Užití Derive ve výuce matematiky

Roman Hašek

České Budějovice 2007

Recenzenti:
prof. RNDr. Pavel Pech, CSc., Mgr. Šárka Voráčová, Ph.D.

ISBN 978-80-239-9054-6

Obsah

Úvod

Kniha je určena všem zájemcům o využití programu Derive v matematice od základní školy až po úvodní vysokoškolské kurzy. Věřím, že ji ocení zvláště vyučující a žáci na základních a středních školách, stejně jako studenti fakult připravujících učitele. Měla by jim pomoci rychleji se zorientovat v možnostech využití programu ve výuce.

Kniha přináší výběr řešených úloh a zkušeností, které jsem nashromáždil za uplynulé čtyři roky, během nichž jsem působil jako školitel seminářů a kurzů věnovaných využití Derive ve výuce matematiky. Na Pedagogické fakultě Jihočeské univerzity jsou s tímto programem seznamováni jak studenti učitelství matematiky tak i učitelé v rámci programu dalšího vzdělávání učitelů.

Předkládané úlohy pokrývají větší část středoškolských matematických osnov. Najdeme zde však i témata, která odpovídají náplni úvodních vysokoškolských kurzů z lineární algebry, matematické analýzy a geometrie. Některým tématům je věnován větší prostor, například funkcím, řešení soustav lineárních rovnic, grafům relací, programování nebo řešení geometrických úloh. V odpovídajících kapitolách najde čtenář kompletní návody na využití ve výuce. Jiná témata, jako třeba kombinatorika či množiny, jsou pouze naznačena. To je odrazem toho, že posláním knihy není jenom představit soubor řešení konkrétních úloh. Měla by čtenáře též inspirovat.

Program Derive.

Programem Derive se v celé knize rozumí česká verze Derive 6.1, která byla vydána na podzim roku 2004. Tím se čeština stala, vedle angličtiny, němčiny, francouzštiny, španělštiny, italštiny, holandštiny, maďarštiny a slovenštiny, již devátým jazykem v němž je program Derive uživatelům nabízen. Derive je program typu CAS (computer algebra system), v němž je možno provádět symbolické i numerické výpočty a kreslit dvourozměrné i třírozměrné grafy. Při výpočtech má uživatel k dispozici velké množství matematických funkcí, případně si může programovat funkce vlastní. Historie programu začíná rokem 1979, v němž byl vydán jeho předchůdce, program muMATH. První verze Derive byla vydána v roce 1988. Po odkoupení firmou Texas Instruments byla v roce 1996 vydána i verze pro Windows. „Dosovská" verze programu se stala na dlouhou dobu populární pro nezvyklou kombinaci malé velikosti a velké výpočetní síly. V současné době je program využíván na řadě středních i vysokých škol v USA a v Evropě. V některých zemích je Derive oficiálním programem pro podporu výuky matematiky na školách, například v Rakousku nebo na Slovensku. Uplatnění má i mimo vzdělávací oblast, v inženýrské praxi. Program Derive 6.1 je určen pro operační systém Windows, verze XP, 2000, Millennium a 98.

Pro využití na základních a středních školách je program dle mého názoru vhodný především díky jednoduchosti obsluhy, jejímž důsledkem je snadná a krátká cesta od zadání k výsledku. Program je totiž vybaven rozhraním, které nám umožňuje provádět většinu výpočtů na úrovni matematiky základní a střední školy bez znalosti potřebných příkazů. Program tak se svými příkazy ustupuje do pozadí -

není třeba se je učit a uvolňuje místo řešenému problému. K výsledku, ať již v symbolické, numerické či grafické podobě, navíc vede ve většině případů jen velmi malý počet úkonů.

Strukura knihy.

Kniha je členěna do dvou částí a 26 kapitol.

V první části zvané „Obsluha programu“, která obsahuje deset kapitol, je čtenář na řešených příkladech seznámen se základními rysy a možnostmi programu Derive. Jsou zde popsány různé užitečné postupy, například zjednodušování výrazů, tvorby grafu, editace dokumentu či programování vlastních funkcí a vytváření jejich balíčků. Uvedené postupy jsou potom uplatněny při řešení příkladů ve druhé části knihy. Hlavním účelem úvodní části je tedy připravit čtenáře tak, aby byl schopen aktivně a s porozuměním studovat řešení úloh ve druhé, rozsáhlejší, části knihy. Vedle toho může být tato část knihy chápána samostatněji, jako stručný úvod do programu Derive. V žádném případě však nelze tuto část, stejně jako celou knihu, chápat jako kompletní uživatelskou příručku Derive. Tou je pouze Nápověda programu. Na to by čtenář neměl zapomenout a i při studiu této knihy by měl Nápovědu aktivně využívat.

Zájemci o program Derive rovněž doporučuji pečlivé prostudování knihy Bernharda Kutzlera *Derive 6, Pokročilá matematika pro vaše PC,* [10], která je dodávána spolu s programem.

Druhá, podstatně rozsáhlejší, část knihy je tvořena šestnácti kapitolami. Každá kapitola je věnována vybranému tématu z matematiky. Jak už bylo řečeno, některému tématu je věnováno více prostoru, jiná jsou probrána stručněji.

Kniha je vybavena podrobným obsahem a abecedním rejstříkem. V případě zájmu o konkrétní téma se čtenář kombinací těchto dvou prostředků snadno dostane k cíli.

Mé upřímné poděkování patří oběma recenzentům, prof.RNDr. Pavlovi Pechovi a Mgr. Šárce Voráčové, Ph.D. za pečlivé prostudování rukopisu a za cenné rady a připomínky, kterými přispěli ke zkvalitnění konečné podoby této knihy.

Doufám, že Vás kniha zaujme a shledáte ji užitečnou.

V každém případě uvítám jakékoliv připomínky, nápady či sdělení Vašich osobních zkušeností.

Informace o programu Derive spolu s dalšími příklady jeho využití jsou průběžně uváděny na webové stránce: www.pf.jcu.cz/~hasek/derive

V Českých Budějovicích

Roman Hašek

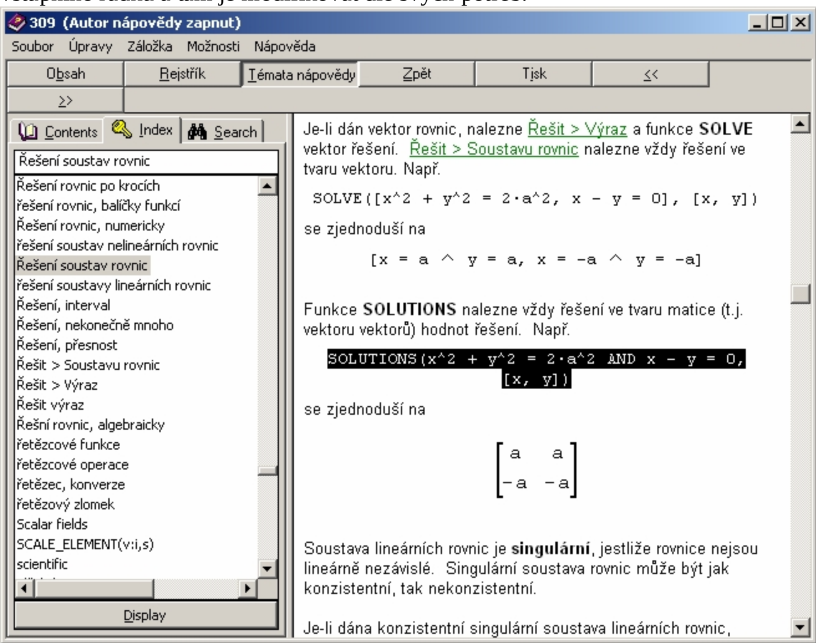
hasek@pf.jcu.cz

I. OBSLUHA PROGRAMU

První část knihy přináší stručný úvod do práce s programem Derive. Čtenář je na příkladech uveden do základních rysů programu, se kterými se potom setká při řešení konkrétních úloh v druhé části knihy. Jedná se především o pravidla komunikace s rozhraním programu, tvorbu dokumentů, úpravy výrazů, zobrazování dvourozměrných a třírozměrných grafů a programování vlastních funkcí. Při aktivním prostudování tohoto oddílu knihy získá čtenář poměrně zevrubný přehled o možnostech programu a bude připraven na jeho samostatné využití. Připomínám však, že se v žádném případě nejedná o náhradu kompletní uživatelské příručky. Naopak, předpokládá se, že uživatel bude během studia knihy aktivně pracovat s Nápovědou, která přináší úplný přehled funkcí programu.

1. Práce s nápovědou

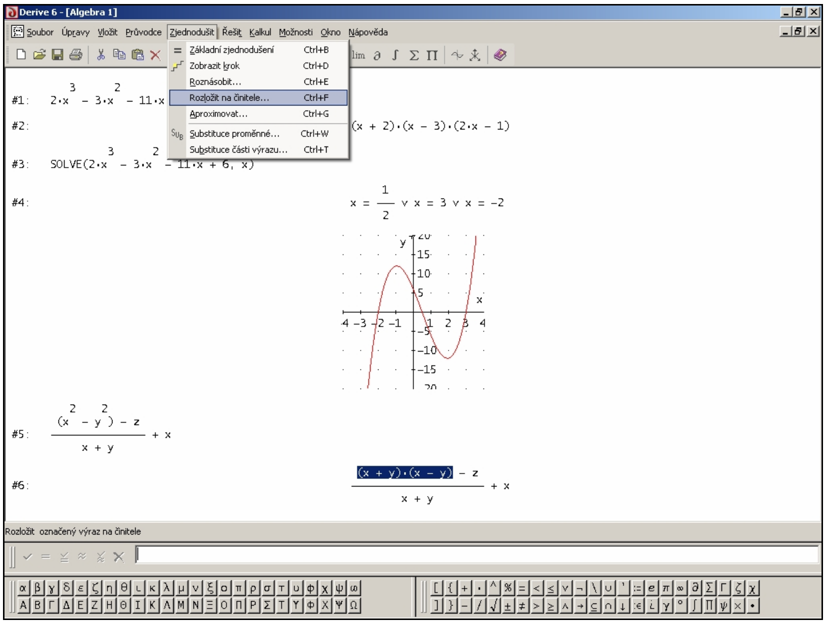
Jak bylo naznačeno výše, nápověda programu Derive představuje jediný dostupný přehled všech funkcí programu, doplněný vysvětlujícími komentáři a příklady použití. Neexistuje tištěná podoba takovéhoto úplného přehledu funkcí programu. Proto doporučuji čtenáři nevyhýbat se nápovědě a aktivně s ní pracovat. Nejenom, že se dozví téměř vše o funkcích programu, ale najde tam i různé příklady užití (i když je třeba upřímně přiznat, že ne mnoho a ne na všechno), které může pomocí kombinací kláves Ctrl+C (Kopírovat), Ctrl+V (Vložit) kopírovat z nápovědy do vstupního řádku a tam je modifikovat dle svých potřeb.



Představme si například, že nás zajímají možnosti řešení soustav rovnic v Derive. Vyvoláme tedy nápovědu (posloupností příkazů **Nápověda → Témata nápovědy** nebo klávesou F1). Potom stačí zvolit rejstřík (index) klíčových slov a napsat heslo „řešení“. Hned se nám objeví nabídka souvisejících témat. Vybrali jsme si téma „Řešení soustav rovnic“. Zde nás zaujal příkaz `SOLUTIONS`. Místo, abychom se snažili si zapamatovat jeho syntaxi, myší vyznačíme v textu nápovědy příklad použití příkazu kombinací kláves Ctrl+C, Ctrl+V ho překopírujeme do příkazového řádku Derive. Zde ho jenom přizpůsobíme svým představám a odešleme ke zjednodušení na pracovní plochu.

2. Grafické rozhraní

S programem komunikujeme prostřednictvím grafického rozhraní (Obr. 1). Řada jeho prvků je známa i z jiných programů pracujících pod operačním systémem Windows. Najdeme zde **nabídku akcí** (Obr. 1, druhý řádek), pod nimiž se skrývají **roletová menu** s konkrétními příkazy.



Obr. 1: Grafické rozhraní programu Derive 6

Ukážeme-li v menu na některý příkaz myší, objeví se pod spodním okrajem bílé pracovní plochy jeho stručná **charakteristika** (Obr. 1, text „Rozložit označený ...“). Nechybí ani **tlačítka** pro **rychlé provádění** některých příkazů (Obr. 1, třetí řádek).

Najdeme zde i **panely nástrojů** pro rychlé vkládání písmen řecké abecedy či matematických symbolů. Samozřejmě, podoba rozhraní se dá **změnit** (**Okno** → **Přizpůsobit...**). **Nápověda** programu je tradičně přístupná z nabídky **Nápověda** nebo klávesou (F1).

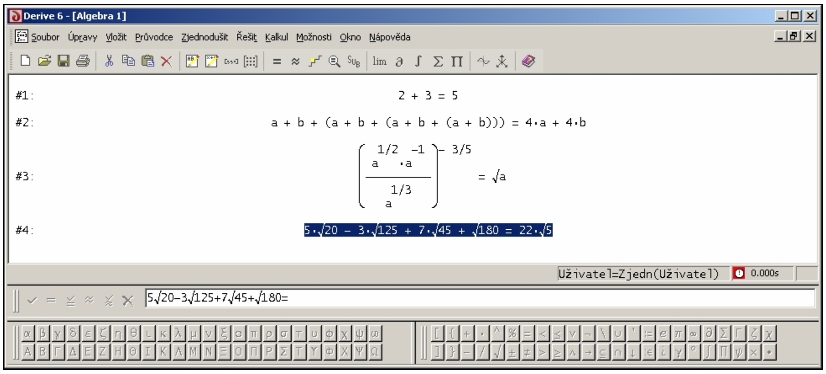
3. Vstup a výstup

Výrazy vkládáme do programu prostřednictvím **příkazového (vstupního) řádku** (**Obr. 1**, úzký bílý řádek při dolním okraji). Po napsání výrazu a stisknutí klávesy (¢) (Enter) nebo tlačítka ✓ (nalevo od příkazového řádku) se výraz objeví na **pracovní ploše** (v okně **Algebra**), která celému rozhraní programu dominuje. Každý výraz na pracovní ploše má své **číslo** (např. #1), které slouží i jako **proměnná**, pomocí níž se můžeme na výraz odvolávat. Chceme-li například sečíst dva výrazy, jež se na pracovní ploše vyskytují pod čísly #5 a #13, stačí na příkazový řádek napsat #5 + #13 = a stisknout (¢).
Často budeme potřebovat **přecházet mezi příkazovým řádkem a pracovní plochou**. Můžeme to řešit klikáním myší nebo použitím kláves (Esc) (nahoru) a (F2) (dolů).

4. Manipulace s výrazy

V rámci **pracovní plochy se pohybujeme** opět pomocí myši, nebo pomocí kláves (½)(¼). Zaznamenání historie naší práce na pracovní ploše nám může podstatně ušetřit čas při vkládání nových výrazů. Buď tak, že se odkazujeme na „staré“ výrazy pomocí jejich čísla (viz předchozí kapitola), nebo tak, že tyto výrazy, případně jenom jejich části, přeneseme na vstupní řádek následujícím způsobem. Výraz najdeme a **zvýrazníme** kliknutím myší. Potom ho můžeme **přenést** na příkazový řádek - **v původním tvaru** stisknutím klávesy (F3), **uzavřený v závorce** pak klávesou (F4). Tuto možnost manipulace oceníme hlavně při potřebě využít jenom část již existujícího výrazu. Při pohybu po pracovní ploše (myší nebo šipkovými klávesami) se zvýrazňují celé výrazy. **Zvýraznění jenom části výrazu** docílíme buď opakovaným klikáním myší s ukazatelem umístěným nad příslušnou částí výrazu (tato část musí mít sama o sobě smysl jako výraz), nebo pomocí **kombinace kláves** (½)(¼)(Æ)(æ) **při současném držení** klávesy (Shift). Výrazy můžeme po pracovní ploše **přemisťovat** (tj. měnit jejich pořadí) jednoduše **uchopením myší a přetažením** na požadované místo. To, zda jim zůstane jejich původní **číslo**, nebo se změní dle nového pořadí, ovlivníme volbou **Možnosti** → **Zobrazení** → **Přečíslovat výrazy**. **Čísla výrazů** (označení) **můžeme skrýt** volbou **Možnosti** → **Skrýt** → **Označení**.
Při manipulaci s výrazy oceníme skupinu tlačítek ✓ = ≦ ≈ ≨ ☒ vlevo od příkazového řádku. Jejich význam je v pořadí zleva doprava tento: **zobrazit výraz** na pracovní ploše, výraz **zjednodušit a výsledek zobrazit** na ploše, na ploše **zobrazit výraz i výsledek jeho zjednodušení**, výraz **aproximovat**, výraz **zobrazit na ploše spolu s jeho aproximací** a **smazat výraz** na příkazovém řádku. Tlačítka =, ≈ najdeme ještě na horní liště tlačítek. Odpovídající akce můžeme provést také volbami **Zjednodušit** → **Základní zjednodušení, Zjednodušit** →

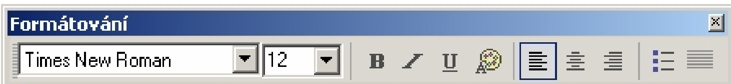
Aproximovat.... Chceme-li mít výraz i jeho zjednodušení na jednom řádku, napíšeme při vkládání za výraz znaménko = (rovná se) a odešleme stisknutím Enter (¢) (Obr. 2).



Obr. 2: Přímé zjednodušení výrazu

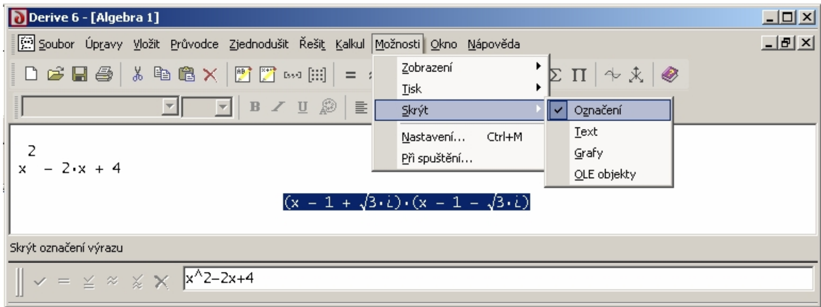
5. Tvorba dokumentů

Kromě výrazů a s nimi spojených výpočtů můžeme na pracovní ploše zobrazit **text** (**Vložit** → **Text...**) , **obrázky** (**Úpravy** → **Kopírovat** , **Vložit)**, **OLE** (Object Linking and Embedding) objekty všech myslitelných aplikací (**Vložit** → **OLE Objekt...**) a samozřejmě **grafy** vytvořené samotným programem (**Vložit** → **2D-graf...**, **Vložit** → **3D-graf...**), jak vidíme na Obr.1. Do textu můžeme vkládat i **odkazy na webové stránky** – začneme psát http://... a program pochopí. Formát textu můžeme měnit přímo v programu Derive. Příslušný panel nástrojů se jmenuje „Formátování" a vypadá takto:



Zobrazíme ho buď zatrhnutím na kartě „Panely nástrojů" dialogového okna **Okno** → **Přizpůsobit...** nebo zatrhnutím v nabídce, která se objeví po poklepání pravým tlačítkem myši s ukazatelem umístěným někde v prázdné oblasti šedivého rámu pracovní plochy programu. Skromnou nabídku panelu „Formátování" můžeme doplnit o některé chybějící formáty (například horní a dolní index) tak, že text vytvoříme ve Wordu a do textového pole Derive vložíme pomocí klávesových zkratek Ctrl+C (kopírovat), Ctrl+V (vložit). Toto vše nám umožňuje vytvářet v programu Derive **kompletní dokumenty**, například zadání samostatných prací, podklady pro vyučovací hodinu, vypracování laboratorní práce apod. Program umožňuje uložit dokument ve formátu RTF (Rich Text Format) (posloupností příkazů **Soubor** → **Exportovat** → **RTF...**). Tak nejsme vázáni jenom na prostředí Derive, ale můžeme dokument otevřít a dále upravovat třeba ve Wordu případně převést do formátu PDF. Bohužel, dle mé zkušenosti se při převodu do formátu RTF ztratí vložené OLE objekty. Pak nezbývá nic jiného, než je do RTF souboru vložit znovu.

Různé formáty můžeme volit i v případě ukládání grafů z 2D (3D) grafického okna. Program dovoluje měnit pomocí příkazu **Možnosti** různé parametry algebraického okna a tím i výsledného dokumentu. Pro příklad uveďme skrytí označení výrazů, tj. čísel (přesněji řečeno proměnných) ve tvaru #n na začátku každého řádku (Obr. 3).



Obr. 3: Možnosti okna Algebra

Nutno přiznat, že možnosti tvorby dokumentů v Derive mají své hranice. Je to pochopitelné. Program Derive slouží především k výpočtům, nikoliv k editaci dokumentů. Navíc pořád pracuje v řádkovém režimu (tj. objekty jsou řazeny svisle pod sebou) a neumožňuje, jako třeba program Mathcad, přemisťování výrazů po ploše a jejich řazení ve vodorovném směru.

Přesto můžeme i v Derive vytvořit dokument, například pracovní list, který má přijatelnou grafickou i informační úroveň, je kompaktní a zachovává si vlastnosti okna Algebra, tj. možnost bezprostředního počítání nebo grafického znázornění.

Jako příklad uvádím na následující straně pracovní list věnovaný křivce řetězovka. Textové části dokumentu i matematické výrazy jsou psány rovnou v Derive. K jejich vhodnému umístění ve vodorovném směru můžeme využít příkaz **Možnosti** → **Zobrazení** → **Zarovnání nových objektů...**

Student dokument otevře v Derive a přímo v něm nebo v dalších oknech řeší užitím nástrojů programu dané úkoly (Obr. 4). Nakonec výsledek uloží ve formátu DFW nebo RTF.

Řetězovka

PROBLÉM: Kabel elektrického vedení je mezi dvěma sloupy vzdálenými od sebe 100 metrů prověšen o 10 metrů. Vypočítejte, jak dlouhý je kabel mezi těmito dvěma sloupy?

Nápověda: Ohebné vlákno (tedy i kabel el. vedení) volně zavěšené ve dvou bodech, téže nebo i různé výšky, zaujímá tvar křivky, která se nazývá **řetězovka** (angl. Catenary).

Řetězovka má rovnici:

$$y(x) := \frac{a\cdot\left(e^{x/a} + e^{-x/a}\right)}{2}$$

kde **a je konstanta, jejíž hodnota závisí na fyzikálních vlastnostech vlákna a jeho zavěšení**, konkrétně na délkové hustotě vlákna a tahové síle, s kterou je nataženo. Rovnici řetězovky můžeme zapsat i pomocí funkce hyperbolický kosinus:

$$y(x) := a\cdot \mathrm{COSH}\left(\frac{x}{a}\right)$$

ÚKOLY:

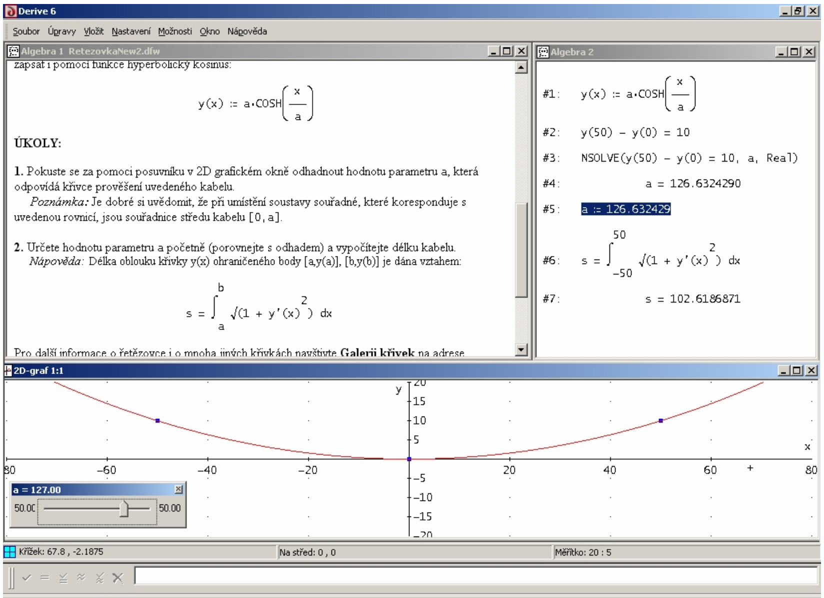
1. Pokuste se za pomoci posuvníku v 2D grafickém okně odhadnout hodnotu parametru a, která odpovídá křivce prověšení uvedeného kabelu.
Poznámka: Je dobré si uvědomit, že při umístění soustavy souřadné, které koresponduje s uvedenou rovnicí, jsou souřadnice středu kabelu [0,a].

2. Určete hodnotu parametru a početně (porovnejte s odhadem) a vypočítejte délku kabelu.
Nápověda: Délka oblouku křivky y(x) ohraničeného body [a,y(a)], [b,y(b)] je dána vztahem:

$$s = \int_a^b \sqrt{1 + y'(x)^2}\, dx$$

Pro další informace o řetězovce i o mnoha jiných křivkách navštivte **Galerii křivek** na adrese

<http://curvebank.calstatela.edu/home/home.htm>



Obr. 4: Užití pracovního listu Řetězovka

6. Import dat

Derive umožňuje načtení dat z tzv. datového souboru. Jedná se o textový soubor, v němž jsou data uspořádána do tabulky (Příklad 6.1). Takový soubor vytvoříme například v poznámkovém bloku, jako oddělovače na řádku použijeme mezeru nebo tabulátor. Můžeme pracovat i s daty z tabulkového procesoru, když je uložíme jako textový soubor (s výše uvedenými oddělovači). Program sice očekává u takovéhoto souboru příponu DAT, ale lze načíst i soubory s příponami TXT či PRN. Příponu můžeme navíc jednoduše přepsat provedením změny jména souboru. Pro využití ve třídě, například pro sběr dat při laboratorních pracích, se jeví jako užitečná možnost vytvořit datový soubor v **grafické kalkulačce** (TI 89, TI 92+, Voyage 200) a do počítače přenést pomocí datového kabelu. Ten se dá k uvedeným kalkulačkám dokoupit. Po načtení datového souboru posloupností příkazů **Soubor** → **Importovat** → **Datový soubor…** se data zobrazí jako prvky matice v okně Algebra.

Příklad 6.1: Do okna Algebra načtěte naměřená data ze souboru *Data.dat*, graficky je zobrazte a proložte jimi funkci, která co nejlépe popisuje zkoumanou závislost

Řešení:

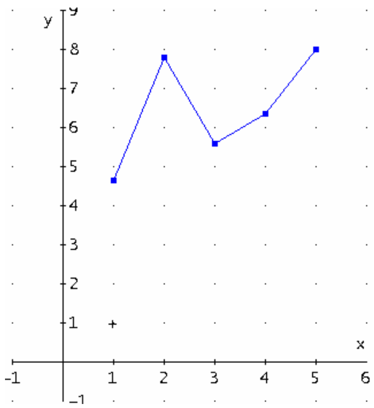
Data, která představují výsledky měření jakési závislosti, jsou výše uvedeným způsobem uložena do souboru *Data.dat* ve formě takovéto tabulky:

1	4.65
2	7.8
3	5.6
4	6.34
5	8

Po načtení souboru posloupností příkazů **Soubor** → **Importovat** → **Datový soubor…** se v okně algebra objeví následující matice.

$$\#1: \quad \begin{bmatrix} 1 & 4.65 \\ 2 & 7.8 \\ 3 & 5.6 \\ 4 & 6.34 \\ 5 & 8 \end{bmatrix}$$

Tu pak obvyklým způsobem zvýrazníme a zobrazíme v 2D grafickém okně (viz kapitola 9, Tvorba grafu). Graf na obrázku (Obr. 5) odpovídá nastavení voleb **Spojovat** - **Ano** a **Velikost** - **Velká** na kartě **Možnosti** → **Zobrazení** → **Body**.



Obr. 5: Grafické znázornění dat

Takto importovaná data můžeme v Derive statisticky zpracovat či dále použít. Přitom jistě oceníme příkazy pro přístup k prvkům matice (viz Nápověda a kapitola 18.2, Užití matic...) a statistické funkce programu Derive (`AVERAGE`, FIT, `STDEV` atd., viz Nápověda, heslo „statistické funkce“).

Naměřenými hodnotami proložíme polynomickou funkci čtvrtého stupně užitím funkce FIT. Pro další zpracování je vhodné matici dat pojmenovat, tj. uložit ji do proměnné, například `M`:

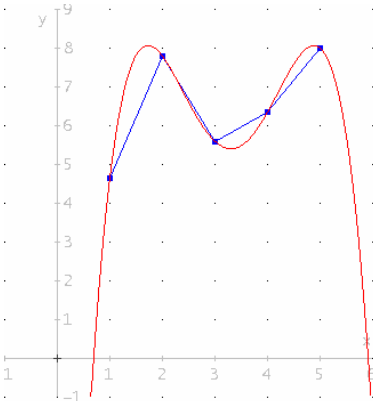
$$\#2: \quad M := \begin{bmatrix} 1 & 4.65 \\ 2 & 7.8 \\ 3 & 5.6 \\ 4 & 6.34 \\ 5 & 8 \end{bmatrix}$$

Použijeme funkci FIT (syntaxe a příklady použití viz Nápověda):

$$\#3: \quad \mathrm{FIT}\left(\left[x,\ a\cdot x^4 + b\cdot x^3 + c\cdot x^2 + d\cdot x + e\right],\ M\right)$$

a výsledný polynom zobrazíme v 2D-grafickém okně:

$$\#4: \quad -\frac{1031\cdot x^4}{2400} + \frac{2271\cdot x^3}{400} - \frac{62401\cdot x^2}{2400} + \frac{19141\cdot x}{400} - \frac{449}{20}$$



Poznámka: Funkci `FIT` můžeme použít i k prokládání ploch danými body v trojrozměrném prostoru.

7. Příkazy

Zadání příkazu je v programu Derive chápáno jako zápis výrazu. Chceme-li, aby program zadaný příkaz vykonal, musíme jemu odpovídající výraz **zjednodušit** (tj. použijeme $\boxed{=}$ nebo **Zjednodušit → Základní zjednodušení**). Program Derive podporuje dvojí způsob **zadávání příkazů**. Buď příkaz zadáme prostřednictvím dialogových oken programu, nebo ho napíšeme na příkazový řádek.

7.1. Zadání výběrem z nabídky

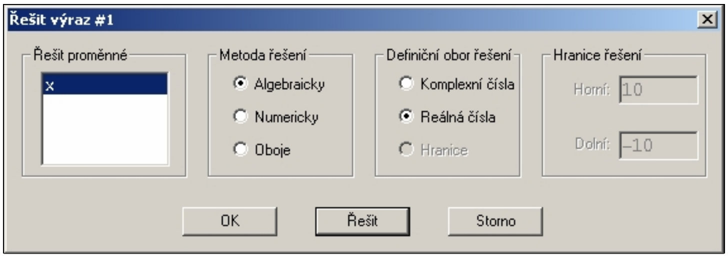
Jak ukazuje následující příklad, nemusíme znát příslušné příkazy. Stačí z nabídky rozhraní vybrat požadovanou akci a program nás sám vede.

Příklad 7.1: V R řešte rovnici $x^2 - 2x - 3 = 0$

Řešení:

Rovnici napíšeme na příkazovém řádku ve tvaru x^2-2x-3=0 a stisknutím (¢)(Enter) odešleme na pracovní plochu.

V nabídce potvrdíme příkaz **Řešit**, v příslušném roletovém menu pak potvrdíme **Výraz...** (zkráceně **Řešit → Výraz...**). Otevře se následující dialogové okno



Obr. 6: Řešení rovnice

V něm zvolíme metodu řešení a definiční obor a potvrdíme tlačítko Řešit. Chceme-li znát výsledek i ve formě desetinného čísla, jednoduše zvýrazníme na pracovní ploše výsledek algebraického řešení rovnice a stiskneme tlačítko $\boxed{\approx}$ (Aproximovat) na horní liště tlačítek.

Možnost řešit úlohy výběrem z nabídky je pohodlnější a přibližuje program uživateli. Pokrývá však jen část funkcí programu. Ten je **mnohem mocnějším nástrojem**, než by se zdálo při pouhém prozkoumání nabídky příkazů jeho rozhraní. Řadu speciálních příkazů, například na řešení diferenciálních rovnic, počítání s čísly, zkoumání geometrických vztahů apod., prostě musíme na příkazový řádek napsat. Množství funkcí programu můžeme posoudit prolistováním obsahu Nápovědy (**Nápověda → Témata nápovědy → Obsah**).

Kromě vestavěných funkcí pracuje program Derive se systémem **balíčků funkcí** (**Nápověda → Témata nápovědy → Uživatelské balíčky matematických**

funkcí), což jsou soubory typu MTH (soubory obsahující funkce naprogramované přímo v Derive), které se **automaticky** načítají do paměti, jakmile je volána některá z funkcí v nich definovaných.

7.2. Zadání zápisem na příkazový řádek

Tato druhá možnost zadávání příkazů je rychlejší, pokud známe syntaxi příkazu. Výše uvedenou rovnici jsme například mohli vyřešit zápisem příkazu `SOLVE(x^2–2x–3=0,x,Real)` na příkazovém řádku, jeho odesláním na pracovní plochu a následným zjednodušením. Ještě rychleji by to šlo odesláním příkazu ve tvaru `SOLVE(x^2–2x–3=0,x,Real)=` (¢) (všimněte si znaménka = na konci). Jak bylo již řečeno, někdy je zadání příkazu jedinou cestou k získání výsledku, protože příslušná funkce není „pokryta“ grafickým rozhraním. Jednou z takových funkcí je `GCD(n1, n2, …, nk)` pro nalezení největšího společného dělitele čísel n_1, n_2, …, n_k.

Příklad 7.2: *Určete největšího společného dělitele čísel 754 a 221*

Řešení:

Na příkazový řádek napíšeme příkaz `GCD(754,221)=` a odešleme ho na pracovní plochu stisknutím (¢).

8. Deklarace proměnných

Při tvorbě vlastních příkazů a vůbec při práci na nějakém rozsáhlejším projektu v Derive nám mohou značně zkomplikovat život nějaká zapomenutá nastavení hodnot a oborů proměnných, které nevědomky použijeme znovu za jiným účelem. Těmto komplikacím, které jinak mohou přerůst až v naprostou ztrátu důvěry v sebe i v program, se vyhneme dodržováním následujících zásad. Pokud to jde, používáme **vždy nový soubor**, který načítá firemní nastavení z inicializačního souboru Derive6.ini. Pokud to nejde, hlídáme si již použité proměnné. K ověření jejich **hodnot a oborů** použijeme volby **Průvodce → Hodnota proměnné...** a **Průvodce → Definiční obor proměnné...**. Konečně se nabízí postup běžný z programování. Každou proměnnou, kterou chci použít, **deklarovat**, tj. vymezit její obor, případně hodnotu. Pokud např. chceme, aby proměnná **A** byla bez hodnoty, tj. prázdná, použijeme **prázdný přiřazovací příkaz A:=** . Jsme-li tedy na konci nějakého rozsáhlejšího souboru a chceme použít proměnnou **A**, jejímž obsahem si nejsme jisti, provedeme nejprve tento příkaz.

9. Tvorba grafu

V programu Derive je možno vedle symbolických a numerických výpočtů kreslit dvourozměrné a třírozměrné grafy. Jednoduché propojení algebraických a grafických funkcí umožňuje bezprostřední grafickou reprezentaci většiny zobrazitelných formulí. Zobrazit lze množiny bodů dané předpisem i parametricky. Můžeme volit z nabídky několika souřadnicových soustav.

Grafy budeme hojně využívat v celé knize. Rozhodně však nepůjde o vyčerpávající přehled všech možností programu. Ten získá čtenář při podrobném studiu nápovědy programu spolu s připojenými balíčky funkcí. Grafickým možnostem Derive se podrobně věnuje i kniha [10].
V této kapitole si na dvou příkladech ukážeme základní algoritmus (posloupnost úkonů) tvorby 2D a 3D grafů, na který se budeme v dalším odkazovat.

9.1. 2D grafy

Příklad 9.1: Znázorněte graficky hodnoty výrazu $\sqrt{2|x|-x^2}$

Naformátováno: Řádkování: jednoduché, Ohraničení: Pole: (stínované jednoduché, Automatická, 1 b. šířka čáry)

Řešení:

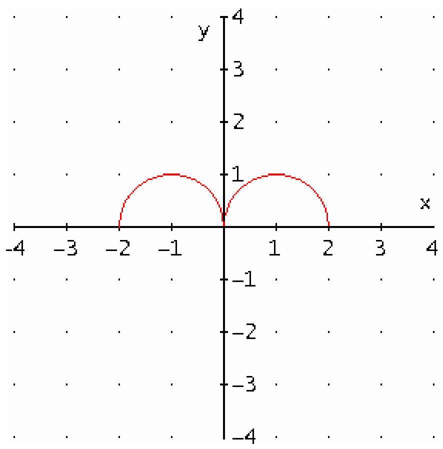
Zadáme výraz:
√(2|x|−x^2) (¢)
V okně Algebra ho zvýrazníme:

```
            2
#1:  √(2·|x| - x )
```

Kliknutím na tlačítko otevřeme 2D grafické okno.

Pro větší názornost můžeme rozdělit pracovní plochu na dvě části, jednu pro okno 2D (3D) grafu, druhé pro okno Algebra. Použijeme posloupnost příkazů **Okno** → **Vertikální (Horizontální) dlaždice.**

Při aktivním 2D grafickém okně (má modrý horní pruh a nabídka příkazů je jiná) ještě jednou stiskneme tlačítko (teď je trochu jinde, protože je jiná nabídka příkazů). Objeví se graf.



Poznámky:

1) Pokud není příslušné okno aktivní, umístíme do něj ukazatel myši a stiskneme levé tlačítko.
2) Tlačítko z panelu nástrojů grafického okna představuje zkrácenou formu posloupnosti příkazů **Vložit** → **Graf**. V následujícím příkladu uvidíme, že u 3D grafů je mezi těmito dvěma způsoby zobrazení rozdíl. Pokud totiž místo tlačítka použijeme uvedenou posloupnost příkazů, můžeme hned nastavit parametry vznikajícího grafu (3D).
3) Abychom dostali graf složený skutečně ze dvou půlkružnic, je třeba posloupností příkazů grafického okna **Nastavení** → **Poměr stran** → **Obnovit** → **OK** zajistit, aby nebylo grafické okno v jednom směru „protažené“.

