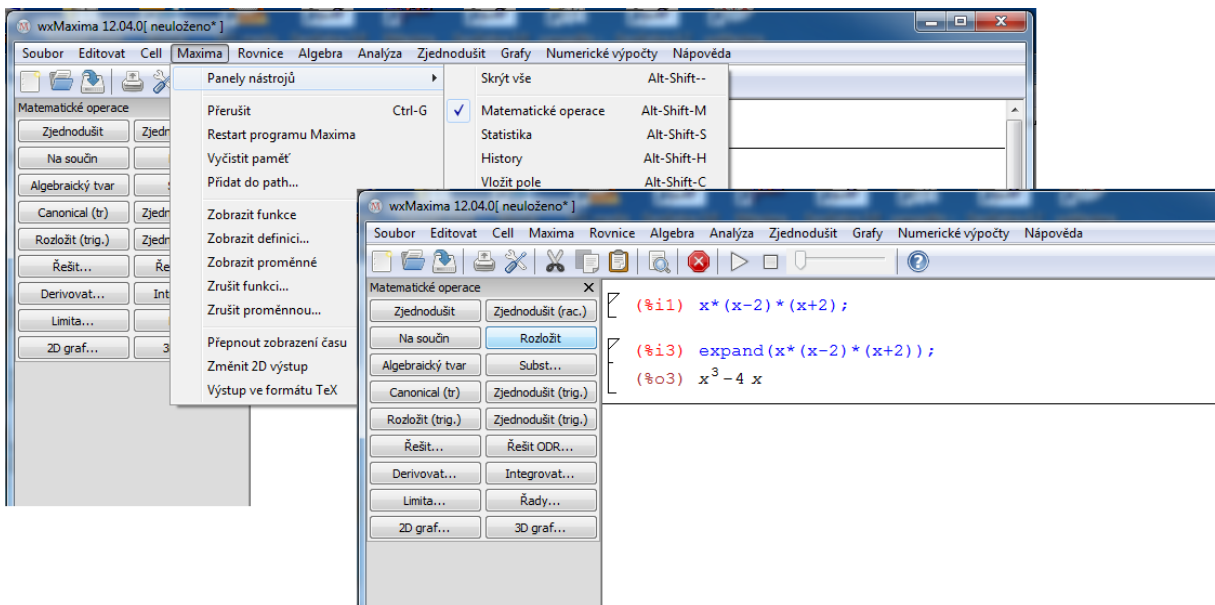


Obr. 5

Vložení příkazu pomocí panelů nástrojů

Zobrazíme si panel nástrojů *Matematické operace* (podmenu *Maxima*) – viz obrázek 6.

Zapišeme výraz $x*(x-2)*(x+2)$ do vstupního pole, označíme ho a v panelu *Matematické operace* klikneme na tlačítko s nápisem příkaz *Rozložit*. Výsledek ukazuje druhý snímek na obrázku 6.



Obr. 6

Poté, kdy několikrát zvolíme stejný příkaz, si už budeme jeho název nejspíš pamatovat a při další práci s programem zvolíme rychlejší cestu k získání požadovaného součinu nebo rozkladu – příkaz rovnou zapíšeme a jeho argument vložíme do kulatých závorek za jménem příkazu.

