

PROGRAM WXMAXIMA VE VÝUCE MATEMATIKY

Roman Hašek, Michaela Noruláková

Jihočeská Univerzita v Českých Budějovicích

Abstrakt. Článek představuje vybrané příklady použití počítačového algebraického systému wxMaxima ve výuce matematiky spolu s možností jeho integrace s programem GeoGebra a s prostředím interaktivní tabule.

Klíčová slova: wxMaxima, systém počítačové algebry, GeoGebra, interaktivní tabule, matematika.

USE OF WXMAXIMA IN MATHEMATICS EDUCATION

Abstract. The paper presents selected examples of the use of computer algebra system wxMaxima in mathematics education. It also introduces ways of its integration into the interactive whiteboard environment and its application in association with GeoGebra software.

Klíčová slova: wxMaxima, computer algebra system, GeoGebra, interactive whiteboard, mathematics.

1 Úvod

Cílem článku je, prostřednictvím konkrétních příkladů, ukázat některé možnosti využití programu wxMaxima ve výuce matematiky. Vychází z workshopu *Úvod do programu wxMaxima* pořádaného se stejným cílem na 5. konferenci Užití počítačů ve výuce matematiky v listopadu 2011 v Českých Budějovicích.

Program wxMaxima je volně šiřitelným reprezentantem kategorie systémů počítačové algebry (dále nahrazujeme zkratkou CAS pocházející z anglického „computer algebra system“), který umožňuje realizovat symbolické a numerické výpočty a grafické znázornění 2D a 3D objektů.

Jak ukázala i skladba příspěvků na výše zmíněné konferenci, jedničkou mezi programy používanými pro podporu výuky matematiky se v současnosti stává program GeoGebra. Nutno říci, že právem. Článek se zaměřuje na situace ve školní matematice, v nichž je,

alespoň dle mínění autorů, vhodné použití CAS programu, konkrétně wxMaximy, přínosné. Napomůže studentovi v pochopení učiva, přispěje k rychlému překonání výpočetní náročnosti problému, dovoluje aplikovat symbolické řešení, poskytne trojrozměrný graf, poslouží k ověření správnosti výsledku apod. Pro úplné vyřešení některých problémů, většinou vycházejících z reálného světa, je pak optimální použití kombinace programu dynamické geometrie s programem počítačové algebry.

2 Program wxMaxima

Program wxMaxima je rozhraním pro práci s volně stažitelným open source CAS programem Maxima [3, 4]. Některé jeho vlastnosti, například částečná lokalizace do češtiny, systém nástrojů, které dovolují program použít i při neznalosti syntaxe vybraných příkazů, možnost vytváření dokumentů a jejich exportu do formátu „tex“ či „html“, a samozřejmě také nulová cena, činí z programu wxMaxima zajímavého reprezentanta kategorie CAS programů ve školách.

2.1 Instalace

Program wxMaxima je volně stažitelný open source program, dostupný na stránce [4]. Jeho instalace je provedena téměř samostatně instalačním programem, bez nároků na speciální dovednosti a znalosti uživatele. Program je po nainstalování připraven k použití. Některé rysy jeho počátečního nastavení lze snadno změnit pomocí grafického rozhraní, které vidíme na obrázku 1.