9.2. 3D grafy

PŘÍKLAD 9.2: Znázorněte graficky plochu danou předpisem $z = y^6 - x^2 + x^6$

Řešení:

Výraz zadáme a v okně Algebra zvýrazníme (můžeme zadat celý předpis $z = y^6 - x^2 + x^6$, jak je uvedeno níže, nebo jenom pravou stranu $y^6 - x^2 + x^6$):

```
        6    2    6
#1:  z = y  - x  + x
```

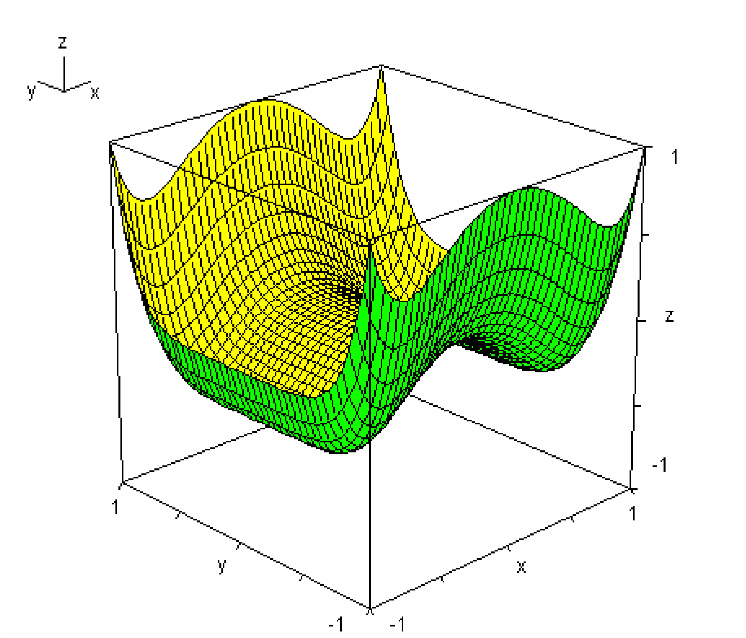
Kliknutím na tlačítko otevřeme 3D grafické okno.

Máme dvě možnosti jak zobrazit graf, které se u 3D grafů liší:

i) Použijeme opět ikonu . Dostaneme graf, jehož parametry odpovídají implicitnímu nastavení nebo parametrům předchozího grafu..
ii) Použijeme posloupnost příkazů 3D grafického okna **Vložit** → **Graf**. To nám umožňuje ovlivnit parametru grafu před jeho vykreslením.

Poznámka:
Při zápisu výrazu jsme mohli použít i jiná jména proměnných. Musíme však dbát na zachování abecedního pořadí, tj. trojici proměnných [x, y, z] můžeme nahradit trojicí [u, v, w], v uvedeném pořadí, a dostaneme stejný graf. Více viz **Nápověda** → **Index**, heslo „pořadí proměnných“.

Výsledek zobrazení v 3D grafickém okně:



9.3. Tvorba grafu - shrnutí

Tvorba 2D (3D) grafu

Grafické znázornění nějakého **výrazu** (výsledek rovnice či nerovnice, předpis funkce, podmínky definující relaci, …) dostaneme následujícím způsobem:

1) Výraz v okně Algebra zvýrazníme.

2) Klikneme na ( ), objeví se okno 2D (3D) grafu.

3) Při aktivním grafickém okně klikneme ještě jednou na ( ) (tlačítko je teď na jiném místě) nebo z nabídky vybereme posloupnost příkazů **Vložit** → **Graf**. Nakreslí se graf.

4) Pro větší názornost můžeme rozdělit pracovní plochu na dvě části, jednu pro okno 2D (3D) grafu, druhé pro okno Algebra: **Okno** → **Vertikální (Horizontální) dlaždice**.

Naformátováno: zarovnání na střed, Ohraničení: Pole: (jednoduché, Automatická, 0,25 b. šířka čáry)

Program dovoluje zobrazit jenom **část výrazu** (když jí zvýrazníme). Tím je usnadněno **grafické řešení rovnic a nerovnic**. Jednoduše zobrazíme do jednoho grafického okna zvlášť levou a zvlášť pravou stranu rovnice.
Grafické řešení pak můžeme přenést na pracovní plochu do okna Algebra příkazem **Soubor → Přemístit**. Tento příkaz patří do nabídky grafického okna, můžeme ho tedy vybrat a provést jenom při aktivním grafickém okně. Okna aktivujeme poklepáním myší.
Při **změně velikosti** grafu přeneseného do okna Algebra se zhoršuje kvalita obrázku. **Původní kvality**, při nové velikosti, dosáhneme příkazem **Soubor → Aktualizovat** z nabídky grafického okna.

Doporučení

Pro plné využití grafů je dobré vědět, co všechno u nich můžeme měnit a nastavit. V některých situacích oceníme například možnost změny barvy os a jejich popisu, případně možnost umístění obrázku na pozadí grafu a podobně. Doporučuji proto detailně prostudovat všechny karty dialogového okna **Možnosti → Zobrazení…** z nabídky příkazů 2D (3D) grafického okna. Až budete experimentovat se změnami parametrů grafu, mějte na paměti, že **každá změna se projeví až po zopakování vložení grafu**, tj. po opětovném stisknutí tlačítka ().

9.4. Změna typu čáry 2D grafu.

U spojitých dvourozměrných grafů Derive nenabízí žádnou možnost měnit tloušťku a typ čáry. Jinak je tomu u grafů bodových (viz **Nápověda → Index**, hesla „bodový graf“, „bodové grafy“ a „parametrické grafy“), u kterých jisté možnosti změny parametrů jsou. Lze u nich nastavit tři velikosti bodů, počet bodů a zda je chceme spojovat či nikoliv a jakým typem čáry.
Chceme-li tedy měnit tloušťku a typ čáry grafu, znázorníme prostě spojitou křivku bodovým grafem. Při dostatečně velkém počtu bodů (bohužel, maximum je 640) získáme iluzi spojité křivky. Přitom můžeme docílit tří různých tloušťek „čáry“, dle volby jedné ze tří velikostí bodů (Malá, Střední, Velká). Pokud množství bodů vhodně snížíme, dostaneme čáru tečkovanou.

Dalším způsobem, jak změnit typ čáry grafu (nabízí se plná, tečkovaná, čárkovaná a čerchovaná), ovšem bez možnosti změnit její tloušťku, je nastavení režimu spojování bodů a typu příslušné čáry na kartě **Možnosti → Zobrazení → Body** (viz obrázek 8).

Vedle uvedených možností změny tloušťky a typu čáry grafu se z následujících ukázek ještě dozvíme, jak kreslit graf funkce na omezeném intervalu. Více o těchto grafech je pak uvedeno v kapitole 13 (Funkce).

Ukázky konkrétních řešení

Tři různé způsoby zobrazení grafu funkce, lišící se od výše uvedeného běžného způsobu, si ukážeme na příkladu funkce dané předpisem $f: y = x^2 - 3$.

a) Tlustá čára

Použijeme parametrický graf (parametrem bude x) v režimu **Body** (Obr. 7).

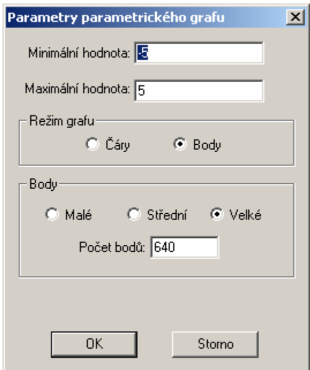
Zadáme předpis funkce:

```
               2
#1:   f(x) := x  - 3
```

Potom zapíšeme křivku grafu parametricky, s parametrem x:

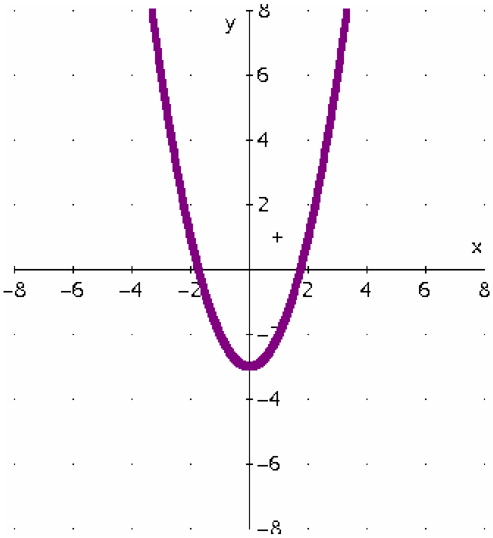
```
#2:   [x, f(x)]
```

Zadáme-li nyní pokyn k zobrazení výrazu #2 v 2D grafickém okně, program sám identifikuje tento výraz jako parametrický a zobrazí dialogové okno „Parametry parametrického grafu“ k upřesnění některých údajů. V okně zaškrtneme režim **Body**, vybereme velikost `Velké`, zadáme jejich počet (640) a potvrdíme (`OK`).



Obr. 7: Parametry grafu

Výsledný graf funkce f(x), vyvedený tlustou čarou, vidíme na následující straně.

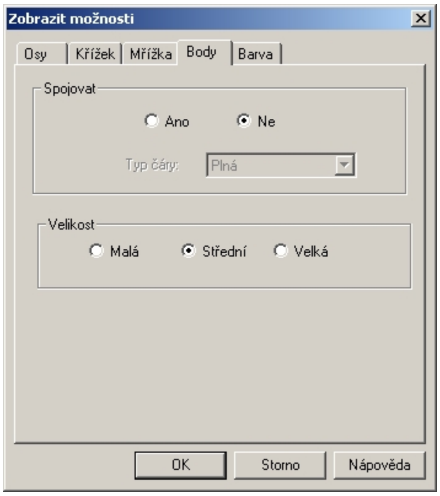


b) Středně tlustá čára
Tentokrát použijeme bodový graf zadaný výčtem bodů.

K získání tohoto výčtu bodů, přesněji vektoru bodů, použijeme funkci `VECTOR` (nebo `TABLE`, viz další ukázka). Podrobné informace o této funkci najdeme v Nápovědě. Zde si jenom připomeňme význam pátého parametru této funkce (viz následující výraz #3), který představuje krok, s nímž se mění hodnota řídící proměnné (v našem případě je řídící proměnnou x). Jeho velikostí ovlivníme hustotu bodů grafu.

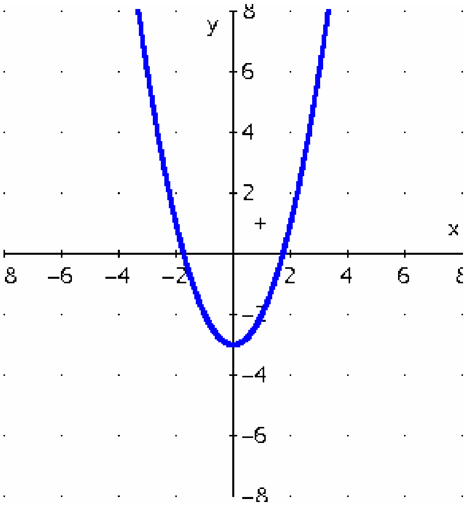
```
#3:    VECTOR([x, f(x)], x, -5, 5, 0.01)
```

Chceme-li čáru střední tloušťky, zatrhneme na kartě **Možnosti → Zobrazení → Body** (**Obr. 8**) v poli **Spojovat** volbu **Ne** a v poli **Velikost** volbu **Střední**.



Obr. 8: Možnosti zobrazení

Výraz #3 musí být před zobrazením v 2D grafickém okně zjednodušen (základní zjednodušení). Máme dvě možnosti, buď ho zjednodušíme v algebraickém okně a výsledek zobrazíme, nebo aktivujeme volbu **Možnosti → Zjednodušit před vykreslením** a zobrazíme rovnou #3.



Poznámka: Stejného efektu dosáhneme, když místo funkce VECTOR použijeme funkci TABLE. Tak tomu bude v následující ukázce.

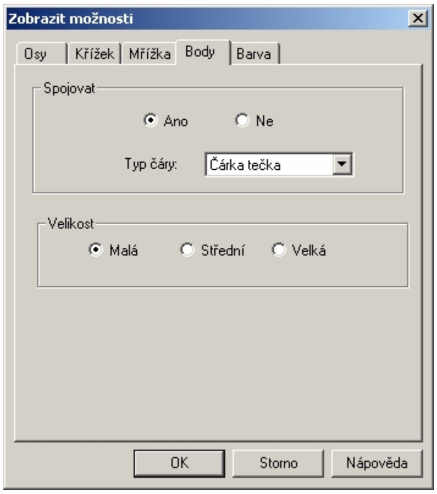
c) Změna typu čáry
Opět použijeme bodový graf. Tentokrát však budeme jeho body spojovat. Využijeme totiž toho, že to lze provést čtyřmi různými typy čar.

Zadáme funkci f(x) předpisem

```
                 2
#1:    f(x) := x   - 3
```

K vytvoření posloupnosti bodů použijeme tentokrát funkci TABLE (více o této funkci viz Nápověda)

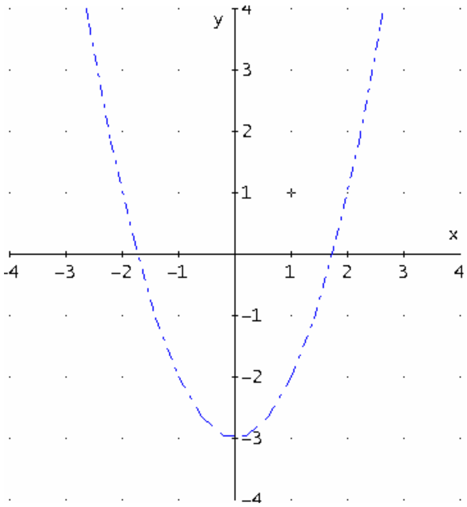
```
#2:    TABLE(f(x), x, -5, 5, 0.4)
```



Obr. 9: Možnosti zobrazení

Pro zobrazení výrazu #2 opět platí pravidla zmíněná v předchozí ukázce. Buď výraz zjednodušíme (základní zjednodušení) v algebraickém okně a výsledek zobrazíme, nebo aktivujeme volbu **Možnosti → Zjednodušit před vykreslením** a zobrazíme rovnou #2.

Výsledkem je následující graf.



Poznámka: Místo funkce TABLE jsme mohli použít i funkci VECTOR. Stejným způsobem, jako v předchozí ukázce.

Užití příkazů TABLE a VECTOR při kreslení částí grafů funkcí různými typy čar si ilustrujme ještě jedním příkladem.

Příklad 9.3: Sestrojte v R^2 graf relace S:

$$S = \{[x, y]; y \geq \log_2 x \wedge y < 2^x \wedge x \geq 1 \wedge y < -x + 6\}.$$

Řešení:

```
                            x
#1:    y ≥ LOG(x, 2) ∧ y < 2   ∧ x ≥ 1 ∧ y < −x + 6

#2:    x = 1
```

K numerickému výpočtu x-ových souřadnic průsečíků hraničních křivek, které jsou dány jednotlivými podmínkami, použijeme funkci `NSOLUTIONS`. Takto získané hodnoty pak použijeme jako meze příkazů `TABLE`. Hodnotou funkce `TABLE` je tabulka (matice) bodů, která se dá zobrazit jako bodový graf. Nezapomeňme na to, že výraz `TABLE` musíme před zobrazením zjednodušit. Máme dvě možnosti.

1. Provedeme základní zjednodušení výrazu `TABLE` jeho zvýrazněním a stisknutím $\boxed{=}$. Výsledek pak zobrazíme v 2D grafickém okně.
2. Otevřeme 2D grafické okno a zatrhneme volbu **Možnosti → Zjednodušit před vykreslením**. Potom zobrazíme rovnou příkaz `TABLE`, bez jeho zjednodušování.

Efektu plné nebo tečkované čáry docílíme vhodnou volbou velikosti kroku v příkazu

`TABLE` (je to jeho pátý parametr). Jakou hodnotu zvolit závisí na konkrétní úloze. V našem případě dostaneme plnou čáru při velikosti kroku 0.01 (viz #3), tečkovanou pak při velikosti kroku 0.1 (viz #4, #5).

```
#3:
TABLE(LOG(x, 2), x, 1, (NSOLUTIONS(LOG(x, 2) = -x + 6, x))  , 0.01)
                                                          1
```

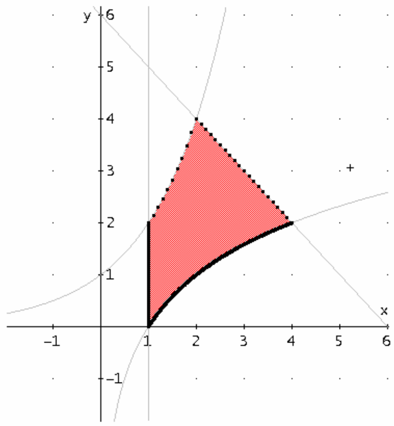
$$\#4:\ \mathrm{TABLE}\left(2^x,\ x,\ 1,\ (\mathrm{NSOLUTIONS}(2^x = -x + 6,\ x))_1,\ 0.1\right)$$

$$\#5:\ \mathrm{TABLE}\left(-x + 6,\ x,\ (\mathrm{NSOLUTIONS}(2^x = -x + 6,\ x))_1,\ (\mathrm{NSOLUTIONS}(-x + 6 = \mathrm{LOG}(x,\ 2),\ x))_1,\ 0.1\right)$$

Pro zobrazení části přímky x = 1 použijeme funkci VECTOR

```
#6:   VECTOR([1, y], y, 0, 2, 0.01)
```

Předpisy pro zobrazení grafů funkcí, které tvoří hranice množiny, najdeme například jako první parametry funkcí `TABLE` (#3, #4, #5). Jejich barvu volíme světle šedou.



Další tipy užitečné pro tvorbu speciálních grafů najdeme v kapitolách věnovaných relacím a funkcím. Konkrétně se jedná o grafy relací, funkcí definovaných na intervalu a funkcí definovaných po částech. V kapitole 16 (Integrál) potom zmíníme a v několika příkladech použijeme speciální funkce Derive pro kreslení obrazců omezených danými křivkami (příkladem může být funkce `AreaBetweenCurve`).

10. Programování

Derive umožňuje uživateli vytvářet vlastní **funkce**. Disponuje programovacím jazykem, který je poměrně jednoduchý a nabízí pouze několik obvyklých příkazů a syntaktických pravidel. V této kapitole se při řešení konkrétních úloh s většinou z nich seznámíme. Pro získání podrobnějších informací o programování v Derive doporučuji čtenáři prostudování nápovědy programu (**Nápověda → Obsah → Vybraná témata → Programování v Derive**) spolu se studiem zdrojových kódů uživatelských funkcí, které jsou součástí instalace programu (viz adresář Users). Z internetových zdrojů dostupných ke dni vydání této knihy bych zmínil stránku www.cms.livjm.ac.uk/deriveprogramming.

10.1. Definice jednoduché funkce

Za nejjednodušší program můžeme pokládat definici funkce bez použití speciálních příkazů programovacího jazyka.

Příklad 10.1: Definice funkce `VzdBp(D,a,b,c)` pro výpočet vzdálenosti bodu `D` od přímky zadané obecnou rovnicí `ax+ by+ c = 0`.

Řešení:
Definici funkce zapíšeme na příkazovém řádku ve tvaru:
`VzdBp(D,a,b,c):=|aD↓1+bD↓2+c|/sqrt(a^2+b^2)`,
kde `D↓1`, `D↓2` jsou zápisy první a druhé souřadnice bodu `D`. Po odeslání klávesou Enter dostaneme na pracovní ploše zápis, v němž poznáváme příslušný vzorec:

$$\mathrm{VzdBp}(D,\ a,\ b,\ c) := \frac{|a\cdot D_1 + b\cdot D_2 + c|}{\sqrt{(a^2 + b^2)}}$$

10.2. Prostředí pro zápis programu

Jsou možné dva způsoby zadávání programu – řádkový režim nebo strukturovaný zápis ve víceřádkovém režimu. Výstup do algebraického okna je vždy strukturován do více řádků. Mezi jednořádkovým a víceřádkovým (čtyřřádkovým) oknem pro vkládaní a úpravu výrazů přepínáme příkazem **Možnosti → Zobrazení → Zadání víceřádkových výrazů.** Ve 4-řádkovém okně můžeme pomocí šipkových kláves pohybovat kurzorem nahoru, dolů, doleva i doprava. Nový řádek vložíme pomocí kombinace kláves Alt + Enter.

10.3. Speciální funkce pro tvorbu a ladění programu

Programovací jazyk Derive používá několik následujících funkcí, které odpovídají běžným příkazům známým i z jiných programovacích jazyků.

Složený příkaz **PROG**

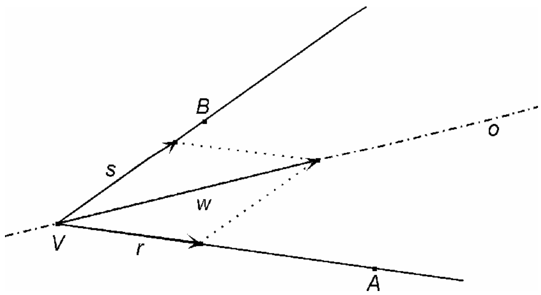
Syntaxe: PROG(příkaz 1, příkaz 2, …, příkaz n)

Výstupem složeného příkazu je hodnota posledního příkazu v něm obsaženého, pokud ovšem uvnitř nepoužijeme příkazy EXIT nebo RETURN (více v Nápovědě).

Příklad 10.2: Definice funkce `OsaUhlu(A,V,B)`, jejíž hodnotou je parametrické vyjádření osy úhlu daného vrcholem V a body A, B na ramenech.

Řešení:

Směrovým vektorem $\vec{w}$ osy úhlu je součet jednotkových směrových vektorů $\vec{r}=\dfrac{(A-V)}{|A-V|}$, $\vec{s}=\dfrac{(B-V)}{|B-V|}$ jeho ramen, $\vec{w}=\vec{r}+\vec{s}$.



Parametrické vyjádření osy má potom tvar: $X=V+t\cdot\vec{w}$.

Zápis definice funkce v jednořádkovém režimu:

```
OsaUhlu(A, V, B, r, s, w) := Prog(r := (A - V)/ABS(A - V),s := (B -
V)/ABS(B - V), w := r + s,V + t·w)
```

Zápis ve víceřádkovém režimu:

```
OsaUhlu(A, V, B, r, s, w) :=
        Prog(
          r := (A - V)/ABS(A - V),
          s := (B - V)/ABS(B - V),
          w := r + s,
          V + t·w)
```

Poznámka: Kromě použití příkazu PROG nám výše uvedený příklad ukazuje způsob **deklarace lokálních proměnných** v programovacím jazyku Derive. Vidíme, že lokální proměnné `r, s, w` jsou uvedeny v závorce za jménem spolu se vstupními parametry `A, V, B` programu.

Podmíněný příkaz **IF**

Syntaxe: IF(test, potom, jinak, neznámo)

Příklad 10.3: Definice funkce `Matice(M)`, jejíž hodnotou je informace o tom, zda je matice M regulární či singulární. Pokud není čtvercová, nebo nelze určit hodnotu determinantu, výsledkem je slovo „nelze”.

Řešení:

Zápis v jednořádkovém režimu:

```
Matice(M):=IF(DET(M)≠0,WRITE("regulární"),WRITE("singulární"),WRITE
("nelze"))
```

Zápis ve víceřádkovém režimu:

```
Matice(M) :=
  IF(DET(M) ≠ 0,
     WRITE("regulární"),
     WRITE("singulární"),
     WRITE("nelze"))
```

Cyklus **LOOP**

Syntaxe: LOOP(příkaz 1, příkaz 2, …, příkaz n)

Příkaz LOOP ukončí svou činnost, pokud narazí na příkaz EXIT nebo RETURN (více viz nápověda). Kombinací příkazu LOOP s podmíněným příkazem IF můžeme realizovat příkazy cyklu s podmínkou na začátku (tj. příkaz typu While) nebo na konci (tj. příkaz typu Repeat … Until).

Příklad 10.4: Funkce `Del2(x)` nejprve posoudí paritu vstupní hodnoty `x`. Pokud je liché, funkce vypíše hlášení na obrazovku a ukončí činnost. Pokud je `x` sudé, funkce provádí jeho opakované dělení dvěma, dokud není výsledek lichý. Potom opět vypíše hlášení a ukončí činnost. Dílčí výsledky dělení se vypisují na obrazovku.

Řešení:

Zápis v jednořádkovém režimu:

```
Del2(x) := Loop(If(MOD(x, 2) ≠ 0, RETURN "liché"), x :/ 2,
DISPLAY(x))
```

Zápis ve víceřádkovém režimu:

```
Del2(x) :=
    Loop(
      If( MOD(x, 2) ≠ 0,
          RETURN "liché"),
      x :/ 2,
      DISPLAY(x))
```

Poznámka: Všimněme si zápisu `x :/ 2` pro dělení hodnoty proměnné `x` dvěma. Derive používá podobné operátory i pro ostatní způsoby změny hodnoty proměnné (více viz nápověda, kapitola „Funkce pro tvorbu a ladění programu“).

Příklad 10.5: Funkce `OdhadPi(n)` je realizací Archimédovy metody určení přibližné hodnoty π pomocí obvodů n-úhelníků vepsaných a opsaných kružnici. Vstupním údajem `n` je počet desetinných míst, na která chceme určit hodnotu π. Výstupem, tj. hodnotou funkce, je informace o počtu úhlů použitého pravidelného k-úhelníka a hodnoty příslušného dolního a horního odhadu π.

Řešení:

Zatímco v předchozím příkladě jsme využili cyklus s podmínkou na začátku (tj. While), nyní bude podmínka pro ukončení na konci cyklu. Jedná se tedy o analogii cyklu Repeat. Program už je trochu složitější. Uvedeme proto jenom jeho zápis při aktivním režimu zadávání víceřádkových výrazů:

```
OdhadPi(n, k := 5, d, h) :=
     Loop(
       d := k·SIN(π/k),
       h := k·TAN(π/k),
       If(FLOOR(d·10^n) = FLOOR(h·10^n),
             Prog(
               Precision := Approximate,
               DISPLAY(APPEND(STRING(k), "-úhelník")),
               RETURN APPROX([d, h], 10))),
       k :+ 1)
```

Syntaxe funkce OdhadPi(n) a její výstup:

```
#2:   OdhadPi(2)

57-úhelník

#3:                          [3.14000234, 3.144777633]

#4:   OdhadPi(4)

1187-úhelník

#5:                          [3.141588985, 3.141599989]
```

<u>Ukončení běhu **RETURN, EXIT**</u>. <u>Výstup **WRITE, DISPLAY**</u>

Význam těchto příkazů je zřejmý z ukázek.

Další funkce, které mohou najít uplatnění při tvorbě programu v Derive, najde čtenář v nápovědě programu. Patří mezi ně například funkce **ASSIGN**, **ITERATE**, **ITERATES**, **VECTOR**. Jinak při psaní programu můžeme samozřejmě využít všech dostupných funkcí a operátorů programu.

10.4. Ukládání programů - funkcí

Naprogramované funkce je nejlepší uložit do souboru typu .mth (**Soubor → Uložit jako ... → Math soubor (*.mth))**. Vhodné je do jednoho souboru uložit funkce tématicky příbuzné a soubor („balíček funkcí“) dle onoho tématu pojmenovat. Chceme-li potom některou funkci ze souboru - balíčku použít, musíme ho, na rozdíl od vestavěných balíčků funkcí, načíst do paměti příkazem **Soubor → Importovat → Balíček funkcí...(Matematický soubor...)**.
V případě, že použijeme termín **Balíček funkcí...**, definice funkcí ze souboru se natáhnou do paměti počítače a na pracovní ploše programu Derive se objeví pouze hlášení podobné tomuto: `LOAD(C:\Program Files\TI Education\Derive 6\Math\EnglishUnits.mth).` Pokud budeme importovat **Matematický soubor...**, nahrají se definice do paměti i na pracovní plochu programu.

10.5. Další příklady programů

Uveďme si ještě dva příkladů programů vytvořených v Derive. Další příklady pak najdeme roztroušeny po celé knize.

Příklad 10.6: Napište program pro řešení kvadratické rovnice v oboru komplexních čísel.

Řešení:

```
       KvadratickaRovnice(rovnice, promenna) :=
         Prog
           r := rovnice
           p := promenna
           If POLY_DEGREE(r, p) ≠ 2
              RETURN "Rovnice není kvadratická"
              Prog
                s := SOLVE(r, p, Complex)
                a := POLY_COEFF(r, p, 2)
                b := POLY_COEFF(r, p, 1)
                c := POLY_COEFF(r, p, 0)
#1:             d := b^2 - 4·a·c
                If d < 0
                   Prog
             DISPLAY("Rovnice má dva komplexně sdružené kořeny:")
                     RETURN s
                   If d = 0
                      Prog
                   DISPLAY("Rovnice má dvojnásobný reálný kořen:")
                       RETURN s
                      Prog
                   DISPLAY("Rovnice má dva různé reálné kořeny:")
                       RETURN s

                                  2
#2:    KvadratickaRovnice(3·x  - 4·x + 5, x)
```

Rovnice má dva komplexně sdružené kořeny:

#3: $$x = \frac{2}{3} - \frac{\sqrt{11}\cdot i}{3} \lor x = \frac{2}{3} + \frac{\sqrt{11}\cdot i}{3}$$

#4: KvadratickaRovnice($3\cdot x^2 - 4\cdot x - 5$, x)

Rovnice má dva různé reálné kořeny:

#5: $$x = \frac{2}{3} - \frac{\sqrt{19}}{3} \lor x = \frac{\sqrt{19}}{3} + \frac{2}{3}$$

#6: KvadratickaRovnice($x^2 - 4\cdot x + 4$, x)

Rovnice má dvojnásobný reálný kořen:

#7: $$x = 2$$

#8: KvadratickaRovnice($3\cdot x^3 - 4\cdot x + 5$, x)

#9: Rovnice není kvadratická

Příklad 10.7: Napište program pro určení vzájemné polohy dvou přímek v prostoru .

Řešení:

#1: $$\frac{|(A - B)\cdot \mathrm{CROSS}(u, v)|}{|\mathrm{CROSS}(u, v)|}$$

#2: $$\mathrm{VzdBp}(D, a, b, c) := \frac{|a\cdot D_1 + b\cdot D_2 + c|}{\sqrt{(a^2 + b^2)}}$$

#3: $$\mathrm{VzdMimo}(A, u, B, v) := \frac{|(A - B)\cdot \mathrm{CROSS}(u, v)|}{|\mathrm{CROSS}(u, v)|}$$

#4: $$\mathrm{VzdBpP}(A, B, u) := \frac{\sqrt{(|A - B|^2\cdot u^2 - ((A - B) \bullet u)^2)}}{|u|}$$

```
      PrusPrP(A, u, B, v, k, l, p) :=
        Prog
#5:       p := (SOLUTIONS(A + k·u = B + l·v, [k, l]))↓1↓1
          MAP_LIST(SUBST(x, k, p), x, A + k·u)


      VzajemnaPoloha2(A, u, B, v) :=
        Prog
          If RANK([u, v]) = 1
             Prog
               DISPLAY("Rovnoběžky, vzdálenost = ")
               RETURN VzdBpP(A, B, v)
#6:          If RANK([u, v, B - A]) = 2
                Prog
                  DISPLAY("Různoběžky, společný bod =")
                  RETURN PrusPrP(A, u, B, v)
                Prog
                  DISPLAY("Mimoběžky, vzdálenost = ")
                  RETURN VzdMimo(A, u, B, v)
```

#7:VzajemnaPoloha2([0, 0, 0], [2, 3, -5], [0, 0, 7], [-4, -6, 10])

Rovnoběžky, vzdálenost =

#8: $$\frac{7\cdot\sqrt{494}}{38}$$

#9:VzajemnaPoloha2([0, 0, 0], [2, 3, -5], [0, 0, 7], [-3, 2, 1])

Mimoběžky, vzdálenost =

#10: $$\frac{7\cdot\sqrt{3}}{3}$$

#11: VzajemnaPoloha2([1, 2, 4], [2, 3, -5], [0, 0, 7], [-3, 2, 1])

Různoběžky, společný bod =

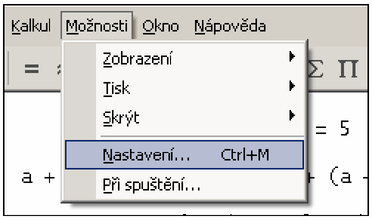
#12: $$\left[-\frac{3}{13}, \frac{2}{13}, \frac{92}{13}\right]$$

II. ŘEŠENÍ ÚLOH

Tato druhá, rozsáhlejší, část knihy je věnována řešení konkrétních úloh užitím Derive. Řešení úloh jsou okomentována. Komentáře mají různou úroveň podrobnosti. Tam, kde je použito nových funkcí a postupů, které nebyly předvedeny v první části, je komentář náležitě zevrubný. V ostatních případech je komentář více nebo méně stručnější, jak se očekává čtenářova zběhlost v užití programu. Snadnější orientaci napomáhají odkazy na části textu, které se příslušnému tématu věnují podrobněji nebo na nápovědu programu.

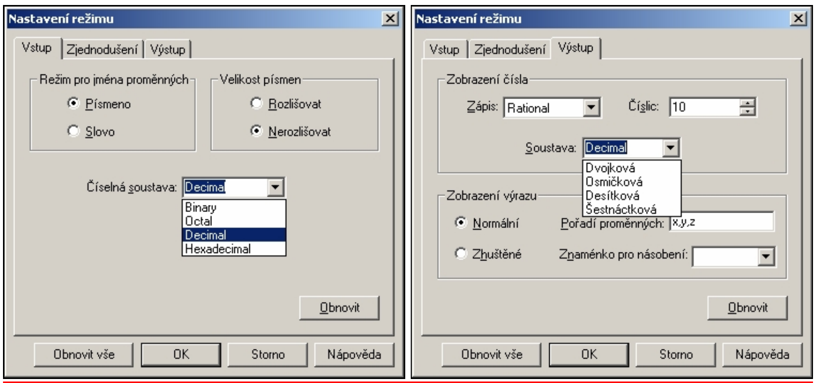
11. Úpravy matematických výrazů

Pro **nastavení vstupních a výstupních podmínek, definic a pravidel** budeme hojně využívat nabídku **Možnosti → Nastavení...** (Obr. 10).



Obr. 10: Nastavení režimu práce

Naformátováno: Písmo: 11
Naformátováno: Písmo: 11
Naformátováno: Písmo: 11 b., není Tučné
Naformátováno: Písmo: 11 b., není Tučné



Obr. 11 a 12: Nastavení vstupních a výstupních podmínek

Naformátováno: Písmo: 11
Naformátováno: Písmo: 11
Naformátováno: Písmo: 11 b., není Tučné
Naformátováno: Písmo: 11 b., není Tučné
Naformátováno: Písmo: 11 b.

Podívejme se znovu na Obr. 11. Kromě volby základu číselné soustavy zápisu vkládaných hodnot tam najdeme volbu **Režim** pro jména proměnných. Zvolíme-li **Písmeno**, program uvažuje každé písmeno jako zvláštní proměnnou. To nám umožňuje, stejně jako při zápisu na papír, **vynechávat symbol násobení** (∗). Program chápe při nastavení volby **Písmeno** zápis `abc` jako výraz `a∗b∗c`. Zvolíme-li **Slovo**, chápe zápis `abc` jako symbol jedné proměnné. Volba **Velikost písmen** ovlivňuje, zda záleží (**Rozlišovat**) či nezáleží (**Nerozlišovat**) na velikosti písmene (tj., zda X, x znamená to samé, či nikoliv). Pokud nastavíme **Rozlišovat**, je třeba vědět, že jména vestavěných funkcí Derive se zapisují **velkými písmeny**.

11.1. Algebraické výrazy

Při úpravách algebraických výrazů využijeme volby **Zjednodušit** → **Základní zjednodušení**, **Roznásobit...**, **Rozložit na činitele...**. Význam voleb **Typ rozkladu...** v příslušných dialogových oknech ilustruje následující příklad. Zároveň vidíme způsob zadání funkce `FACTOR` ve tvaru `FACTOR(f(x), Typ rozkladu, x)`. Typy rozkladu jsou definovány klíčovými slovy programu Derive, proto musí být v případě přímého zadání příkazu uvedeny v originálním znění (**Trivial, Squarefree, ...**).

Příklad 11.1: Rozložte v součin polynom:

$$v(x) = 3x^9 + 6x^8 - 6x^7 - 6x^6 - 15x^5 - 48x^4 + 66x^3 + 72x^2 - 72x$$

Řešení:

#1: $v := 3\cdot x^9 + 6\cdot x^8 - 6\cdot x^7 - 6\cdot x^6 - 15\cdot x^5 - 48\cdot x^4 + 66\cdot x^3 + 72\cdot x^2 - 72\cdot x$

#2: `FACTOR(v, Trivial, x)`

#3: $3\cdot x\cdot(x^8 + 2\cdot x^7 - 2\cdot x^6 - 2\cdot x^5 - 5\cdot x^4 - 16\cdot x^3 + 22\cdot x^2 + 24\cdot x - 24)$

#4: `FACTOR(v, Squarefree, x)`

#5: $3\cdot x\cdot(x^2 + x - 2)^2\cdot(x^4 + x^2 - 6)$

#6: `FACTOR(v, Rational, x)`

#7: $3\cdot x\cdot(x - 1)^2\cdot(x + 2)^2\cdot(x^2 - 2)\cdot(x^2 + 3)$

#8: `FACTOR(v, Radical, x)`

#9: $3\cdot x\cdot(x - 1)^2\cdot(x + 2)^2\cdot(x + \sqrt{2})\cdot(x - \sqrt{2})\cdot(x^2 + 3)$

#10: `FACTOR(v, Complex, x)`

#11: $3\cdot x\cdot(x - 1)^2\cdot(x + 2)^2\cdot(x + \sqrt{2})\cdot(x - \sqrt{2})\cdot(x + \sqrt{3}\cdot i)\cdot(x - \sqrt{3}\cdot i)$

Naformátováno: Řádkování: jednoduché, Ohraničení: Pole: (stínované jednoduché, Automatická, 1 b. šířka čáry)

Příklad 11.2: Zjednodušte lomený výraz $\dfrac{2x-1}{x+1}$.

