



O PROGRAMU MAXIMA

ON PROGRAMME MAXIMA

Lukáš Honzík

Abstract: Since teachers get used to use computers in their math classes, it becomes almost necessary for them to handle some computer programmes that are useful in teaching process. There are quite a lot of dynamic geometry programmes to choose from but much less computer algebra systems (CAS) that can be used for free. Let us introduce one of the free programmes of computer algebra – Maxima which can be very helpful although it cannot match with any commercial CAS.

Key words: Maxima, wxMaxima, computer algebra, CAS, equation solving

Úvod

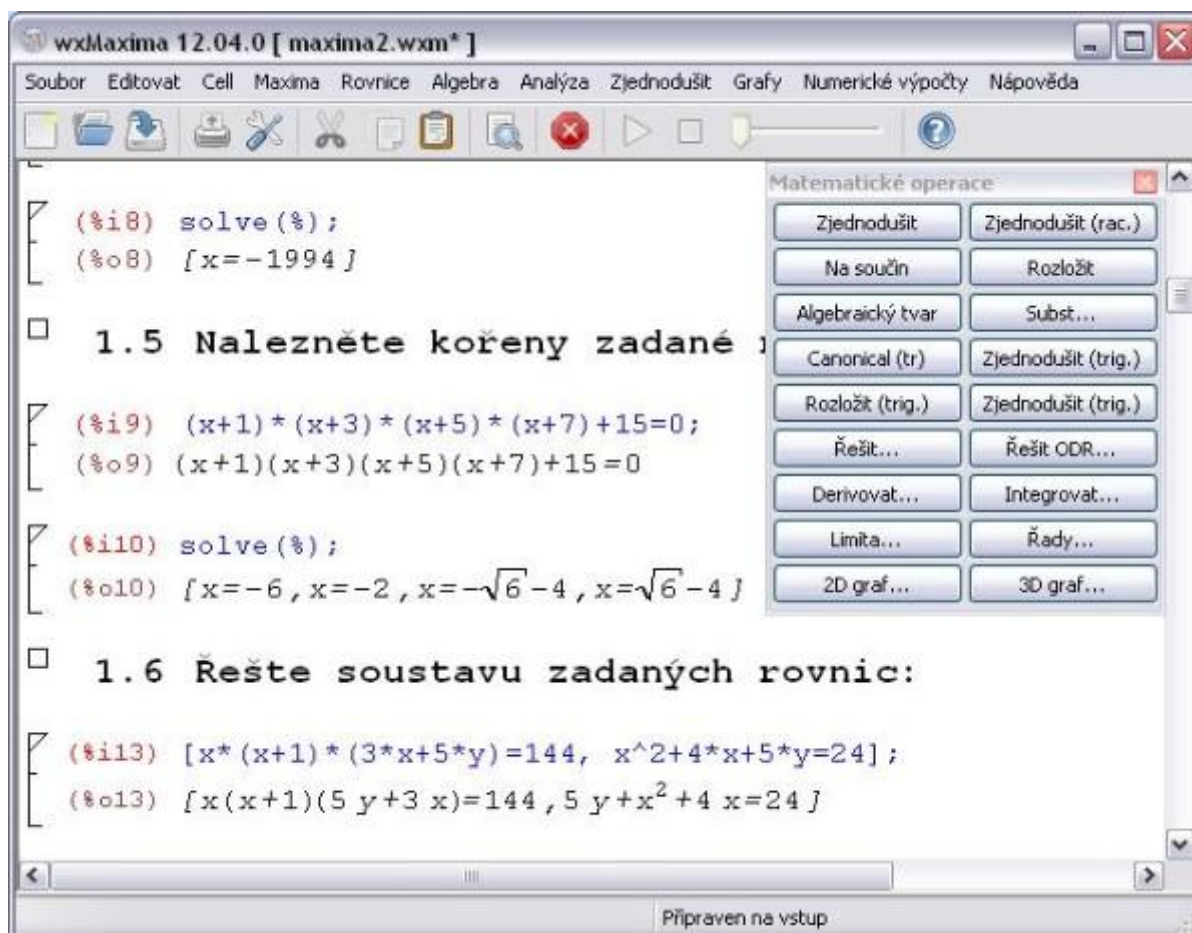
V súčasnej dobe se učiteľ matematiky pri práci již téměř neobejde bez počítače, což platí jak v práci administratívnej, tak v časti přímo spjaté s výukou. V prípade výuky (dynamické) geometrie je zde poměrně velká nabídka programů dostupných zadarmo (připomeňme například programy GeoGebra, GEONExT nebo starší verze programu Cinderella), naproti tomu u programů počítačové algebry je situace o něco složitější. Licence pro programy Mathematica a Maple jsou i pro školy poměrně drahou záležitostí, program Derive je také stále zpoplatněn, přestože jeho další vývoj byl již ukončen...