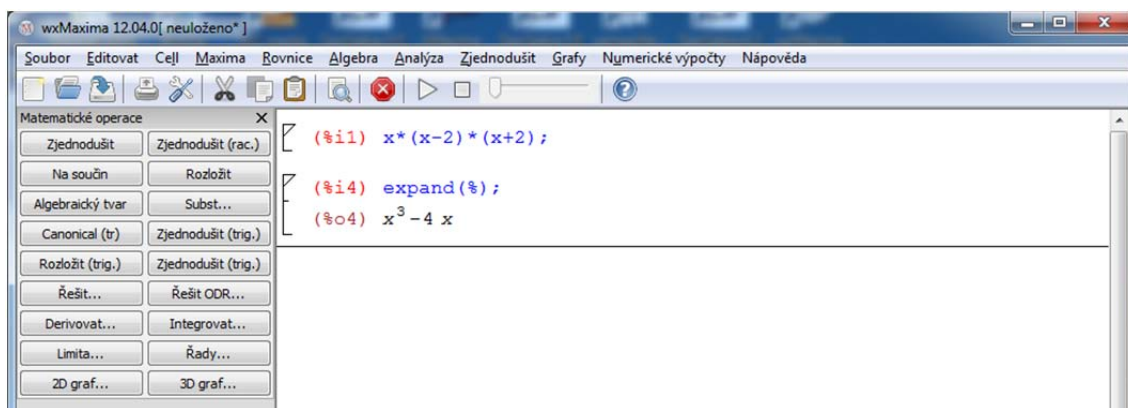
Využití předchozích vstupů či výstupů

Chceme-li jako vstup (či jeho část) použít hodnotu nebo výraz, který je obsažen v některé proměnné, odkážeme se na něho jeho jménem.

Příklad: vložíme příkaz $expand(x*(x-2)*(x+2))$; Je-li například označen jako %i2, bude výsledný roznásobený výraz označen %o2.

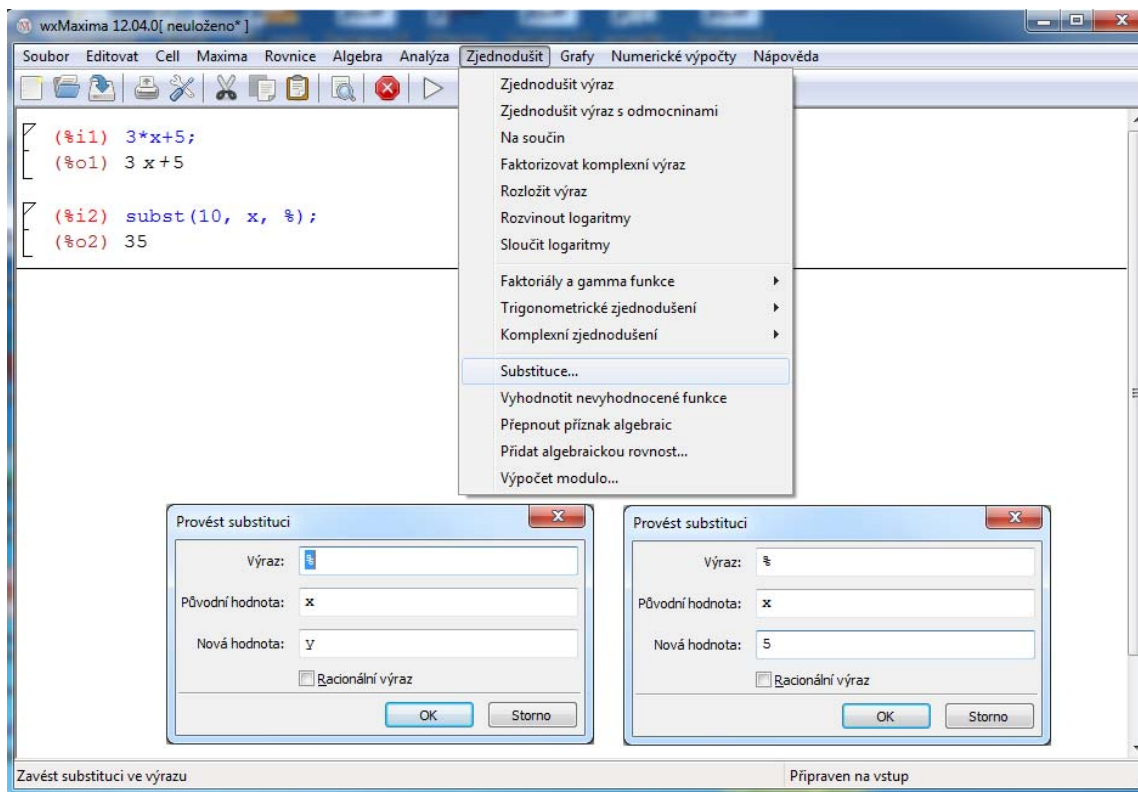
Následující příkaz $factor(%o2)$; vrátí jako výsledek původní výraz $(x-2)*x*(x+2)$.

Samotný znak % má význam “posledního výrazu” – viz obrázek 7.