Řešení:

Zjednodušením se v tomto případě rozumí rozklad lomeného výrazu na součet konstanty a zlomku s konstantním čitatelem. Tento rozklad je obvyklý při kreslení grafu racionální lomené funkce. V Derive je náležitá akce ošetřena funkcí EXPAND, která je v nabídce **Zjednodušit** uvedena pod heslem **Roznásobit...**
Postupujeme tedy tak, že výraz zvýrazníme a provedeme posloupnost příkazů **Zjednodušit** → **Roznásobit...** , potvrdíme tlačítkem (Roznásobit)

#1: $\dfrac{2\cdot x - 1}{x + 1}$

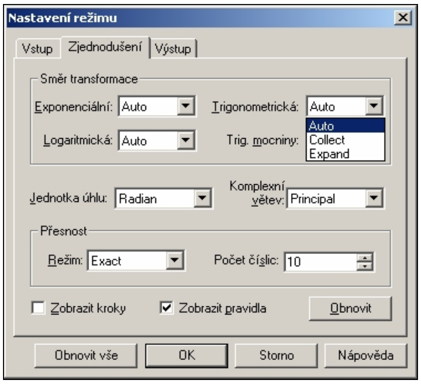
#2: $\mathrm{EXPAND}\left(\dfrac{2\cdot x - 1}{x + 1}, \mathrm{Trivial}, x\right)$

#3: $2 - \dfrac{3}{x + 1}$

Naopak, pro převedení zlomku na společného jmenovatele použijeme příkaz FACTOR (**Rozložit na činitele...**).

11.2. Goniometrické výrazy

Chceme-li upravovat goniometrické výrazy, můžeme (někdy musíme) sdělit programu, jakou úpravu požadujeme. Program sice „zná“ goniometrické vzorce, ale „neví“, jakým směrem je chceme uplatnit. To mu sdělíme volbou **Možnosti** → **Nastavení...** → **Zjednodušení** (Obr. 10), která otevře dialogové okno (Obr. 13),



Obr. 13: Nastavení režimu zjednodušování

v němž jsou nabídnuty různé možnosti a režimy úprav výrazů:

Trigonometrická → Auto, Collect, Expand
- určuje směr využití goniometrických vzorců,

Trig. mocniny → Auto, Cosines, Sines
- reprezentace sudých mocnin gon. funkcí,

Jednotka úhlu → Radian, Degree
- jednotka velikosti úhlu,

Komplexní větev → Principal, Real, Any
- volba větve řešení odmocniny v oboru komplexních čísel.

Poslední volbu oceníme při výpočtu „liché“ odmocniny ze záporného čísla, například $\sqrt[3]{-8}$, nebo při grafickém znázornění „liché“ odmocniny, více v kapitole 13.7, Graf iracionální funkce.

Příklad 11.3: Zjednodušte výraz $\dfrac{(\sin a + \sin b)^2}{\cos^2 a - \cos^2 b}$

Řešení:

#1: $$\frac{(\mathrm{SIN}(\alpha) + \mathrm{SIN}(\beta))^2}{\mathrm{COS}(\alpha)^2 - \mathrm{COS}(\beta)^2}$$

#2: $$\mathrm{FACTOR}\left(\frac{(\mathrm{SIN}(\alpha) + \mathrm{SIN}(\beta))^2}{\mathrm{COS}(\alpha)^2 - \mathrm{COS}(\beta)^2}, \mathrm{Radical}, \alpha, \beta\right)$$

#3: $$\frac{\mathrm{SIN}(\alpha) + \mathrm{SIN}(\beta)}{\mathrm{SIN}(\beta) - \mathrm{SIN}(\alpha)}$$

Příklad 11.4: Rozložte výraz $\sin(x+y)$

Řešení:
Výraz budeme upravovat v režimu „zobrazit krok“ (režim není pro řešení úlohy nezbytný, zde je použit na ukázku), který nastavíme takto:
Možnosti → Nastavení... → Zjednodušení → Zobrazit kroky

Pokračujeme rozkladem výrazu. Nejprve nastavíme požadovaný směr transformace:
Možnosti → Nastavení... → Zjednodušení → Trigonometrická → Expand
a potom teprve daný výraz podrobíme základnímu zjednodušení:

```
#1:   Trigonometry := Expand

#2:   SIN(x + y)

SIN(z + w) → SIN(z)·COS(w) + COS(z)·SIN(w)

#3:                  SIN(x)·COS(y) + COS(x)·SIN(y)
```

Naformátováno: Písmo: 11 b.

Naformátováno: Písmo: 11 b.

Příklad 11.5: Upravte výraz $\sin(x)^2 + 2\cos(x)^2 - 1$.

Řešení:
Tentokrát je jedno, zda ponecháme originální nastavení trigonometrických transformací **Auto** nebo využijeme výše uvedený režim **Expand**.

$$\#1:\quad \mathrm{SIN}(x)^2 + 2\cdot\mathrm{COS}(x)^2 - 1$$

$$\mathrm{SIN}(z)^2 + \mathrm{COS}(z)^2 \rightarrow 1$$

$$\#2:\quad \mathrm{COS}(x)^2$$

Naformátováno: Písmo: 11

Naformátováno: Ohraničení: Nahoře: (bez ohraničení), Dole: (bez ohraničení), Vlevo: (bez ohraničení), Vpravo: (bez ohraničení)

Naformátováno: Písmo: 11 b.

Příklad 11.6: Zjednodušte výraz $\cos^2 x - \sin^2 x$

Naformátováno: Řádkování: jednoduché, Ohraničení: Pole: (stínované jednoduché, Automatická, 1 b. šířka čáry)

Řešení:
Výraz se dá samozřejmě upravit na cos2x. Pokud ale ponecháme originální nastavení trigonometrických transformací **Auto**, tento výsledek, jak vidíme v následující ukázce, nedostaneme.

$$\#1:\quad \mathrm{COS}(x)^2 - \mathrm{SIN}(x)^2$$

$$\mathrm{SIN}(z)^2 + \mathrm{COS}(z)^2 \rightarrow 1$$

$$\#2:\quad 2\cdot\mathrm{COS}(x)^2 - 1$$

Naformátováno: Písmo: 11

Naformátováno: Písmo: 11 b.

Je totiž nutné nastavit režim (**Collect**):
Možnosti → Nastavení... → Zjednodušení → Trigonometrická → Collect

Potom znovu označíme výraz `#1` a opakovaně (protože máme nastaven režim **Zobrazit kroky**) provádíme jeho základní zjednodušení:

`cos(x)^2-sin(x)^2` $\boxed{=}$ $\boxed{=}$

```
#3:   Trigonometry := Collect
```

`COS(z + π) → - COS(z)`

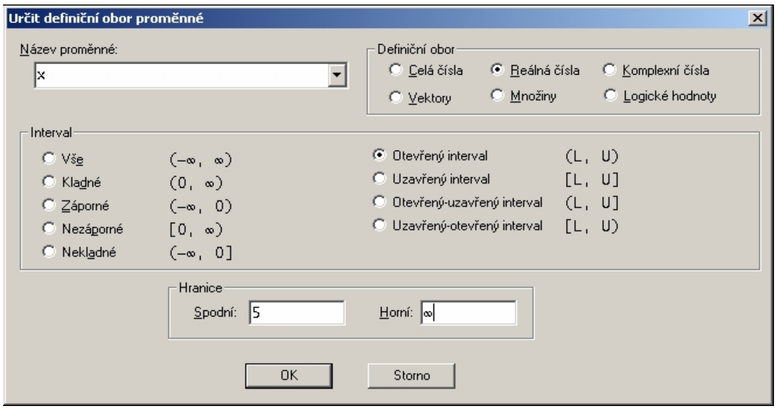
$$\#4: \quad \frac{\mathrm{COS}(2\cdot x)}{2} + \frac{1}{2} - \mathrm{SIN}(x)^2$$

$$\#5: \quad \mathrm{COS}(2\cdot x)$$

Poznámka: Nesmíme zapomenout, že při nastaveném režimu **Zobrazit kroky** odpovídá **jednomu zjednodušení**, tj. jednomu stisknutí $\boxed{=}$, **jeden krok**. Proto je většinou potřeba tlačítko $\boxed{=}$ mačkat opakovaně (viz řešení předchozího příkladu).

11.3. Logaritmické výrazy

`LN(x)` nebo `LOG(x)` jsou zápisy **přirozeného logaritmu** čísla *x*. Logaritmus *x* při základu *z* zapisujeme ve tvaru `LOG(x,z)`. Při úpravách logaritmických výrazů může hrát roli definiční obor logaritmické funkce. Program tuto skutečnost zohledňuje a proto musí některým úpravám předcházet náležité vymezení definičních oborů příslušných proměnných. K tomu používáme posloupnost příkazů
Průvodce → Definiční obor proměnné ... (Obr. 14)
Stejně jako u goniometrických můžeme i u logaritmických výrazů specifikovat způsob úpravy volbami **Možnosti → Nastavení... → Zjednodušení → Logaritmická → Auto, Collect, Expand**



Obr. 14: Definiční obor proměnné … $x \in (5, \infty)$

Naformátováno: Písmo: 11 b.

Naformátováno: Písmo: 11

Naformátováno: Písmo: 11

Naformátováno: Písmo: 11 b., není Tučné

Příklad 11.7: Rozložte na součet logaritmů výrazy $\ln(xy)$, $\ln((x-5)(y-10))$.

Řešení:

Aby se logaritmus součinu rozložil na součet logaritmů, musíme náležitě nastavit způsob (směr) úprav logaritmických výrazů a určit definiční obor alespoň jednoho z činitelů.

Naformátováno: Zarovnat do bloku

```
#1:    DisplaySteps := true

#2:    LOG(x·y)

LOG(z) → LN(z)

#3:                                    LN(x·y)

#4:    Logarithm := Expand

#5:    x :∈ Real (0, ∞)
```

Naformátováno: Písmo: (výchozí) Times New Roman, 11 b.

Naformátováno: Písmo: 11 b.

Naformátováno: Písmo: 11 b., Barva písma: Modrá

Protože máme nastaven režim **Zobrazit kroky**, zjednodušení se provede ve dvou krocích. Z komentáře řešení (máme zapnutou i volbu **Zobrazit pravidla**) je vidět, že program bere při úpravě v potaz definiční obor jedné z proměnných (`If x>0 …`).

Naformátováno: Zarovnat do bloku

```
LOG(z) → LN(z)

#6:                                    LN(x·y)

If x>0,

LN(x·z) → LN(x) + LN(z)

#7:                                 LN(x) + LN(y)

#8:    LN((x - 5)·(y - 10))

#9:    x :∈ Real (5, ∞)

If x>0,

LN(x·z) → LN(x) + LN(z)

#10:                           LN(x - 5) + LN(y - 10)
```

Naformátováno: Písmo: 11 b.

Přestože přistupuje Derive k úpravám logaritmických výrazů takto „rozumně“, má definice logaritmu v tomto programu jistá nečekaná specifika, o kterých je dobré vědět. Jedná se o zápis logaritmu a jeho definiční obor při řešení rovnic.

Symbolický zápis logaritmu: Funkce `LOG(x)` a `LN(x)` jsou identické, jedná se o přirozený logaritmus. Logaritmus o základu `n` zapíšeme výrazem `LOG(x,n)`. Dekadický logaritmus se tedy zadá výrazem `LOG(x,10)`.

Definiční obor: Logaritmy jsou v Derive definovány v oboru komplexních čísel a při řešení rovnic není bohužel možné nastavit obor rovnice tak, jak jsme to udělali při úpravě výše uvedeného výrazu. Dostáváme tak někdy výsledky, které nejsou řešením dané rovnice v oboru reálných čísel. Proto je nutné provádět v případě logaritmických a iracionálních rovnic zkoušku. Více v kapitole 17.6 (Logaritmické rovnice). Výše uvedené skutečnosti jsou ilustrovány následujícími třemi příklady výpočtu logaritmu záporného čísla.

#1: $$\mathrm{LOG}(-5) = \mathrm{LN}(5) + \pi\cdot i$$

#2: $$\mathrm{LN}(-5) = \mathrm{LN}(5) + \pi\cdot i$$

#3: $$\mathrm{LOG}(-1000,\ 10) = 3 + \frac{\pi\cdot i}{\mathrm{LN}(10)}$$

11.4. Úprava výrazu po krocích

Při řešení předchozího příkladu 11.7. jsme využili režimů **Zobrazit kroky** a **Zobrazit pravidla.** Úpravy po krocích můžeme provádět buď pomocí tlačítka [⌐], které najdeme na panelu nástrojů nebo tak, že potvrdíme volbu **Zobrazit kroky** na kartě **Možnosti** → **Nastavení...** → **Zjednodušení** (Obr. 13), kde najdeme i volbu **Zobrazit pravidla** prováděných úprav.

11.5. Číselné výrazy - přibližné a přesné hodnoty

Číselné výrazy mohou být reprezentovány jak přesnými zápisy (např. $2\hat{}(1/2)=\sqrt{2}$) tak i přibližnými hodnotami ($2\hat{}(1/2)=1.414213562$). Zápis desetinným číslem přitom může mít libovolný počet míst (jsme omezeni jenom možnostmi počítače).
Při úpravách výrazů tyto dvě možnosti volíme ve výše uvedeném pořadí pomocí příkazů (nebo pomocí tlačítek) **Zjednodušit** → **Základní zjednodušení** ([=]) a **Zjednodušit** → **Aproximovat...** ([≈]).
Pro trvalé nastavení režimu reprezentace hodnot použijeme volbu (Obr. 13) **Možnosti** → **Nastavení...** → **Zjednodušení** → **Přesnost** → **Režim** → **Approximate, Exact, Mixed** (v režimu **Mixed** se aproximují pouze iracionální čísla).
Počet míst číselného zápisu nastavíme opět na kartě **Zjednodušení**, tentokrát v poli **Počet číslic**: **Nastavení...** → **Zjednodušení** → **Přesnost** → **Počet číslic**. Tak můžeme zobrazit třeba π klidně i na 1000 desetinných míst apod. Konstanty jako je

Ludolfovo nebo Eulerovo číslo vkládáme z příslušného panelu matematických symbolů nebo použijeme kombinace kláves Ctrl + P (E).

Příklad 11.8: Jaké jsou poslední dvě číslice čísla 3^{1234} ?

Naformátováno: Řádkování: jednoduché, Ohraničení: Pole: (stínované jednoduché, Automatická, 1 b. šířka čáry)

Řešení:

```
       1234
#1:   3
```

Výraz #1 zvýrazníme a stiskneme tlačítko základního zjednodušení $\boxed{=}$

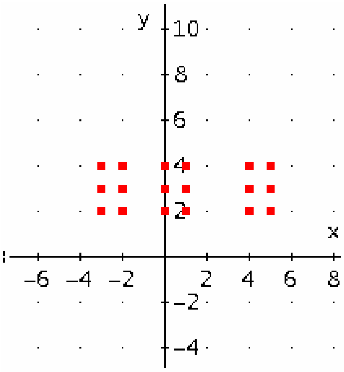
#2:
5856367529932071269049608726415028439757143623443063785468784843536946634255003858796842453459905186846046851774292445753149532138648409140561593823222428559200283816367593571556942770711203894947318922509986128995034233177140247221695941679146171215675158154358685224309073135481486540427499644363484815218991488230440687359021965115067678545212365725678242878485920885436927030372580147256775788882374301866845441065076726499613479868210410548366056401445912760529623658838510267592199677099676046090140514526228797525926977514932394954796457727416398081482661262807288229389463819882569

12. Binární relace

12.1. Kartézský součin

Kartézský součin množin A, B se v Derive zapíše jako běžný součin A*B. Grafické znázornění prvků kartézského součinu dostaneme pouhým zobrazením tohoto výrazu nebo výsledku jeho zjednodušení (tj. výsledné množiny uspořádaných dvojic) v 2D grafickém okně. Pro větší názornost obrázku doporučuji potvrdit na kartě **Možnosti** → **Zobrazení** → **Body** (viz Obr. 8) v poli **Velikost** volbu **Velká**.

```
#1: A ≔ {-3, 4, 0, -2, 1, 5}

#2: B ≔ {2, 3, 4}

#3: A·B = {[-3, 2], [-3, 3], [-3, 4], [-2, 2], [-2, 3], [-2, 4],
[0, 2], [0, 3], [0, 4], [1, 2], [1, 3], [1, 4], [4, 2], [4, 3],
[4, 4], [5, 2], [5, 3], [5, 4]}
```



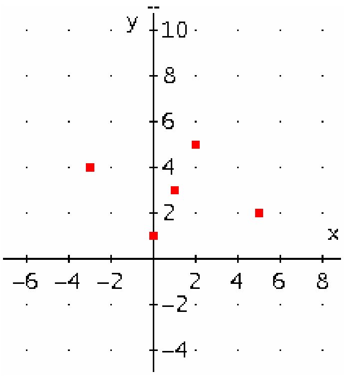
12.2. Zadání relace

Binární relaci můžeme zadat jako množinu uspořádaných dvojic. Z následující ukázky je patrné, že zadáme-li první prvek z uspořádané dvojice jako první index množiny dvojic, výsledkem je druhý prvek z dvojice. Bohužel, pokud je jeden prvek prvního oboru v relaci s více prvky druhého oboru, tj. v množině je několik uspořádaných dvojic se stejným prvním prvkem, výsledkem zmíněné operace je vždy druhý prvek z první takovéto dvojice uvedené v množině. Proto tuto operaci oceníme spíše v případě funkce zadané množinou uspořádaných dvojic, nikoliv relace.

#1: $S := \{[1, 3], [2, 5], [-3, 4], [0, 1], [5, 2]\}$

#2: $S_0 = 1$

#3: $S_{-3} = 4$



#4: $R := \{[2, a], [4, b], [2, c], [-1, a], [2, b]\}$

#5: $R_4 = b$

#6: $R_2 = a$

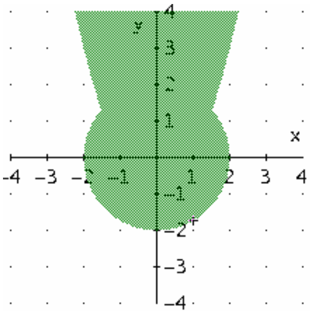
#7: $R := \{[2, b], [2, a], [4, b], [2, c], [-1, a]\}$

#8: $R_2 = b$

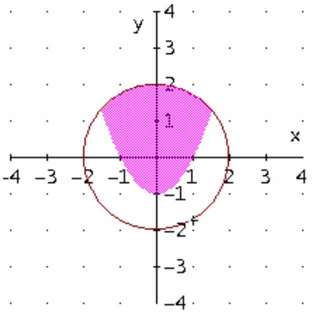
12.3. Kartézský graf binární relace

Program umožňuje graficky znázornit množiny bodů zadané různými podmínkami, které musí splňovat jejich souřadnice. Podmínky mají podobu nerovností či rovností a spojujeme je logickými spojkami.

#1: $y \geq x^2 - 1 \vee x^2 + y^2 < 4$



#2: $(y \geq x^2 - 1 \wedge x^2 + y^2 < 4) \vee x^2 + y^2 = 4$



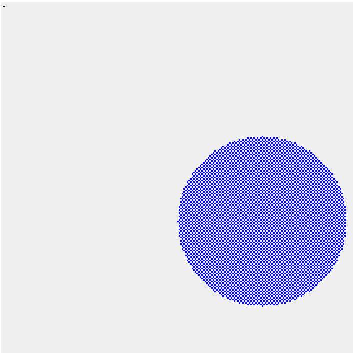
12.4. Grafické znázornění množinových operací

Jednotlivé nerovnosti (podmínky) můžeme pomocí přiřazovacích příkazů pojmenovat a pro jejich spojení použít množinové operace. Dostaneme tak prostředek pro znázornění množinových operací

Pokud chceme znázornit zápisy množinových operací (např. #3, #5) rovnou, bez jejich předchozího zjednodušení (základní), musíme v nabídce 2D grafického okna aktivovat volbu
Možnosti → Zjednodušit před vykreslením

Základní množiny M, N

#1: $M := (x - 2)^2 + (y + 1)^2 \le 4$

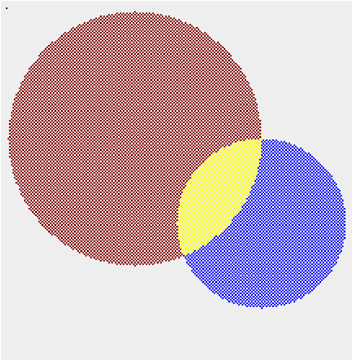


#2: $N := (x + 1)^2 + (y - 1)^2 \le 9$

Průnik

#3: M ∩ N

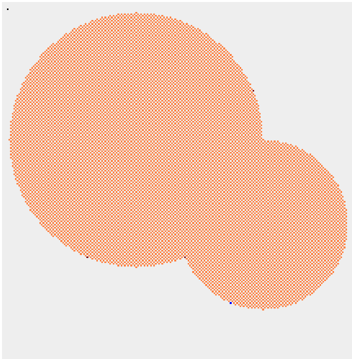
#4: $x^2 - 4\cdot x + y^2 + 2\cdot y \le -1 \wedge x^2 + 2\cdot x + y^2 - 2\cdot y \le 7$



Sjednocení

#5: M ∪ N

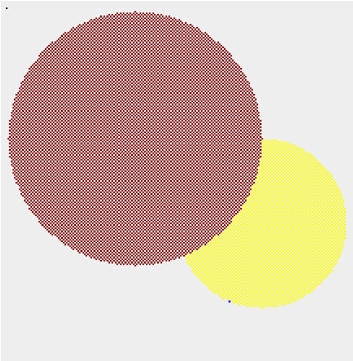
#6: $x^2 - 4\cdot x + y^2 + 2\cdot y \le -1 \vee x^2 + 2\cdot x + y^2 - 2\cdot y \le 7$



Rozdíl

#7: M \ N

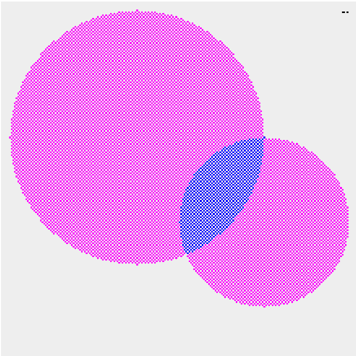
#8: $x^2 - 4\cdot x + y^2 + 2\cdot y \le -1 \wedge x^2 + 2\cdot x + y^2 - 2\cdot y > 7$



Symetrický rozdíl

#9: M \ N ∪ N \ M

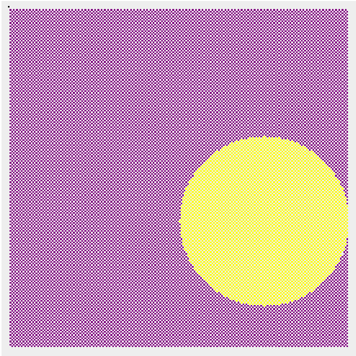
#10: $(x^2 - 4\cdot x + y^2 + 2\cdot y \le -1 \wedge x^2 + 2\cdot x + y^2 - 2\cdot y > 7) \vee$
$(x^2 + 2\cdot x + y^2 - 2\cdot y \le 7 \wedge x^2 - 4\cdot x + y^2 + 2\cdot y > -1)$



Doplněk

```
#11:   M`
```

$$x^2 - 4\cdot x + y^2 + 2\cdot y > -1$$



Bohužel, takto jednoduše nejde zobrazit každou podobně zadanou množinu. Problémy jsou s křivkami, tj. s množinami zadanými rovnicemi. Ty se buď zobrazí celé nebo vůbec. Chceme-li zobrazit část křivky, například jako hranici grafu binární relace, nelze použít symbol konjunkce (či průniku) pro stanovení požadované části. V takovém případě použijeme podmíněný příkaz IF (následující Příklad 12.1) nebo příkazy TABLE a VECTOR.

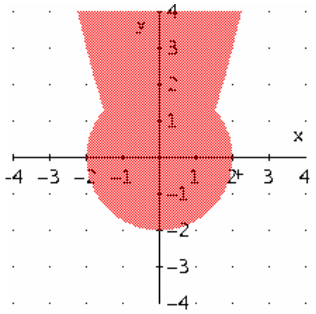
Příklad 12.1: Nakreslete hranici množiny A ∪ B. Množiny A, B jsou dány podmínkami: $A=\{[x,y]\in R^2; y\geq x^2-1\}$, $B=\{[x,y]\in R^2; x^2+y^2\leq 4\}$.

Řešení:

$$\#1:\quad A := y \geq x^2 - 1$$

$$\#2:\quad B := x^2 + y^2 \leq 4$$

$$\#3:\quad A \cup B$$



Pomocí příkazu SOLUTIONS určíme společné body hraničních křivek. Výhodou příkazu SOLUTIONS oproti příkazu SOLVE je to, že jeho výsledkem je vektor řešení příslušné rovnice nebo soustavy rovnic. Tato forma zápisu usnadňuje další manipulaci s výsledky. V našem případě, kdy řešíme soustavu dvou rovnic o dvou neznámých, dostaneme souřadnice hledaných bodů jako řádky matice Sp typu (2,2). Každá ze souřadnic je potom přístupna pomocí dvojice dolních indexů, jak je běžné u prvků takovéto matice (viz následující partie zápisu řešení)

#4: $y = x^2 - 1$

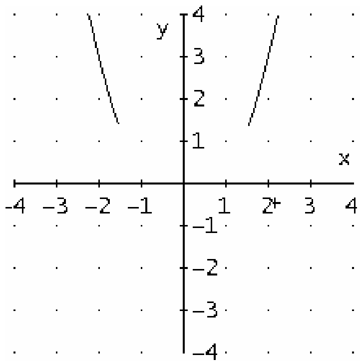
#5: $x^2 + y^2 = 4$

#6: $Sp := \mathrm{SOLUTIONS}([y = x^2 - 1, x^2 + y^2 = 4], [x, y], \mathrm{Real})$

#7: $$Sp := \begin{bmatrix} \dfrac{\sqrt{(2\cdot\sqrt{13} + 2)}}{2} & \dfrac{\sqrt{13}}{2} - \dfrac{1}{2} \\ -\dfrac{\sqrt{(2\cdot\sqrt{13} + 2)}}{2} & \dfrac{\sqrt{13}}{2} - \dfrac{1}{2} \end{bmatrix}$$

Souřadnice společných bodů pak použijeme v omezujících podmínkách pro zobrazení částí hraničních křivek, které tvoří hranice sjednocení množin A, B.

#8: $\mathrm{IF}\left(x > Sp_{1,1} \vee x < Sp_{2,1}, x^2 - 1\right)$



#9: $x^2 + y^2 = 4$

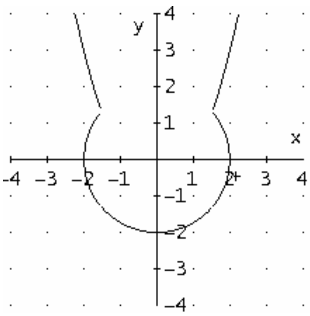
#10: $\text{SOLVE}(x^2 + y^2 = 4, y, \text{Real})$

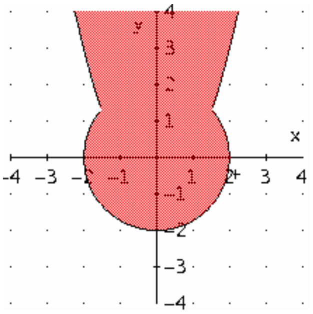
#11: $y = -\sqrt{(4 - x^2)} \vee y = \sqrt{(4 - x^2)}$

#12: $\text{IF}\left(x > Sp_{1,1} \vee x < Sp_{2,1}, -\sqrt{(4 - x^2)}\right)$

#13: $\text{IF}\left(x > Sp_{1,1} \vee x < Sp_{2,1}, \sqrt{(4 - x^2)}\right)$

#14: $\text{IF}\left(Sp_{2,1} < x < Sp_{1,1}, -\sqrt{(4 - x^2)}\right)$





Často jsou množiny, na nichž jsou zadány binární relace, zadány jako interval nebo výčtem prvků. Chceme-li takto zadané relace zobrazit, oceníme příkazy MAP a VECTOR. Jak ukazuje následující příklad, můžeme si užitím těchto příkazů definovat jednoduché procedury, jejichž zobrazením získáme požadované grafy.

Definice procedur pro zobrazení relací

Pro správné pochopení těchto procedur doporučuji prostudovat nápovědu k příkazům MAP_LIST a VECTOR

```
Relace_MM(A, B) := MAP_LIST(MAP_LIST([x, y], x, A), y, B)

Relace_MI(A, I) := MAP_LIST(VECTOR([x, y], y, I , I , 0.01), x, A)
                                               1   2

Relace_IM(I, A) := MAP_LIST(VECTOR([x, y], x, I , I , 0.01), y, A)
                                               1   2
Relace_II(I, J) :=
VECTOR(VECTOR([x, y], x, I , I , 0.05), y, J , J , 0.05)
                          1   2               1   2
```

Použití:
Předpokládejme, že uvedené procedury jsou uloženy v balíčku „`Relace.mth`“ (více o tvorbě balíčků viz kapitola 10.4. a nápověda programu). Nejprve je načteme do paměti

```
#1: LOAD(C:\Derive\Derive_Balicky\Relace.mth)
```

Než přistoupíme ke grafickému znázornění procedur, je vhodné provést následující nastavení 2D-grafického okna:

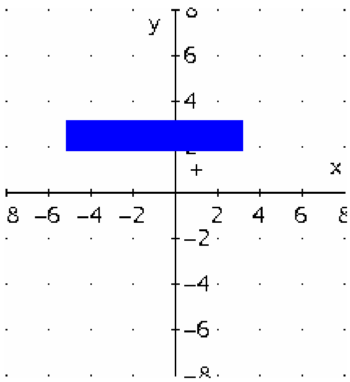
1) Abychom nemuseli zadání procedury zjednodušovat (základní zjednodušení) v okně Algebra, aktivujeme volbu:
Možnosti → Zjednodušit před vykreslením

2) Aby byly grafy jednobarevné, doporučuji vypnout režim automatického střídání barev grafů:
Možnosti → Zobrazení → Barva → Automaticky měnit barvu nových grafů NE

3) Pokud je graf tvořen izolovanými body, je třeba mít vypnuto spojování bodů
Možnosti → Zobrazení → Body → Spojovat NE

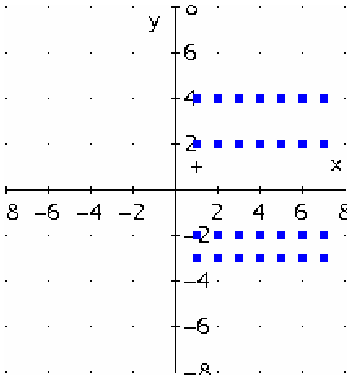
4) Pokud je součástí grafu souvislá plocha, je vhodné nastavit velkou velikost bodu. Pokud je i potom plocha kostkovaná, je třeba v odpovídající proceduře zmenšit velikost kroku v příkazu VECTOR.
Možnosti → Zobrazení → Body → Velikost Velká

```
#2: Relace_II([-5, 3], [2, 3])
```

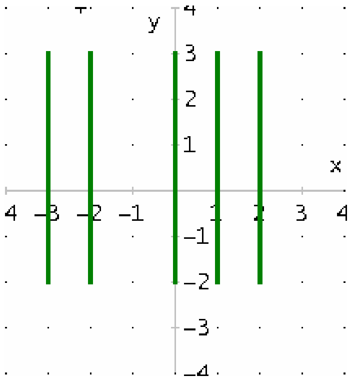



```
#3: Relace_MM({1, 2, ..., 7}, {-2, -3, 2, 4})
```

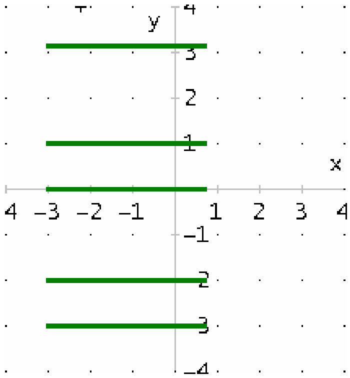
(Pozn.: V tomto případě ale nemusíme používat zvláštní proceduru,stačí provést a zobrazit kartézský součin množin výše uvedeným způsobem, viz kap ***12.1.****)*



#4: Relace_MI({-2, -3, 0, 1, 2}, [-2, 3])



#5: Relace_IM$\left(\left[-3, -\frac{5}{7}\right], \{-2, -3, 0, 1, \pi\}\right)$



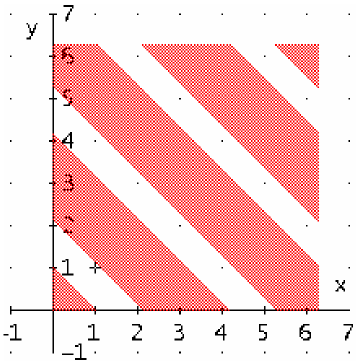
Příklad 12.2: Vyšetřete množinu bodů v rovině, jejichž pravoúhlé souřadnice *x*, *y* splňují soustavu daných nerovnic. Načrtněte obrázek.

Naformátováno: Ohraničení: Pole: (stínované jednoduché, Automatická, 1 b. šířka čáry)

a) $\left|\sin\left(x+y+\frac{p}{2}\right)\right| \ge \frac{1}{2} \wedge 0 \le x \le 2p \wedge 0 \le y \le 2p$

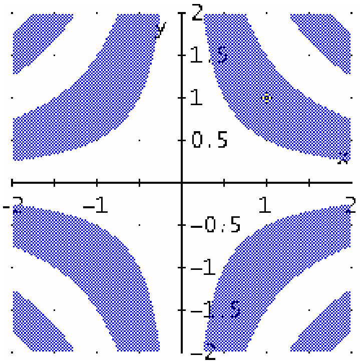
#1: $\left|\mathrm{SIN}\left(x + y + \frac{\pi}{2}\right)\right| \ge \frac{1}{2} \wedge 0 \le x \le 2\cdot\pi \wedge 0 \le y \le 2\cdot\pi$

#2: $\left(\mathrm{COS}(x + y) \le -\frac{1}{2} \vee \mathrm{COS}(x + y) \ge \frac{1}{2}\right) \wedge x \le 2\cdot\pi \wedge y \le 2\cdot\pi$
$\wedge\ x \ge 0 \wedge y \ge 0$



b) $|x| < 2 \wedge |y| < 2 \wedge \cos(p\,xy) \le 0$

#1: $|x| < 2 \wedge |y| < 2 \wedge \mathrm{COS}(\pi\cdot x\cdot y) \le 0$



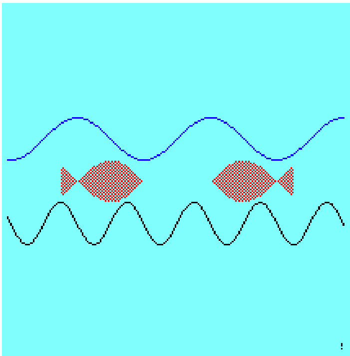
c) Setkání v akváriu

$$\left(|x| \geq \frac{p}{2} \wedge |x| \leq \frac{7p}{2} \wedge |y| \leq |\cos x|\right) \vee y = 2 + \sin x \vee y = -2 + \sin 2x$$

(Emil Calda: *Z umělecké dílny profesora Ypsilona*, Rozhledy M-F, č. 9-10, 1983)

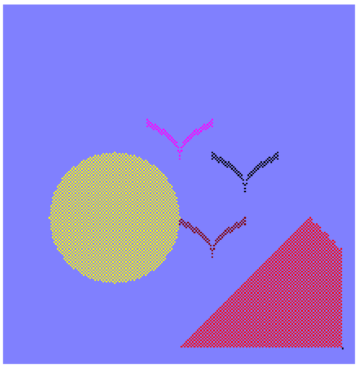
$$\#1:\ \left(|x| \geq \frac{\pi}{2} \wedge |x| \leq \frac{7 \cdot \pi}{4} \wedge |y| \leq |\mathrm{COS}(x)|\right) \vee y = 2 + \mathrm{SIN}(x)$$

$$\vee\ y = -2 + \mathrm{SIN}(2 \cdot x)$$



d) Měsíční noc

Opět relace z dílny profesora Ypsilona. Tentokrát se pokuste sami najít k obrázku odpovídající matematické vyjádření. Derive při tom může být dobrým pomocníkem.



13. Funkce

Pojem funkce se v Derive objevuje na dvou různých úrovních. Jednak jako funkce (přiznejme, že komplexní) jedné nebo více proměnných, jednak v obecnějším významu, jako akce na nějakých vstupních údajích, kterou si může uživatel nadefinovat sám (viz kapitola 10, Programování).
Zde, v této kapitole, budeme pojem funkce chápat oním prvním způsobem, jako funkci reálné proměnné.

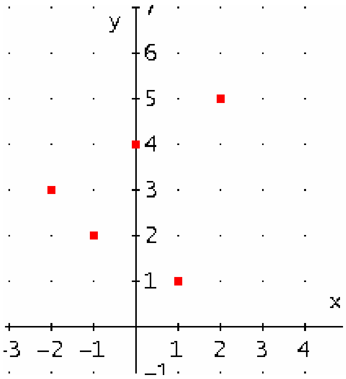
13.1. Funkce zadaná výčtem uspořádaných dvojic

Funkci, která je tvořena jenom vybranými uspořádanými dvojicemi [vzor, obraz] můžeme zadat výčtem těchto dvojic, přesněji jako jejich množinu. Hodnotu obrazu dostaneme tak, že vzor použijeme jako první index jména této množiny. Například výraz `#2` zadáme na příkazový řádek ve tvaru „`f↓-2 =`" a odešleme stisknutím (¢)(Enter). Symbol ↓ se používá k uvedení dolního indexu. Jeho alternativou je klíčové slovo `SUB` (více viz Nápověda).

#1: $f := \{[-2, 3], [-1, 2], [0, 4], [1, 1], [2, 5]\}$

#2: $f_{-2} = 3$

#3: $f_{0} = 4$



13.1.1. Posloupnosti

Jako funkci zadanou výčtem uspořádaných dvojic můžeme chápat i konečnou posloupnost. K její reprezentaci je vhodné použít funkci TABLE.

Příklad 13.1: Je dána posloupnost $\left\{\dfrac{n}{n+1}\right\}_{n=1}^{\infty}$. Znázorněte graficky několik počátečních členů této posloupnosti a určete její limitu pro $n \to \infty$.