Málokterý učitel matematiky však zná program Maxima, jenž je spolu s grafickým uživatelským rozhraním wxMaxima vcelku rozumnou náhradou právě za komerční programy. A to i přesto, že jeho vývoj sahá poměrně daleko do minulosti, konkrétně do 70. let 20. století. Původně čistě textově ovládaný (tak jak to znají pamětníci z práce s operačním systémem MS DOS či zarytí odpůrci grafických rozhraní systému Linux) program prošel v průběhu času nejrůznějšími inovacemi a vylepšeními, a proto je možné jej úspěšně využívat i dnes, a to i navzdory tomu, že přeci jen nedosáhne kvalit svých komerčních konkurentů.

Program i s grafickým uživatelským rozhraním je ke stažení na stránkách vývojářského týmu <http://andrejv.github.com/wxmaxima/>, a to jak ve verzi pro systémy Windows, tak pro Linux.

Prostředí wxMaxima vypadá na první pohled poněkud stroze, avšak to není na závadu využitelnosti. Ve většině práce si uživatel vystačí se samotným oknem aplikace, kam jsou zapisovány jednotlivé příkazy, v případě potřeby je pak možné všechny příkazy najít v hlavní nabídce, či v panelech nástrojů a odtud je i spustit, přičemž jejich argumenty jsou zadány prostřednictvím dialogového okna.

O programu Maxima



Obr. 1: Prostředí grafického rozhraní wxMaxima

1 Základy práce s programem

V programu jdou přirozeně vyhodnocovat jednoduché i složitější matematické operace, lze provádět jednodušší symbolické výpočty (zjednodušovat zadané výrazy, vytýkat, násobit, ap.), řešit rovnice a jejich soustavy, nebo třeba nechat vykreslit dvourozměrné a trojrozměrné grafy.

Podívejme se na práci s programem z praktické stránky. Chceme například vyhodnotit výraz $2 \cdot 9^2 - 5^{-\frac{2}{3}}$. Napíšeme tedy do okna programu příkaz ve tvaru `2*9^2-5^(-2/3)` a ukončíme jej středníkem (oficiálně je tímto způsobem ukončován příkaz, avšak pokud jej uživatel nenapíše, program jej pak sám doplní), potvrzení příkazu provedeme stiskem klávesy Enter na numerické klávesnici, případně kombinací kláves Shift + Enter. Program pak vypíše výsledek výpočtu.

Jak již bylo zmíněno výše, program ale zvládá i složitější výkony než každý normální kapesní kalkulátor. Umí se vypořádat s jednoduššími symbolickými výpočty, takže mu nečiní problémy například rozklady mnohočlenů s rozumnými kořeny („rozumným kořenem“ je v tomto případě myšlen kořen celočíselný), určení největšího společného dělitele, derivování a integrování a podobně, taktéž mu není cizí práce s funkcemi a základní operace s komplexními čísly.

2 Užitečné poznámky k práci

Právě ve spojitosti s komplexními čísly je dobré se zmínit o některých zvláštnostech (nebo spíše věcech, které je dobré mít na paměti) programu. Například imaginární jednotka i musí být zadána jako **%i**, kdy znak **%** slouží jako jakýsi identifikátor. Stejně tak je tomu při zápisu Ludolfova čísla **%pi** a Eulerova čísla **%e**. Bez znaku **%** by všechny takto zapsané výrazy byly programem přečteny jako normální proměnné i , pi a e .