Obr. 7

Hodnoty proměnných, substituce, vyhodnocení



Obr. 8

Substituce

Chceme-li provést substituci ve výrazu, použijeme příkaz *subst*. Buď jej zapíšeme přímo do vstupní řádky, nebo, častěji, vložíme z panelu nástrojů (tlačítko *Subst...*) či z menu. Místo čtení podrobného návodu se podívejte na postup a jeho výsledek na obrázek 8. Za proměnnou x jsme dosadili hodnotu 5. Znak $\%$ v dialogovém okně říká, že se bude dosazovat do naposledy zpracovávaného výrazu.

Přiřazení hodnoty proměnné

Chceme-li dosadit do nějaké proměnné hodnotu, uděláme to nejsnáze znakem $:$ (dvojtečka). Do obrázku 9 se podívejte, jak pak probíhá vyhodnocení výrazu s takovou proměnnou. Chceme-li zabránit vyčíslení hodnoty výrazu (či jeho části), uvedeme před výraz znak $'$ (apostrof).

```

wxMaxima 12.04.0[ neuloženo* ]
Soubor Editovat Cell Maxima Rovnice Algebra Analýza Zjednodušit Grafy Numerické výpočty Nápověda

[ (%i1) 3*x+5;
  (%o1) 3 x + 5

[ (%i2) subst(10, x, %);
  (%o2) 35

[ (%i3) a: 10;
  (%o3) 10

[ (%i4) a^2-3;
  (%o4) 97

[ (%i5) 'a^2-3;
  (%o5) a2-3

Připraven na vstup

```

Obr. 9

Poznámka 1: Pokud budeme definovat vlastní funkce, budeme předpis přiřazovat názvu funkce dvojznakem $:=$. Pro různá přiřazení jsou v Maximě ještě další operátory, ty zde nekomentujeme.

Poznámka 2: Pokud ukončíme příkaz znakem dolaru, pak se sice provede, ale nevytiskne se jeho výstup (viz obrázek 10).

```

wxMaxima 12.04.0[ neuloženo* ]
Soubor Editovat Cell Maxima Rovnice Algebra Analýza Zjednodušit Grafy Numerické výpočty Nápověda

[ (%i1) a:10$ b:5$
  '(a+b)^2;
  (a+b)^2;

[ (%o3) (b+a)2
  (%o4) 225

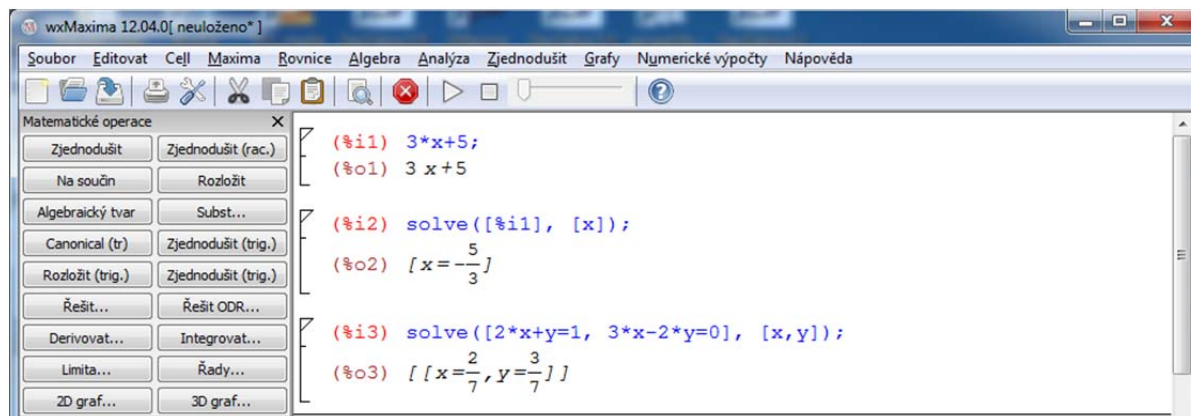
Vítejte v programu wxMaxima Připraven na vstup

```

Obr. 10

Dialogy

Dialogová okna poskytují prostředí pro zadávání složitějších příkazů (tím myslíme takových příkazů, které mají více parametrů, které je nutno zadat). I takové příkazy můžeme, pokud to umíme, rovnou zapisovat do vstupního pole, použití dialogu však bývá podstatně pohodlnější a zabrání chybám. Jako ukázkou jsme zvolili dialog pro řešení rovnic a jejich soustav. Na obrázku 11 vidíme jednu vyřešenou lineární rovnici a jednu soustavu dvou lineárních rovnic.