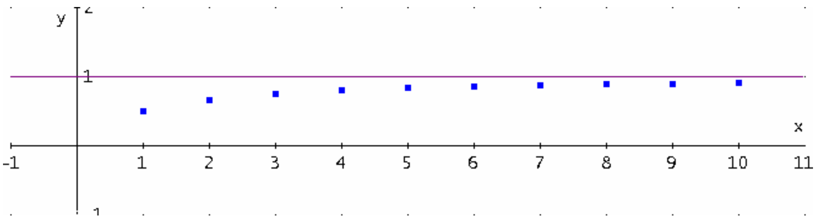
Řešení:

```
                n
#1: a(n) := ─────
              n + 1
```

```
#2: TABLE(a(n), n, 1, 10)
```

Před zobrazením výrazu #2 nastavíme vhodný rozsah os 2D grafického okna a aktivujeme tyto jeho volby:

Možnosti → Zjednodušit před vykreslením
Možnosti → Zobrazení → Body → Velikost Velká



K zadání limity využijeme buď grafické rozhraní (**Kalkul**) nebo na příkazový řádek napíšeme `LIM(a(n),n,∅)=` a odešleme (¢)

```
#3:                          lim a(n) = 1
                             n→∞
```

Příklad 13.2. : Legenda o vynálezci šachu

Známá legenda o vynálezci šachu vypráví, že byl uveden k vládci a dotázán, jakou odměnu si žádá za svůj vynález. Jeho odpověď každého překvapila. Místo zlata, drahých kamenů či nemovitostí požádal o hrstku obilí. Vyslovil totiž přání, aby dostal počet zrnek 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 pokryto všech 64 políček. Vládce, překvapený skromností požadavku, dal okamžitě příkaz k přinesení pytle obilí a vydání odpovídajícího množství zrn. K údivu všech se ukázalo, že ke splnění vynálezcova jednoduchého přání nebude stačit nejen přinesený pytel, ale ani veškeré obilí z královských sýpek.

Řešení:

Počty zrn na jednotlivých políčkách šachovnice získáme užitím příkazu

$$\#1: \quad \mathrm{TABLE}(2^{n-1}, n, 1, 64)$$

Výsledkem jeho zjednodušení je tabulka #2:

$$\#2: \quad \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 4 \\ \cdots & \cdots \\ 32 & 2147483648 \\ 33 & 4294967296 \\ \cdots & \cdots \\ 63 & 4611686018427387904 \\ 64 & 9223372036854775808 \end{bmatrix}$$

Vidíme, že počty zrn na jednotlivých políčkách tvoří konečnou geometrickou posloupnost s prvním členem 1, kvocientem 2 a počtem členů 64.
Její součet uložíme do proměnné C pomocí příkazu `C:=sum(2^(n–1),n,1,64)`.

$$\#3: \quad C := \sum_{n=1}^{64} 2^{n-1}$$

$$\#4: \quad C = 18446744073709551615$$

Rychlou představu o řádu čísla C dostaneme jeho aproximací

$$\#5: \quad C = 1.844674407 \cdot 10^{19}$$

Program nám dovoluje s tímto údajem různě experimentovat, abychom získali představu o velikosti množství.

1. Zajímá nás délka v km řady všech zrn položených za sebou

Předpokládejme, že zrno je dlouhé 0.5 cm

#6: $D := \dfrac{C\cdot 0.5}{100000}$

#7: $D = \dfrac{36893488147419103232}{40000}$

#8: $D = 9.223372036\cdot 10^{13}$

Délku řady můžeme porovnat se vzdáleností Země - Slunce (150 000 000 km), případně s délkou světelného roku (365*24*3600*3*10^5 km).

#9: $\dfrac{D}{150000000} = 6.148914691\cdot 10^{5}$

#10: $\dfrac{D}{365\cdot 24\cdot 3600\cdot 3\cdot 10^{5}} = \dfrac{1229782938247303441}{126144000000000000}$

#11: $\dfrac{D}{365\cdot 24\cdot 3600\cdot 3\cdot 10^{5}} = 9.749040289$

2. Pokud zjistíme hmotnost jednoho zrnka, můžeme snadno porovnat hmotnost všech zrnek s hmotností Země, případně Slunce.

13.2. Funkce definované na intervalu

Při zápisu funkcí definovaných na nějaké množině, například intervalu, využíváme funkce `IF` nebo `TABLE`.

Příklad 13.3: Sestrojte grafy funkcí

$f(x) = x^2 - 2$, $h(x) = x^2 - 4$ a $g(x) = \dfrac{1}{2}(x+1)^2 - 6$ na intervalu $\left(-2;3\right)$.

#1: $f(x) := x^{2} - 2$

#2: $g(x) := x^2 - 4$

#3: $h(x) := \frac{1}{2}\cdot(x + 1)^2 - 6$

K omezení hodnot nezávisle proměnné *x* daných funkcí použijeme funkci IF (podmíněný příkaz, více viz Nápověda nebo kapitola 10, Programování).

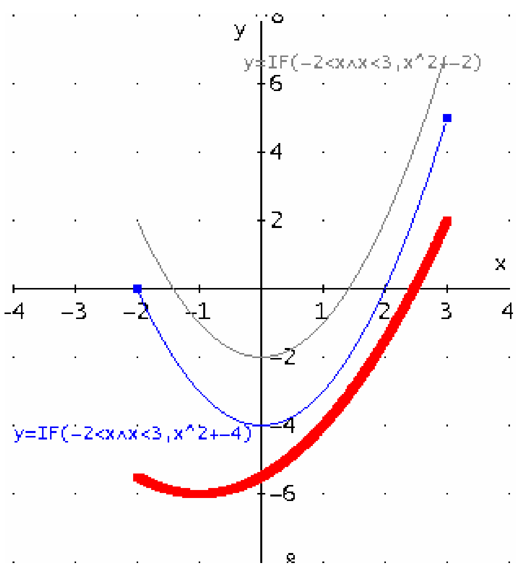
#4: IF(−2 < x < 3, f(x))

Výraz #4 potom jednoduše zobrazíme v 2D grafickém okně. Výsledek vidíme na obrázku níže. Bohužel, stejný graf bychom dostali i pro uzavřený interval. Chceme-li zdůraznit, že funkce je definována i v krajních bodech uzavřeného intervalu, použijeme třeba vektorový zápis jako na řádku #5:

#5: [IF(−2 < x < 3, g(x)), [−2, g(−2)], [3, g(3)]]

Graf o jiné tloušťce čar získáme pomocí funkce TABLE (jak je podrobně uvedeno v kapitole 9.4. Změna typu čáry ...).

#6: TABLE(h(x), x, −2, 3, 0.01)



Funkci definovanou na intervalu můžeme také pojmenovat a potom počítat její funkční hodnoty v různých bodech. Po odeslání výrazu `f(x) := IF(−2 < x < 3, x^2 − 2)` se v okně Algebra objeví definice funkce v této podobě:

```
       f(x) :=
#1:      If -2 < x < 3
            x^2 - 2
```

Potom se můžeme odkazovat na hodnotu funkce v bodě `a` výrazem `f(a)`.
Vyzkoušejte příkazy `f(−2)=`, `f(1)=` a `f(5)=`.

13.3. Funkce definované po částech

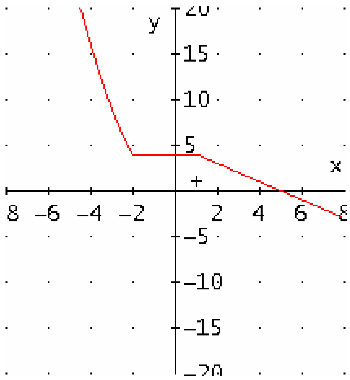
Ze známých funkcí tohoto typu má Derive implementovánu například funkci signum „`SIGN(x)`", charakteristickou funkci „`CHI(a,x,b)`" nebo funkci jednotkového kroku „`STEP(x)`". K zadání obecné funkce definované po částech opět použijeme podmíněný příkaz IF.

Příklad 13.4: Sestrojte graf funkce *f(x)*, která je dána předpisem

$$f(x)=\begin{cases} x^2; & x<-2,\\ 4; & -2\le x<1,\\ 5-x; & x\ge 1.\end{cases}$$

Řešení:

```
                      2
#1:   IF(x < -2, x , IF(x < 1, 4, 5 - x))
```

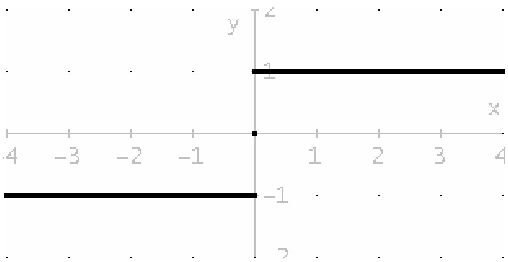


Funkce signum

Definice funkce signum se v Derive liší od běžného chápání této funkce hodnotou v bodě 0, která je `SIGN(0) = ±1`. Pokud nám tato odlišnost vadí, můžeme si výrazem `Signum(x) := IF(x < 0, –1, IF(x = 0, 0, 1))` definovat funkci vlastní. Ke správnému zobrazení takovéto funkce je vhodné použít funkci `TABLE`, jak vidíme z následující ukázky.

```
          Signum(x) :=
            If x < 0
               –1
#2:            If x = 0
                  0
                  1

#3:                                Signum(–5) = –1

#4:                                 Signum(5) = 1

#5:                                 Signum(0) = 0

#6:       Signum(x)

#7:       TABLE(Signum(x), x, –5, 5, 0.01)
```

Před vykreslením #7 aktivujeme volbu **Možnosti → Zjednodušit před vykreslením**



13.4. Celá část

Funkci celá část proměnné x (zapisujeme $\left[x\right]$) odpovídá v Derive výraz `FLOOR(x)`.

PŘÍKLAD 13.5: V oboru reálných čísel řešte rovnici

$$x\left[x\right] - 5x + 7 = 0\,.$$

Naformátováno: Řádkování: jednoduché, Ohraničení: Pole: (stínované jednoduché, Automatická, 1 b. šířka čáry)

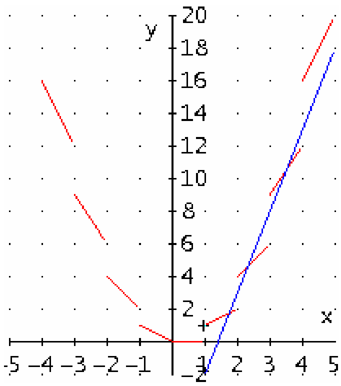
Řešení:

Úlohu vyřešíme nejprve graficky. Potom na vybraných intervalech provedeme numerické řešení. Rovnici řešíme numericky tak, že při již zmiňované proceduře řešení rovnice (kapitola 7.1.) zatrhneme v příslušném dialogovém okně (Obr. 6) volbu **Metoda řešení → Numericky**. V tom samém okně jsme dotázáni na meze intervalu, v němž se má řešení hledat **Hranice → Horní (Dolní)**.

```
#1:    x·FLOOR(x) - 5·x + 7 = 0
```

Rovnici upravíme do tvaru, který je pro grafické řešení vhodnější:

```
#2:    x·FLOOR(x) = 5·x - 7
```




```
#3:    NSOLVE(x·FLOOR(x) - 5·x + 7 = 0, x, 1, 2)

#4:                              x = 1.75

#5:    NSOLVE(x·FLOOR(x) - 5·x + 7 = 0, x, 2, 3)

#6:                           x = 2.333333333

#7:    NSOLVE(x·FLOOR(x) - 5·x + 7 = 0, x, 3, 4)

#8:                               x = 3.5
```

Poznámka: Protikladem funkce `FLOOR(x)` je funkce `CEILING(x)`

13.5. Skládání funkcí

V Derive snadno ukážeme, že skládání dvou funkcí není komutativní.

Příklad 13.6: Jsou dány funkce *f(x)* a *g(x)*. Složte tyto funkce v obou možných pořadích, tj. *f(g(x))* a *g(f(x))* a vzniklé funkce zapište předpisem a graficky znázorněte.

Řešení:

#1: $f(x) := x^2$

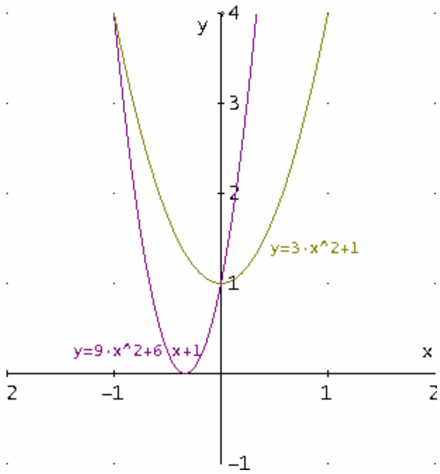
#2: $g(x) := 3\cdot x + 1$

#3: $$f(g(x)) = (3\cdot x + 1)^2$$

#4: $$f(g(x)) = 9\cdot x^2 + 6\cdot x + 1$$

#5: $$g(f(x)) = 3\cdot x^2 + 1$$

Obě složené funkce na řádcích #4 a #5 zobrazíme do jedné soustavy souřadné. Pozor, při zvoleném způsobu zápisu musíme zobrazit jenom jednu stranu (levou nebo pravou) příslušné rovnosti (#4, #5). Pro popis grafů aktivujeme v nabídce grafického okna volbu **Možnosti → Popsat nové grafy**



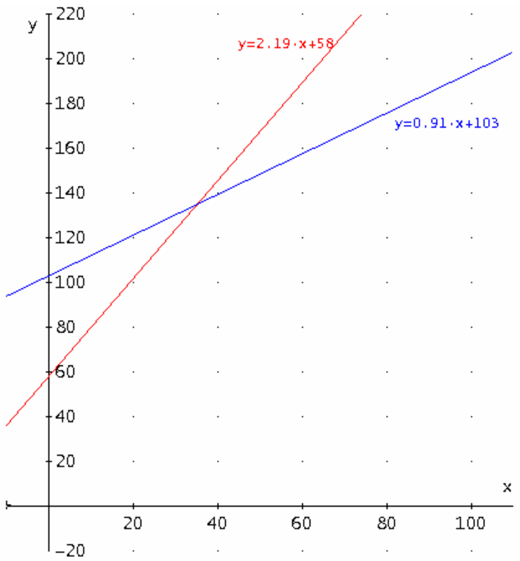
13.6. Lineární funkce

Příklad 13.7: Existuje několik poplatkových sazeb za odběr elektrické energie. Nejčastěji se vyskytuje sazba B a sazba BN. V roce 1999 se v sazbě B platilo 2,19 Kč za 1 kWh a měsíční poplatek 58 Kč bez ohledu na spotřebu. V sazbě BN se platilo 0,91 Kč za 1 kWh a měsíční poplatek byl 103 Kč bez ohledu na spotřebu. Určete, při jaké měsíční spotřebě (vyjádřené s přesností na desetiny kWh) byla finančně výhodnější sazba B. Úlohu řešte graficky i početně.
(Šarounová, A. a kol.: Matematika 9, II. díl, Prométheus Praha, 2000, str. 135, Cv.1)

Řešení:

Definujeme funkce B a BN (viz řádky #1, #2 okna Algebra).
Program Derive je po instalaci nastaven tak, že nepracuje s víceznakovými jmény proměnných. Chceme-li je používat, musíme nejprve změnit toto nastavení. To provedeme příkazem **Možnosti → Nastavení... → Vstup → Režim pro jména proměnných → Slovo**, nebo na příkazový řádek napíšeme přiřazovací příkaz `InputMode:=Word`.
Tato opatření provádět **nemusíme**, pokud se víceznaková proměnná vyskytuje **v levé části přiřazovacího příkazu** (viz #2).

```
#1:    B := 2.19·x + 58

#2:    BN := 0.91·x + 103
```



Z grafu žák snadno pozná úskalí jednotlivých sazeb, pochopí souvislost grafu s předpisem lineární funkce, o které se učí a najde řešení úlohy B < BN. K tomu zřejmě užije režim **Trasování grafů**, aktivovaný tlačítkem grafického okna. Takovýto výzkum je dobré podpořit i výpočtem. Řešíme tedy nerovnici B < BN.

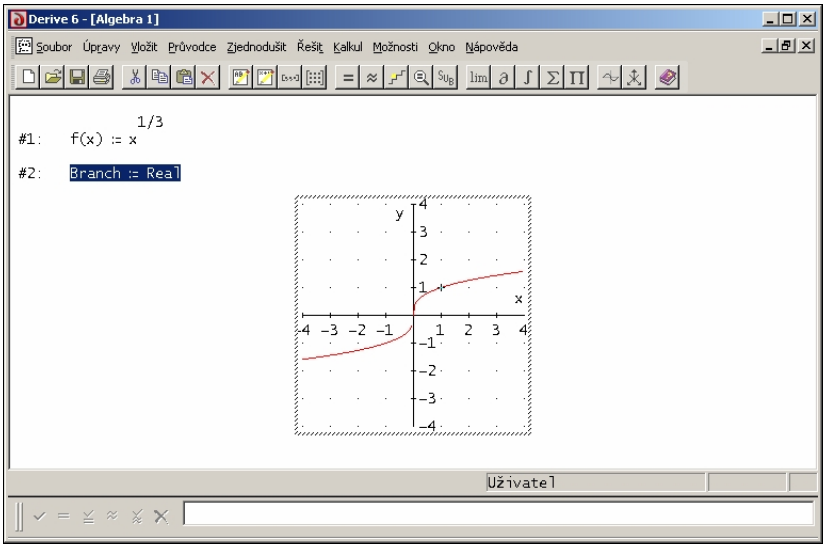
```
#3:   B < BN

#4:   SOLVE(B < BN, x, Real)

                                   1125
#5:                            x < ————
                                    32

#6:                           x < 35.15625
```

13.7. Graf iracionální funkce

Příklad 13.8: Sestrojte graf funkce f: $y = \sqrt[3]{x}$

Naformátováno: Řádkování: jednoduché, Ohraničení: Pole: (stínované jednoduché, Automatická, 1 b. šířka čáry)

Řešení:
Pro vykreslení grafu funkce i pro záporné hodnoty x je předem náležitě nastavit obor řešení odmocniny. Použijeme k tomu posloupnost příkazů
Možnosti → **Nastavení** → **Komplexní větev** → **Real** (Obr. 14).
Graf přeneseme do okna Algebra dříve uvedeným příkazem z nabídky 2D-grafického okna (musí být aktivní) **Soubor** → **Přemístit**



Obr. 15: Graf třetí odmocniny

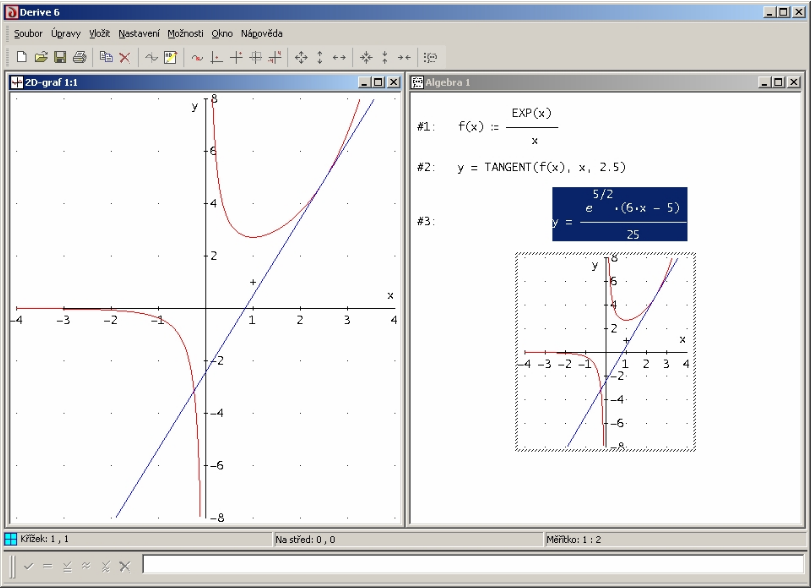
Příklad 13.9: Načrtněte graf funkce $f : y = \dfrac{e^x}{x}$, určete rovnici tečny grafu v bodě $x = 2.5$ a tečnu sestrojte.

Naformátováno: Řádkování: jednoduché, Ohraničení: Pole: (stínované jednoduché, Automatická, 1 b. šířka čáry)

Řešení:

Ukážeme si dva způsoby řešení úlohy. Nejprve využijeme prostředky programu, aniž bychom se zaměřili na podstatu problému. V druhém případě to bude naopak.

I. Použijeme příkaz `TANGENT(f(x),x,x0)`. Jeho výsledkem je rovnice tečny grafu funkce *f(x)* v bodě *x* = *x0* (Obr. 16).



Obr. 16: Tečna grafu

Naformátováno: Písmo: 11

Naformátováno: Písmo: 11 b., není Tučné

Poznámka: V následující partii 13.8, věnované racionální lomené funkci, poznáme, že funkce `TANGENT` akceptuje jako *x*-ovou souřadnici bodu dotyku tečny s grafem i nekonečno (`inf`).

II. Postupujeme jako při řešení na papíře. Derivaci funkce *f(x)* zapíšeme pomocí apostrofu: `f'(x)`, `f''(x)` atd. (o derivaci více v kapitole 15, Derivace ...).

Definujeme funkci *f(x)*:

#1: $f(x) := \frac{EXP(x)}{x}$

Zapíšeme rovnici tečny v bodě a:

#2: $y = f'(a)\cdot(x - a) + f(a)$

Proměnné a přiřadíme hodnotu (viz #3). Potom se vrátíme k výrazu #2 a zjednodušíme ho základním zjednodušením (tlačítko $\boxed{=}$). Výsledek #4 tohoto zjednodušení, tj. rovnici hledané tečny, pak zobrazíme.

#3: a := 2.5

#4: $y = \frac{e^{5/2}\cdot(6\cdot x - 5)}{25}$

13.8. Racionální lomená funkce

Příklad 13.10: Grafem funkce f, dané předpisem

$$f : y = \frac{2x-1}{x+2},$$

je rovnoosá hyperbola. Nakreslete tuto hyperbolu i s jejími asymptotami.

Naformátováno: Řádkování: jednoduché, Ohraničení: Pole: (stínované jednoduché, Automatická, 1 b. šířka čáry)

Řešení:

Pro sestrojení grafu racionální lomené funkce i s jeho asymptotami si můžeme definovat jednoduchou funkci. Nazveme ji `ASYMPTOTY(f(x))` a jejím jediným parametrem bude předpis funkce *f(x)*.

Hodnotou funkce `ASYMPTOTY` bude vektor, jehož složkami jsou v uvedeném pořadí předpis funkce *f(x)*, rovnice vodorovné asymptoty a rovnice svislé asymptoty.

Předpis funkce `ASYMPTOTY` je uveden níže na řádku #1: ve stejné podobě, v jaké ho píšeme na příkazový řádek.

Předpis **vodorovné asymptoty** získáme příkazem `TANGENT(f(x),x,∞)`, jako předpis tečny grafu funkce *f(x)* v nekonečnu.

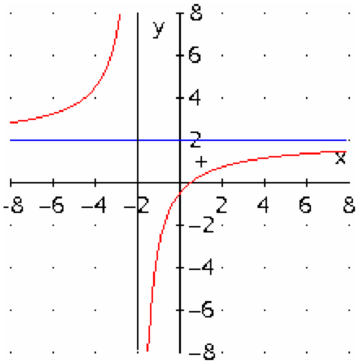
Předpis **svislé asymptoty** je výsledkem řešení rovnice, v níž položíme jmenovatel výrazu definujícího funkci *f(x)* roven nule. Rovnici řešíme příkazem `SOLVE(`*`jmenovatel`*`=0,x)`. Jmenovatele zlomku z získáme příkazem `DENOMINATOR(z)`. Protože předpis funkce *f(x)* nemusí být v ryze lomeném tvaru, raději ho rovnou do tohoto tvaru, tj. na společného jmenovatele, převedeme příkazem `FACTOR(f(x),x,Trivial)`.

```
#1: ASYMPTOTY(f) := [f, TANGENT(f, x, ∞),
                    SOLVE(DENOMINATOR(FACTOR(f, x, Trivial)) = 0, x)]
```

#2: ASYMPTOTY$\left(\dfrac{2\cdot x-1}{x+2}\right)$ $\boxed{=}$ (základní zjednodušení)

#3: $\left[\dfrac{2\cdot x-1}{x+2},\ 2,\ x=-2\right]$

Výraz #3 zobrazíme:



13.9. Graf funkce s parametrem

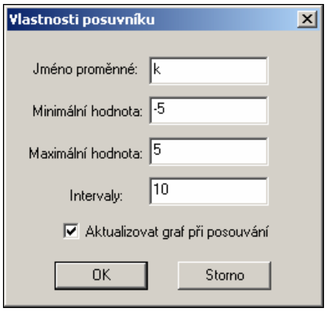
13.9.1. Užití posuvníku

Nástroj „posuvník“ nám umožňuje pouhým pohybem myší měnit hodnoty parametru daného výrazu a okamžitě zobrazovat grafy příslušných funkcí.

Příklad 13.11: Vyzkoumejte, jak ovlivní hodnoty parametru $k \in R$ průběh funkce $g: y = kx - 3$

Naformátováno: Řádkování: jednoduché, Ohraničení: Pole: (stínované jednoduché, Automatická, 1 b. šířka čáry)

Řešení:
Při aktivním grafickém okně a zvýrazněném výrazu v okně Algebra provedeme posloupnost příkazů **Vložit** → **Posuvník...**. Objeví se dialogové okno (Obr. 17), v němž vyplníme potřebné údaje.

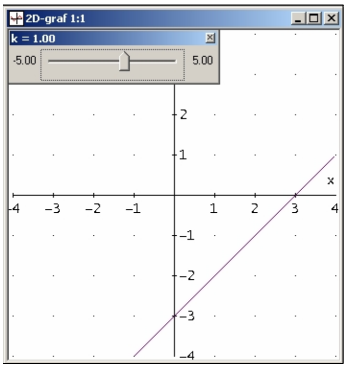


Obr. 17: Vlastnosti posuvníku

Naformátováno: Písmo: 11 b.

Naformátováno: Písmo: 11 b., není Tučné

Po potvrzení se objeví posuvník (Obr. 18). Potom teprve stisknutím tlačítka nakreslíme graf a sledujeme, jak se pohybem posuvníku mění jeho průběh v závislosti na odpovídající hodnotě parametru.



Obr. 18: Posuvník

Naformátováno: Písmo: 11 b.

Posuvník je velice silným nástrojem „dynamického" zkoumání vlastností funkcí nebo geometrických vztahů. Jeho využití v těchto oblastech se budeme věnovat v samostatných kapitolách (kap. 14. a 20.3.). Uvidíme, že vhodné použití posuvníku přibližuje možnosti systému počítačové algebry (CAS) Derive možnostem, kterými disponují programy z kategorie systémů dynamické geometrie (DGS).

13.9.2. Užití funkce VECTOR

Tento příkaz nám dává možnost zobrazit do jedné soustavy souřadnic více funkcí, které se vzájemně liší hodnotou jistého parametru. Funkce můžeme zobrazit najednou, nebo i postupně, abychom lépe postřehli vliv konkrétní hodnoty parametru na průběh grafu funkce.

Příklad 13.12: Vyzkoumejte, jak ovlivní hodnoty parametru $b \in R$ průběh funkce

$$f : y = x^2 + bx + 2$$

Naformátováno: Řádkování: jednoduché, Ohraničení: Pole: (stínované jednoduché, Automatická, 1 b. šířka čáry)

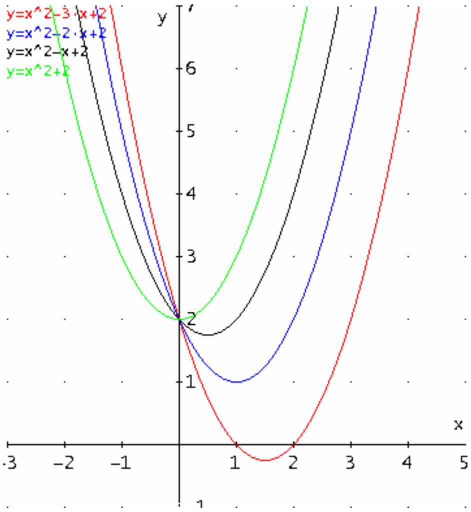
Řešení:

1) Zadáme funkci f(x) (#1).
2) Definujeme vektor funkcí lišících se hodnotou parametru (#2).
3) Příkaz `VECTOR(…)` na ploše zvýrazníme a zjednodušíme. Výsledkem je vektor funkcí #3.
4) Při zvýraznění všech složek vektoru #3 se zobrazí všechny funkce najednou. Pokud zvýrazníme jednu složku, zobrazí se jenom jí odpovídající funkce.
5) Provedeme zobrazení zvýrazněného výrazu v okně 2D-graf. Doporučuji předtím aktivovat volbu **Možnosti → Popsat nové grafy**. Ke každé funkci je tak v její barvě do grafického okna přiřazen příslušný předpis. Tyto popisky můžeme v rámci okna přemisťovat a tím ještě zvýšit názornost grafů.

#1: $f(x) := x^2 + b \cdot x + 2$

#2: `VECTOR (f(x), b, -3, 0)`

#3: $\left[x^2 - 3 \cdot x + 2,\ x^2 - 2 \cdot x + 2,\ x^2 - x + 2,\ x^2 + 2\right]$



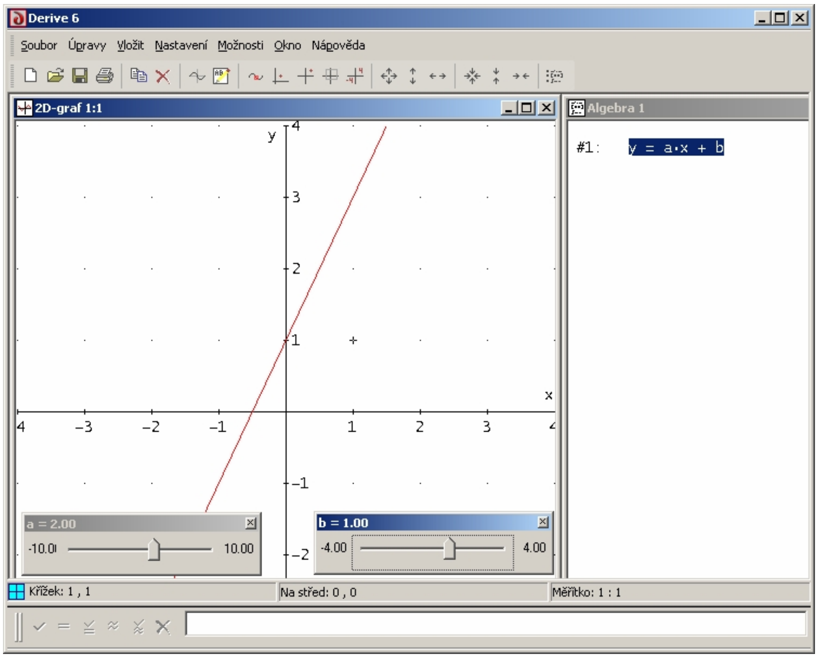
14. Využití posuvníku při zkoumání grafu funkce

14.1. Graf lineární funkce

Příklad 14.1: Lineární funkce je každá funkce daná rovnicí $y = ax + b$, kde a, b jsou libovolná reálná čísla a definičním oborem je množina všech reálných čísel. Vyzkoumejte, jaký vliv mají hodnoty čísel a, b na podobu a průběh grafu lineární funkce.

Řešení:

Pokud bychom se snažili funkci $f\colon y = ax + b$ zobrazit běžným způsobem, odměnou by nám bylo hlášení „**Příliš mnoho proměnných pro toto grafické okno!**“ Protože však z kapitoly 13.9 víme o existenci „posuvníku“, nebudeme postupovat běžným způsobem. Pro zobrazení funkce s parametry a, b totiž použijeme dva posuvníky. Výsledek vidíme na následujícím obrázku 19. Vlastní výzkum závislosti průběhu funkce na hodnotách parametrů a, b je zřejmý. Střídavou manipulací s oběma posuvníky a pozorným sledováním grafu brzy zjistíme, co má u grafu ten který parametr na starosti.

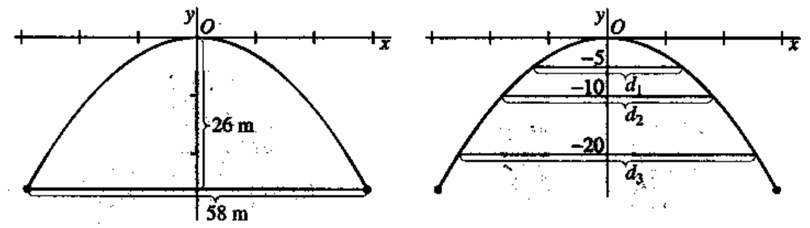


Obr. 19: Graf lineární funkce – využití posuvníku

14.2. Kvadratická funkce

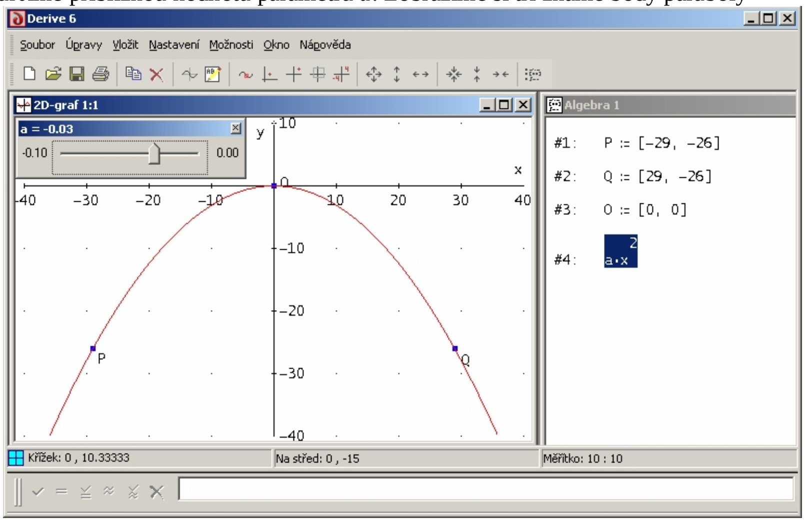
„Experimentální“ vyřešení úlohy s použitím posuvníku může být impulsem k pokusu o vyřešení početní.

Příklad 14.2: Architekt navrhuje lávku pro pěší. Stavitel lávky požaduje podrobnější informace o nosných obloucích. Architekt předkládá nákres nosného oblouku a doplňuje údaje: „Oblouk má rozpětí 58 metrů a výšku 26 metrů. V soustavě souřadnic na obrázku vlevo jde o část paraboly, která je grafem kvadratické funkce $y = ax^2$.



a) Určete ve vzorci $y = ax^2$, kterým je tato kvadratická funkce vyjádřena, hodnotu a.
b) Vypočítejte pro stavitele délky d_1, d_2, d_3, které vyznačil v obrázku vpravo.
(Odvárko, O., Kadleček, J.: Matematika pro 9. ročník z.š., 2. díl, Prométheus Praha, 2000, str. 34, Cv. 9)

Řešení:
Použijeme informaci o předpisu kvadratické funkce $y = ax^2$ a pomocí posuvníku určíme přibližnou hodnotu parametru a. Zobrazíme si tři známé body paraboly



Obr. 20: Proložení paraboly danými body

a užitím posuvníku se pokusíme těmito body proložit parabolu, která je dána předpisem ax^2. Z posuvníku na Obr. 20 vidíme, že hodnota parametru a bude přibližně -0.03. Bohužel, číselný údaj na posuvníku je omezen na setiny.

Pro přesnější vyjádření hodnoty a definujeme funkci

#5: $f(x) := a \cdot x$

a řešíme rovnice

#6: $f(-29) = -26$

#7: SOLVE(f(-29) = -26, a, Real)

#8: $$a = -\frac{26}{841}$$

#9: $$a = -0.03091557669$$

K proložení funkce danými body můžeme použít i speciální funkci FIT, jejímž výstupem je rovnice hledané křivky.

Dané body zapíšeme do matice B:

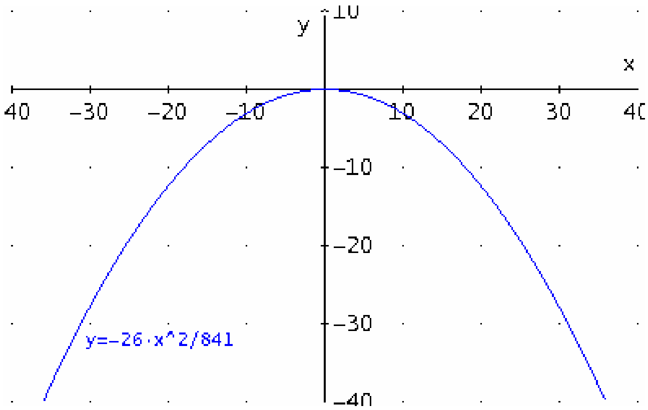
#1: $$B := \begin{bmatrix} -29 & -26 \\ 0 & 0 \\ 29 & -26 \end{bmatrix}$$

Pomocí FIT definujeme funkci f(x) (použití FIT viz Nápověda):

#2: $f(x) := \mathrm{FIT}([x, a \cdot x^2], B)$

#3: $$f(x) := -\frac{26 \cdot x^2}{841}$$

Tuto funkci pak zobrazíme:



Nakonec spočítáme hodnotu d_1 z bodu **b)** zadání:

#4: $f\left(\frac{d}{2}\right) = -5$

#5: $\mathrm{SOLVE}\left(f\left(\frac{d}{2}\right) = -5,\ d,\ \mathrm{Real}\right)$

#6: $d = -\frac{29\cdot\sqrt{130}}{13} \vee d = \frac{29\cdot\sqrt{130}}{13}$

#7: $d = -25.43468255 \vee d = 25.43468255$

Vidíme, že s pomocí nástroje posuvník se takováto úloha stává pro žáka snadněji uchopitelná a řešení rovnice pro určení hodnoty parametru *a* přirozeně vyplyne ze situace. Použití příkazu `FIT` může mít smysl po absolvování úvodu do funkcí. Pak můžeme se studenty hledat odpovědi na různé otázky. Kolik bodů stačí, co když budeme chtít proložit jinou funkci než kvadratickou, zda řešení závisí na umístění v soustavě souřadnic apod.

14.3. Graf goniometrické funkce

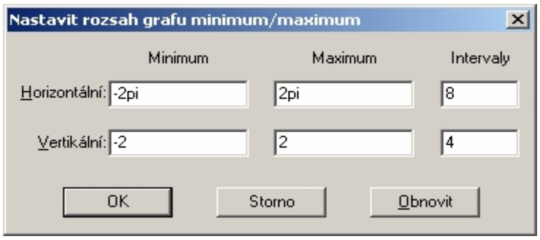
Příklad 14.3: Vyšetřete vliv hodnot parametrů a, b na průběh funkce dané předpisem

$$y = \sin(ax + b).$$

Řešení:

Nejprve si na funkci $y = \sin x$ ukážeme, jak získáme podobu osového kříže obvyklou pro znázorňování goniometrických funkcí, tj. na vodorovné ose jsou násobky π.