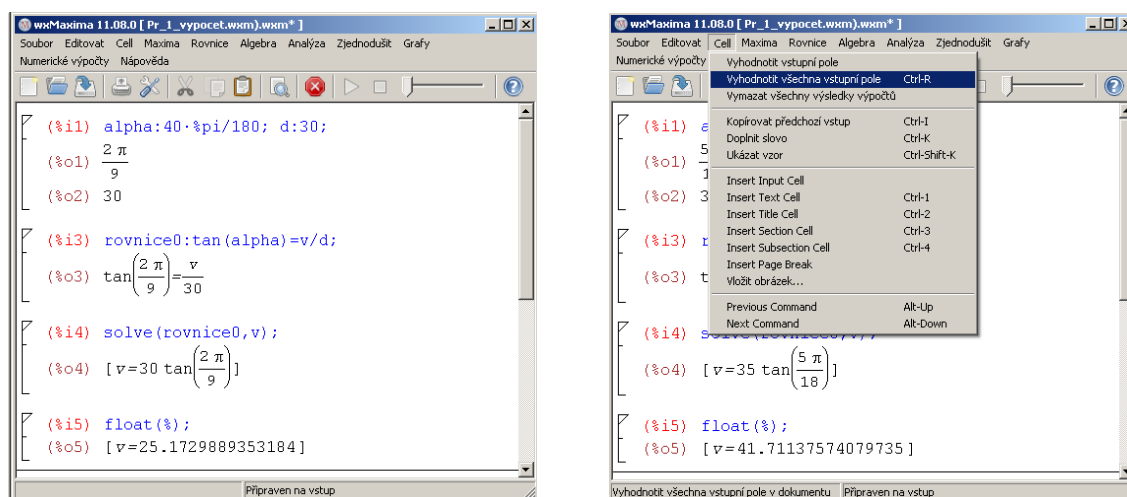
2.2 Nastavení

Všechna užitečná nastavení programu provedeme prostřednictvím dialogového okna **Konfigurace programu wxMaxima**, které je přístupné prostřednictvím posloupnosti akcí **Editovat - Nastavení**. Zde můžeme především nastavit jazyk, kterým s námi komunikuje grafické rozhraní programu. Poznamenejme, že česká lokalizace grafického uživatelského rozhraní není úplná, nápověda pak je dostupná pouze v angličtině. Dále lze v okně **Konfigurace programu wxMaxima** doporučit zaškrtnutí volby **Enter vyhodnocuje výraz**. Tím zajistíme, že obsah řádku se odešle ke zpracování do paměti pouhým stisknutím klávesy „Enter“, na rozdíl od implicitního nastavení, které vyžaduje stisknutí dvojice kláves „Ctrl“ a „Enter“. Za zmínku stojí ještě volby **Použít centrovanou tečku jako symbol násobení**, **Uložit velikost/pozici okna wxMaxima** nebo **Kontrola závorek v textovém vstupu**. Poslední jmenovaná volba zajistí, že spolu s levou závorkou se automaticky píše i pravá, kurzor přitom zůstává mezi nimi. Pokud chceme v tomto režimu uzavřít do závorek již existující výraz, nejprve ho označíme přetažením myši, potom už stačí jenom napsat levou závorku. Po kliknutí na ikonu **Styl** u levého okraje okna

Konfigurace programu wxMaxima můžeme měnit parametry písma v jednotlivých komponentách programu. To nám umožní například zvětšit písmo před použitím programu na interaktivní tabuli.

2.3 Výstup

Při uložení dokumentu s přednastavenou příponou „wxm“ se ztrácí výstupy zapsaných příkazů. Při následujícím otevření dokumentu se zobrazí pouze tyto příkazy bez jejich výsledků. Pro jejich získání tak musíme provést opětovné potvrzení všech příkazů, nejlépe pomocí volby **Cell - Vyhodnotit všechna vstupní pole** nebo příslušné klávesové zkratky „Ctrl+R“. Uložíme-li dokument s druhou nabízenou příponou „wxmx“, výstupy v něm zůstanou zachovány. Nezůstanou ale v paměti. Tam je dostaneme pouze opětovným potvrzením příslušných příkazů. Uvedená volba **Cell - Vyhodnotit všechna vstupní pole** („Ctrl+R“) tak spolu s volbou **Maxima - Restart programu Maxima** hraje důležitou roli při opakovaném používání již hotového dokumentu. Užitečnost možnosti vyhodnocení celého dokumentu najednou ilustruje obrázek 1. Předpokládejme, že máme, například pro práci na interaktivní tabuli, předpřipraven dokument pro realizaci nějakého výpočtu. V našem případě se jedná o výpočet výšky stromu. Potom stačí, poté co se studenty příslušné vztahy odhalíme rozborem obrázku na tabuli, odpovídající dokument vyvolat, dosadit aktuální hodnoty vstupních parametrů (elevační úhel a vzdálenost pozorovatele od paty stromu) a stisknout **Ctrl+R**. Celý dokument je ihned přepočítán pro nové vstupní hodnoty.



Obrázek 1: Výpočet výšky stromu pro různé vstupní hodnoty

2.3.1 Export

Při přípravě výukových materiálů oceníme možnost exportu do formátů „html“ a „tex“. Takto vytvořené soubory můžeme buď rovnou použít k prezentaci na webu, respektive pro generování dokumentu „pdf“, nebo je můžeme prostřednictvím příslušných editorů dále upravovat. Mohou se také stát součástí rozsáhlejšího textu či www stránky. Export dokumentu do „html“ rovněž představuje rychlou cestou, jak vytvářet kvalitní a všem dostupné instruktážní materiály pro práci se samotným programem wxMaxima.

2.4 Náповěda

Program wxMaxima disponuje podrobnou nápovědou, která obsahuje i příklady použití příslušných příkazů. Jak již bylo řečeno, nápověda není lokalizována do češtiny. Kompletní text nápovědy, spolu s příklady použití a odkazem na webovou stránku [4], kde najdeme online výukové materiály pro zvládnutí konkrétních dovedností s programem, najdeme pod heslem **Nápověda** v nabídce příkazů. Nápovědu ke konkrétnímu příkazu vyvoláme odesláním výrazu ve tvaru `? příkaz` (pozor, mezi otazníkem a příkazem musí být mezerka). Pokud je příkaz již zapsán v dokumentu, stačí na něj přemístit kurzor a stisknout klávesu „F1“. Pokud něco nevíme a nápověda nepomůže, je možné vznést dotaz na poměrně aktivní fórum uživatelů, které je přístupné přes stránku programu [4].