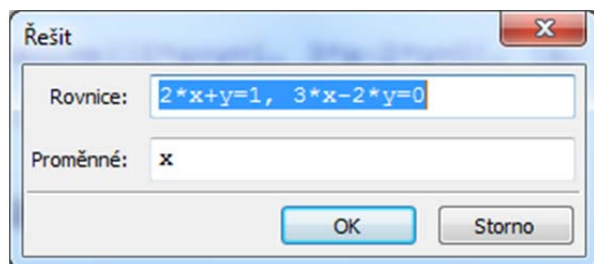
Obr. 11

Prvou rovnici jsme nejprve zapsali do vstupu (ne jako rovnici, ale jen výraz, který bude dále považován za levou stranu anulované rovnice), soustavu jsme zadávali rovnou pomocí dialogu. Dialogy, které jsme použili, jsme vyvolali tlačítkem *Řešit...* z panelu nástrojů *Matematické operace*. Oba (již vyplněné) dialogy ukazuje obrázek 12.



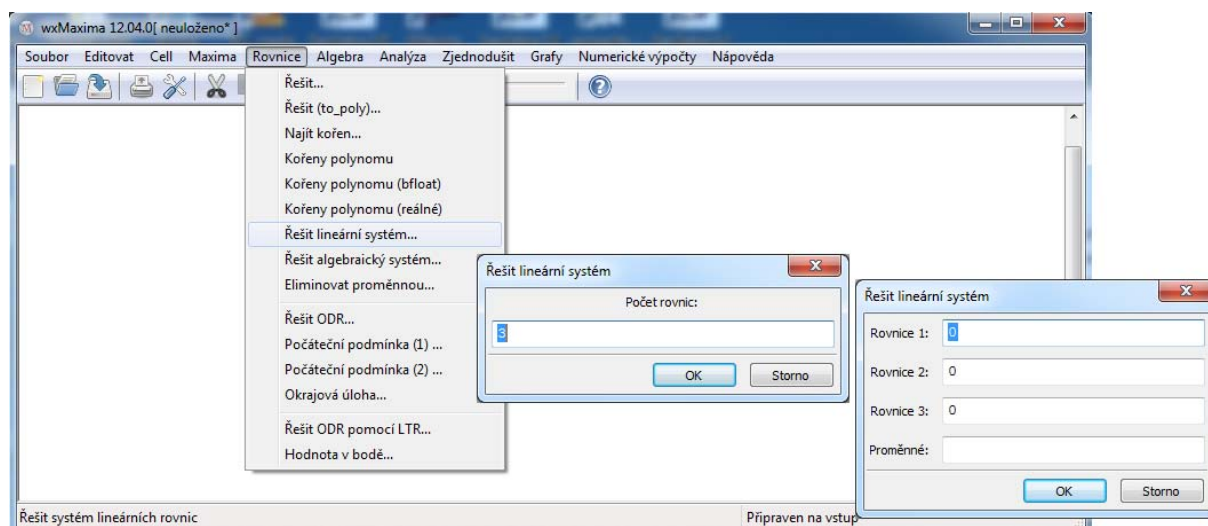
Obr. 12a, b

Pokud bychom zapsali soustavu do vstupní řádky rovnou, pak by řádku „obsloužil“ i výchozí dialog, do kterého bychom již nic nevpisovali. Pokud výraz vybereme, objeví se po volbě příkazu *Řešit...* dialog již částečně vyplněný (obrázek 12c).



Obr. 12c

Pro řešení soustav lineárních rovnic je výhodné použít dialog *Řešit lineární systém* z podmenu *Rovnice* (v *Hlavním menu*), který nás zadáním provede podrobněji (viz kombinovaný obrázek 13). Do jeho polí můžeme – stejně jako v ukázkách uvedených výše – vkládat (zapisovat) jak přímo jednotlivé rovnice, tak i odkazy na rovnice uložené ve vstupních či výstupních proměnných.