1) Při aktivním 2D grafickém okně otevřeme dialogové okno příkazu **Nastavení** → **Rozsah grafu** → **Minimum/maximum...** a vyplníme ho tak, aby byly meze horizontální osy zadány v násobcích π:

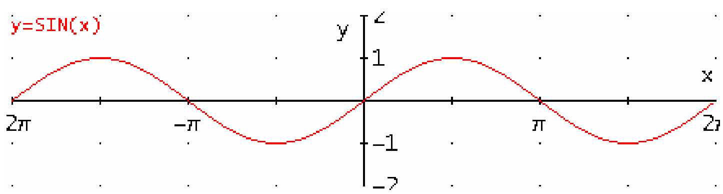


2) Na kartě **Možnosti** → **Zobrazení...** → **Osy** vepíšeme do pole **Vodorovné násobky:** hodnotu `pi` (napíšeme slovo „pi" nebo použijeme panel mat. symbolů).

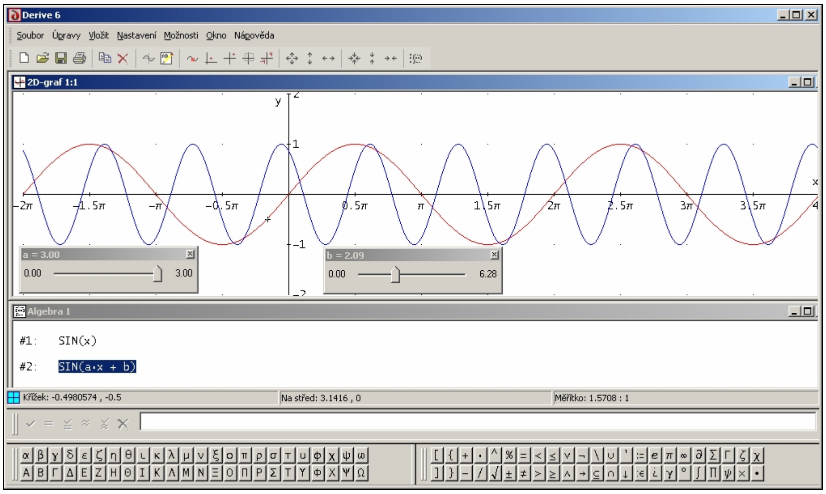
3) Pro zobrazení funkcí sinus a kosinus doporučuji zvolit horizontální uspořádáni oken.

Po zobrazení funkce sin(x) na řádku `#1:` dostaneme graf:

```
#1:   SIN(x)
```



Vliv hodnot parametrů a, b na průběh grafu funkce $y = \sin(ax + b)$ nyní můžeme vyšetřit tak, že do 2D grafického okna vložíme dva posuvníky, jeden pro parametr a, druhý pro b. Další postup je zřejmý (Obr. 21).



Obr. 21: Graf harmonické funkce

Naformátováno: Písmo: 11

Naformátováno: Písmo: 11

Naformátováno: Písmo: 11 b., není Tučné

Naformátováno: Písmo: 11 b., není Tučné

14.4. Experimentální určení rovnice křivky pomocí posuvníku

Příklad 14.4: Kabel elektrického vedení je mezi dvěma sloupy vzdálenými od sebe 100 metrů prověšen o 10 metrů. Vypočítejte, jak dlouhý je kabel mezi těmito dvěma sloupy? (Na str. 12 je uveden vzorový pracovní list, věnovaný této úloze)

Řešení:

Ohebné vlákno (tedy i kabel el. vedení) volně zavěšené ve dvou bodech, téže nebo i různé výšky, zaujímá tvar křivky, která se nazývá řetězovka (angl. Catenary). Řetězovka má rovnici:

$$y(x) := \frac{a\cdot\left(e^{x/a} + e^{-x/a}\right)}{2}$$

kde a je konstanta, jejíž hodnota závisí na fyzikálních vlastnostech vlákna a jeho zavěšení, konkrétně na délkové hustotě vlákna a tahové síle, s kterou je nataženo. Rovnici řetězovky je možno zapsat i pomocí funkce hyperbolický kosinus:

$$y(x) := a\cdot \mathrm{COSH}\left(\frac{x}{a}\right)$$

Úlohu můžeme samozřejmě vyřešit početně. Nejprve určíme hodnotu parametru a. Potom vypočítáme délku kabelu. Užijeme vzorec pro výpočet délky oblouku křivky y(x) vymezeného body [a,y(a)], [b,y(b)]:

$$s = \int_a^b \sqrt{(1 + y'(x)^2)}\, dx$$

Užití posuvníku

Nástroj posuvník nám umožňuje „experimentální" určení přibližné hodnoty parametru *a*. Provedeme to tak, že v 2D grafickém okně vyznačíme tři body, kterými kabel musí procházet (vrcholy dvou sousedních sloupů a střed kabelu, jenž je 10 metrů pod jejich úrovní), zavěsíme ho v těch dvou krajních a posuvníkem měníme hodnotu parametru *a* tak dlouho, dokud kabel neprochází i třetím prostředním bodem. Při realizaci tohoto postupu v 2D grafickém okně je dobré si uvědomit, že při umístění soustavy souřadné, které koresponduje s výše uvedenou rovnicí řetězovky, jsou souřadnice středu kabelu [0, *a*], zatímco souřadnice jeho konců jsou [-50, *y*(-50)] a [50, *y*(50)]. Tomu přizpůsobíme definice zobrazovaných objektů. Můžeme postupovat například následujícím způsobem (Řešení vidíme i na následujícím obrázku 22).

V okně Algebra nejprve obvyklým způsobem definujeme funkci y(x):

$$y(x) := a \cdot \mathrm{COSH}\left(\frac{x}{a}\right)$$

Potom zadáme tři body S1, S2 a P, jimiž musí procházet hledaná křivka:

S1 := [−50, 10]

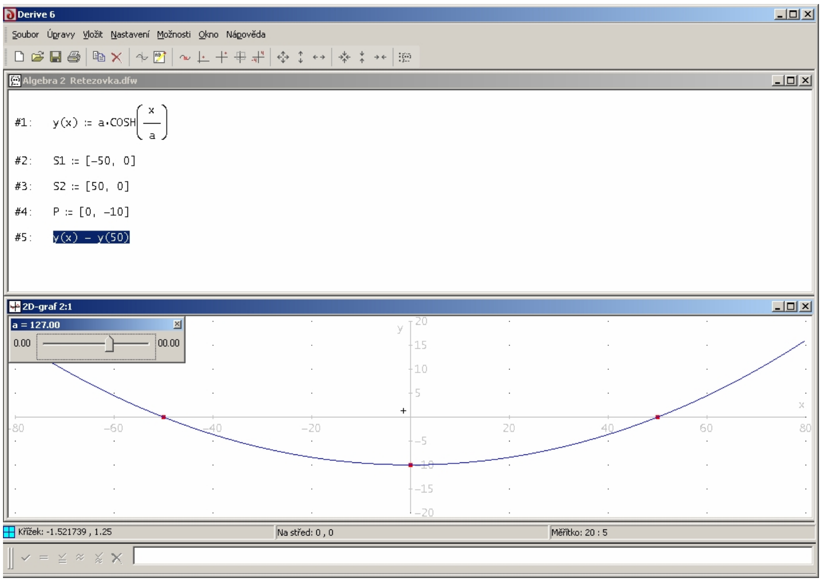
S2 := [50, 10]

P := [0, 0]

Nakonec otevřeme 2D grafické okno a vložíme do něj posuvník pro parametr a.
V tomto okně zobrazíme dané tři body spolu s grafem funkce dané předpisem:

y(x) − y(50)

Pohybem posuvníku se snažíme docílit toho, aby křivka grafu co nejlépe procházela všemi třemi body S1, S2 a P. Potom odečteme z posuvníku příslušnou hodnotu parametru a. Na obrázku 22 vidíme, že tento odhad činí $a \approx 127$.



Obr. 22: Odhad hodnoty parametru a užitím posuvníku

Grafické řešení příkladu 14.4 doplňme řešením početním.
Při vhodném umístění soustavy souřadné můžeme kabel popsat rovnicí:

#1: $y(x) := a \cdot \mathrm{COSH}\left(\dfrac{x}{a}\right)$

Podmínka pro určení hodnoty parametru *a* vyplývá ze zadání - mezi vrcholem sloupu a středem kabelu je výškový rozdíl 10 metrů:

#2: $y(50) - y(0) = 10$

#3: $\mathrm{NSOLVE}(y(50) - y(0) = 10,\ a,\ \mathrm{Real})$

#4: $a = 126.6324290$

#5: $a := 126.632429$

#6: $s = \displaystyle\int_{-50}^{50} \sqrt{(1 + y'(x)^2)}\ dx$

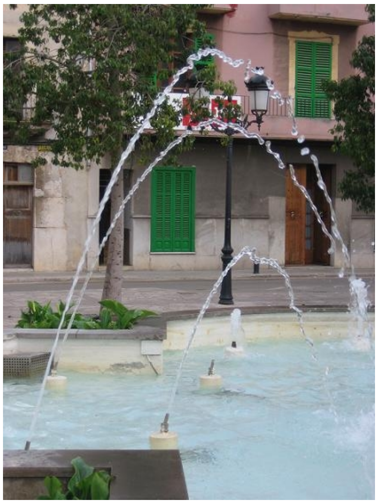
Potom délka kabelu v metrech činí:
#7: $s = 102.6186871$
Na řádku #4 vidíme, že odhad hodnoty *a* pomocí posuvníku (tj. $a \approx 127$) byl poměrně přesný.

14.5. Rovnice křivek kolem nás

V této kapitole si dovoluji představit a podrobněji rozvést myšlenku Josefa Böhma z Rakouska, kterou tento průkopník užití Derive ve výuce představil ve svém vystoupení na konferenci DES-TIME 2006 v Drážďanech (viz [3]). Další, neméně zajímavé, příspěvky tohoto zkušeného učitele najde čtenář na stránkách [18] a [19].
Kolem nás se vyskytuje řada tvarů, které lze popsat pomocí elementárních funkcí. K jednoduché analýze takových křivek v Derive stačí pořídit fotografii, umístit ji na pozadí 2D grafického okna a vyslovit hypotézu o tvaru rovnice funkce, která by příslušný tvar nejlépe popisovala. Potom už jenom vhodně umístíme počátek soustavy souřadné vzhledem ke křivce a pomocí jednoho nebo i více posuvníků se snažíme grafem zadané funkce co nejlépe kopírovat tvar křivky.
Tato aktivita může žákům ukázat, že elementární funkce probírané ve škole mají také své využití. Žáci budou zároveň motivováni k pozornějšímu sledování svého okolí a lze čekat, že někteří z nich dokonce začnou ve svém okolí tvary vyhledávat a pokoušet se je popsat rovnicemi. Přitom se nemusíme omezit jenom na hledání rovnice dané křivky. Jak uvidíme na příkladu logaritmické spirály, která popisuje tvar hlemýždí ulity, můžeme se přes křivky dostat k dalším tématům, například posloupnostem nebo komplexním číslům.

14.5.1. Parabola

Příklad 14.5: Pokuste se najít předpis funkce jedné proměnné, jejíž graf co nejlépe odpovídá křivce vykreslené tryskající vodou na obrázku **23**. Dle vlastního uvážení volte jednu z křivek na obrázku.

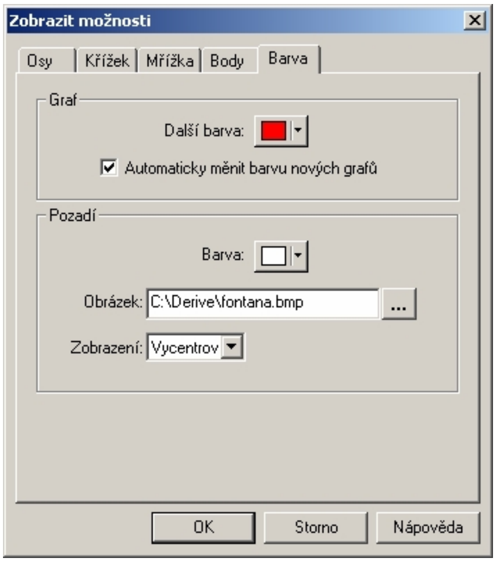


Obr. 23: Fontána

Řešení:
Postup řešení můžeme rozdělit do následujících bodů:

A. Umístění obrázku na pozadí 2D grafického okna
Program Derive umožňuje umístit na pozadí grafického okna (2D i 3D) obrázek formátu BMP. Postup je následující:

1) Příslušný soubor s příponou BMP umístíme do vybraného adresáře.
2) Otevřeme 2D grafické okno.
3) Posloupností akcí **Nastavení** → **Poměr stran…** → **Obnovit** (OK) zajistíme nezkreslené zobrazení 2D grafického okna.
4) V nabídce příkazů 2D grafického okna volíme posloupnost **Možnosti** → **Zobrazení** → **Barva**
5) Na kartě **Barva** uvedeme do pole **Obrázek** cestu k souboru BMP (Kliknutím na tlačítko se třemi tečkami se aktivuje režim vyhledání souboru), v poli **Zobrazení** volíme režim **Vycentrovat** a potvrdíme (OK).



B. Změna barvy os
Pokud nejsou osy souřadnicové soustavy kvůli barvě obrázku dostatečně čitelné, navrhuji změnit jejich barvu:

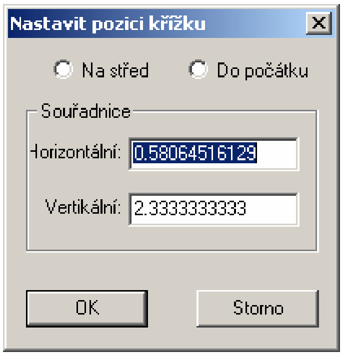
6) V nabídce příkazů 2D grafického okna volíme posloupnost příkazů **Možnosti** → **Zobrazení** → **Osy**
Na kartě **Osy** potom změníme barvu os (**Čáry**) a jejich popisu (**Měřítko**).

C. Posunutí počátku souřadné soustavy

Vhodné umístění počátku souřadnicové soustavy může podstatně zjednodušit rovnici křivky. V našem případě se snažíme přesunout počátek do vrcholu vyšetřované paraboly. Postupujeme takto:

7) Klikneme na tlačítko (**Vycentrovat do počátku**).
8) V 2D grafickém okně umístíme křížek tam, kde má být nově počátek.
9) Z nabídky volíme posloupnost **Nastavení → Pozice křížku…**

   Objeví se dialogové okno



   V tomto okně změníme v polích **Horizontální** i **Vertikální** znaménka souřadnic křížku na opačná. Tak dostaneme souřadnice bodu, který je s aktuální pozicí křížku středově souměrný podle počátku. Potvrdíme (OK).

10) Klikneme na tlačítko (**Vycentrovat na křížek**).

D. Analýza křivky užitím posuvníku

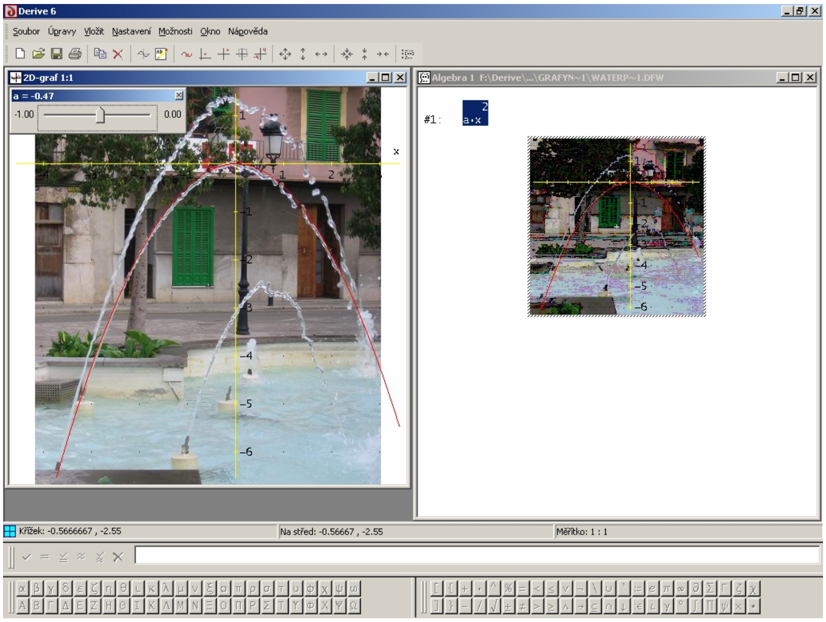
Po vhodném umístění počátku souřadnicové soustavy už nám zbývá jenom navrhnout rovnici zkoumané křivky v obecném tvaru a pomocí posuvníku určit co nejpřesněji její konkrétní podobu. V případě tryskající vody na obrázku XY bude zřejmě na místě uvažovat kvadratickou funkci s předpisem $y = ax^2$.

Do algebraického okna tedy zadáme výraz

```
         2
#1:   a·x
```

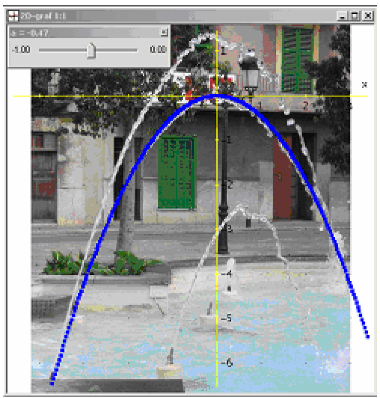
a do 2D grafického okna s obrázkem vložíme posuvník se jménem proměnné a. Rozsah hodnot proměnné a zvolíme dle našeho prvotního odhadu. Další postup je zřejmý. Posuvníkem měníme hodnotu a, dokud nemáme pocit, že se graf funkce tím nejlepším možným způsobem kryje se zkoumanou křivkou. Výsledný graf vidíme na obrázku 24. Odpovídající hodnota parametru je $a = -0.47$.

Je možné očekávat, že při cestě za co nejdokonalejším výsledkem budeme průběžně upravovat nastavení posuvníku nebo korigovat umístění počátku soustavy souřadné. I tak bude výsledek našeho snažení jenom přibližný. Přínosem takovéto aktivity je však, dle mého názoru, spíše cesta k výsledku než samotný výsledek.



Obr. 24: Je to parabola

Pro větší názornost můžeme graf zobrazit silnější čarou. Postupujeme dle návodu uvedeného v kapitole 9.4 a parabolu zobrazíme parametrickým grafem.



Za tím účelem užijeme parametrické rovnice paraboly ve formě uspořádané dvojice.

#2: $\left[x,\ a\cdot x^2\right]$

Program dle syntaxe rozpozná parametrický výraz, parametrem bude souřadnice *x*. Před zobrazením výrazu #2 v 2D grafickém okně je třeba aktivovat volbu **Možnosti → Zjednodušit před vykreslením** a, pokud jsme tak již neučinili, vložit do okna posuvník pro ovládání hodnot parametru a.

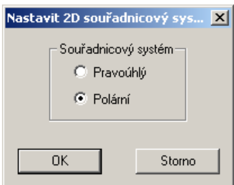
14.5.2. Logaritmická spirála

Příklad 14.6: Existuje rovnice, která by popisovala tvar hlemýždí ulity?

Řešení:

Na první pohled by se mělo jednat o spirálu. Jsou ale všechny spirály stejné? Nebo se nějak mohou odlišovat svým průběhem?

Pro znázornění a zkoumání spirály jsou ideální polární souřadnice [r, φ]. Definice těchto souřadnic není nikterak složitá a jejich použití ve výuce nemusí předcházet žádná průprava. Derive podporuje zobrazování v této souřadnicové soustavě. Stačí v nabídce 2D grafického okna provést posloupnost akcí **Nastavení → Souřadnicový systém…** a v následujícím okně zatrhnout potřebné.



Při pohledu na spirálu je zřejmé, že s rostoucí odchylkou (úhlem φ) roste délka průvodiče *r*. Otázkou je jenom, jak. Můžeme vyzkoušet závislost lineární i exponenciální.

Rovnoměrná (Archimédova) spirála

Průvodič r roste úměrně s odchylkou φ. Jinak řečeno, bod spirály se vzdaluje úměrně s otáčením. Rovnice této spirály v polárních souřadnicích $[r, \varphi]$ je

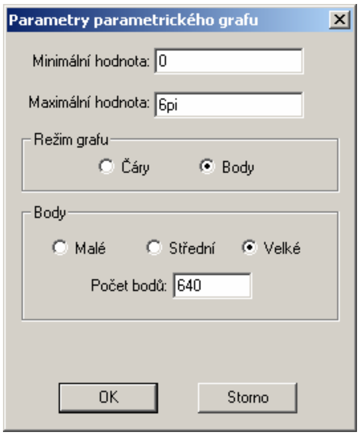
$$r = aj \; .$$

Napíšeme tedy pravou stranu této rovnice do algebraického okna a pomocí posuvníku hledáme optimální hodnotu a.

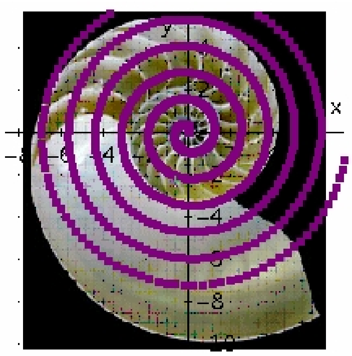
```
#1:   a·φ
```

Postup analýzy křivky je téměř stejný jako v případě pravoúhlých souřadnic. Z odlišnosti polárních souřadnic plynou akorát tato doporučení:

1) Umístění obrázku na pozadí 2D grafického okna a potřebné posunutí počátku souřadné soustavy provedeme při nastaveném **pravoúhlém** systému souřadnic. Až potom změníme soustavu na **polární**.
2) Při umisťování obrázku na pozadí může být v případě spirály vhodnější použít na kartě **Barva** v poli **Zobrazení** volbu **Rozprostřít**.
3) Nezapomeneme aktivovat volbu **Možnosti** → **Zjednodušit před vykreslením.**
4) Rovnice #1 je parametrická s parametrem φ. Proto jsme před vykreslením grafu automaticky požádáni o zadání potřebných údajů v tomto dialogovém okně:



Výsledek našeho průzkumu je patrný z následujícího obrázku 25. Je zřejmé, že tvar hlemýždí ulity neodpovídá rovnoměrné spirále pro žádnou hodnotu koeficientu a.



Obr. 25: Rovnoměrná spirála

Zkusíme tedy závislost exponenciální.

Exponenciální (logaritmická) spirála

Délka průvodiče r se pro stejné přírůstky úhlu φ zvětšuje se stejným poměrem. Rovnice této spirály je

$$r = a \cdot b^{j} \ .$$

Odtud název „exponenciální“. Hojně používaný název „logaritmická spirála“ pochází z tvaru závislosti úhlu φ na délce průvodiče r.

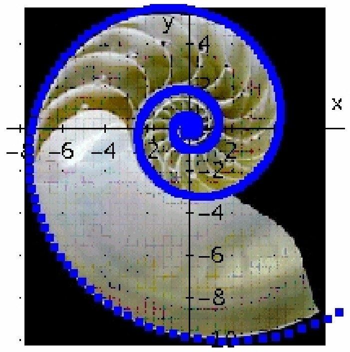
Poměr dvou průvodičů svírajících stejné úhly jako jiné dva je stálý. Roste-li odchylka aritmeticky (s konstantním rozdílem), roste průvodič geometricky (se stejným poměrem). Protože protíná všechny průvodiče pod stejným úhlem, nazývá se tato spirála též „rovnoúhlá“.

Do algebraického okna zadáme pravou stranu rovnice logaritmické spirály:

#2: $a \cdot b^{\phi}$

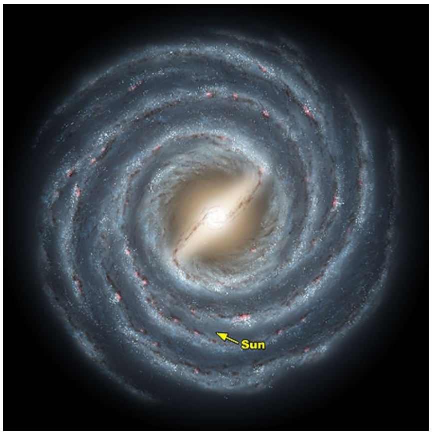
Potom vložíme do 2D grafického okna dva posuvníky pro parametry a, b a manipulací s nimi se pokusíme spirálu ulity překrýt spirálou grafu.

Tentokrát je výsledek příznivější. Hlemýžď se při stavbě své ulity nepochybně řídí rovnicí logaritmické spirály. Rovnice spirály na následujícím obrázku 26 odpovídá těmto hodnotám parametrů: `a = 0.14, b = 1.20`.



Obr. 26: Logaritmická spirála

Kde ještě v přírodě najdeme spirálu?



Obr. 27: Mléčná dráha

Poznámky na závěr
Zdrojem použitých obrázků je rubrika Obrázky na webové stránce www.google.cz. Pro vtažení žáků do děje je zřejmě účelnější, když obrázky nafotí sami ve svém okolí, jak předpokládá i autor této školní aktivity J. Böhm (viz [3]).

Připomínám, že je třeba mít na paměti, že při řešení těchto úloh jde spíše o postup, než o výsledek. Stačí například, abychom měli jinak nastavená měřítka na souřadnicových osách a už dostaneme jiný výsledek!

Analýza křivek kolem nás je dle mého názoru zajímavá sama o sobě. Vedle toho nám ale nabízí řadu tématických odboček. Například, při zkoumání uvedených dvou typů spirál uplatníme poznatky o posloupnostech, konkrétně o aritmetické a geometrické posloupnosti.
Přímo v názvu druhé ze spirál je slovo „exponenciální“. Nabízí se ukázat význam exponenciálních funkcí. Přijde někdo na to, proč zrovna tato závislost hraje tak významnou roli v některých jevech kolem nás? Proč si hlemýžď staví ulitu podle rovnice s exponenciální závislostí? Podívejme se na tvar těch dvou spirál. Odpověď se nabízí (uvážíme-li, že ulita roste, protože roste hlemýžď).

14.5.3. Logaritmická spirála v Gaussově rovině komplexních čísel

Když se zabýváme logaritmickou spirálou, byla by škoda nezmínit možnost jejího znázornění v Gaussově rovině komplexních čísel, které úzce souvisí se znázorněním komplexní mocniny.
Parametrické vyjádření logaritmické spirály v oboru komplexních čísel C má tvar z^t, kde z je komplexní číslo, pro které platí Im(z)≠0, |z|≠1 a t je reálný parametr. Potom je zřejmé, že celočíselné mocniny komplexního čísla z leží na této spirále. Pojďme si je, spolu se spirálou, znázornit.

Příklad 14.7: V Gaussově rovině komplexních čísel znázorněte mocniny komplexního čísla $(1+i)^n$ pro $n \in \{1, 2, ..., 9\}$ spolu s logaritmickou spirálou, na které příslušné body leží.

Řešení:
Definujeme číslo z:

```
#1:   z ≔ 1 + ί
```

Pokud chceme komplexní číslo znázornit jako **bod** v Gaussově rovině, provedeme to prostřednictvím zobrazení následující uspořádané dvojice v kartézské soustavě souřadnic:

$$\left[\mathrm{RE}(z^k),\ \mathrm{IM}(z^k)\right].$$

V případě, že chceme komplexní číslo znázornit i s jeho **průvodičem**, zobrazíme vektor, jehož první složkou jsou souřadnice počátku [0,0], druhou potom uspořádaná dvojice [Re(z^t),Im(z^t)]. Tento vektor se v algebraickém okně zobrazí jako matice

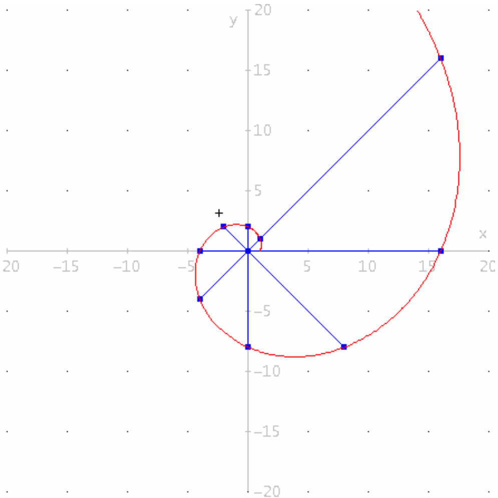
$$\begin{bmatrix} 0 & 0 \\ \mathrm{RE}(z^k) & \mathrm{IM}(z^k) \end{bmatrix}.$$

Pokračujeme v rozboru kódu řešení. Matice, odpovídající požadovaným mocninám získáme pomocí funkce VECTOR. Než následující výrazy #2 a #3 zobrazíme v 2D grafickém okně, nezapomeneme aktivovat volby **Možnosti → Zjednodušit před vykreslením** a **Možnosti → Zobrazení... → Body → Spojovat → Ano**

#2: $\mathrm{VECTOR}\left(\begin{bmatrix} 0 & 0 \\ \mathrm{RE}(z^k) & \mathrm{IM}(z^k) \end{bmatrix}, k, 1, 9\right)$

#3: $\left[\mathrm{RE}(z^k), \mathrm{IM}(z^k)\right]$

Grafické znázornění vidíme na následujícím obrázku. Zobrazením výrazu #2 dostaneme devět bodů spojených průvodičem s počátkem. Grafem výrazu #3 je pak celá spirála.



K zobrazení tabulky hodnot příslušných mocnin použijeme funkci TABLE:

#4: $\mathrm{TABLE}(z^k, k, 1, 9)$

#5: $$\begin{bmatrix} 1 & 1 + i \\ 2 & 2 \cdot i \\ 3 & -2 + 2 \cdot i \\ 4 & -4 \\ 5 & -4 - 4 \cdot i \\ 6 & - 8 \cdot i \\ 7 & 8 - 8 \cdot i \\ 8 & 16 \\ 9 & 16 + 16 \cdot i \end{bmatrix}$$

15. Derivace funkce

15.1. Zápis a výpočet derivace

Derivaci výrazu nebo funkce provedeme pomocí příkazu `DIF`, který zadáme prostřednictvím rozhraní (tlačítkem $\boxed{\partial}$ nebo posloupností příkazů **Kalkul** → **Derivace...**), případně rovnou zapíšeme na příkazový řádek ve tvaru `DIF(v,x,n)`, kde `v` je výraz nebo funkce, `x` je proměnná a `n` je řád derivace. Například, první derivaci výrazu $3x^4 - 5x^2 + 1$ vypočítáme zadáním příkazu `DIF(3x^4–5x^2+1,x)= ( ¢ )`. Výsledek dostaneme ve tvaru

$$\frac{d}{dx}(3 \cdot x^4 - 5 \cdot x^2 + 1) = 12 \cdot x^3 - 10 \cdot x$$

Druhou derivaci funkce $f(x, y) = xy^3 - \cos xy$ podle proměnné y pak určíme příkazem `DIF(f(x,y),y,2)= ( ¢ )`, jehož výsledkem je rovnost

$$\left(\frac{d}{dy}\right)^2 f(x, y) = x^2 \cdot \mathrm{COS}(x \cdot y) + 6 \cdot x \cdot y$$

Chceme-li s derivací dále pracovat jako s funkcí, oceníme možnost zápisu pomocí apostrofu (jednoduchá uvozovka – single quote, ASCII kód 39). Tímto způsobem však můžeme zapsat pouze derivace pojmenované funkce, nikoliv samotného výrazu. Například druhá derivace funkce g(x)=3x^2 se napíše `g''(x)`, zápis (3x^2)`''` ale smysl nemá.

Více o počítání derivací najdeme v nápovědě programu (**Příkazy z Nabídky** → **Okno Algebra** → **Kalkul**).

15.2. Tečna grafu funkce

Zápisu rovnice tečny grafu funkce, užitím funkce `TANGENT` nebo pomocí derivace, se věnují příklady 13.9 a 13.10 na stranách 68 až 70. Zde si připomeneme ten druhý způsob, který je názornější a vyžaduje porozumění významu derivace funkce. Zároveň si ukážeme, jak se dá pomocí posuvníku animovat pohyb tečny podél grafu funkce.

Příklad 15.1: Sestrojte graf funkce *f(x)*, která je dána předpisem $f: y = \dfrac{4x}{x^2+1}$.
Napište rovnici tečny grafu této funkce nejprve v obecném bodě $x = a$, potom v konkrétních bodech $x = \dfrac{3}{4}$ a $x = 1$. Příslušné tečny zobrazte.

Řešení:

#1: $f(x) := \frac{4 \cdot x}{x^2 + 1}$

Rovnici tečny zapíšeme jako na papíře:

#2: $y = f'(a) \cdot (x - a) + f(a)$

Zadáme konkrétní hodnotu parametru a:

#3: $a := \frac{3}{4}$

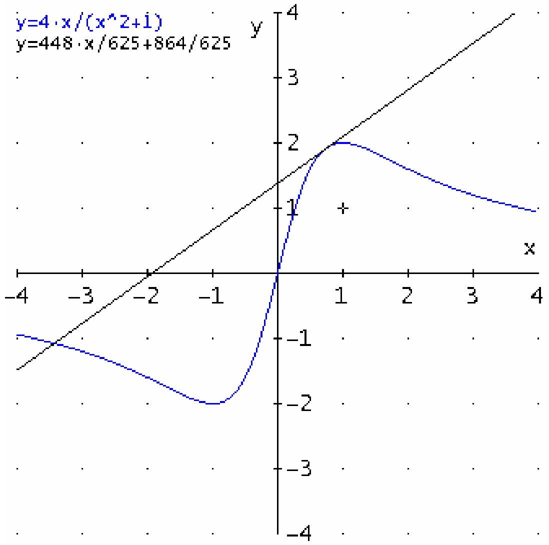
Potom zvýrazníme výraz #2 a provedeme jeho základní zjednodušení. Dostaneme rovnici tečny:

#4: $$y = \frac{32 \cdot (14 \cdot x + 27)}{625}$$

Po úpravě (**Zjednodušit → Roznásobit…**):

#5: $$y = \frac{448 \cdot x}{625} + \frac{864}{625}$$

Výrazy #1 a #5 zobrazíme v 2D-grafickém okně:

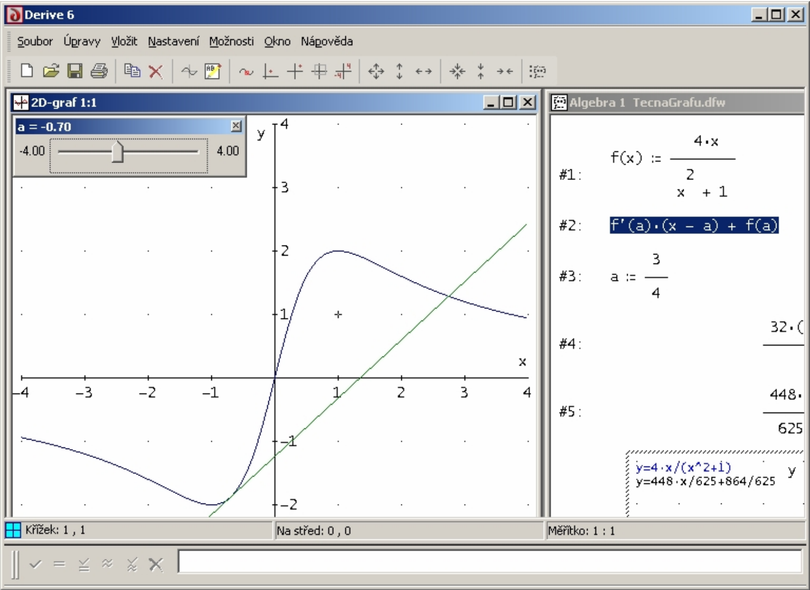


Jak bylo řečeno výše, posuvník nám umožňuje provést animaci pohybu tečny podél grafu. Tak můžeme například zkoumat souvislost mezi průběhem derivace funkce a změnami sklonu tečny jejího grafu.

Budeme zobrazovat rovnici tečny #2 s parametrem a. Abychom mohli tento parametr ovládat posuvníkem, musíme ho zbavit hodnoty. Použijeme prázdný přiřazovací příkaz:

#6: a :=

Potom vložíme do 2D-grafického okna posuvník pro ovládání hodnoty parametru a. Nakonec výraz #2 zvýrazníme, v nabídce 2D-grafického okna potvrdíme volbu **Možnosti → Zjednodušit před vykreslením** a výraz (rovnici tečny) #2 znázorníme.
Při manipulaci s posuvníkem se tečna pohybuje podél grafu funkce. Na následujícím obrázku je zachycena situace pro $a = -0.70$:



15.3. Užití derivace

Příklad 15.2: Při jaké hodnotě základu $a > 0$ je exponenciální funkce a^x rovna své derivaci?

Řešení:

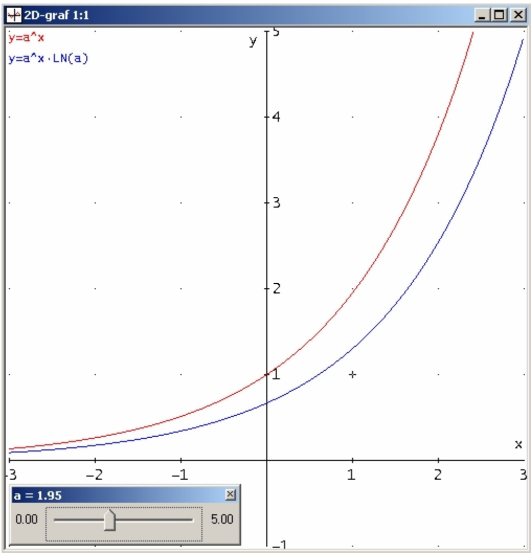
#1: $f(x) := a^x$

#2: $f'(x)$

Funkci i její derivaci můžeme zobrazit současně, užitím jednoho posuvníku pro parametr a. Můžeme to udělat tak, že při vloženém posuvníku zobrazíme v 2D grafickém okně výrazy #1 a #2. Při pohybu posuvníku se pak mění oba grafy. Další možností je zapsat výrazy do matice 2×1, tj. z příkazového řádku odeslat na pracovní plochu výraz `[f(x);f'(x)]` (pozor na středník, jinak můžeme stejnou matici zadat také výrazem `[[f(x)],[f'(x)]]`), a zobrazit je najednou.

#3: $\begin{bmatrix} f(x) \\ f'(x) \end{bmatrix}$

Pro správné zobrazení je třeba při aktivním 2D grafickém okně nastavit režim **Možnosti → Zjednodušit před vykreslením**. Pro dosažení větší přehlednosti doporučuji ještě režim **Možnosti → Popsat nové grafy** spolu s nastavením automatického střídání barev grafů.