2.5 Režimy práce s programem

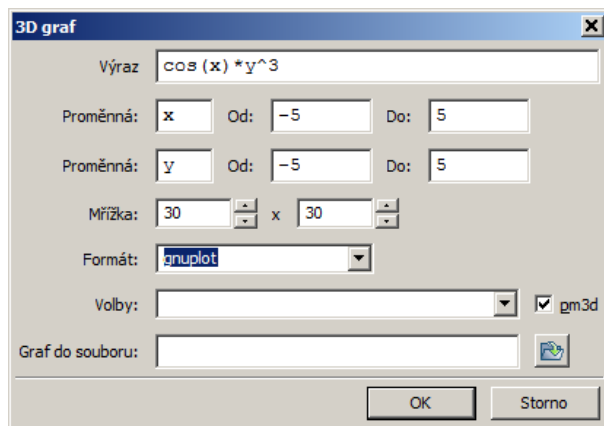
Funkce programu wxMaxima, které pokrývají velkou část potřeb matematiky na základní a střední škole, můžeme provádět dvěma způsoby: 1) s využitím nabídky grafického uživatelského rozhraní programu, bez nutnosti znát potřebné příkazy a jejich syntaxi, 2) prostřednictvím vstupního řádku programu, s nutností znát potřebné příkazy a jejich syntaxi.

Druhý způsob je samozřejmě univerzální a umožňuje nám využít všechny příkazy a funkce programu. Při volbě prvního způsobu můžeme program použít k základním výpočtům, bez nutnosti znát příkazy, ale s omezením jenom na redukovanou nabídku příkazů interpretovaných grafickým rozhraním. Každý režim práce s programem má tak své zřejmé výhody i nevýhody. Rozhodně je však třeba mít stále na paměti, že možnosti programu daleko převyšují nabídku akcí zprostředkovaných jeho grafickým rozhraním. Postup při řešení úlohy prostřednictvím grafického rozhraní je ilustrován následujícím příkladem.

Příklad 1. *Zobrazte graf funkce dvou proměnných, která je dána předpisem $f(x, y) = \cos x \cdot y^2$.*

Z nabídky příkazů grafického rozhraní vybereme posloupnost **Grafy - 3D graf...** Objeví se srozumitelný formulář (viz Obr. 2) obsahující volby, které mohou ovlivnit podobu a formát výstupu grafu.

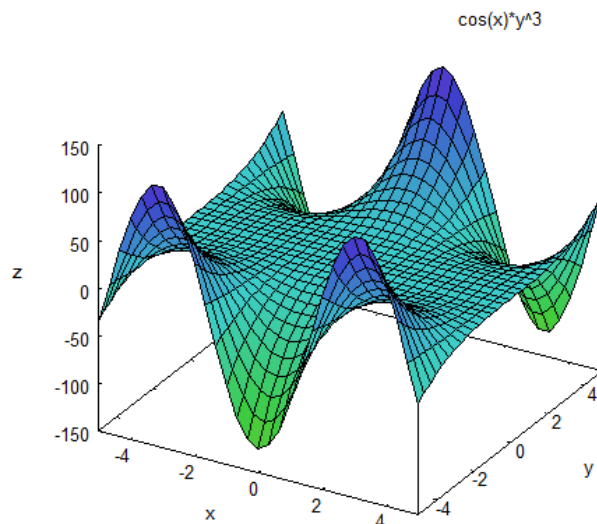
Po jeho vyplnění a potvrzení se ve zvláštním okně zobrazí graf dané funkce, který vidíme na obrázku 3.



Obrázek 2: Formulář grafického rozhraní pro zadání 3D grafu

Zároveň s vykreslením grafu se do aktivního dokumentu otevřeného v okně programu přidá vstupní řádek s příslušným příkazem:

```
(%i1) plot3d(cos(x)*y^3, [x,-5,5], [y,-5,5], [plot_format,gnuplot])$
```



Obrázek 3: 3D graf

Příkazy grafického rozhraní můžeme aplikovat i na výrazy, které jsou již zapsány v dokumentu. Při vyvolání vhodného příkazu z nabídky uživatelského rozhraní, například *Rovnice - Řešit...*, se tento příkaz provede buď na posledním potvrzeném (např. klávesou „Enter“) výrazu, nebo, přemístíme-li kurzor, na výrazu v řádku s kurzorem.

3 Vybrané úlohy

3.1 Základní výpočty

Symbolické i numerické možnosti programu oceníme při řešení úloh, které jsou sice jednoduché, jejichž úplné řešení je však výpočetně náročné.