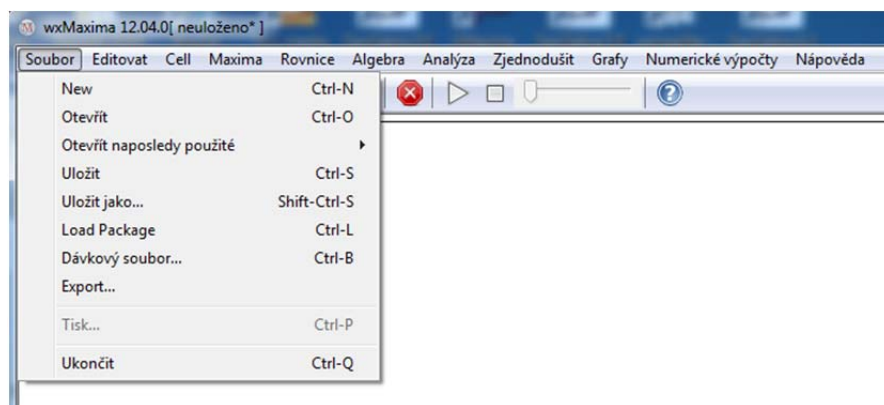


Obr. 13

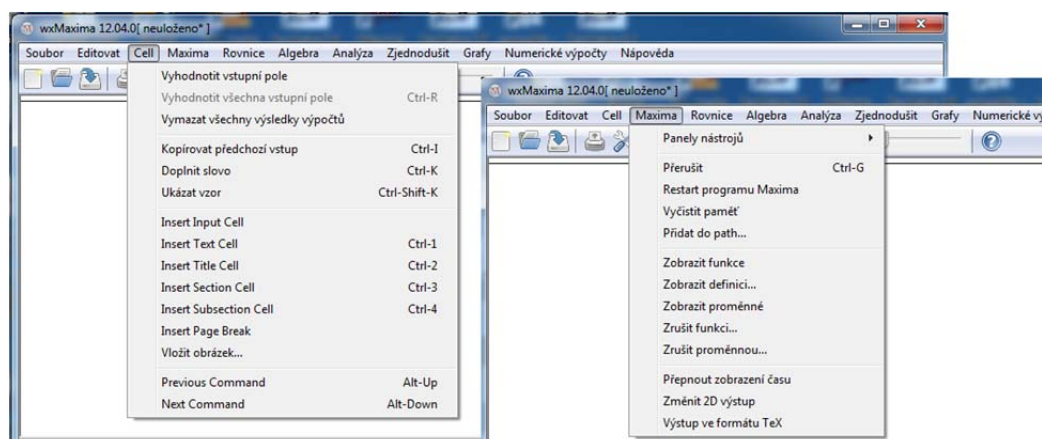
K dalším často používaným dialogům patří dialogy pro tvorbu grafů. Ty si podrobně ukážeme v jednom ze tří příkladů poslední kapitoly.

Klávesové zkratky a užitečné poznámky

Pro často používané nástroje jsou k dispozici klávesové zkratky. Kromě obvyklých zkratk používaných ve Windows pro práci se soubory (obrázek 14), jsou to zejména zkratky pro akce, které jsou v prostředí wxMaxima k dispozici pro práci se zobrazením pracovního listu, pro práci s jednotlivými částmi listu, s buňkami a akce ovlivňující operace prováděné programem Maxima (obrázek 15).