Toto ale není jediné využití znaku **%** v práci, samostatně zapsaný znak **%** odkazuje na hodnotu či výraz aktuálně načtený v paměti počítače. Řeší-li tedy uživatel například složitější úlohu s větším množstvím mezikroků, je možné se na právě spočítané výsledky v dalším počítání odkázat právě tímto způsobem. Zároveň se lze obdobně odkázat na libovolný výpočet provedený v aktuálně otevřeném souboru, a sice prostřednictvím příslušných „pojmenování“, která se objevují na začátku řádky u každého vstupu i výstupu (ve tvaru např. **%i5** nebo **%o21**).

3 Několik řešených příkladů

Podívejme se na několik příkladů a jejich řešení, začněme s rovnicemi a jejich soustavami (nerovnice nechme prozatím stranou, řeč o nich bude později). V tomto případě si vystačíme s příkazem **solve ()**, a to jak při práci s rovnicemi, tak s jejich soustavami. Je však třeba mít na paměti, že pro obě zmíněné situace může mít daný příkaz různé množství argumentů. Zadáme-li soustavu rovnic, jejichž počet odpovídá počtu neznámých, pak lze použít příkaz ve tvaru **solve (%)**, kde **%** odkazuje na soustavu načtenou v paměti. Na druhou stranu je-li neznámých více než zadaných rovnic, je třeba specifikovat, pro které neznámé chceme soustavu řešit.

Tedy řešíme-li například rovnici $\frac{x+11}{1983} + \frac{x+9}{1985} + \frac{x+7}{1987} = \frac{x+1991}{3} + \frac{x+1989}{5} + \frac{x+1987}{7}$, či soustavu dvou rovnic o dvou neznámých $xy^2 + x^2y = 4$, $xy + x + y = 4$, stačí použít příkaz **solve (%)** a dostaneme správné výsledky $x = -1994$, resp. $[x, y] \in \{[1 + i, 1 - i], [1 - i, 1 + i]\}$.

Naproti tomu pokud zadáme rovnici $ax^2 + bx + c = 0$ a použijeme příkaz **solve (%)**, obdržíme chybové hlášení, protože program není schopen v zadaném zápisu schopen zjistit, co má brát jako neznámou a co naopak jako parametr. V takovém případě je nutné zapsat příkaz k řešení rovnice či soustavy jako **solve (% , x)**, kde argument **x** určuje, že x je neznámou. Tentokrát je výstup

řešení ve správném tvaru $x = \frac{b \pm \sqrt{b^2 - 4ac}}{2a}$.

Výše zjištěná skutečnost tak upozorňuje na další dovednost, kterou Maxima nabízí, a tou je řešení některých matematických problémů s parametry. Zde je ale třeba se mít na pozoru, bohužel ne vždy je výsledek naprosto správně.

Vezměme kupříkladu poměrně jednoduchou rovnici $(a - 1)x^2 - (a - 2)x + 2a - 1 = 0$, kde x je neznámá. Když ji budeme řešit „lidským“ přístupem (a pomocí tužky a papíru), dojdeme nakonec k řešení, které nejlépe vyjádříme zápisem do tabulky:

O programu Maxima

a	řešení
$(-\infty, 0) \cup (\frac{8}{7}, \infty)$	nemá řešení
$\langle 0, 1 \rangle \cup \langle 1, \frac{8}{7} \rangle$	$\left\{ \frac{a-2 \pm \sqrt{a(8-7a)}}{2(a-1)} \right\}$
$\{1\}$	$\{-1\}$

Tab. 1: Tabulka výsledků v závislosti na parametru a

Program naproti tomu po načtení rovnice a použití příkazu **solve** (**%**, **x**) vypíše jediný výsledek, a sice $x = \frac{a-2 \pm \sqrt{a(8-7a)}}{2(a-1)}$, ale o tom, že by výsledek mohl nabývat jiné hodnoty nebo nemusel vůbec existovat, pomlčí.

4 Vyšetření průběhu funkce

Nakonec můžeme program zkusit využít i pro takovou věc, jakou je vyšetření průběhu funkce. Provést kompletní vyšetření tak, jak jsme zvyklí, sice nebude možné, pokusme se však využít všech dostupných nástrojů, abychom o zadané funkci zjistili co nejvíce. Vyzkoušejme to na funkci s předpisem $f(x) = x^3 - 6x^2 + 9x + 2$.