Příklad 2. Legenda o tvůrci šachu [2]. *Když tvůrce šachu předvedl novou hru svému vládci, ten byl tak nadšen, že ho vyzval, ať si sám určí odměnu za tento vynález. Odpověď každého přítomného překvapila. Tvůrce hry totiž vyslovil přání, aby dostal počet zrněk obilí, který je určen tak, že na první políčko šachovnice se položí jedno zrnko, na druhé dvě, na třetí čtyři a tak dále, tj. na každé následující políčko dvojnásobek počtu zrn z předchozího, dokud není vyčerpáno všech 64 políček. Vládce, skoro až uražený skromností tohoto požadavku, dal svému pokladníkovi okamžitý příkaz k spočítání a následnému vydání odpovídajícího množství zrn. Po více než týdenním počítání pokladník zjistil, že ke splnění vynálezceva jednoduchého přání nebude stačit ani veškeré obilí z královských sýpek. Kolik zrn obilí čítala požadovaná odměna? Vyjádřete toto množství v kilogramech, tunách či jiným způsobem, který by nám napomohl vytvořit si jeho názornou představu.*

Program wxMaxima nám umožňuje provést symbolický zápis řešení úlohy a spočítat součet příslušné posloupnosti:

```
(%i1) 'sum(2^k, k, 0, 63)=sum(2^k, k, 0, 63);
```

```
(%o1) 
$$\sum_{k=0}^{63} 2^k = 18446744073709551615$$

```

Pomocí příkazu `makelist` si můžeme nechat vypsat obsahy všech 64 políček. Zde je z důvodu úspory místa tento výstup zkrácen:

```
(%i2) makelist(2^k,k,0,63);
```

```
(%o2) [1, 2, 4, 8, 16, 32, 64, 128, ..., 4611686018427387904, 9223372036854775808]
```

Tak již víme, že vládce by musel tvůrci hry vydat 18446744073709551615 zrn obilí. Máme ale představu o tomto množství? Kolik je to kilogramů? Kolik nákladních automobilů Tatra 815 by musel vládce nechat zrním naložit? Kdyby je měl, samozřejmě. Uvažujte, že 1000 zrn obilí má hmotnost přibližně 50 gramů a že nejběžnější modifikace Tatry 815 má nosnost 15 tun.

```
(%i3) Zrn:sum(2^k, k, 0, 63);
```

```
(%o3) 18446744073709551615
```

```
(%i6) Zrn_kg:float(Zrn/20000);
```

```
(%o6) 9.223372036854776 1014
```

```
(%i7) Zrn_Aut:Zrn_kg/15000;
```

```
(%o7) 6.148914691236517 1010
```

Vládce by tedy na transport daru potřeboval přibližně $6 \cdot 10^{10}$ Tatrovek. S tak silným výpočetním prostředkem v ruce můžeme zcela pustit uzdu své fantazie a začít tvořit v úlohách tohoto typu hojně užívané vláčky (z Tatrovek) či řetězy (ze zrn) a jejich délky porovnávat s různými pozemskými či vesmírnými rozměry. Tak můžeme, snad i s jistou dávkou zábavy, dospět k celkem nenásilnému poučení o rozměrech světa, který nás obklopuje.

Příklad 3. Narozeninový problém [1]. *Jaká je pravděpodobnost, že ve skupině n lidí, například ve školní třídě s $n = 30$ žáky, najdeme alespoň dvě osoby, které mají stejný den a měsíc narození?*

Uvažovaná pravděpodobnost je překvapivě vysoká již pro poměrně malé hodnoty n , proto se ve spojení s touto úlohou často používá termín „paradox“. Jinak ale postup řešení tohoto problému není nijak složitý a pracuje pouze se základními pojmy pravděpodobnosti na úrovni střední školy. Vycházíme z klasické definice pravděpodobnosti a pro snazší početní uchopení pracujeme s jevem doplňkovým. Řešíme proto nejprve otázku, jaká je pravděpodobnost $np(n)$, že v dané skupině n lidí nemají žádní dva narozeniny ve stejný den. Hledaná pravděpodobnost je potom dána vztahem $p(n) = 1 - np(n)$. Jedná se tak o pěknou aplikaci středoškolského učiva pravděpodobnosti, jejímž jediným úskalím může být konečný numerický výpočet. Z následující ukázky je zřejmé, že použití wxMaximy nám umožní dojít se studenty ke konečnému výsledku, případně dále zkoumat vývoj hodnot $p(n)$ v závislosti na n .

```
(%i1) np(n):=product(365-k+1, k, 1, n)/365^n;
```

```
(%o1) np(n) :=  $\frac{\prod_{k=1}^n 365 - k + 1}{365^n}$ 
```

```
(%i2) p(n):=1-np(n);
```

```
(%o2) p(n) := 1 - np(n)
```

```
(%i3) p(30), numer;
```

```
(%o3) 0.706316242719269
```

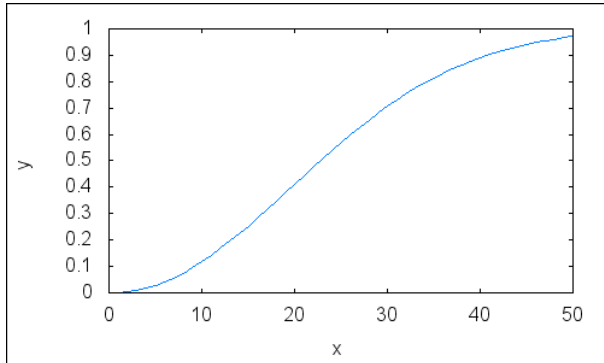
Pravděpodobnost, že ve skupině 30 lidí najdeme alespoň dvě osoby, které mají stejné datum narození, je tedy přibližně 71%.