Po vykreslení grafů funkce `f(x)` a její derivace je naším cílem manipulací s posuvníkem tyto grafy ztotožnit. V levém horním rohu okna posuvníku potom odečteme odpovídající hodnotu parametru `a`. Při nastavení počtu intervalů v okně **Vlastnosti posuvníku** tak, abychom využili maximální přesnosti údaje posuvníku, dostaneme hodnotu a = 2.72. Toto číslo dle očekávání odpovídá hodnotě Eulerova čísla.

Po grafickém průzkumu můžeme úlohu vyřešit početně. Při výpočtu v symbolickém režimu je výsledkem Eulerovo číslo `e`.

#4: $f(x) - f'(x) = 0$

#5: $$a^x \cdot (1 - \mathrm{LN}(a)) = 0$$

#6: $\mathrm{SOLVE}(a^x \cdot (1 - \mathrm{LN}(a)) = 0, a, \mathrm{Real})$

#7: $$a = e$$

#8: $$a = 2.718281828$$

Eulerovo číslo můžeme najít také postupnou analýzou vztahu $\lim\limits_{h \to 0} \frac{a^h - 1}{h} = 1$.

Například užitím funkce `TABLE`.

#1: $f(x) := a^x$

#2: $df(x) := \lim\limits_{h \to 0} \frac{f(x + h) - f(x)}{h}$

#3: $$df(x) := \lim_{h \to 0} a^x \cdot \left(\frac{a^h}{h} - \frac{1}{h} \right)$$

Výraz #3 zjednodušíme na:

#4: $$df(x) := \lim_{h \to 0} a^x \cdot \frac{a^h - 1}{h}$$

Potom je zřejmé, že rovnice

#5: $f(x) = df(x)$

je splněna pro takové a, pro které je $\lim_{h\to 0}\dfrac{a^h-1}{h}=1$. K určení jeho přibližné hodnoty použijeme funkci TABLE. Tu použijeme třeba i opakovaně, dokud nedosáhneme vhodné přesnosti vyjádření a. Z tabulky, která je výsledkem zjednodušení následujícího výrazu #6, je zřejmé, že a je přibližně rovno 2.72.

#6: $\text{TABLE}\left(\lim_{h\to 0}\dfrac{a^h-1}{h},\ a,\ 2,\ 3,\ 0.01\right)$

#7:

2	0.6931471805
2.01	0.6981347220
....	....
2.71	0.9969486348
2.72	**1.000631880**
2.73	1.004301609
....	....
3	1.098612288

Vidíme, že funkci TABLE můžeme využít k nalezení přibližného řešení rovnice prostým dosazováním postupně se zpřesňujících odhadů. S tímto využitím funkce TABLE se setkáme ještě v kapitole 15.3, věnované vyšetřování extrémů funkce bez použití derivace.

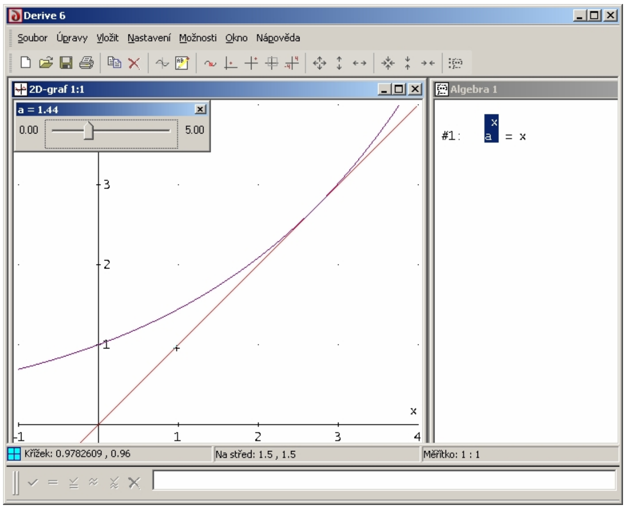
Příklad 15.3: Pro které hodnoty reálného parametru $a > 1$ má rovnice $a^x = x$ jediné řešení?

Řešení:

#1: $a^x = x$

Grafické řešení.

Řešme úlohu nejprve graficky. Tak, že zobrazíme zvlášť levou a zvlášť pravou stranu rovnice. K zobrazení levé strany s parametrem `a` použijeme posuvník. Jak je patrné z následujícího obrázku, snažíme se pohybem posuvníku dosáhnout toho, aby měly oba grafy právě jeden společný bod. V okně posuvníku pak najdeme odpovídající přibližnou hodnotu `a`, `a = 1.44`.

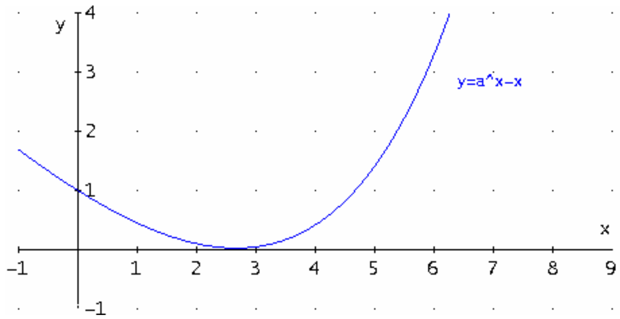


Početní řešení.

Rovnici upravíme do homogenního tvaru a definujeme odpovídající funkci `f(x)`. Hledáme tedy hodnotu parametru `a`, pro kterou má funkce `f(x)` jediný nulový bod.

```
             x
#2:   f(x) := a  - x
```

Opět můžeme využít posuvník a s jeho pomocí prozkoumat závislost průběhu funkce `f(x)` na hodnotách parametru `a`. Pokud jsme neprovedli předchozí grafické řešení, zjistíme tak také, zda má úloha smysl. Z grafu na následujícím obrázku je patrné, že úloha řešení má a cesta k němu vede přes minimum funkce `f(x)`.



Stejnou informaci dostaneme i z předpisů funkce a její druhé derivace. Funkce je spojitá a konvexní na R. Má-li mít jeden nulový bod, musí tento bod odpovídat bodu dotyku grafu funkce s osou x, tj. minimu funkce.

#3: $$f'(x) = a \cdot LN(a) - 1$$

#4: $$f''(x) = a^{x} \cdot LN(a)^{2}$$

Hledáme tedy bod, v němž je první derivace funkce f(x) rovna nule. Řešením je výraz #6 s proměnnou a.

#5: $SOLVE(a^{x} \cdot LN(a) - 1, x, Real)$

#6: $$x = -\frac{LN(LN(a))}{LN(a)}$$

Tento výraz dosadíme do předpisu funkce f(x) místo proměnné x:

#9: $SUBST\left(f(x), x, -\frac{LN(LN(a))}{LN(a)}\right)$

#10: $$\frac{LN(LN(a))}{LN(a)} + \frac{1}{LN(a)}$$

Výsledný výraz položíme roven nule a řešíme jako rovnici s neznámou a:

#11: $SOLVE\left(\frac{LN(LN(a))}{LN(a)} + \frac{1}{LN(a)}, a, Real\right)$

#12: $a = \infty \vee a = 0 \vee a = e^{e^{-1}}$

Výsledkem, který nás zajímá a tudíž řešením úlohy je $e^{\frac{1}{e}}$. Pokud tento výraz v zápisu řešení zvýrazníme a stiskneme tlačítko aproximace $\boxed{\approx}$, dostaneme přibližnou hodnotu a:

#13: 1.444667861

Vidíme, že výsledek grafického řešení dobře odpovídá této hodnotě.

Poznámka:
Nesmíme se nechat zmást tím, že v Derive mají smysl i logaritmy z nekladných čísel, z komplexních čísel a dokonce i z nevlastních hodnot (±∞). Vyzkoušejte:

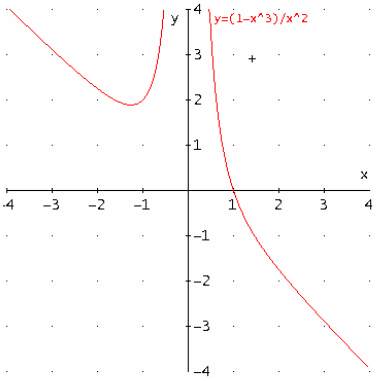
$\mathrm{LN}(0) = -\infty$, $\mathrm{LN}(\infty) = \infty$, $\mathrm{LN}(-4) = 2 \cdot \mathrm{LN}(2) + \pi \cdot i$,

$\mathrm{LN}(3 \cdot i) = \mathrm{LN}(3) + \dfrac{\pi \cdot i}{2}$.

Příklad 15.4: Vyšetřete průběh funkce dané předpisem $f(x) = \dfrac{1-x^3}{x^2}$.

Řešení:

$$f(x) := \frac{1 - x^3}{x^2}$$



První a druhá derivace:

$$f'(x) = -\frac{x^3 + 2}{x^3}$$

$$f''(x) = \frac{6}{x^4}$$

Limity v krajních bodech definičního oboru:

$$\lim_{x \to -\infty} f(x) = \infty$$

$$\lim_{x \to \infty} f(x) = -\infty$$

$$\lim_{x \to 0-} f(x) = \infty$$

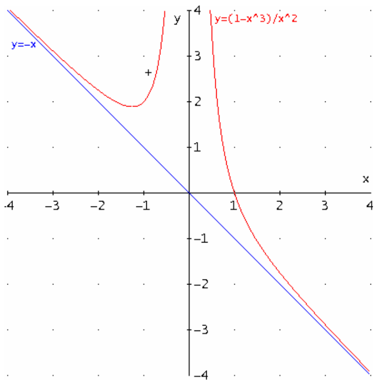
$$\lim_{x \to 0+} f(x) = \infty$$

$$\lim_{x \to 0} f(x) = \infty$$

Asymptota:

TANGENT(f(x), x, ∞) = −x

TANGENT(f(x), x, −∞) = −x



Stacionární body:

```
SOLVE(f'(x), x, Real)
```

$$x = -2^{1/3}$$

Abychom mohli se stacionárními body dále manipulovat, použijeme místo funkce `SOLVE` funkci `SOLUTIONS`, jejímž výstupem je vektor řešení. Tento vektor uložíme do proměnné `SB`:

```
SB := SOLUTIONS(f'(x), x, Real)
```

$$SB := \left[-2^{1/3} \right]$$

Potom podezřelý bod snadno dosadíme do druhé derivace:

$$f''(SB_1) = \frac{3 \cdot 2^{2/3}}{2}$$

$$f''(SB_1) = 2.381101577$$

Ze znaménka druhé derivace je zřejmé, že v podezřelém bodě je lokální minimum. Hodnotu funkce v tomto bodě zjistíme dosazením:

$$f(SB_1) = \frac{3 \cdot 2^{1/3}}{2}$$

$$f(SB_1) = 1.889881574$$

Konvexnost, konkávnost:

```
SOLVE(f''(x) > 0, x)
```

$$\mathrm{true}$$

```
SOLVE(f''(x) < 0, x)
```

$$\mathrm{false}$$

Druhá derivace je v celém definičním oboru kladná, funkce je tedy v každém z intervalů definičního oboru konvexní.

V následujícím příkladu uvidíme, jak si Derive poradí s bodem, v němž je funkce spojitá, ale derivace v něm neexistuje.

Příklad 15.5: Vyšetřete derivaci funkce $f(x) = \dfrac{15}{1+|x-1|} + x^2 + 2x - 1$ v bodě $x = 1$.

Řešení:

$x :\in \text{Real}$

$$f(x) := \frac{15}{1 + |x - 1|} + x^2 + 2\cdot x - 1$$

$$f'(x) = \frac{(4\cdot x^2 - 19)\cdot \text{SIGN}(x - 1) + 2\cdot(x^3 - x^2 + 2)}{(|x - 1| + 1)^2}$$

První derivace v bodě x = 1:

$$f'(1) = \pm 15 + 4$$

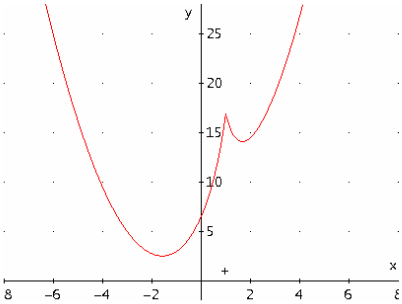
Z definice derivace pomocí limity určíme postupně derivaci zprava a zleva v bodě 1. Připomeňme si, že limitu funkce `f(x)` v bodě `a` zprava (zleva) určíme pomocí funkce `LIM(f(x),x,a,p)`, kde pátý parametr `p` má hodnotu libovolného kladného (záporného) čísla, např. 1 (-1).

$$\lim_{x\to 1+} \frac{f(x) - f(1)}{x - 1} = -11$$

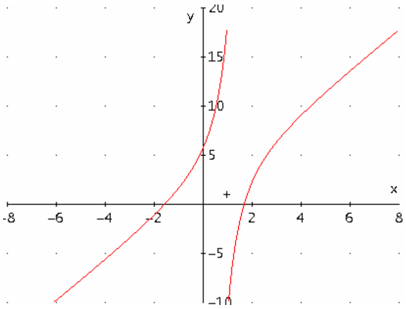
$$\lim_{x\to 1-} \frac{f(x) - f(1)}{x - 1} = 19$$

Potom je význam zápisu $f'(1) = \pm 15 + 4$ zřejmý.

Srovnejme ještě graf funkce `f(x)` a její derivace:



`f(x)`



`f'(x)`

15.4. Vyšetření extrému bez užití derivace

Program Derive nám nabízí prostředky, které můžeme využít k přibližnému určení extrémů funkce bez užití derivace. Konkrétně si ukážeme použití režimu **Trasovat grafy** a funkce **TABLE**.

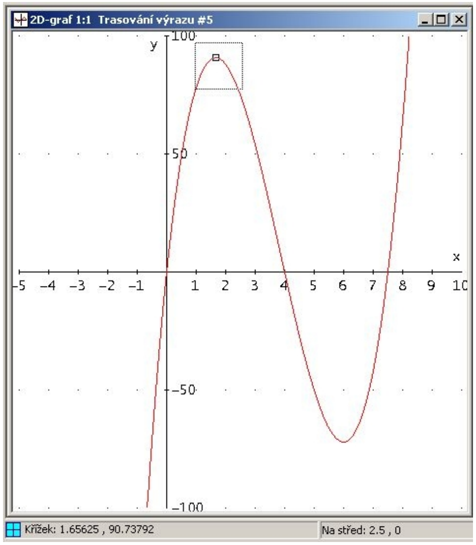
Příklad 15.6: Z lepenky tvaru obdélníku o stranách *a*, *b* (*a*>*b*) se mají v rozích vyříznout čtverce o straně velikosti *x* tak, aby vznikla síť bez horní podstavy pro kvádr s maximálním objemem. a) Určete velikost *x* strany čtverce. b) Dokažte, že pro *x* platí vztah *x*<*(a*+*b)/6*. c) Řešte pro *a=15* cm, *b=8* cm.

(*Ivan Bušek: Řešené maturitní úlohy z matematiky, SPN Praha, 1985*)

Řešení:

```
#1:   V(x) := (a - 2·x)·(b - 2·x)·x

#3:   a := 15

#4:   b := 8

#5:                       V(x) = 2·x·(x - 4)·(2·x - 15)
```

Nejprve použijeme k vyšetření extrému funkce *V(x)* režim **Trasovat grafy**, který aktivujeme tlačítkem (stejným tlačítkem ho i deaktivujeme). Ukazatel myši se změní na čtvereček přichycený na grafu, jak je patrné z následujícího obrázku.



Souřadnice středu čtverečku jsou zobrazeny v levém dolním rohu grafického okna. Můžeme s ním hýbat pomocí šipkových kláves nebo myši. K získání většího detailu, a tedy i přesnějšího určení extrému, využijeme tlačítko (Nastavit rozsah).

Funkce `TABLE` umožňuje vytvářet tabulky libovolného počtu hodnot dané funkce na vybraném intervalu. Využití je zřejmé. V první tabulce vyhledáme extrémní hodnoty funkce V(x), stanovíme intervaly, v nichž se určitě vyskytují a vytvoříme nové tabulky na těchto intervalech. Tento proces opakujeme, dokud nejsme spokojeni s přesností určení polohy extrému.

```
#6:   TABLE(V(x), x, 1, 3, 0.2)
```

#7:

$$\begin{bmatrix} 1 & 78 \\ 1.2 & 84.672 \\ 1.4 & 88.816 \\ 1.6 & 90.624 \\ 1.8 & 90.288 \\ 2 & 88 \\ 2.2 & 83.952 \\ 2.4 & 78.336 \end{bmatrix}$$

$$\begin{bmatrix} 2.6 & 71.344 \\ 2.8 & 63.168 \\ 3 & 54 \end{bmatrix}$$

#8: TABLE(V(x), x, 1.4, 1.8, 0.05)

#9:
$$\begin{bmatrix} 1.4 & 88.816 \\ 1.45 & 89.4795 \\ 1.5 & 90 \\ 1.55 & 90.3805 \\ 1.6 & 90.624 \\ 1.65 & 90.7335 \\ 1.7 & 90.712 \\ 1.75 & 90.5625 \\ 1.8 & 90.288 \end{bmatrix}$$

#10: TABLE (V(x), x, 1.6, 1.7, 0.01)

#11:
$$\begin{bmatrix} 1.6 & 90.624 \\ \ldots & \ldots \\ 1.66 & 90.739584 \\ 1.67 & 90.74045199 \\ 1.68 & 90.736128 \\ 1.69 & 90.726636 \\ 1.7 & 90.712 \end{bmatrix}$$

#12: TABLE (V(x), x, 1.66, 1.68, 0.005)

#13:
$$\begin{bmatrix} 1.66 & 90.739584 \\ 1.665 & 90.7406685 \\ 1.67 & 90.74045199 \\ 1.675 & 90.7389375 \\ 1.68 & 90.736128 \end{bmatrix}$$

#14: TABLE (V(x), x, 1.66, 1.67, 0.001)

#15:

1.66	90.739584
....	
1.665	90.7406685
1.666	90.74072918
1.667	90.74073785
1.668	90.74069452
1.669	90.74059923
1.67	90.74045199

#16: TABLE (V(x), x, 1.666, 1.668, 0.0005)

#17:

1.666	90.74072918
1.6665	**90.74074001**
1.667	90.74073785
1.6675	90.74072268
1.668	90.74069452

Potom můžeme své výsledky porovnat s výpočtem:

#18: $V'(x) = 4\cdot(3\cdot x^2 - 23\cdot x + 30)$

#19: $\text{SOLVE}(4\cdot(3\cdot x^2 - 23\cdot x + 30), x, \text{Real})$

#20: $x = \frac{5}{3} \vee x = 6$

#21: $V''(x) = 4\cdot(6\cdot x - 23)$

16. Integrál

K výpočtu integrálu, neurčitého i určitého, použijeme buď grafické rozhraní (tlačítko ∫ nebo posloupnost příkazů **Kalkul** → **Integrál...**) a nebo funkci `INT` (syntaxe viz **Nápověda**). Kromě této funkce je v Derive definována řada dalších, které umožňují například výpočet integrálu konkrétní metodou nebo grafické znázornění obrazců, jejichž obsahy počítáme určitým integrálem. Samostatnou skupinu potom tvoří aplikační funkce založené na integrálním počtu. Jmenujme například funkce `VOLUME_OF_REVOLUTION` nebo `AREA_OF_REVOLUTION` pro výpočet objemu, respektive obsahu plochy rotačního tělesa. I když zde budou některé funkce zmíněny, pro získání úplného přehledu doporučuji čtenáři prostudovat nápovědu.

16.1. Neurčitý integrál

Při výpočtu neurčitého integrálu využíváme několika metod. Volba konkrétní metody je dána podobou integrovaného výrazu. Málo zkušenému jedinci často činí potíže již tento výběr metody. Mohlo by tedy být zajímavé zjistit, jak „přemýšlí“ program Derive. To nám částečně odhalí režim **Zobrazit krok** ().

Příklad 16.1: Vypočtěte $\int \sin(x)\cos(x)dx$.

Řešení:

Zadáme výraz, který chceme integrovat:

```
#1:   SIN(x)·COS(x)
```

vyznačíme ho a stiskneme tlačítko ∫, vyplníme údaje v dialogovém okně a potvrdíme tlačítkem (OK). Objeví se zvýrazněný zápis integrálu:

```
#2:   ∫(SIN(x)·COS(x), x, c)
```

Potom už jenom mačkáme tlačítko a sledujeme jednotlivé kroky řešení. Končíme až tehdy, když se výsledek začne opakovat.

```
∫(F(x), x, c) → ∫ F(x) dx + c

#3:                    ∫ SIN(x)·COS(x) dx + c

                          n + 1
      n               F(x)
∫ F(x) ·F'(x) dx → ——————————
                       n + 1

                                  2
                            SIN(x)
#4:                         ——————— + c
                               2
```

Vedle funkce INT jsou v Derive k dispozici speciální funkce pro počítání integrálu metodou per partes (INT_PARTS) a metodou substituční (INT_SUBST).

16.2. Určitý integrál

Jak bylo řečeno v úvodu kapitoly, Derive disponuje funkcemi nejenom pro výpočet integrálu, ale i pro znázornění ploch, jejichž obsahy se užitím určitého integrálu počítají. Pro příklad zde uvedeme funkce PlotInt a AreaBetweenCurves.

Příklad 16.2: Určete obsah obrazce, ohraničeného grafem funkce $h: y = x - x^2\sqrt{x}$ a osou *x*.

Řešení:

```
          2
#1:   h(x) ≔ x - x ·√x
```

Nakreslíme graf funkce (viz Obr. 28). Potom určíme její nulové body. Pro účely jejich dalšího použití jako mezí určitého integrálu je výhodnější využít k jejich výpočtu funkci SOLUTIONS. Ta dovoluje uložit řešení příslušné rovnice do vektoru N, na jehož složky se můžeme odkazovat pomocí dolních indexů (operátor ↓ nebo SUB).

```
#2:   N ≔ SOLUTIONS(h(x), x)

#3:                                N ≔ [0, 1]
```

Složky N↓1, N↓2 vektoru řešení použijeme jako meze určitého integrálu. K jeho zadání použijeme buď grafické rozhraní (**Kalkul → Integrál...**) nebo na příkazový řádek zapíšeme funkci INT(h(x),x,N↓1,N↓2):

```
       N↓2
#4:   ∫     h(x) dx
       N↓1
```

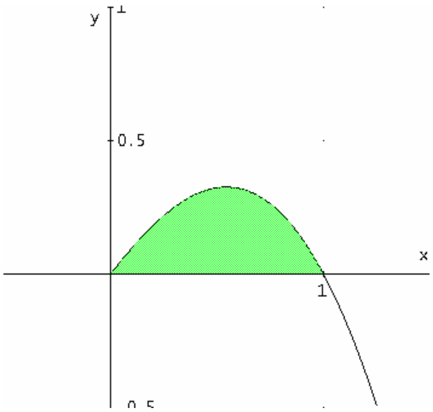
Základním zjednodušením #4 dostaneme výsledek:

$$\#5: \quad \frac{3}{14}$$

Funkce PlotInt má stejné parametry jako odpovídající funkce INT. Lze ji zadat jenom přes příkazový řádek ve tvaru PlotInt(h(x),x,N↓1,N↓2):

```
      PlotInt(h(x), x, N , N )
#6:                      1   2
```

Chceme-li výraz #6 rovnou zobrazit, musíme nejprve aktivovat volbu 2D grafického okna: **Možnosti → Zjednodušit před vykreslením**



Obr. 28: Funkce `PlotInt(x - x^2·√x, x, 0, 1)`

Příklad 16.3: Určete obsah obrazce ohraničeného grafy funkcí
$f: y = e^x$, $g: y = e^{-x}$ a přímkou $x = 1$. Načrtněte obrázek.

Řešení:

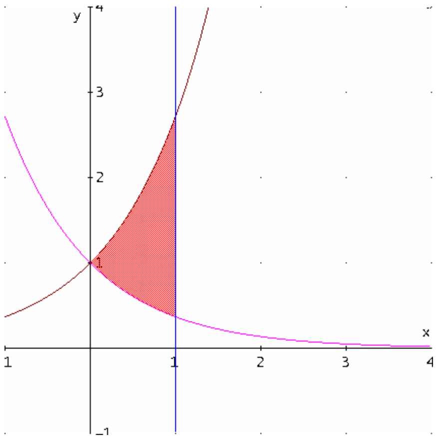
```
#1:   f(x) := EXP(x)

#2:   g(x) := EXP(-x)

#3:   x = 1
```

Z hraničních křivek stačí zobrazit jenom `#3`. Grafy funkcí `f(x)`, `g(x)` jsou součástí zobrazení následující funkce `AreaBetweenCurves`.

```
#4:   AreaBetweenCurves(f(x), g(x), x, 0, 1)
```

Opět platí, že chceme-li výraz `#4` rovnou zobrazit, musíme nejprve aktivovat volbu 2D grafického okna: **Možnosti → Zjednodušit před vykreslením**. Výsledek vidíme na následujícím obrázku 29.

Do balíčku funkcí `PlotInt` a `AreaBetweenCurves` patří ještě podobné funkce `AreaUnderCurves` a `AreaOverCurves`. Každá z nich se dá použít různými způsoby. Pro další informace doporučuji prostudovat příslušnou kapitolu nápovědy (**Obsah Nápovědy (Contents) → Knihovna funkcí → Grafické funkce → Zobrazení oblastí grafů a určitých integrálů**).



Obr. 29: Funkce `AreaBetweenCurves(f(x),g(x),x,0,1)`

Příklad 16.4: Určete objem tělesa vytvořeného rotací rovinného obrazce ohraničeného grafy funkcí $f: y = x^2$, $g: y = 1 - x^2$ kolem osy x.

Řešení:

#1: $f(x) := x^2$

#2: $g(x) := 1 - x^2$

Určíme x-ové souřadnice průsečíků křivek:

#3: `P := SOLUTIONS(f(x) = g(x), x)`

#4: $$P := \left[\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}\right]$$

Pokud nám nevyhovuje pořadí, můžeme složky vektoru uspořádat funkcí `SORT`. Vznikne tak nový vektor R, jehož složky použijeme jako meze integrálu.

#5: `R := SORT(P)`

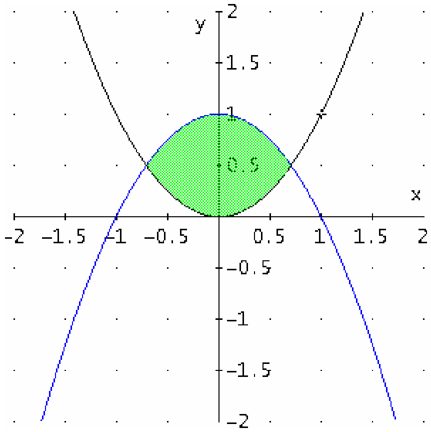
#6: $$\int_{R\downarrow 1}^{R\downarrow 2} \pi\cdot(g(x)^2 - f(x)^2)\, dx$$

Objem tělesa je

#7: $$\frac{2\cdot\sqrt{2}\cdot\pi}{3}$$

K zobrazení tvořícího obrazce tělesa využijeme funkci `AreaBetweenCurves`:

#8: $\text{AreaBetweenCurves}(g(x),\ f(x),\ x,\ R_1,\ R_2,\ y)$

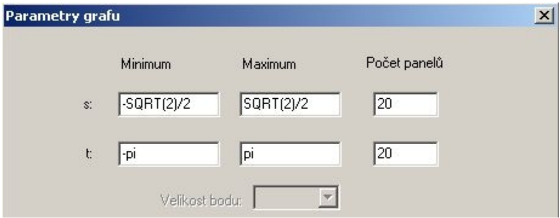


Samotné rotační těleso zobrazíme pochopitelně v 3D grafickém okně. Použijeme k tomu parametrickou reprezentaci dvou rotačních ploch, které tvoří jeho plášť. Každá z těchto ploch vznikne rotací grafu jedné z funkcí `f(x)`, `g(x)` kolem osy `x`.

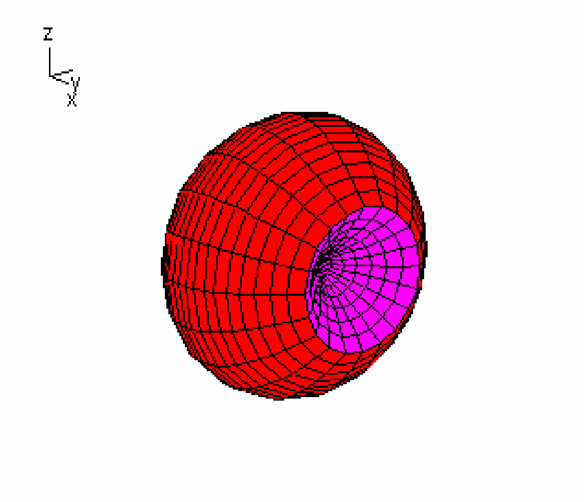
#9: $[s,\ g(s)\cdot\text{COS}(t),\ g(s)\cdot\text{SIN}(t)]$

#10: $[s,\ f(s)\cdot\text{COS}(t),\ f(s)\cdot\text{SIN}(t)]$

Abychom dostali skutečně jenom zkoumané těleso, musíme nastavit meze parametru `s` (který plní roli *x*–ové souřadnice). Proto k zobrazení výrazů #9, #10 v 3D grafickém okně použijeme místo tlačítka ⍊ raději posloupnost příkazů **Vložit** → **Graf...** z nabídky tohoto grafického okna. Tak se nám totiž otevře dialogové okno „Parametry grafu“, v němž vyplníme potřebné hodnoty:



V dialogu, který následuje, můžeme, ale nemusíme, změnit i barevné vlastnosti grafu. Po dokončení procedury dostaneme vyšetřované těleso.



Obr. 30: Rotační těleso – řešení Příkladu 16.4

16.3. Integrální součty. Balíček funkcí „Integraly.mth“

Díky programovacímu jazyku Derive (viz kapitola 10) můžeme vestavěné funkce programu obohatit o funkce vlastnoručně naprogramované a tím program co nejvíce přizpůsobit našim záměrům. Takové funkce nám umožní soustředit sílu programu i žákovu pozornost rovnou na probíranou skutečnost. Veškeré pomocné výpočty, jejichž provádění by pro daný vzdělávací účel nebylo efektivní, jsou skryté. V této kapitole si představíme balíček takovýchto funkcí, které byly vytvořeny na podporu výkladu definice určitého integrálu pomocí dolního a horního integrálního součtu.

Balíček, kterému budeme říkat „Integrály“, je tvořen čtyřmi funkcemi pro zobrazení a výpočet dolního, resp. horního integrálního součtu:

```
DolniS_G(f,a,b,n),    DolniS(f,a,b,n),
HorniS_G(f,a,b,n),    HorniS(f,a,b,n),
```

kde `f` je předpis funkce, `a`, `b` představují dolní a horní mez a n je počet intervalů příslušného dělení. Definice funkcí uložíme do balíčku funkcí, který pojmenujeme `Integraly.mth` (kompletní kód je uveden v přílohách). Nyní můžeme graficky i početně demonstrovat význam dolních a horních integrálních součtů pro výpočet určitého integrálu spojité funkce.

Uvažujme například určitý integrál od dvou do deseti funkce `m(x) := LOG(x)`. Platí

$$\int_{2}^{10} m(x)\, dx = 13.63955656.$$

Po načtení balíčku funkcí `Integraly.mth` můžeme začít zkoumat, jak se mění hodnoty dolního a horního integrálního součtu s rostoucím počtem intervalů dělení a jak souvisejí s výše uvedenou hodnotou odpovídajícího určitého integrálu.

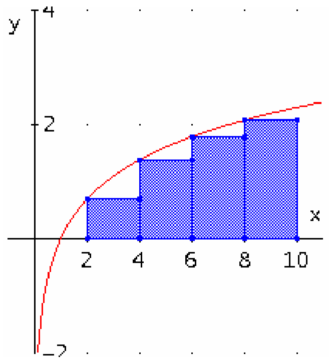
Na následujících obrázcích vidíme výsledky zobrazení funkcí

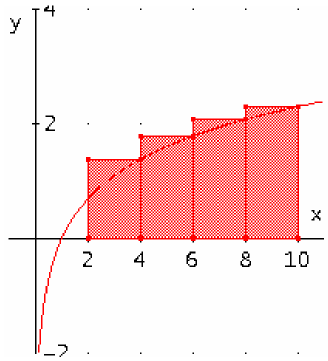
```
DolniS_G(m(x),2,10,4), HorniS_G(m(x),2,10,4),
DolniS_G(m(x),2,10,8), HorniS_G(m(x),2,10,8)
```

v 2D-grafickém okně spolu s výpočty hodnot odpovídajících součtů.

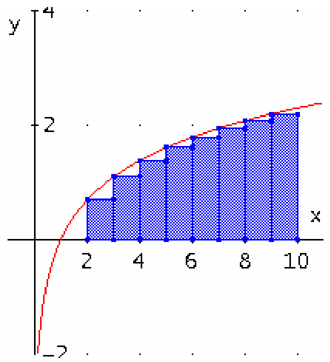
Pro správnou činnost grafických funkcí je třeba nastavit tyto volby 2D grafického okna:

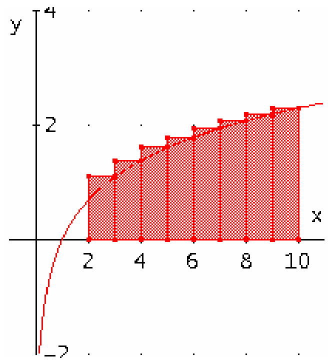
Možnosti → Zjednodušit před vykreslením
Možnosti → Zobrazení → Barva → vypnout automatickou změnu barvy
Možnosti → Zobrazení → Body → Spojovat Ano





`DolniS(m(x),2,10,4) = 11.90128510,` `HorniS(m(x),2,10,4) = 15.12016093`





`DolniS(m(x),2,10,8) = 12.80182748,` `HorniS(m(x),2,10,8) = 14.41126539`

Již z těchto dvou ukázek je patrné, že při postupném zjemňování dělení intervalu se hodnoty dolního a horního součtu přibližují, každý ze své strany, k hodnotě vypočítané.

Poznámka: Kompletní kód funkcí definovaných v balíčku `Integraly.mth` je uveden v přílohách

17. Rovnice a nerovnice

Z nástrojů Derive se při řešení rovnic kromě vestavěných funkcí (`SOLVE`, `SOLUTIONS`, `NSOLVE`, `NSOLUTIONS`), které jsou určeny pro symbolické i numerické řešení, dobře uplatní zejména grafické funkce programu a možnost provádět jednotlivé úpravy rovnice krok za krokem

17.1. Úpravy rovnic

Derive umožňuje provádět úpravy rovnic krok za krokem, tak, jak postupujeme při počítání na papíře či tabuli. Velice jednoduše můžeme provádět různé početní operace s oběma stranami rovnice současně a neustále sledovat vliv těchto úprav na dílčí rovnice, od zadání až po výsledek. Pomocí příkazu pro substituci můžeme potom snadno vykonat zkoušku.

Příklad 17.1: Řešte rovnici $2x + 5 = 27$

Řešení:

```
#1: 2x + 5 = 27
```

Jak bylo řečeno výše, program Derive umožňuje provádět úpravy rovnic přímo na pracovní ploše.

Chceme-li například od obou stran výše uvedené rovnice odečíst 5, napíšeme na příkazový řádek jednoduše `#1–5` (tj. na rovnici se odkazujeme prostřednictvím proměnné `#1`) nebo rovnici zkopírujeme do závorek pomocí `F4` a odečteme od ní onu pětku.

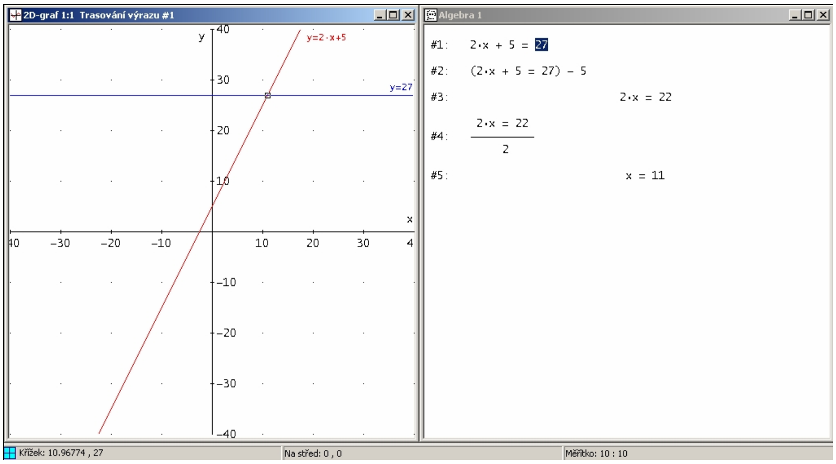
Na následujícím řádku okna Algebra potom dostaneme (též Obr. 30):

```
#2: (2x + 5 = 27) – 5
```

Po zjednodušení příkazem **Zjednodušit → Základní zjednodušení** ($\boxed{=}$) dostaneme výsledek

```
#3:                              2x = 22
```

V dalších úpravách pokračujeme ve stejném duchu (Obr. 30). Početní výsledek potom můžeme konfrontovat s grafickým řešením rovnice. To dostaneme tak, že zobrazíme zvlášť levou a zvlášť pravou stranu rovnice (Obr. 30, grafické okno).



Obr. 30: Úpravy rovnice a grafické řešení

17.2. Symbolické a grafické řešení rovnic.

Program Derive umožňuje jednoduše a rychle kombinovat symbolické, numerické a grafické řešení rovnic.

Příklad 17.2: V R řešte rovnici $\sqrt{2|x| - x^2} = \frac{1}{2}$

Naformátováno: Řádkování: jednoduché, Ohraničení: Pole: (stínované jednoduché, Automatická, 1 b. šířka čáry)

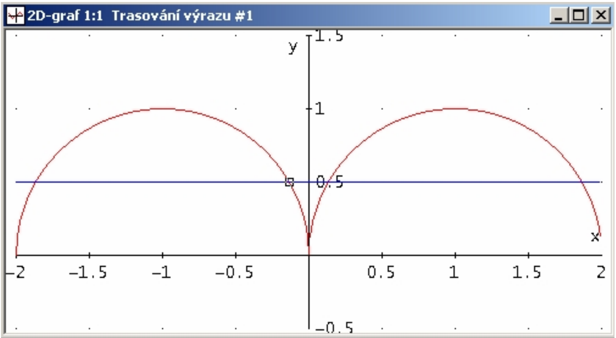
Řešení:

Odešleme rovnici do okna Algebra a necháme ji zvýrazněnu

#1: $\sqrt{(2\cdot|x| - x^2)} = \frac{1}{2}$

V 2D grafickém okně zobrazíme zvlášť levou a zvlášť pravou stranu rovnice. K odhadu řešení použijeme režim Trasovat grafy, který aktivujeme (i deaktivujeme) stisknutím tlačítka . Grafické řešení rovnice vidíme na následujícím obrázku.