Definiční obor a obor hodnot nechme stranou, Maxima s nimi má problémy, ze zkušenosti však víme, že pro tuto funkci platí definiční obor $D = R$ a obor hodnot $H = R$. Jako první tedy pomocí programu určíme průsečíky s osami. Pro jejich nalezení stačí jednak nalézt funkční hodnotu pro $f(0) = 0^3 - 6 \cdot 0^2 + 9 \cdot 0 + 2$ a dále řešit rovnici $x^3 - 6x^2 + 9x + 2 = 0$. V první případě stačí pouze dosadit hodnotu 0, ve druhém načteme do paměti výraz $x^3 - 6x^2 + 9x + 2$ zápisem **x^3-6*x^2+9*x+2** a řešíme příslušnou rovnici příkazem **solve** (**%**). Výsledek je sice trochu krkolomný, program vypočte všechny kořeny včetně komplexních, nicméně nalezne i jeden reálný kořen ve tvaru $x = (\sqrt{3}-2)^{\frac{1}{3}} + \frac{1}{(\sqrt{3}-2)^{\frac{1}{3}}} + 2$. Jeho hodnotu můžeme vyjádřit v

jednodušším tvaru (po jeho načtení do paměti) příkazem **factor** (**%**) jako $x = \frac{\left((\sqrt{3}-2)^{\frac{1}{3}} + 1 \right)^2}{(\sqrt{3}-2)^{\frac{1}{3}}}$.

Dále se podíváme na limity v krajních bodech definičního oboru, v našem případě v $\pm\infty$. Po načtení předpisu funkce do paměti uijeme příkaz **limit** (**%**, **x**, **inf**) a **limit** (**%**, **x**, **minf**), kde **inf** a **minf** jsou výrazy pro ∞ a $-\infty$. Postupně tak dostaneme $\lim_{x \rightarrow \infty} f(x) = \infty$, $\lim_{x \rightarrow -\infty} f(x) = -\infty$.

Pro zjištění intervalů monotonie využijeme derivaci funkce, pro tu je zde příkaz **diff** (**%**, **x**, **1**), kde **x** je proměnná, podle které derivujeme, a **1** je řád hledané derivace. Výstup je $3x^2 - 12x + 9$. Nulové body této derivace dokážeme získat opět příkazem **solve** (**%**), dostaneme hodnoty $x = 3$ a $x = 1$. Tím ale bohužel v podstatě končíme, program Maxima ve své základní verzi s nerovnostmi

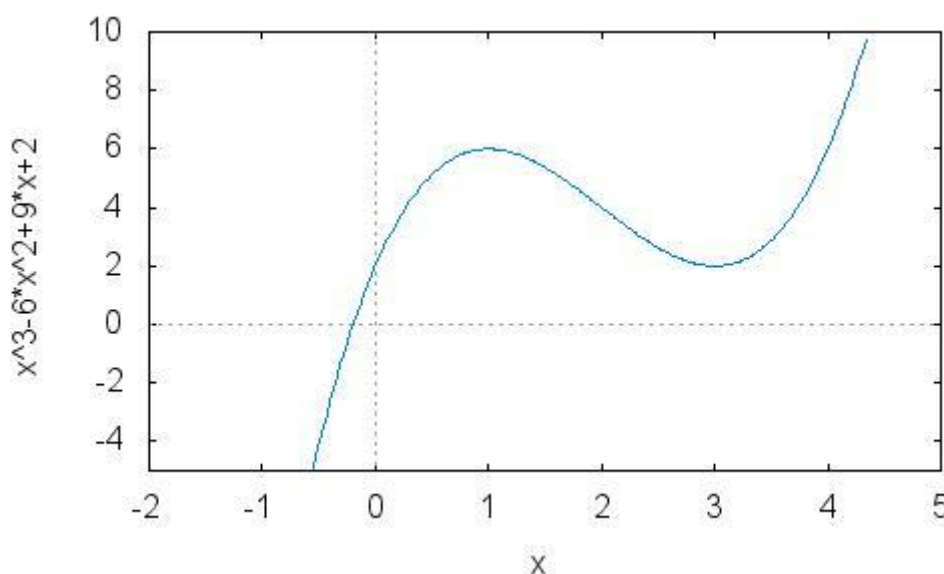
neumí pracovat. Pomoci nám však může například stažení balíčku `solve_rat_ineq` pomocí příkazu `load ("solve_rat_ineq")`. Po jeho načtení využijeme příkaz `solve_rat_ineq ()` a jako argument dosadíme výraz derivace, odpovědí bude dvojice nerovností $x < 1$, $x > 3$. Tedy v intervalech $(-\infty; 1)$ a $(3; \infty)$ je funkce $f(x)$ rostoucí, v intervalu $(1; 3)$ je klesající.