Abychom měli lepší představu o vývoji závislosti $p(n)$ na n můžeme si nechat programem spočítat hodnoty $[n, p(n)]$ pro nějaký počáteční interval hodnot n , např. od 1 do 50 (výstup programu je pro úsporu místa zkrácen), a tyto hodnoty pak znázornit graficky.

```
(%i4) listp: makelist([k,p(k)], k, 1, 50), numer;
```

```
(%o4) [[1, 0], [2, 0.00273972602739725], ..., [49, 0.965779609322676], [50, 0.970373579577988]]
```

```
(%i5) wxplot2d([discrete, listp], [x, 0, 50]);
```



3.2 Rovnice

Příklad 4. Je vám nabídnuta půjčka 103 000 Kč s dobou splácení 12 let při roční splátce 10 000 Kč. Spočítejte roční úrokovou míru „ i “ této půjčky.

Řešení úlohy vede na polynomickou rovnici (třináctého stupně) s neznámou „ i “. Tuto rovnici odvodte, graficky znázorněte, určete všechna její reálná řešení a nakonec vyberte to, které se vztahuje k zadání příkladu.

Program CAS dovoluje provést symbolický zápis úlohy, grafické znázornění příslušného polynomu i jeho numerické řešení.

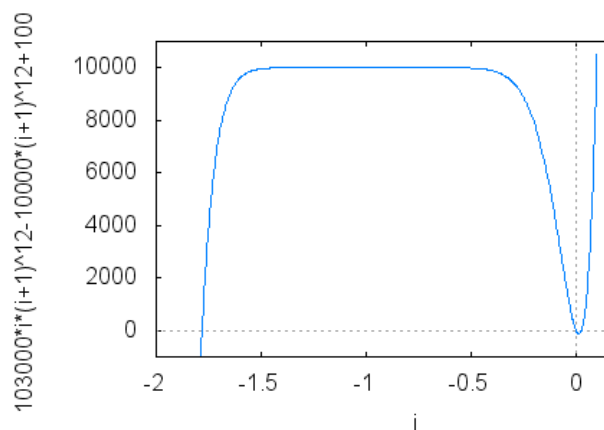
```
(%i1) polynom:103000*i*(1+i)^12-10000*(1+i)^12+10000;
```

```
(%o1) 103000 i (i + 1)12 - 10000 (i + 1)12 + 10000
```

```
(%i2) expand(polynom);
```

```
(%o2) 103000 i13 + 1226000 i12 + 6678000 i11 + 22000000 i10 + 48785000 i9 + 76626000 i8 + 87252000 i7 + 72336000 i6 + 43065000 i5 + 17710000 i4 + 4598000 i3 + 576000 i2 - 17000 i
```

```
(%i3) wxplot2d(polynom, [i, -2, 0.5], [y, -1000, 11000]);
```



Pohledem na graf zjistíme, že polynom má nulový bod, jehož poloha odpovídá očekávané hodnotě úrokové míry „ i “. Hodnotu tohoto nulového bodu potom spočítáme numericky:


```
(%i4) find_root(polynom,i,0.001,0.5);
```

```
(%o4) 0.0243220266129531
```

Roční úroková míra „*i*“ uvažované půjčky činí po zaokrouhlení 2.4 %.

3.3 Goniometrické rovnice

U některých funkcí programu není z jejich jména na první pohled zřejmé, k čemu všemu je můžeme využít. To je i případ použití funkce `to_poly_solve` k řešení goniometrických rovnic, které je ilustrováno následujícím příkladem. Tento jev je zřejmě důsledkem otevřeného vývoje programu. Je dobré si na to vzpomenout vždy, když nás první neúspěch s programem svádí k prohlášení, že „to program neumí“. Vždy stojí za to se před takovýmto prohlášením ještě poptat na uživatelském fóru [4].

Příklad 5. Řešte v \mathbb{R} goniometrickou rovnici: $\frac{1}{2} \sin x + \frac{\sqrt{3}}{2} \cos x = -\frac{1}{2}$.

Pro řešení goniometrické rovnice použijeme funkci `to_poly_solve`, která je součástí balíčku `to_poly_solver`. Proto musíme tento balíček ze všeho nejdříve nahrát do paměti příkazem `load`.

```
(%i1) load(to_poly_solver);
```

```
(%i2) rov2:1/2*sin(x)+sqrt(3)/2*cos(x)=-1/2;
```

```
(%o2)  $\frac{\sin(x)}{2} + \frac{\sqrt{3} \cos(x)}{2} = -\frac{1}{2}$ 
```

```
(%i3) to_poly_solve([rov2],[x]);
```

```
(%o3) %union  $\left( [x = 2\pi \%z2 + \frac{5\pi}{6}], [x = 2\pi \%z3 - \frac{\pi}{2}] \right)$ 
```

Na posledním řádku vidíme, jak funkce `to_poly_solve` dokáže poměrně přehledným způsobem zapsat obecné řešení dané rovnice, kde `%union` je symbolem sjednocení a `%z2`, `%z3` jsou celočíselné parametry.