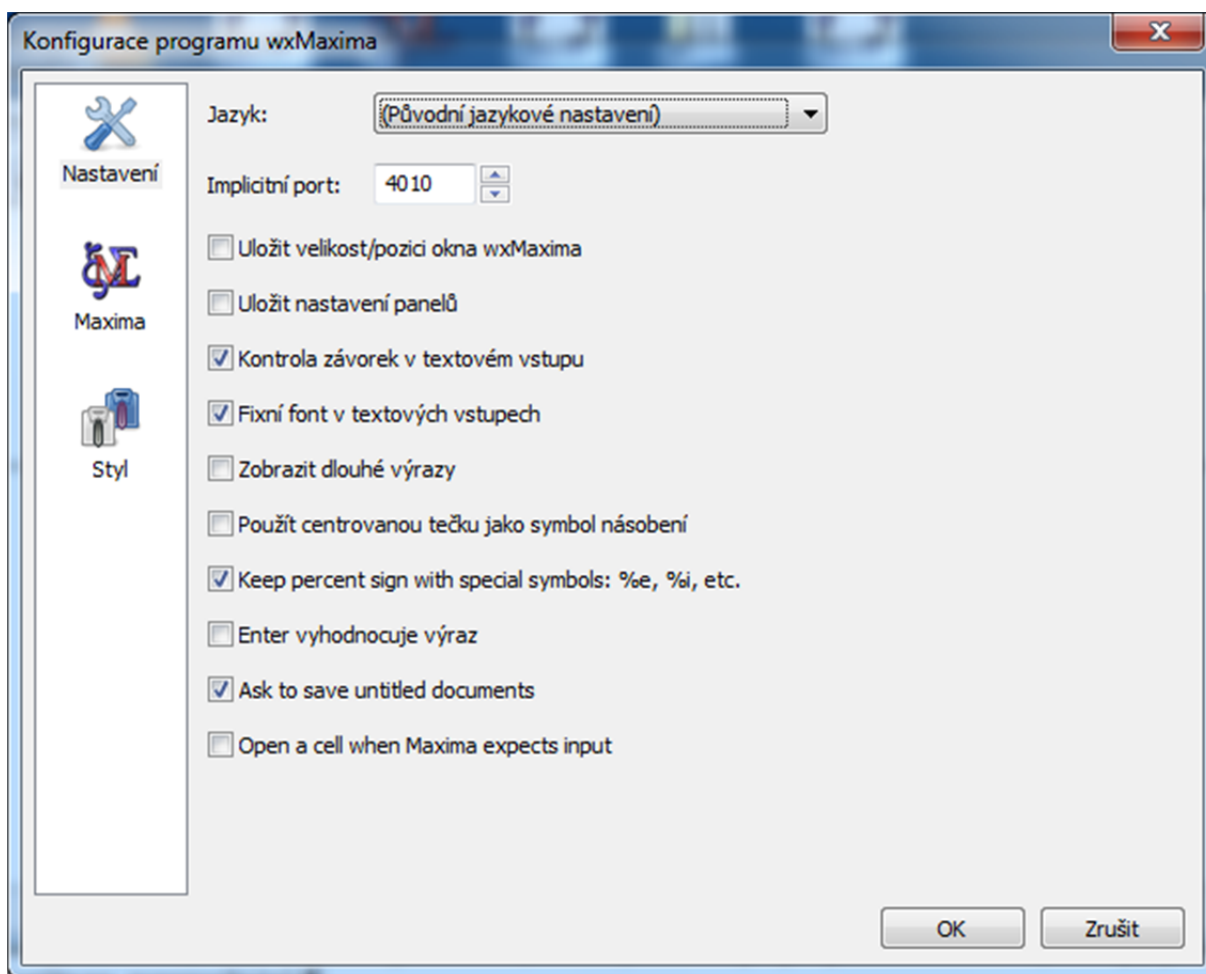
Obr. 14



Obr. 15

K akcím, které budeme pro výpočetní pole na listu prostředí nejčastěji používat, určitě patří:
Vymazat všechny výsledky výpočtů a následně
Vyhodnotit všechna vstupní pole (**Ctrl + R**),
vlození jednotlivých částí listu (*Insert ... Cell*)
a pro rychlou editaci vkládaného obsahu vkládání předchozího a následujícího vstupu.
Pro běh výpočtu a samotného programu Maxima pak v první řadě *přerušeni* dlouho trvajících výpočtů (**Ctrl + G**) a *vyčištění paměti* (to ve výsledku po přepočtení všechny vstupy přečísluje – neobávejte se, nesmaže je).

Pokud se program nechová tak, jak je v tomto textu popsáno, zkuste se podívat do dialogu *Nastavení* (podmenu *Editovat* v *Hlavním menu* programu). Obrázek 16 ukazuje, že mnohé může být jinak, než jsme zde napsali.



Obr. 16

Závěrem textu

Pokusili jsme se komentovat základní rysy programu Maxima a principy práce a s prostředím wxMaxima. Při práci s ním jistě sami najdete spoustu užitečných postupů, které Vám budou vyhovovat. V následující (poslední) kapitole najdete tři příklady a k nim komentované soubory, příklady vytvořených didaktických materiálů a postupů, které jsme v nich použili.