Podívejme se ještě na druhou derivaci funkce, resp. na intervaly, v nichž je funkce $f(x)$ konvexní a konkávní. Po načtení předpisu funkce $f(x)$ použijeme příkaz `diff (% , x, 2)`, přičemž dostaneme odpověď $6x - 12$, nulový bod (příkaz `solve (%)`) je jediný a sice $x = 2$. Užijeme-li opět příkaz `solve_rat_ineq ()`, zjistíme, že druhá derivace $f''(x)$ je záporná pro $x < 2$, tedy v intervalu $(-\infty; 2)$ je funkce $f(x)$ konkávní, v intervalu $(2; \infty)$ je konvexní.

Podívejme se nyní na možnosti nalezení extrémů funkce. K tomu poslouží příkaz `lbfgs ()`, který je však kvůli většímu počtu argumentů asi vhodnější zadávat pomocí hlavní nabídky (Analýza → Najít minimum...). Pomocí dialogového okna pak můžeme zadat výraz popisující zadanou funkci, proměnnou, počáteční odhad, kde minimum hledat, a hodnotu epsilon (minimum je totiž dopočítáváno numericky). Příkaz pro nalezení maxima neexistuje, ale bez problému se bez něj obejdeme, protože jej můžeme hledat také jako minimum funkce dané opačným výrazem k našemu výrazu $x^3 - 6x^2 + 9x + 2$. S hledáním a výpočtem nám pomůžou nulové body první derivace, které jsme určili již dříve a existuje pravděpodobnost, že budou hledanými extrémy. Proto po načtení výrazu $x^3 - 6x^2 + 9x + 2$ do paměti zadejme příkaz `lbfgs ()` s příslušnými argumenty nebo vyberme příkaz `Najít minimum...` z hlavní nabídky a v dialogovém okně kromě jiného vyplňme do kolonky počátečního odhadu hodnotu 2. Program pak ví, že má minimum hledat právě v okolí hodnoty $x = 2$, a odpoví výstupem $x = 3$.

Podobně dokážeme určit i maximum, které však hledáme jako minimum funkce $-f(x)$ s předpisem $-x^3 + 6x^2 - 9x - 2$. Hledáme jej v okolí bodu $x = 1$, můžeme tedy opět použít počáteční hodnotu 2. Výstupem bude tvrzení $x = 1$, jedná se tedy o maximum funkce $f(x)$.

Jako poslední přichází na řadu vykreslení grafu funkce, pro to použijeme příkaz `wxplot2d (% , [x, -2, 5], [y, -5, 10])`, kde v hranatých závorkách jsou uvedeny hodnoty omezující vykreslení grafu. Možná trochu jednodušeji lze graf vykreslit pomocí příkazu `2D graf...` v hlavní nabídce (Grafy → 2D graf...), zde je možné zadat omezení grafu prostřednictvím dialogového okna.



Obr. 2: Vykreslení grafu funkce $f(x) = x^3 - 6x^2 + 9x + 2$

O programu Maxima

Závěr

Program Maxima určitě nelze srovnávat s komerčními programy počítačové algebry typu Mathematica, bezpečně prohraje na celé čáře, i tak je ale na závěr nutné vyzdvihnout, že nabízí vcelku široké spektrum funkcí a nástrojů, které uživatel shledá užitečnými, a navíc je zadarmo téměř vždy po ruce, pokud se uživatel rozhodne pro jeho nainstalování.

Literatura

Maxima 5.28.0 Manual [online]. 2012 [cit. 2013-02-11]. Dostupné z:
<http://maxima.sourceforge.net/docs/manual/en/maxima.html>

Materiály seminářů předmětu KMT/ŘUZ2 Metody řešení matematických úloh 2, Katedra matematiky, fyziky a technické výchovy, FPE ZČU v Plzni.

Mgr. Lukáš Honzík
Katedra matematiky, fyziky a technické výchovy
Fakulta pedagogická Západočeské univerzity v Plzni
luky21@kmt.zcu.cz, <http://fpe.zcu.cz/kmt/kmt>