Pro samotné úpravy výrazů s goniometrickými funkcemi nám wxMaxima nabízí explicitněji pojmenované funkce, například `trigsimp` či `trigreduce`.

3.4 Soustavy rovnic

Program wxMaxima nabízí celou řadu funkcí pro řešení soustav lineárních rovnic, od jejich přímého řešení, přes použití matic a determinantů až po grafické znázornění rovnic, pokud má smysl. Jedná se například o funkce `solve`, `linsolve`, `coefmatrix`, `augcoefmatrix`, `triangularize`, `echelon`, `col`. V článku není prostor pro podrobné představení všech těchto funkcí, ani to není jeho účelem. V následujícím příkladu se proto zaměříme pouze na ukázkou grafické reprezentace rovnic o třech neznámých s využitím funkce `wxdraw3d`.

Příklad 6. Řešte následující soustavu lineárních rovnic:

$$\begin{aligned}x + 2y + z &= 2 \\2x + 6y + z &= 7 \\x + y + 4z &= 3.\end{aligned}$$

Pokud nevyužijeme služeb grafického rozhraní (volba **Rovnice - Řešit lineární systém**), začneme třeba uložением rovnic do proměnných. Pro přímé vyřešení soustavy potom nasadíme funkci `linsolve`.

```
(%i1) r1:2*x-2*y+z=-9; r2:2*x+6*y-z=5; r3:x+y+4*z=3;
```

```
(%i4) linsolve([r1,r2,r3],[x,y,z]);
```

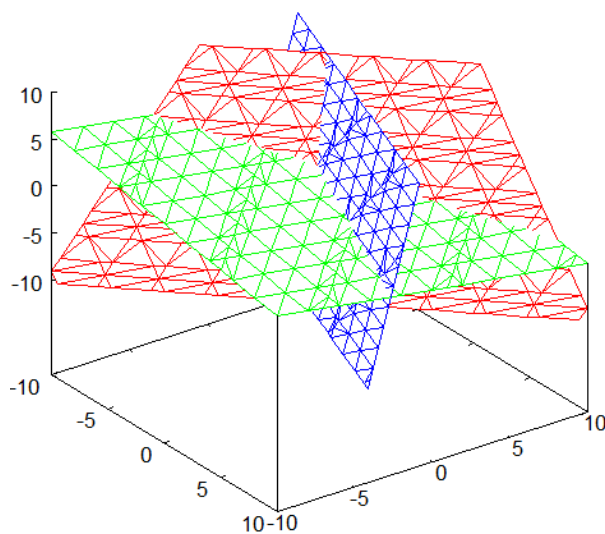
```
(%o4) [x = -3, y = 2, z = 1]
```

Funkce `wxdraw3d` náleží do balíčku `draw`, který musíme nejprve nahrát do paměti:

```
(%i5) load(draw)$
```

Pro zobrazení rovnic bez jejich předchozí úpravy na explicitní tvar převedeme každou rovnici funkcí `implicit` na tzv. grafický objekt, na který potom aplikujeme funkci `wxdraw3d`:

```
(%i12) wxdraw3d(surface_hide=true,  
color=red,implicit(r1,x,-10,10,y,-10,10,z,-10,10),  
color=blue,implicit(r2,x,-10,10,y,-10,10,z,-10,10),  
color=green,implicit(r3,x,-10,10,y,-10,10,z,-10,10));
```



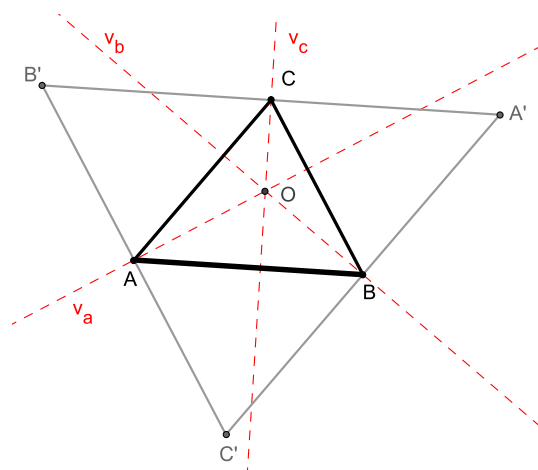
Obrázek 4: Grafické řešení soustavy tří lineárních rovnic o třech neznámých

3.5 Geometrie

Použitím wxMaximy lze dobře doplnit stále ještě chybějící symbolickou složku programu GeoGebra.

Příklad 7. *Dokažte, že všechny tři výšky v libovolném trojúhelníku mají jeden společný bod.*

Existují různé důkazy této vlastnosti. Provedení jednoho z nich v programu GeoGebra je znázorněno na obrázku 5. Podstatou tohoto důkazu je převedení otázky průsečíku výšek trojúhelníku ABC na otázku průsečíku os stran trojúhelníku $A'B'C'$.