Z obrázku také hned vidíme, že zajímavou úlohou by bylo řešení příslušné parametrické rovnice $\sqrt{2|x| - x^2} = a$ s reálným parametrem *a*.



Symbolické řešení provedeme již známým způsobem pomocí posloupnosti příkazů **Řešit → Výraz…**

V příslušném dialogovém okně „Řešit výraz" (Obr. 6) vybereme pro symbolické řešení metodu **Algebraickou** a obor reálných čísel.
Chceme-li znát výsledek i ve formě desetinného čísla (viz #4), zvýrazníme na pracovní ploše výsledek symbolického řešení #3 a stiskneme tlačítko $\boxed{\approx}$ (**Aproximovat**).

#2: $\mathrm{SOLVE}\left(\sqrt{(2\cdot|x| - x^2)} = \frac{1}{2},\ x,\ \mathrm{Real}\right)$

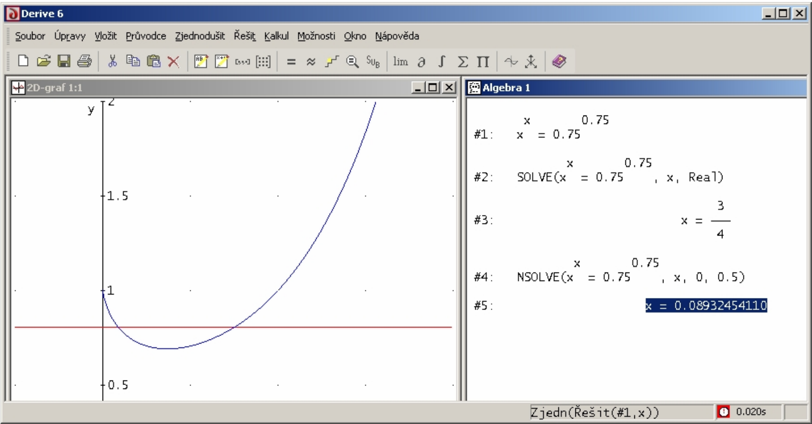
#3: $x = -\frac{\sqrt{3}}{2} - 1 \vee x = 1 - \frac{\sqrt{3}}{2} \vee x = \frac{\sqrt{3}}{2} - 1 \vee x = \frac{\sqrt{3}}{2} + 1$

#4: $x = -0.1339745962 \vee x = 0.1339745962 \vee x = -1.866025403 \vee x = 1.866025403$

17.3. Numerické řešení rovnice

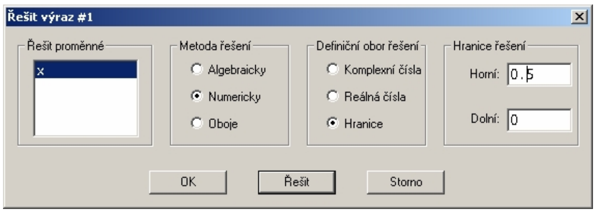
PŘÍKLAD 17.3: Řešte v R rovnici $x^x = 0.75^{0.75}$.

Řešení:
Rovnici řešíme nejprve přímo, pomocí posloupnosti **Řešit** → **Výraz...**
Výsledkem je jenom jedno řešení x = 3/4, které je ale zřejmé již při letmém pohledu na rovnici (Obr.31, řádky #2, #3). Otázkou je, zda je to jediné řešení dané rovnice. Odpověď nám pomůže najít grafické řešení rovnice. Postup známe z předchozího příkladu. Výsledek vidíme na Obr.31. Teď už tedy víme, že úloha má řešení dvě.



Obr. 31: Numerické řešení rovnice

Vzhledem k povaze rovnice (transcendentní) je zřejmé, že ono druhé řešení lze určit pouze přibližně, numericky. Abychom numerickou metodou dostali všechna řešení rovnice, je většinou nutné programu v dialogovém okně „Řešit výraz“



zadat pro každý kořen hranice intervalu, v němž má být hledán (při řešení příkladu dle Obr. 31 jsme použili hranice 0 a 0.5). Pro určení všech těchto intervalů je ideální použít grafické řešení rovnice.
Dalším příkladem řešení rovnice kombinací grafické a numerické metody je Příklad 13.5 uvedený na straně 63.

17.4. Goniometrické rovnice

Goniometrické rovnice můžeme řešit v radiánech nebo ve stupních. Je třeba vědět, že Derive „neumí“ zapsat nekonečně mnoho řešení rovnice. Vždy vypíše pouze několik kořenů z okolí počátku. Proto je dobré výsledky příkazu `SOLVE` (`SOLUTIONS`) vždy konzultovat s grafem a se svým úsudkem.

PŘÍKLAD 17.4: Řešte v R rovnici $\sin(x+\frac{p}{2})=\frac{1}{2}$

Řešení:

Předpokládáme originální nastavení programu, při kterém je zvolena reprezentace úhlů v radiánech, tj. na kartě **Možnosti** → **Nastavení…** → **Zjednodušení** je v poli **Jednotka úhlu** uvedeno „Radian“ . Řešení provádíme pomocí posloupnosti příkazů **Řešit** → **Výraz...** grafického rozhraní programu.

#1: $\mathrm{SIN}\left(x+\frac{\pi}{2}\right)=\frac{1}{2}$

#2: $\mathrm{SOLVE}\left(\mathrm{SIN}\left(x+\frac{\pi}{2}\right)=\frac{1}{2},\ x,\ \mathrm{Real}\right)$

#3: $x=\frac{5\cdot\pi}{3} \vee x=-\frac{\pi}{3} \vee x=\frac{\pi}{3}$

Řešení stejné rovnice ve stupních:
Pokud nechceme měnit nastavení jednotky úhlu na výše uvedené kartě, stačí použít symbol stupně ° (nebo slovo `DEG`), který napíšeme stisknutím příslušného tlačítka na panelu matematických symbolů. Jinak uvidíme, že je pak jedno, zda napíšeme fázové posunutí pomocí stupňů nebo radiánů.

1. možnost

#4: $\mathrm{SIN}\left(x^{\circ}+\frac{\pi}{2}\right)=\frac{1}{2}$

#5: $\mathrm{SOLVE}\left(\mathrm{SIN}\left(x^{\circ}+\frac{\pi}{2}\right)=\frac{1}{2},\ x,\ \mathrm{Real}\right)$

#6: $x=300 \vee x=-60 \vee x=60$

2. možnost

$$\#7:\quad \mathrm{SIN}(x° + 90°) = \frac{1}{2}$$

$$\#8:\quad \mathrm{SOLVE}\left(\mathrm{SIN}(x° + 90°) = \frac{1}{2},\ x,\ \mathrm{Real}\right)$$

$$\#9:\quad x = 300 \vee x = -60 \vee x = 60$$

Jak uvidíme v následujícím příkladě, někdy je třeba programu pomoci s úpravou rovnice.

Příklad 17.5: Řešte v R rovnici $\cos 3x + 2\cos x = 0$.

Řešení:

$$\#1:\quad \mathrm{COS}(3\cdot x) + 2\cdot \mathrm{COS}(x) = 0$$

Přímé řešení příkazem SOLVE nepřinese výsledek:

$$\#2:\quad \mathrm{SOLVE}(\mathrm{COS}(3\cdot x) + 2\cdot \mathrm{COS}(x) = 0,\ x,\ \mathrm{Real})$$

$$\#3:\quad \mathrm{COS}(3\cdot x) + 2\cdot \mathrm{COS}(x) = 0$$

Pomůžeme tedy programu nastavením vhodného směru úprav goniometrických vzorců:

Možnosti → Nastavení... → Zjednodušení → Trigonometrická → Expand

#4: Trigonometry := Expand

Rovnici poté ještě podrobíme základnímu zjednodušení (=).

$$\#5:\quad 4\cdot \mathrm{COS}(x)^3 - \mathrm{COS}(x) = 0$$

Nyní ji bez problémů vyřešíme:

$$\#6:\quad \mathrm{SOLVE}(4\cdot \mathrm{COS}(x)^3 - \mathrm{COS}(x) = 0,\ x,\ \mathrm{Real})$$

#7:

$$x = \frac{5\cdot\pi}{3} \vee x = \frac{4\cdot\pi}{3} \vee x = -\frac{2\cdot\pi}{3} \vee x = \frac{2\cdot\pi}{3} \vee x = -\frac{\pi}{3} \vee$$

$$x = \frac{\pi}{3} \vee x = \frac{3\cdot\pi}{2} \vee x = -\frac{\pi}{2} \vee x = \frac{\pi}{2}$$

17.5. Iracionální rovnice

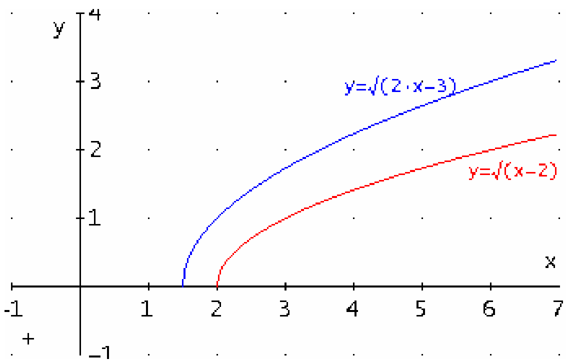
Doporučení z předchozí části textu, konfrontovat výsledky řešení rovnic příkazem `SOLVE` s řešením grafickým a s vlastním úsudkem, platí nejenom pro goniometrické rovnice. V následujících příkladech si ukážeme, že je třeba být obezřetný i v případě iracionálních a logaritmických rovnic.

Příklad 17.6: Řešte v R rovnici $\sqrt{x-2} = \sqrt{2x-3}$

Řešení:

```
#1:    √(x - 2) = √(2·x - 3)

#2:    SOLVE(√(x - 2) = √(2·x - 3), x, Real)

#3:                                 x = 1
```

Výsledek `x = 1` ale přece nepatří do definičního oboru rovnice! Že rovnice nemá v oboru reálných čísel žádné řešení je patrné i z jejího grafického řešení:



Výsledek zkoušky dosazením (**Zjednodušit → Substituce proměnné...**) odhalí příčinu této zdánlivé chyby:

```
#4:    √(1 - 2) = √(2·1 - 3)

#5:                                 i = i
```

Derive totiž řeší všechny rovnice v oboru komplexních čísel. Bohužel, bez ohledu na nastavený definiční obor proměnné (**Průvodce → Definiční obor proměnné...**).

17.6. Logaritmické rovnice

Připomeňme, co již zaznělo v kapitole 11.3, věnované logaritmickým výrazům. Výrazy `LOG(x)` a `LN(x)` shodně reprezentují přirozený logaritmus. Dekadický logaritmus musíme zapsat výrazem `LOG(x,10)`. Pro řešení rovnic je zásadní informací, že logaritmus je v Derive definován v komplexnímu oboru a má smysl i pro záporné hodnoty argumentu. V následující úloze vidíme, co to může způsobit.

Příklad 17.7: Řešte v R rovnici: $\log(x+4)+\log(x-2)=\log(x+3)$

Řešení:

#1: LOG(x + 4) + LOG(x − 2) = LOG(x + 3)

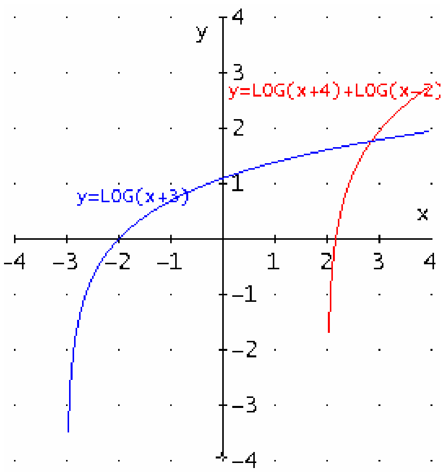
#2: SOLVE(LOG(x + 4) + LOG(x − 2) = LOG(x + 3), x, Real)

#3: $x = -\frac{3\cdot\sqrt{5}}{2} - \frac{1}{2} \vee x = \frac{3\cdot\sqrt{5}}{2} - \frac{1}{2}$

#4: x = −3.854101966 ∨ x = 2.854101966

První řešení evidentně nepatří do definičního oboru rovnice.

Grafické řešení ukazuje, že rovnice má jediné reálné řešení:



18. Soustavy lineárních rovnic

18.1. Přímé řešení

Přímým řešením zde rozumíme řešení s použitím funkce `SOLVE` nebo `SOLUTIONS`. K provedení funkce `SOLVE` použijeme buď rozhraní programu (potom příkaz nemusíme znát), nebo ho rovnou napíšeme do příkazového řádku. V případě funkce `SOLUTIONS` připadá v úvahu jenom druhá možnost, příkazový řádek.

18.1.1. Soustava s jediným řešení

Příklad 18.1: Řešte v R soustavu lineárních rovnic

$$x+3y+z=5$$
$$2x+y+z=2$$
$$x+y+5z=-7$$

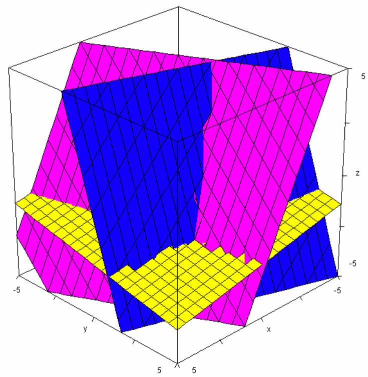
Naformátováno: Řádkování: jednoduché, Ohraničení: Pole: (stínované jednoduché, Automatická, 1 b. šířka čáry)

Řešení:

Posloupnost příkazů **Řešit → Soustavu rovnic...** vede k „formuláři“, do kterého rovnice zapíšeme (pokud je již máme na pracovní ploše, tak je překopírujeme nebo přeneseme pomocí F3). Potom stiskneme tlačítko **Řešit**.

Algebraické řešení můžeme snadno a rychle doplnit obrázkem. Stačí postupně zvýrazňovat rovnice, které se objeví na pracovní ploše jako parametry příkazu `SOLVE` (viz #1), a dříve popsaným způsobem je zobrazovat v 3D grafickém okně.

```
#1: SOLVE([x + 3·y + z = 5, 2·x + y + z = 2, x + y + 5·z = −7],
          [x, y, z])

#2:                       [x = 1 ∧ y = 2 ∧ z = −2]
```



Obr. 32: Grafické řešení soustavy rovnic

Naformátováno: Písmo: 11

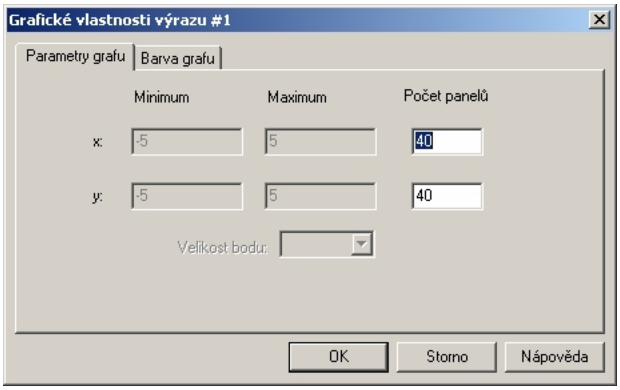
Naformátováno: Písmo: 11

Naformátováno: Písmo: 11 b., není Tučné

Naformátováno: Písmo: 11 b., není Tučné

Naformátováno: Písmo: 11 b.

Úpravy grafu
Jak vidíme na obrázku 32, grafické znázornění řešení soustavy by potřebovalo ještě doladit. Průsečnice rovin jsou zubaté a někomu by mohla vadit i viditelná síťová struktura ploch. Síť snadno skryjeme, zubaté průsečnice můžeme zjemnit, ideálními přímkami ale nikdy nebudou.
Při změně parametrů již existující plochy tuto plochu nejprve označíme kliknutím myší (síť plochy zbělá), potom vyvoláme posloupnost příkazů 3D grafického okna **Úpravy → Graf...** (stejného efektu dosáhneme užitím pravého tlačítka myši).
Objeví se toto dialogové okno:



Pokud chceme dostat hladší plochy a zjemnit jejich průsečnice, zvýšíme na kartě **Parametry grafu** počty panelů, které určují hustotu sítě plochy. Bohužel, při zvyšování počtu panelů se nepříjemně prodlužuje doba zobrazování plochy. Na kartě **Barva grafu** potom můžeme kromě nastavení barvy grafu také vypnout zobrazení sítě plochy.

Pro okamžité nastavení parametrů grafu hned při jeho zobrazování použijeme posloupnost příkazů **Vložit → Graf**, jak je uvedeno v kapitole 9.2.

Funkce `SOLUTIONS`
Pokud chceme s řešením soustavy dále manipulovat, je výhodnější použít funkci `SOLUTIONS`. Ta sice není „ošetřena“ grafickým rozhraním, ale její syntaxe je totožná s funkcí `SOLVE`. Nejsnadnější je tedy výraz #1 přenést na vstupní řádek pomocí klávesy F3 a slovo `SOLVE` přepsat na `SOLUTIONS`:

```
SOLUTIONS([x + 3·y + z = 5, 2·x + y + z = 2, x + y + 5·z = −7],
[x, y, z])
```

Tento výraz potom odešleme na pracovní plochu, kde provedeme jeho základní zjednodušení. Výsledkem je vektor řešení, k jehož složkám se dostaneme pomocí dolního indexu (operátor ↓ nebo `SUB`).

18.1.2. Soustava, která nemá řešení

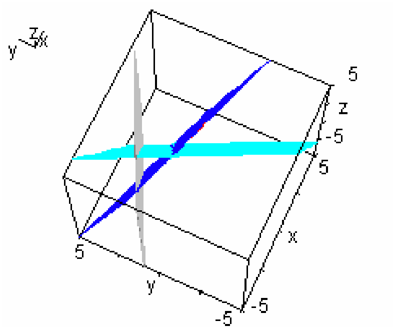
Pokud nemá soustava rovnic řešení, výsledkem funkce `SOLVE` (`SOLUTIONS`) jsou prázdné závorky `[]` (viz výraz `#2` v řešení následujícího příkladu).

Příklad 18.2: Řešte v *R* soustavu lineárních rovnic:

$$
\begin{aligned}
x+3y+z&=5\\
2x+y+z&=2\\
x-2y&=-7
\end{aligned}
$$

Řešení:

```
#1:
SOLVE([x + 3·y + z = 5, 2·x + y + z = 2, x - 2·y = -7], [x, y, z])

#2:                                   []
```



18.1.3. Soustava s nekonečně mnoha řešeními

Příklad 18.3: Řešte v *R* soustavu lineárních rovnic:

$$
\begin{aligned}
x+3y+z&=5\\
2x+y+z&=2\\
-5y-z&=-8
\end{aligned}
$$

Řešení:
Porovnejme dva způsoby řešení této soustavy:

1. Řešíme vzhledem ke všem třem neznámým *x, y, z.*

```
#1:  SOLVE([x + 3·y + z = 5, 2·x + y + z = 2, - 5·y - z = -8],
           [x, y, z])
```

#2: $$\left[x + \frac{2\cdot z}{5} = \frac{1}{5} \wedge y + \frac{z}{5} = \frac{8}{5}\right]$$

2. Řešíme vzhledem k vybraným „základním" neznámým *x, y.*

```
#3:
SOLVE([x + 3·y + z = 5, 2·x + y + z = 2, - 5·y - z = -8], [x, y])
```

#4: $$\left[x = \frac{1 - 2\cdot z}{5} \wedge y = \frac{8 - z}{5}\right]$$

Zápis #4 odpovídá, na rozdíl od výsledku #2, obvyklému způsobu zápisu řešení takovéto soustavy.

Závěr
Jak uvidíme ještě v příkladu 18.5, při řešení soustavy rovnic, která má nekonečně mnoho řešení, je vhodnější druhý z výše uvedených způsobů použití příkazu SOLVE. Ten spočívá v tom, že v příkazu uvádíme jako neznámé pouze tzv. základní neznámé dané soustavy.

18.2. Užití matic při řešení soustav lineárních rovnic

Budeme používat tyto funkce Derive: APPEND_COLUMNS, FORCE0, PIVOT, RANK, SUBRACT_ELEMENTS, SWAP_ELEMENTS, ROW_REDUCE, MINOR, DET.

Frobeniova podmínka.
Ověření řešitelnosti soustavy. Převádíme rozšířenou matici soustavy na Gaussův tvar. Zjišťujeme hodnost matice soustavy a rozšířené matice soustavy.

Příklad 18.4: Řešte v *R* soustavu lineárních rovnic

$$x + 3y + z = 5$$
$$2x + y + z = 2$$
$$x + y + 5z = -7$$

Naformátováno: Řádkování: jednoduché, Ohraničení: Pole: (stínované jednoduché, Automatická, 1 b. šířka čáry)

Řešení:
K dané soustavě musíme sami vytvořit matici soustavy:

#5: $A := \begin{bmatrix} 1 & 3 & 1 \\ 2 & 1 & 1 \\ 1 & 1 & 5 \end{bmatrix}$

a vektor pravých stran:

#6: $B := \begin{bmatrix} 5 \\ 2 \\ -7 \end{bmatrix}$

Vytvoříme rozšířenou matici soustavy:

#7: `Aroz := APPEND_COLUMNS(A, B)`

#8: $\mathrm{Aroz} = \begin{bmatrix} 1 & 3 & 1 & 5 \\ 2 & 1 & 1 & 2 \\ 1 & 1 & 5 & -7 \end{bmatrix}$

a ověříme splnění Frobeniovy podmínky:

#9: `RANK(A) = 3`

#10: `RANK(Aroz) = 3`

Eliminace
Soustavu můžeme řešit přímo, pomocí funkce `ROW_REDUCE`, která odpovídá provedení Gaussovy-Jordanovy eliminace. Výsledný tvar matice se nazývá „row echelon form“.

#11: $\mathrm{ROW\_REDUCE}(A, B) = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & -2 \end{bmatrix}$

Řešení soustavy dostaneme v posledním sloupci této matice, tj. $x = 1$, $y = 2$, $z = -2$.

Program Derive nám nabízí i funkce, s jejichž pomocí můžeme eliminaci rozšířené matice provádět postupně, krok za krokem, stejně jako na papíře. Jedná se o funkce `PIVOT`, `FORCE0`, `SWAP_ELEMENTS` a `SUBTRACT_ELEMENTS`. Jejich přesný význam a způsob použití zde nebudeme objasňovat, vše potřebné najde čtenář v nápovědě.

```
#12:  Aroz1 ≔ PIVOT(Aroz, 1, 1)
```

$$\#13:\quad Aroz1 = \begin{bmatrix} 1 & 3 & 1 & 5 \\ 0 & -5 & -1 & -8 \\ 0 & -2 & 4 & -12 \end{bmatrix}$$

$$\#14:(Aroz2 ≔ PIVOT(Aroz1, 2, 2)) = Aroz2 ≔ \begin{bmatrix} 1 & 3 & 1 & 5 \\ 0 & -5 & -1 & -8 \\ 0 & 0 & \frac{22}{5} & -\frac{44}{5} \end{bmatrix}$$

Užití inverzní matice
V případě regulární soustavy můžeme k rychlému řešení použít inverzní matici k matici soustavy A.

$$\#15:\quad A^{-1}\cdot B = \begin{bmatrix} 1 \\ 2 \\ -2 \end{bmatrix}$$

Dostáváme tak již známé řešení $x = 1, y = 2, z = -2$.

Cramerovo pravidlo
Při odkazování na části matice (prvky, řádky, sloupce) využijeme klíčová slova `SUB`, `SUB SUB` (přesněji řečeno, s nimi alternativní symboly `↓`, `↓↓`), která uvádějí dolní indexy maticové proměnné. Ocenit můžeme i funkci `MINOR` pro vytvoření submatice. Detailní informace o těchto i dalších nástrojích pro práci s maticemi najde čtenář v nápovědě. Zde si jenom pro usnadnění pochopení následujících řádků řekněme, že výrazem `A↓↓[2, 3]` rozumíme matici, která je tvořena jenom 2. a 3. sloupcem matice A, výrazem `A↓↓[1]` pak matici (sloupcový vektor) tvořenou pouze prvním sloupcem matice A.

```
#16:  A1 ≔ APPEND_COLUMNS(B, A↓↓[2, 3])

#17:  A2 ≔ APPEND_COLUMNS(A↓↓[1], B, A↓↓[3])

#18:  A3 ≔ APPEND_COLUMNS(A↓↓[1, 2], B)
```

$$\#19:\quad A1 = \begin{bmatrix} 5 & 3 & 1 \\ 2 & 1 & 1 \\ -7 & 1 & 5 \end{bmatrix}$$

#20: $A2 = \begin{bmatrix} 1 & 5 & 1 \\ 2 & 2 & 1 \\ 1 & -7 & 5 \end{bmatrix}$

#21: $A3 = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 1 & 2 \\ 1 & 1 & -7 \end{bmatrix}$

#22: $\left(x :\!= \dfrac{\mathrm{DET(A1)}}{\mathrm{DET(A)}} \right) = x :\!= 1$

#23: $\left(y :\!= \dfrac{\mathrm{DET(A2)}}{\mathrm{DET(A)}} \right) = y :\!= 2$

#24: $\left(z :\!= \dfrac{\mathrm{DET(A3)}}{\mathrm{DET(A)}} \right) = z :\!= -2$

18.3. Soustava s nekonečně mnoha řešeními

Příklad 18.5: Rozhodněte o řešitelnosti dané soustavy lineárních rovnic. Pokud řešení existuje, určete ho.

$$\begin{aligned} x_1 + 2x_2 - x_3 + 3x_4 &= 1 \\ -3x_1 - 6x_2 + 5x_3 - 10x_4 &= -1 \\ 2x_1 + 4x_2 + 5x_4 &= 4 \\ x_1 + 2x_2 + x_3 + 2x_4 &= 3. \end{aligned}$$

Řešení:

Přímým řešením zjistíme, že soustava má nekonečně mnoho řešení. Výsledek však není ve tvaru, obvyklém pro zápis nekonečně mnoha řešení.

#1: InputMode := Word

#2: SOLVE([x1 + 2·x2 - x3 + 3·x4 = 1, - 3·x1 - 6·x2 + 5·x3 - 10·x4 = -1, 2·x1 + 4·x2 + 5·x4 = 4, x1 + 2·x2 + x3 + 2·x4 = 3], [x1, x2, x3, x4])

#3: $\left[x1 + 2 \cdot x2 + \dfrac{5 \cdot x4}{2} = 2 \wedge x3 - \dfrac{x4}{2} = 1 \right]$

Abychom zjistili, které neznámé lze volit libovolně (nezákladní neznámé) a které ne (základní neznámé), převedeme rozšířenou matici soustavy do Gaussova tvaru:

#4: $A := \begin{bmatrix} 1 & 2 & -1 & 3 \\ -3 & -6 & 5 & -10 \\ 2 & 4 & 0 & 5 \\ 1 & 2 & 1 & 2 \end{bmatrix}$

#5: $B := \begin{bmatrix} 1 \\ -1 \\ 4 \\ 3 \end{bmatrix}$

K matici soustavy A připojíme sloupec pravých stran B příkazem `APPEND_COLUMNS`

#6: `Ar := APPEND_COLUMNS(A, B)`

#7: $Ar := \begin{bmatrix} 1 & 2 & -1 & 3 & 1 \\ -3 & -6 & 5 & -10 & -1 \\ 2 & 4 & 0 & 5 & 4 \\ 1 & 2 & 1 & 2 & 3 \end{bmatrix}$

Gaussovu eliminaci provedeme postupnou aplikací funkce `PIVOT`

#8: `Ar1 := PIVOT(Ar, 1, 1)`

#9: $Ar1 := \begin{bmatrix} 1 & 2 & -1 & 3 & 1 \\ 0 & 0 & 2 & -1 & 2 \\ 0 & 0 & 2 & -1 & 2 \\ 0 & 0 & 2 & -1 & 2 \end{bmatrix}$

#10: `Ar2 := PIVOT(Ar1, 2, 3)`

#11: $$Ar2 := \begin{bmatrix} 1 & 2 & -1 & 3 & 1 \\ 0 & 0 & 2 & -1 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Nyní již je zřejmé, které proměnné volit jako (základní) neznámé (x1, x3) a které nechat jako parametry (nezákladní neznámé) (x2, x4).

```
#12:  SOLVE([x1 + 2·x2 - x3 + 3·x4 = 1, - 3·x1 - 6·x2 + 5·x3 -
10·x4 = -1, 2·x1 + 4·x2 + 5·x4 = 4, x1 + 2·x2 + x3 + 2·x4 = 3],
[x1, x3])
```

#13: $$\left[x1 = -\frac{4\cdot x2 + 5\cdot x4 - 4}{2} \wedge x3 = \frac{x4 + 2}{2}\right]$$

Podobně jako příkaz `PIVOT` můžeme k eliminaci matice použít i příkaz `FORCE0` (`syntaxe FORCE0(A,i,j,p)`), kterým však vynulujeme vždy pouze jeden prvek (prvek a_{ij}) pomocí vybraného řádku (p).

K získání kompletních informací o řešení soustav rovnic doporučuji prostudovat následující partie **Nápovědy** programu:

Témata nápovědy → Vybraná témata → Okno Algebra → Řešení rovnic, → Vektory a matice;

Témata nápovědy → Knihovna funkcí → Řešení rovnic, Lineární algebra;

Témata nápovědy → Uživatelské balíčky matematických funkcí.

Naformátováno: Derive Menu Char Char, Písmo: 10 b., není Tučné

Naformátováno: Derive Menu Char Char, Písmo: 10 b.

Naformátováno: Derive Menu Char Char, Písmo: 10 b., není Tučné

Naformátováno: Derive Menu Char Char, Písmo: 10 b., není Tučné

Naformátováno: Derive Menu Char Char, Písmo: 10 b., není Tučné

Naformátováno: Derive Menu Char Char, Písmo: 10 b., není Tučné

Naformátováno: Derive Menu Char Char, Písmo: 10 b., není Tučné

Naformátováno: Písmo: (výchozí) Arial, 11 b., není Tučné

19. Nerovnice a soustavy nerovnic

Při řešení nerovnic oceníme snadnou dostupnost grafického znázornění.

Příklad 19.1: Řešte početně i graficky nerovnici $x^2 + x - 2 \le 0$.

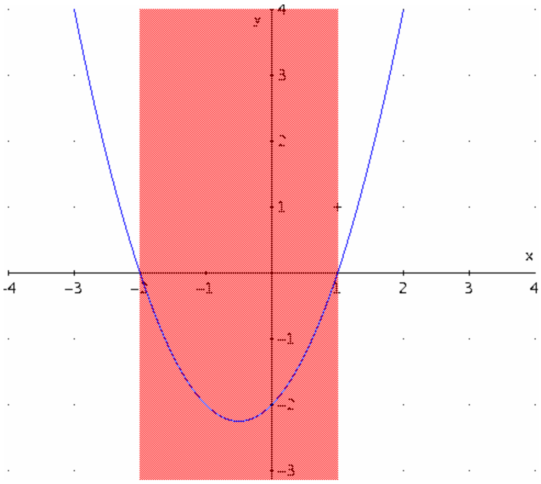
Řešení:

```
      2
#1: x   + x - 2 ≤ 0
```

Řešíme přímo, užitím posloupnosti příkazů **Řešit** → **Výraz...** nebo zápisem odpovídajícího výrazu s funkcí `SOLVE` na příkazový řádek:

```
            2
#2: SOLVE(x   + x - 2 ≤ 0, x, Real)

#3:                         -2 ≤ x ≤ 1
```

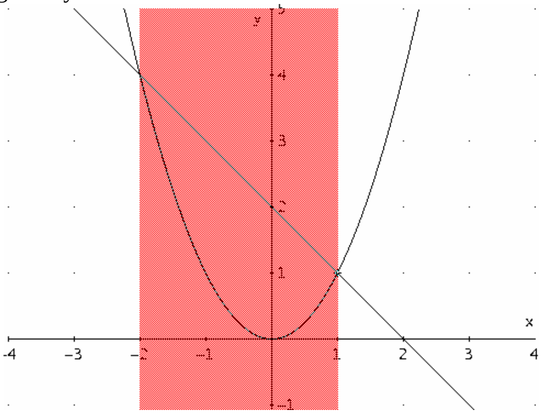
V 2D grafickém okně zobrazíme levou stranu nerovnice (parabola) spolu s jejím řešením ne řádku `#3` (pás):



Nerovnici můžeme také nejprve upravit (viz kapitola 17.1)

```
       2
#4:  (x   + x - 2 ≤ 0) - x + 2

                             2
#5:                         x   ≤ 2 - x
```

a potom řešit graficky:



Případně můžeme levou stranu nerovnice rozložit:

#6: $(x - 1)\cdot(x + 2) \le 0$

Příklad 19.2: V R řešte nerovnici $|x-3|>1$. Řešení graficky znázorněte.

Naformátováno: Řádkování: jednoduché, Ohraničení: Pole: (stínované jednoduché, Automatická, 1 b. šířka čáry)

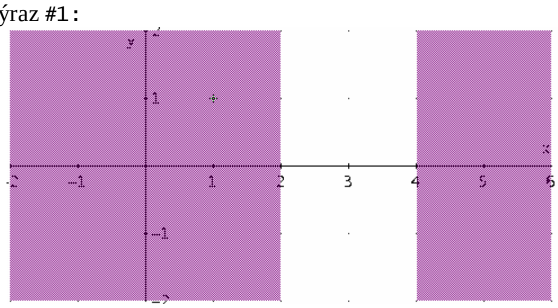
Řešení:

#1: $|x - 3| > 1$

#2: SOLVE($|x - 3| > 1$, x, Real)

#3: $x < 2 \vee x > 4$

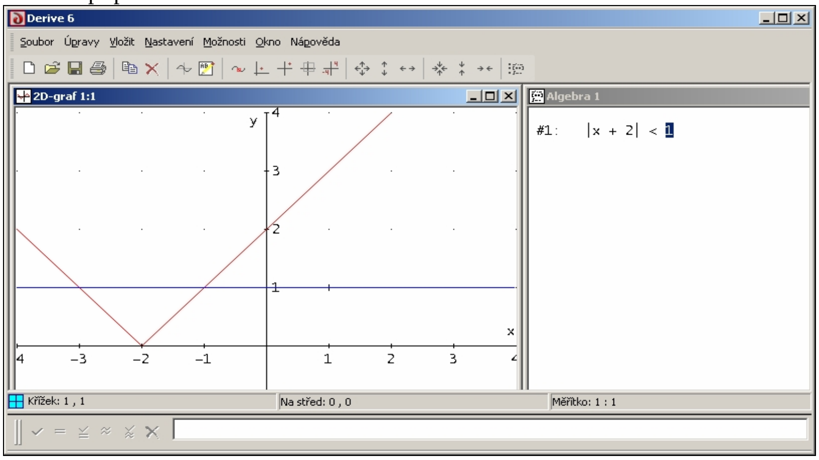
Zobrazíme výraz #1:



Příklad 19.3: Řešte graficky nerovnici $|x+2|<1$

Řešení:

V tomto případě zobrazíme každou stranu nerovnice zvlášť:



Příklad 19.4: Řešte graficky soustavu nerovnic:

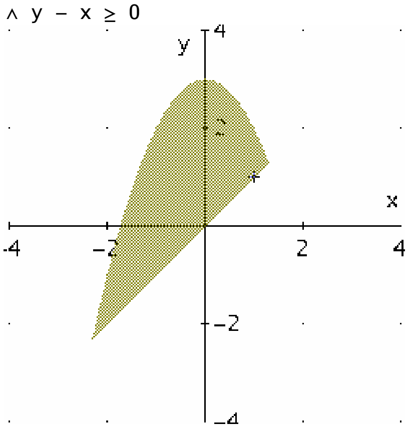
$$x^2 + y \le 3$$

$$y - x \ge 0$$

Řešení:

Soustavu zapíšeme tak, že nerovnice spojíme logickou spojkou `AND`, celý takto vzniklý výraz zvýrazníme a zobrazíme.

$\#1: \quad x^2 + y \le 3 \wedge y - x \ge 0$



Naformátováno: Řádkování: jednoduché, Ohraničení: Pole: (stínované jednoduché, Automatická, 1 b. šířka čáry)

Naformátováno: Písmo: 11

Naformátováno: zarovnání na střed, Řádkování:

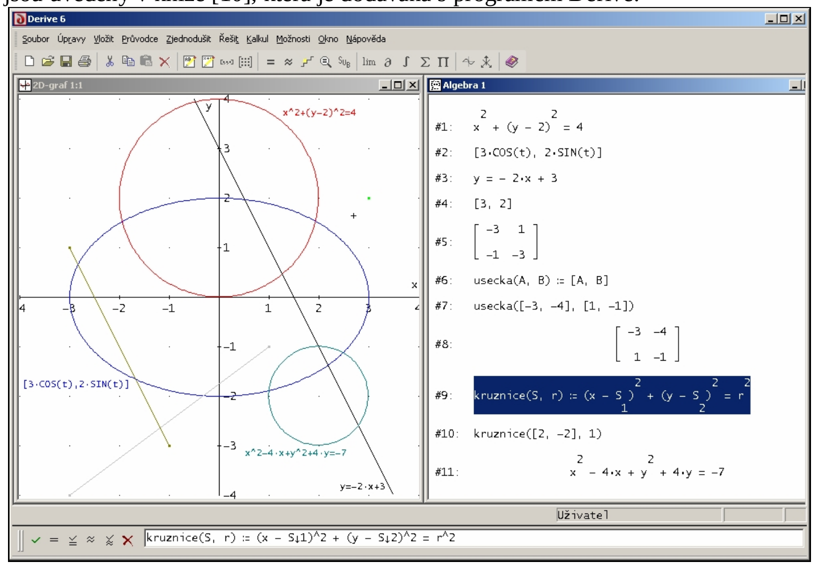
Naformátováno: Řádkování: jednoduché, Ohraničení: Pole: (stínované jednoduché, Automatická, 1 b. šířka čáry)

20. Geometrie

V této kapitole si ukážeme možnosti užití Derive k podpoře výuky geometrie, především analytické metody zkoumání geometrických vztahů. Při výuce geometrie oceníme na Derive jednoduchý přechod od algebraické formule k jejímu grafu, možnost dynamizace grafické reprezentace úlohy užitím posuvníku, srozumitelného zápisu výrazů, který se jenom málo liší od zápisu v učebnicích a možnost programovat vlastní funkce užitím jednoduchého programovacího jazyka. Je zřejmé, že při počítačové podpoře výuky geometrie použijeme především dynamické geometrické programy (DGS - Dynamic Geometry Software) jako například Cabri nebo Cinderella. Má tedy program typu CAS (CAS – computer algebra system), například Derive, v geometrii vůbec nějaké uplatnění? Určitě. Díky úzkému propojení mezi symbolickými formulemi a odpovídajícími geometrickými objekty může sloužit třeba jako nástroj vizualizace analytické metody popisu geometrických vztahů. Nebo, díky možnosti provádět symbolické výpočty, jako nástroj umožňující provést důkazy různých geometrických vlastností. Navíc, pomocí posuvníku můžeme získat dynamickou reprezentaci zkoumaných geometrických vlastností a vztahů.