Obrázek 5: Společný průsečík výšek - syntetický důkaz

Jak ilustruje následující záznam kódu (výstupy některých příkazů nejsou z prostorových důvodů zobrazeny), program wxMaxima nám umožňuje doplnit obvyklý syntetický přístup k důkazům geometrických vlastností o přístup analytický, důsledně symbolický. Vrcholům trojúhelníku ABC a obecnému bodu X jeho roviny přiřadíme symbolické souřadnice. Potom zapíšeme rovnice výšek trojúhelníku ABC jako přímek procházejících vrcholy trojúhelníku a kolmých k protilehlým stranám.

```
(%i4) A: [a1, a2]; B: [b1, b2]; C: [c1, c2]; X: [x, y];
```

```
(%i8) va: (X-A) . (B-C)=0; vb: (X-B) . (C-A)=0; vc: (X-C) . (B-A)=0;
```

```
(%o8) (b2 - c2) (y - a2) + (b1 - c1) (x - a1) = 0
```

```
(%o9) (c2 - a2) (y - b2) + (c1 - a1) (x - b1) = 0
```

```
(%o10) (b2 - a2) (y - c2) + (b1 - a1) (x - c1) = 0
```

Potom určíme společná řešení dvou různých dvojic z těchto tří rovnic a tato řešení porovnáme vzájemným odečtením po souřadnicích.

```
(%i11) Pab: solve([va, vb], [x, y]);
```

```
(%i12) Pac:solve([va,vc],[x,y]);
```

```
(%i13) Pab[1][1]-Pac[1][1]; Pab[1][2]-Pac[1][2];
```

```
(%o13) 0 = 0
```

```
(%o14) 0 = 0
```

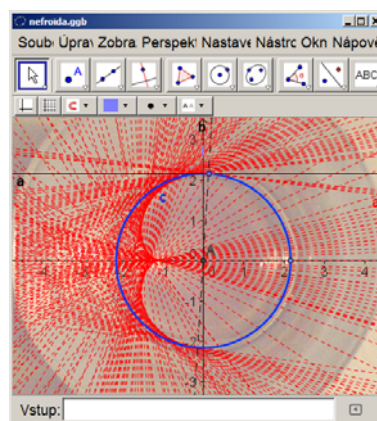
Vyšly nám nuly. To znamená, že obě dvojice výšek mají stejný společný bod. Tím je důkaz proveden.

3.6 Model reálného jevu

I v následujícím příkladě figuruje wxMaxima po boku GeoGebry. Tentokrát se představují jako ideální dvojice pro modelování reálného jevu, se kterým se každý z nás denně setkává.

Příklad 8. *Za určitých okolností můžeme na dně dobře umytého hrnečku nebo na hladině nápoje v něm pozorovat křivku podobnou srdci (viz Obr. 6, vlevo). Tento jev nejprve modelujte v programu GeoGebra, potom v programu wxMaxima odvoďte parametrické rovnice pozorované křivky.*

Dotyčná křivka je polovinou křivky zvané nefroida [6]. Vzniká jako obálka světelných paprsků odražených od vnitřní stěny nádoby. Tím se řadí do rodiny tzv. kaustik. Geometrickou podstatu vzniku této křivky snadno modelujeme v programu GeoGebra. Využijeme při tom možnosti plynulé změny polohy bodu dopadu světelného paprsku a zobrazení stopy paprsku odraženého, jak vidíme na obrázku 6, vpravo.



Obrázek 6: Srdce ve sklenici - geometrický model

Pro odvození parametrických rovnic křivky ve wxMaximě použijeme poznatky diferenciální geometrie o obálce systému křivek v rovině. Jedná se o křivku, která má v každém svém bodě tečnu společnou s jednou z křivek uvažovaného systému. Její rovnice jsou řešením soustavy rovnic, která je tvořena rovnicí parametrického systému křivek $l(x, y, \varphi) = 0$,

kde φ je reálný parametr, a její derivací $\frac{\partial l(x, y, \varphi)}{\partial \varphi} = 0$. Začneme zadáním souřadnic bodu dopadu I a normálových vektorů nd a nr přímeek, které v tomto pořadí reprezentují dopadající a odražený paprsek (výstupy neuvádíme):

```
(%i1) I: [cos(phi+%pi/2), sin(phi+%pi/2)];
```

```
(%i2) nd:matrix([0,1]);
```

```
(%i3) nr: [-sin(2*phi), cos(2*phi)];
```

Poté definujeme uvedené dvě rovnice $l(\varphi)$ a $ld(\varphi)$ a řešíme jejich soustavu. Výsledkem jsou parametrické rovnice zkoumané křivky, o čemž svědčí i závěrečné grafické znázornění.

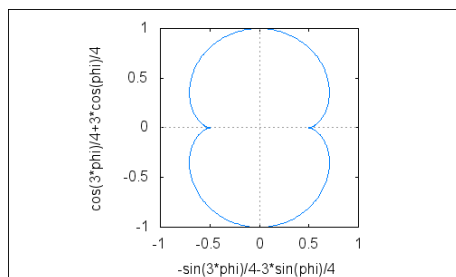
```
(%i4) l(phi):=(x-I[1])*nr[1]+(y-I[2])*nr[2]=0;
```

```
(%i5) ld(phi):=diff(l(phi),phi);
```

```
(%i6) Curve:trigreduce(solve([l(phi)=0,ld(phi)=0],[x,y]));
```

```
(%o6) [[x = -sin(3*phi)/4 - 3*sin(phi)/4, y = cos(3*phi)/4 + 3*cos(phi)/4]]
```

```
(%i7) wxplot2d([parametric,rhs(Curve[1][1]),rhs(Curve[1][2]),[phi,0,2*%pi],
[nticks,400]],[x,-1,1],[y,-1,1],[gnuplot_preamble,"set size ratio
1;"]);
```

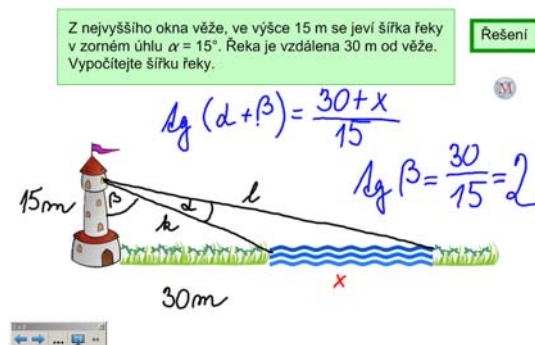


4 Integrace wxMaximy na interaktivní tabuli

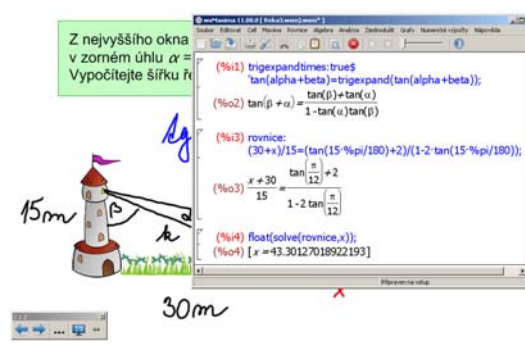
Program wxMaxima najde své uplatnění i při práci s interaktivní tabulí. Samozřejmě ho využijeme jako nástroj pro rychlé symbolické či numerické výpočty a kreslení grafů. Můžeme si v něm ale také předpřipravít nějaké zajímavé, ne však zcela triviální, řešení probíraného problému, které chceme studentům ukázat. Integraci wxMaximy v prostředí interaktivní tabule budeme ilustrovat následujícím příkladem, jehož zadání na tabuli vidíme na obrázku 7 a).

Příklad 9. *Z nejvyššího okna věže ve výšce 15 m se jeví šířka řeky v zorném úhlu $\alpha = 15^\circ$. Řeka je vzdálena 30 m od věže. Vypočítejte šířku řeky.*

Pro řešení příkladu se nabízí kosinová věta, ale také mnohem elegantnější použití součtového vzorce pro funkci tangens. Obě řešení stojí za to, aby si je studenti propočítali. Není však na škodu mít je, pro kontrolu i pro ilustraci, předpřipravené ve wxMaximě, jak ukazují obrázek 7 b).



a) Náčrtek řešení úlohy



b) Výpočet pomocí wxMaximy

Obrázek 7: Využití wxMaximy k výpočtům na interaktivní tabuli

Poděkování

Tento článek vznikl za podpory projektu 089/2010/S hrazeného Grantovou agenturou Jihočeské univerzity.

Literatura

- [1] Birthday problem. In *Wikipedia: The Free Encyclopedia* [online]. Dostupné na adrese http://en.wikipedia.org/wiki/Birthday_problem
- [2] Wheat and chessboard problem. In *Wikipedia: The Free Encyclopedia* [online]. Dostupné na adrese http://en.wikipedia.org/wiki/Wheat_and_chessboard_problem
- [3] Maxima, a Computer Algebra System [online]. Dostupné na adrese <http://maxima.sourceforge.net/>
- [4] wxMaxima [online]. Dostupné na adrese <http://andrejv.github.com/wxmaxima/>
- [5] Leydold, J., Petry, M. *Introduction to Maxima for Economics* [online]. Institute for Statistics and Mathematics, WUWien. 2011. Dostupné na adrese
- [6] Nephroid. In *Wikipedia: The Free Encyclopedia* [online]. Dostupné na adrese <http://en.wikipedia.org/wiki/Nephroid>,

Roman Hašek
 Jihočeská univerzita v Č. B.
 Pedagogická fakulta
 Jeronýmova 10
 371 15 České Budějovice
 e-mail: hasek@pf.jcu.cz

Michaela Noruláková
 Jihočeská univerzita v Č. B.
 Pedagogická fakulta
 Jeronýmova 10
 371 15 České Budějovice
 e-mail: bulvoko@seznam.cz