20.1. Reprezentace geometrických objektů

Uvedeme zde jenom základní skutečnosti. Vyčerpávající přehled možností programu najdeme v Nápovědě. Konkrétní příklady a postupy věnované geometrii jsou uvedeny v knize [10], která je dodávána s programem Derive.



Obr. 33: Geometrické objekty

Připomeňme si, že Derive pracuje se třemi typy oken. Pro symbolické a numerické výpočty a psaní programů využívá okno Algebra, pro dvourozměrné grafy okno 2D-graf (Obr. 33) a pro třírozměrné grafy okno 3D-graf. Jednou z výhod Derive je, že zadání geometrického útvaru je minimálně zatíženo syntaktickými pravidly programu a zhruba odpovídá běžnému zápisu na papíře.

Přímku, kružnici, křivku

zadáme jednoduše rovnicí nebo parametricky (viz Obr. 33, výrazy `#1,#2,#3`). Chceme-li obrázek, toto zadání zvýrazníme a zobrazíme v 2D(3D) grafickém okně (tvorbě grafů je věnována kapitola 9, Tvorba grafu). Je třeba mít na paměti, že pro přímé znázornění parametrického výrazu je většinou nutné aktivovat volbu **Možnosti → Zjednodušit před vykreslením** z nabídky 2D(3D) grafického okna.

Bod

zadáme vektorem souřadnic (Obr. 33, `#4`) a zase obvyklým způsobem zobrazíme.

Úsečku

zadáváme maticí typu (2,2) jejímiž řádkovými vektory jsou souřadnice krajních bodů úsečky. Např. matici/úsečku na řádku `#5`**,** Obr. 33 jsme na příkazový řádek zadali ve tvaru `[[-3,1],[-1,-3]]`. Před vykreslením příslušné úsečky nesmíme zapomenout na volbu (při aktivním okně 2D grafu): **Možnosti → Zobrazení... → Body → Spojovat → Ano**, jinak se nám zobrazí jenom krajní body úsečky. Pokud je grafický objekt spojen s rovnicí, můžeme ho v 2D grafickém okně touto rovnicí popsat užitím volby: **Možnosti → Popsat nové grafy**

Již v kapitole 10 (Programování) bylo naznačeno, že analytická geometrie poskytuje široké pole působnosti pro tvorbu vlastních funkcí. Na Obr. 33 vidíme definice funkcí `usecka(A,B)` a `kruznice(S,r)` pro snazší kreslení těchto útvarů. V definici funkce `kruznice` se objevují dolní indexy u proměnné S pro rozlišení první a druhé souřadnice středu. K zápisu dolních indexů na příkazovém řádku použijeme šipku ↓ z panelu matematických symbolů nebo slovo `SUB`. Abychom dostali rovnici kružnice, musíme příkaz po odeslání na plochu zjednodušit (základním zjednodušením $\boxed{=}$).

Základní geometrické útvary - SHRNUTÍ

Bod

K:= [2, 3]

L:= [2, -3, 4]

více bodů najednou:

$$\#4: \begin{bmatrix} A \\ B \\ C \end{bmatrix}$$

Přímka

Parametrické vyjádření:

#1: A ≔ [2,3]

#2: B ≔ [-1,4]

#3: $A + t\cdot(B - A)$

Obecná rovnice:

#1: $3\cdot x - 2\cdot y + 1 = 0$

Úsečka

#1: A ≔ [0,2]

#2: B ≔ [6,-1]

#3: [A, B] (Body - Spojovat)

Kružnice (elipsa)

Středová rovnice:

#1 S ≔ [m,n]

#2: r ≔

#3: $(x - S_1)^2 + (y - S_2)^2 = r^2$

Parametrické vyjádření:

#1: a ≔ 3

#2: b ≔ 5

#3: [a·COS(t), b·SIN(t)]

Algebraická rovnice:

#1: $x^2 - 4\cdot x\cdot y - y^2 + 4\cdot x + 1 = 0$

Trojúhelník (n-úhelník)

#1: A ≔ [0,2]

#2: B ≔ [6,-1]

#3: C ≔ [-4,3]

#4: [A, B, C, A]

#1: A ≔ [0,2,5]

#2: B ≔ [6,-1,-2]

#3: C ≔ [-4,3,1]

#4: [A, B, C, A]

POLYGON_FILL([A,B,C])

Rovina

Parametrické vyjádření:

#1: A ≔ [1,2,-1]

#2: u ≔ [1,0,2]

#3: v ≔ [0,2,3]

#4: $A + t\cdot u + s\cdot v$

Obecná rovnice:

#1: $x - 2\cdot y + 5\cdot z - 1 = 0$

Koule	Kuželová plocha	Válcová plocha
SPHERE(r, u, t)	CONE(alpha,theta,z)	CYLINDER(r,theta,z)

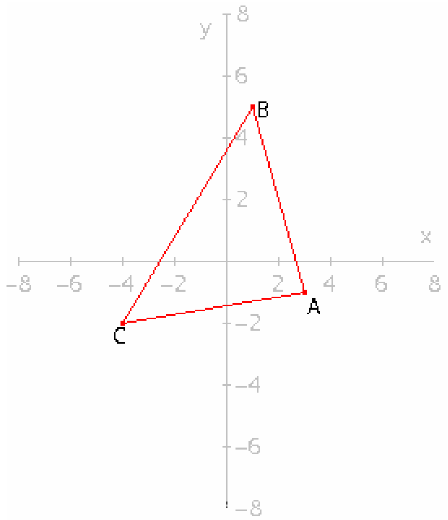
Příklad 20.1: Je dán trojúhelník *ABC*; $A=[3,-1]$, $B=[1,5]$, $C=[-4,-2]$.
Vypočtěte obvod a obsah trojúhelníka *ABC*.

Řešení:

```
#1:    A ≔ [3, -1]

#2:    B ≔ [1, 5]

#3:    C ≔ [-4, -2]
```

Abychom mohli trojúhelník nakreslit, zapíšeme jeho vrcholy do vektoru. Před zobrazením nezapomeneme aktivovat volbu **Možnosti** → **Zobrazení...** → **Body** → **Spojovat** → **Ano**

```
#4:    [A, B, C, A]
```



K popisu grafu použijeme příkaz 2D grafického okna **Vložit** → **Poznámka...** ().

Nastavíme režim „Case sensitive“ pro rozlišování mezi velkými a malými písmeny, aby zápis odpovídal zvyklostem:

```
#5:    CaseMode ≔ Sensitive

#6:    a ≔ |C - B|

#7:    b ≔ |C - A|

#8:    c ≔ |B - A|
```

Obvod:

#9: $O := a + b + c$

#10: $O = \sqrt{74} + 2\cdot\sqrt{10} + 5\cdot\sqrt{2}$

#11: $O = 21.99794839$

Obsah

a) Užitím Heronova vzorce:

#12: $s := \dfrac{O}{2}$

#13: $S := \sqrt{(s\cdot(s - a)\cdot(s - b)\cdot(s - c))}$

#14: $S = 22$

b) Užitím vzorce pro dvě strany a vnitřní úhel jimi sevřený.
Úhel získáme řešením následující rovnice:

#15: $\mathrm{COS}(\alpha) = \dfrac{|(B - A)\cdot(C - A)|}{|B - A|\cdot|C - A|}$

#16: $\mathrm{SOLVE}\left(\mathrm{COS}(\alpha) = \dfrac{|(B - A)\cdot(C - A)|}{|B - A|\cdot|C - A|}, \alpha, \mathrm{Real}\right)$

#17: $\alpha = \mathrm{ACOT}\left(\dfrac{2}{11}\right) \vee \alpha = \mathrm{ATAN}\left(\dfrac{2}{11}\right) + \dfrac{3\cdot\pi}{2} \vee \alpha = - \mathrm{ACOT}\left(\dfrac{2}{11}\right)$

Z nabídnutých řešení (#17) vybereme jednu hodnotu a přiřadíme ji proměnné α:

#18: $\alpha := \mathrm{ACOT}\left(\dfrac{2}{11}\right)$

Potom definujeme hodnotu proměnné obsahu S1 příslušným vzorcem

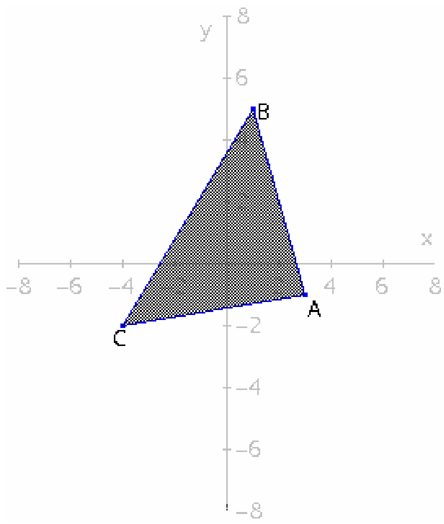
#19: $S1 := \dfrac{1}{2}\cdot b\cdot c\cdot \mathrm{SIN}(\alpha)$

a základním zjednodušením zjistíme její hodnotu:

#20: $S1 = 22$

K zobrazení trojúhelníka můžeme použít i funkci POLYGON_FILL:

#21: POLYGON_FILL([A, B, C])



20.2. Tvorba vlastních uživatelských funkcí

Programování funkcí v Derive se podrobně věnuje samostatná kapitola knihy (kap. 10). Zde jenom na několika příkladech připomeneme, že právě v geometrii se dá možnost vytváření vlastních funkcí dobře využít. Pokud si předem vytvoříme speciální funkce pro dílčí akce a objekty, například funkci `OsaUhlu`, nejsme při řešení složitějších problémů, například při určování středu kružnice vepsané trojúhelníku, obtěžováni zápisem dlouhého kódu a můžeme se soustředit na jádro problému. Na následujících stránkách si představíme balíček takových „pomocných“ geometrických funkcí.

20.2.1. Vzdálenost mimoběžek

Příklad 20.2: Napište v Derive funkci pro výpočet vzdálenosti dvou mimoběžek *p*, *q*, z nichž každá je určena bodem a směrovým vektorem; *p:* $A=[1,0,1]$, $\vec{u}=(-4,2,1)$; q: $B=[2,1,-1]$, $\vec{v}=(-3,1,-1)$.

Řešení:

Funkci nazveme `VM`, bude mít syntaxi `VM(A,u,B,v)` a pro její definici využijeme vzorec

$$\mathrm{VM}(\mathrm{A},\mathrm{u},\mathrm{B},\mathrm{v})=\frac{\left|\overrightarrow{AB}\cdot(\vec{u}\times\vec{v})\right|}{\left|\vec{u}\times\vec{v}\right|},$$

kde VM je vzdálenost dvou mimoběžek, z nichž jedna je dána bodem A a směrovým vektorem $\vec{u}$, zatímco druhá je dána bodem B a směrovým vektorem $\vec{v}$.
Než budeme funkci definovat, můžeme (není to nutné) deklarovat použité proměnné A, B, *u, v* jako vektory:
Průvodce → Definiční obor proměnné... →
napíšeme jméno proměnné a puntíkem vyznačíme **Definiční obor: Vektory → OK**
Toto opakujeme pro všechny čtyři proměnné.
Pro zápis absolutní hodnoty reálného čísla (v čitateli zlomku) i normy vektoru (ve jmenovateli) použijeme funkci `ABS( )` nebo jako závorky svislou dvojčárku ¦ z klávesnice (`|x|`).

Na příkazovém řádku definujeme funkci `VM` takto:
```
VM(A, u, B, v) := ABS((A – B)·CROSS(u, v))/ABS(CROSS(u, v))
```

Řešení v okně Algebra:

#1: A ≔ [1, 0, 1]

#2: u ≔ [–4, 2, 1]

#3: B ≔ [2, 1, –1]

#4: v ≔ [–3, 1, –1]

#5: $$VM(A, u, B, v) := \frac{|(A - B) \times CROSS(u, v)|}{|CROSS(u, v)|}$$

#6: $$VM(A, u, B, v) = \frac{2\cdot\sqrt{682}}{31}$$

#7: VM(A, u, B, v) = 1.684847078

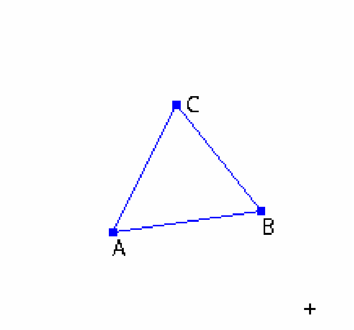
20.2.2. Balíček funkcí „Geometrie“

Příklad 20.3: Je dán trojúhelník ABC svými vrcholy. Určete střed kružnice vepsané trojúhelníku a spolu s touto kružnicí ho zobrazte; A=[-3,-3], B=[4,-2], C=[0,3]

Řešení:
Budeme postupovat stejně jako při konstrukčním řešení. Střed kružnice vepsané je určen průsečíkem os vnitřních úhlů trojúhelníka. Určíme tedy nejprve rovnice os dvou vnitřních úhlů, potom najdeme jejich průsečík a jeho vzdálenost od jedné ze stran trojúhelníka. Nakonec kružnici sestrojíme.

```
#1:    A ≔ [-3, -3]

#2:    B ≔ [4, -2]

#3:    C ≔ [0, 3]

#4:    [A, B, C, A]
```

Výraz #4 zobrazíme v 2D grafickém okně při aktivní volbě **Možnosti** → **Zobrazení...** → **Body** → **Spojovat** → **ANO**. Dostaneme tak trojúhelník *ABC* (Označení vrcholů přidáme pomocí **Vložit** → **Poznámka...** nebo tlačítka):



Osy vnitřních úhlů trojúhelníka zadáme parametricky. Jako směrový vektor osy úhlu použijeme součet jednotkových směrových vektorů jeho ramen (viz obrázek na straně 29, řešení příkladu 10.2).

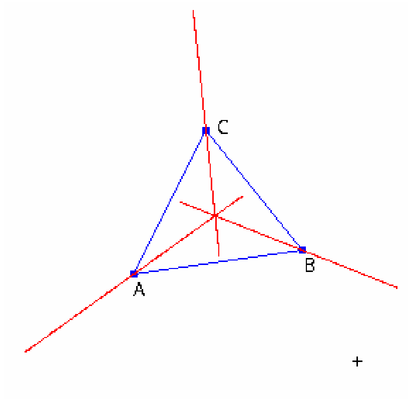
Osy oA, oB, oC vnitřních úhlů při vrcholech *A, B, C*:

$$\#5:\quad oA := A + k\cdot\left(\frac{B - A}{|B - A|} + \frac{C - A}{|C - A|}\right)$$

$$\#6:\quad oB := B + l\cdot\left(\frac{A - B}{|A - B|} + \frac{C - B}{|C - B|}\right)$$

$$\#7:\quad oC := C + m\cdot\left(\frac{A - C}{|A - C|} + \frac{B - C}{|B - C|}\right)$$

Aktivujeme volbu 2D grafického okna **Možnosti** → **Zjednodušit před vykreslením** a zobrazením výrazů `#5, #6, #7` přikreslíme do trojúhelníku *ABC* osy vnitřních úhlů (viz následující obrázek).



Pro určení středu kružnice řešíme následující rovnice s neznámými *k*, *l*, *m* (stačí řešit jenom jednu z nich, když víme, že všechny tři osy mají společný jeden bod):

#8: oA = oB

#9: oA = oC

#10: R1 := SOLVE(oA = oB, [k, l], Real)

#11: $$R1 := k = \frac{45\cdot\sqrt{410}}{338} - \frac{125\cdot\sqrt{41}}{169} + \frac{150\cdot\sqrt{5}}{169} + \frac{575\cdot\sqrt{2}}{338} \wedge l = \frac{115\cdot\sqrt{410}}{1014} + \frac{50\cdot\sqrt{41}}{169} - \frac{1025\cdot\sqrt{5}}{507} + \frac{615\cdot\sqrt{2}}{338}$$

#12: R2 := SOLVE(oA = oC, [k, m], Real)

#13: $$R2 := k = \frac{45\cdot\sqrt{410}}{338} - \frac{125\cdot\sqrt{41}}{169} + \frac{150\cdot\sqrt{5}}{169} + \frac{575\cdot\sqrt{2}}{338} \wedge m = \frac{15\cdot\sqrt{410}}{169} + \frac{115\cdot\sqrt{41}}{338} + \frac{369\cdot\sqrt{5}}{338} - \frac{1025\cdot\sqrt{2}}{338}$$

Střed kružnice vepsané dostaneme tak, že hodnotu jednoho z parametrů *k*, *l*, *m* dosadíme (užijeme třeba tlačítko) do příslušné parametrické rovnice:

#14: S := oA

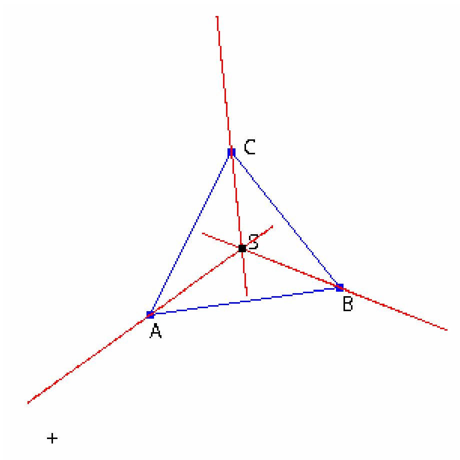
#15: $$S := \left[k\cdot\left(\frac{\sqrt{5}}{5} + \frac{7\cdot\sqrt{2}}{10}\right) - 3,\ k\cdot\left(\frac{2\cdot\sqrt{5}}{5} + \frac{\sqrt{2}}{10}\right) - 3\right]$$

#16:

$$S := \left[\left(\frac{45\cdot\sqrt{410}}{338} - \frac{125\cdot\sqrt{41}}{169} + \frac{150\cdot\sqrt{5}}{169} + \frac{575\cdot\sqrt{2}}{338}\right)\cdot\left(\frac{\sqrt{5}}{5} + \frac{7\cdot\sqrt{2}}{10}\right) - 3,\right.$$
$$\left.\left(\frac{45\cdot\sqrt{410}}{338} - \frac{125\cdot\sqrt{41}}{169} + \frac{150\cdot\sqrt{5}}{169} + \frac{575\cdot\sqrt{2}}{338}\right)\cdot\left(\frac{2\cdot\sqrt{5}}{5} + \frac{\sqrt{2}}{10}\right) - 3\right]$$

#17: S := [0.3777273533, -0.5654718672]

Zobrazením #17 přidáme do obrázku průsečík os vnitřních úhlů *S*:



Řešení úlohy završíme určením rovnice a zobrazením kružnice vepsané trojúhelníku *ABC*. Potřebujeme znát její poloměr, tj. vzdálenost středu *S* od jedné ze stran D*ABC*.

Patu kolmice spuštěné z bodu *S* na stranu *AB* označíme *P*:

#18: P := A + t·(B - A)

Potom musí platit podmínka

#19: (S - P)·(B - A) = 0

Jejím řešením určíme *t* a po jeho substituci do parametrického vyjádření *P* dostaneme souřadnice paty *P*:

#20: SOLVE((S − P)·(B − A) = 0, t, Real)

#21: $$t = \frac{4226120173069}{8102653125300}$$

#22: $$P := A + \frac{4226120173069}{8102653125300}\cdot(B - A)$$

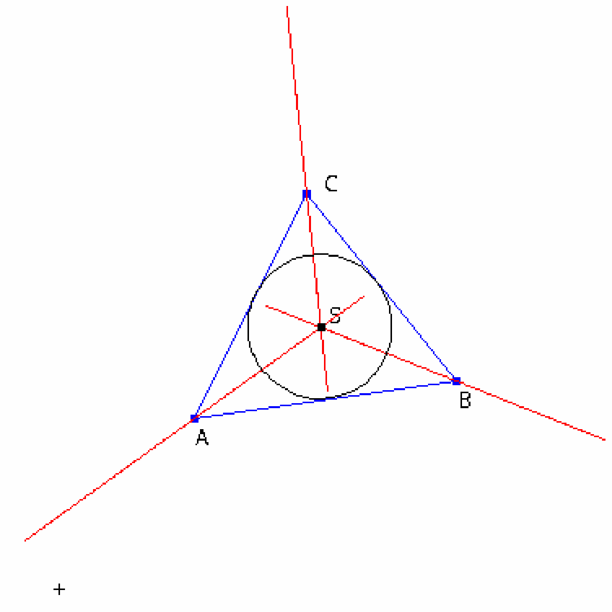
#23: $$P := \left[\frac{5274881835583}{8102653125300}, -\frac{20081839202831}{8102653125300}\right]$$

Poloměr *R* kružnice vepsané je potom roven vzdálenosti bodů *S*,*P*.

#24: $R := |S - P|$

#25: $$R := \frac{15500016810281\cdot\sqrt{2}}{11343714375420}$$

#26: $$(x - S_1)^2 + (y - S_2)^2 = R^2$$



Přidejme k řešení předchozí úlohy ještě řešení analogické úlohy na kružnici opsanou (to čtenář již s přehledem zvládne) a obě řešení rozčleňme na dílčí kroky. Dostaneme seznam akcí, které se zřejmě vyskytují v řešení řady jiných úloh z geometrie. Pokud se nám nechce je pořád opisovat, vyplatí se tyto akce definovat jako samostatné funkce programu Derive. Takto vznikly následující funkce, uložené v balíčku „Geometrie.mth“.

1. *Kolmice z bodu B ke přímce, která má směrový vektor n:*

```
KolmiceVBode(n, B) ≔ n·(B - [x, y]) = 0
```

2. *Střed úsečky AB:*

$$\text{StredUs}(A, B) :=\frac{A + B}{2}$$

3. *Osa úsečky AB:*

```
OsaUs(A, B) ≔ KolmiceVBode(B - A, StredUs(A, B))
```

4. *Osa úhlu AVB:*

```
OsaUhlu(A, V, B, t, r, s) ≔
        Prog
          r ≔ (A - V)/ABS(A - V)
          s ≔ (B - V)/ABS(B - V)
          w ≔ r + s
          V + t·w
```

5. *Vzdálenost bodu B od přímky p. Hodnotou parametru p je obecná rovnice této přímky:*

```
VzdBp(A, g, p, m, n, q) ≔
  Prog
    p ≔ LHS(g) - RHS(g)
    m ≔ POLY_COEFF(p, x, 1)
    n ≔ POLY_COEFF(p, y, 1)
    q ≔ POLY_COEFF(POLY_COEFF(p, x, 0), y, 0)
    ABS(m·A↓1 + n·A↓2 + q)/√(m^2 + n^2)
```

6. *Obecná rovnice přímky AB:*

$$\text{ObRovP}(A, B) := \text{DET}\begin{bmatrix} x & y & 1 \\ A_1 & A_2 & 1 \\ B_1 & B_2 & 1 \end{bmatrix}$$

7. *Rovnice kružnice opsané trojúhelníku ABC:*

```
KrOpsana(A, B, C) :=
  Prog
    sA := OsaUs(B, C)
    sB := OsaUs(A, C)
    sC := OsaUs(A, B)
    S := (SOLUTIONS(sA = sB, [x, y]))↓1
    rO := ABS(S - A)
    (x - S↓1)^2 + (y - S↓2)^2 - rO^2 = 0
```

8. *Rovnice kružnice vepsané trojúhelníku ABC:*

```
KrVepsana(A, B, C) :=
  Prog
    uA := OsaUhlu(B, A, C)
    uB := OsaUhlu(A, B, C)
    uC := OsaUhlu(A, C, B)
    pU := (SOLUTIONS(SUBST(uA, t, k) = SUBST(uB, t, l), [k, l]))↓1
    U := SUBST(uA, t, pU↓1)
    rV := VzdBp(U, ObRovP(A, B))
    (x - U↓1)^2 + (y - U↓2)^2 - rV^2 = 0
```

9. *Střed kružnice opsané trojúhelníku ABC:*

```
StredKrOps(A, B, C) :=
  Prog
    sA := OsaUs(B, C)
    sB := OsaUs(A, C)
    sC := OsaUs(A, B)
    S := (SOLUTIONS(sA = sB, [x, y]))↓1
```

10. *Střed kružnice vepsané trojúhelníku ABC:*

```
StredKrVeps(A, B, C) :=
  Prog
    uA := OsaUhlu(B, A, C)
    uB := OsaUhlu(A, B, C)
    uC := OsaUhlu(A, C, B)
    pU := (SOLUTIONS(SUBST(uA, t, K) = SUBST(uB, t, L), [K, L]))↓1
    U := SUBST(uA, t, pU↓1)
```

20.3. Příklady užití Derive ve výuce geometrie

20.3.1. Nástroj zkoumání a motivace

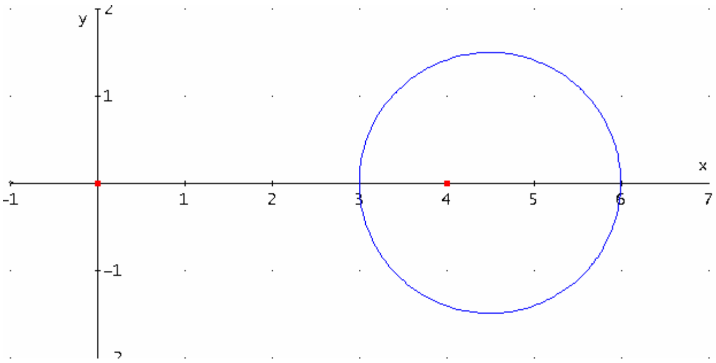
Často probíhá proces vyučování a učení se tak, že se žák nejprve vhodně seznámí s nějakým jevem nebo vlastností a teprve potom, patřičně zmotivován, hledá jejich podstatu. V těchto počátečních fázích můžeme použít Derive jako nástroj typu „černá skříňka“, tj. dostaneme řešení, aniž by žák věděl „jak“ to program dělá. Takovým způsobem můžeme Derive použít třeba při zkoumání množin bodů dané vlastnosti. Nejprve program necháme příslušnou množinu najít a zobrazit, potom teprve hledáme způsob, jak bychom to udělali sami.

Příklad 20.4: Apolloniova kružnice
Jsou dány dva pevné body A, B. Určete množinu (tj. vyšetřete její tvar a rovnici) všech bodů P takových, které splňují rovnost $|PA|=3|PB|$, kde $|PA|$ a $|PB|$ jsou vzdálenosti bodu P od bodu A a od bodu B, v tomto pořadí.

Řešení:

```
#1: A ≔ [0, 0]

#2: B ≔ [4, 0]

#3: P ≔ [x, y]

#4: |P − A| = 3·|P − B|
```

V 2D grafickém okně zobrazíme body A, B (tj. výrazy #1, #2; nastavíme velkou velikost bodů pomocí **Možnosti → Zobrazení...**) spolu s rovnicí #4. Rychlejšímu vykreslení množiny řešení #4 napomůže aktivace režimu **Možnosti → Zjednodušit před vykreslením**.

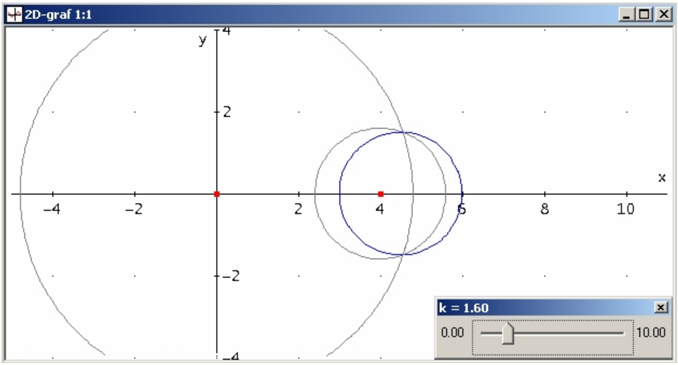


Z výsledku je patrné, že hledanou množinou bodů je kružnice. Chceme-li ukázat cestu jejího vzniku, použijeme posuvník. Každý bod vyšetřované množiny můžeme chápat jako průsečík dvou kružnic, jejichž poloměry jsou v poměru 3:1. Rovnice těchto dvou kružnic zapíšeme ve vektorovém tvaru:

#5: $|P - A| = 3\cdot k$

#6: $|P - B| = k$

Do 2D grafického okna vložíme posuvník pro parametr k, aktivujeme volbu **Možnosti → Zjednodušit před vykreslením** a výrazy #5, #6 zobrazíme. Výsledkem jsou dvě kružnice, jejichž průsečíky je vytvářena námi vyšetřovaná (Apolloniova) kružnice.



V „genealogickém“ způsobu řešení můžeme pokračovat výpočtem a vizualizací oněch průsečíků. Při řešení příslušné soustavy rovnic #5, #6 pomůžeme programu odstraněním odmocnin tak, že obě strany těchto rovnic umocníme na druhou. Na příkazový řádek tedy napíšeme výraz „#5^2 and #6^2” a odešleme (¢):

#7: $(|P - A| = 3\cdot k)^2 \wedge (|P - B| = k)^2$

#8: $x^2 - 8\cdot x + y^2 = k^2 - 16 \wedge x^2 + y^2 = 9\cdot k^2$

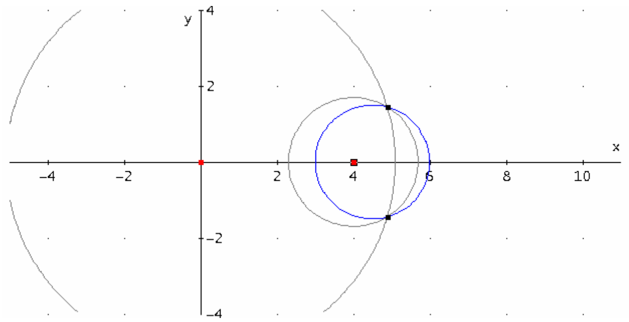
Potom z příkazového řádku odešleme výraz „SOLUTIONS(#8, [x,y])“:

#9: $\mathrm{SOLUTIONS}(x^2 - 8\cdot x + y^2 = k^2 - 16 \wedge x^2 + y^2 = 9\cdot k^2, [x, y])$

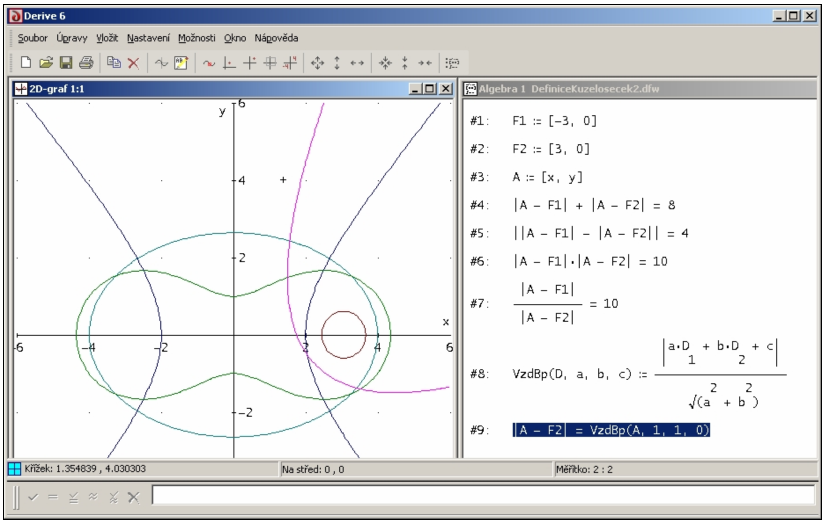
Výsledkem zjednodušení #9 je matice parametrických vyjádření průsečíků:

#10: $$\begin{bmatrix} k^2 + 2 & \sqrt{(- k^4 + 5\cdot k^2 - 4)} \\ k^2 + 2 & - \sqrt{(- k^4 + 5\cdot k^2 - 4)} \end{bmatrix}$$

Výsledkem jejího zobrazení v 2D grafickém okně jsou dva body – průsečíky, které spolu s kružnicemi #5, #6 reagují na pohyb posuvníku.



Řešení předchozího příkladu může ve zvídavém jedinci stimulovat otázku „Co se stane, když změníme tu podmínku?“ Například, když místo |*AP*| = *k*|*BP*| uvažujeme |*AP*|·|*BP*| = *k*, |*AP*| + |*BP*| = *k*, |*AP*| - |*BP*| = *k* atd.
Derive nabízí okamžitou odpověď ve formě obrázku výsledné křivky (Obr. 34) spolu s nástroji pro jejich další výzkum.



Obr. 34: Křivky v rovině

Naformátováno: Písmo: 11

Naformátováno: Písmo: 11

Naformátováno: Písmo: 11 b., není Tučné

Příklad 20.5: Definice kuželosečky užitím řídící přímky a ohniska
Kuželosečka může být definována jako množina bodů P v rovině s konstantním poměrem vzdáleností od daného bodu F (ohnisko) a od dané přímky d (řídící přímka, directrix). Hodnota tohoto poměru, kterou značíme většinou ε a nazýváme ji numerickou výstředností kuželosečky, určuje typ kuželosečky. Pro $0 < e < 1$ dostáváme elipsu, pro $e = 1$ parabolu a pro $e > 1$ hyperbolu.

Řešení:
Při řešení použijeme funkci `VzdBp` z balíčku `Geometrie.mth` (str. 146). Proto tento balíček nahrajeme do paměti:

```
#1: LOAD(C:\Derive\Derive_Packages\Geometrie.mth)
```

Vhodně zvolíme rovnici řídící přímky a souřadnice ohniska:

```
#2: d ≔ x + 2 = 0

#3: F ≔ [0, 0]
```

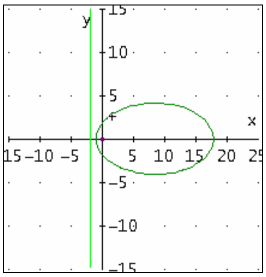
Proměnný bod křivky pojmenujeme *P*, abychom se vyhnuli případné kolizi malého a velkého *x*:

```
#4: P ≔ [x, y]
```

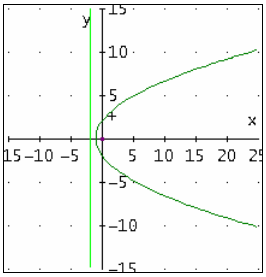
Poměr vzdáleností *P* od *F* a *d* je potom roven numerické výstřednosti:

$$\#5: \quad \frac{|P - F|}{\text{VzdBp}(P,\ d)} = e$$

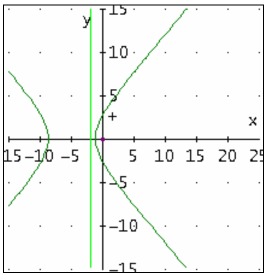
Nejedná se o překlep, místo ε jsme skutečně nuceni použít jiný symbol, např. *e*. Je to kvůli posuvníku, kterým budeme při zobrazení výše uvedeného vztahu ovládat hodnoty této proměnné. Při použití písmen řecké abecedy se posuvník chová nekorektně.
Vložíme tedy do 2D-grafického okna posuvník pro ovládání hodnot *e* a výraz `#5` zobrazíme. Pro různé hodnoty *e* dostaneme odpovídající kuželosečky:



0 < e < 1



e = 1



e > 1

Možnosti symbolické manipulace s výrazy nám dovolují odvodit i algebraickou rovnici kuželosečky s parametrem *e*.

#6: $$\frac{\sqrt{(x^2 + y^2)}}{|x + 2|} = e$$

Rovnici přeneseme užitím klávesy (F4) na příkazový řádek a rozšíříme výrazem $|x + 2|$:

#7: $$\left(\frac{\sqrt{(x^2 + y^2)}}{|x + 2|} = e\right)\cdot|x + 2|$$

#8: $$\sqrt{(x^2 + y^2)} = e\cdot|x + 2|$$

Opět použijeme (F4) a obě strany rovnice tentokrát umocníme na druhou:

#9: $$(\sqrt{(x^2 + y^2)} = e\cdot|x + 2|)^2$$

#10: $$x^2 + y^2 = e^2\cdot(x + 2)^2$$

#11: $$x^2 + y^2 - e^2\cdot(x + 2)^2 = 0$$

Nakonec roznásobíme vzhledem k proměnným *x*, *y*:

#12: $$x^2\cdot(e^2 - 1) + 4\cdot e^2\cdot x - y^2 + 4\cdot e^2 = 0$$

20.3.2. Názorná pomůcka

I v analytické geometrii je pro většinu lidí důležitá názorná představa. Zvláště pak při budování základů této metody. Někdy není pro učitele jednoduché vytvořit tu správnou představu u všech žáků ve třídě. Nepomohla by v tu chvíli „počítačová simulace“ zkoumaného jevu?

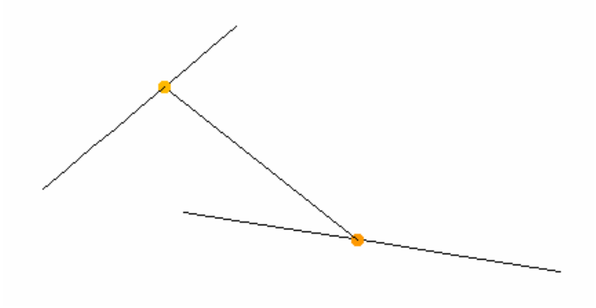
Příklad 20.4: Vzdálenost mimoběžek
Určete vzdálenost dvou mimoběžných přímek p, q. Každá z přímek je zadána bodem a směrovým vektorem: $p=[A,\vec{u}]$, $q=[B,\vec{v}]$; $A=[6,1,10]$, $B=[10,-1,-2]$, $\vec{u}=(1,2,-1)$, $\vec{v}=(-7,2,3)$.

Řešení:
Stejný příklad, akorát s jiným přímkami jsme již řešili (Př. 20.2). Pojďme se teď pokusit vhodným znázorněním objasnit použitou formuli.

```
#1:    A ≔ [6, 1, 10]

#2:    B ≔ [10, -1, -2]

#3:    u ≔ [1, 2, -1]

#4:    v ≔ [-7, 2, 3]

#5:    P ≔ A + s·u

#6:    Q ≔ B + t·v

#7:    [P, Q]
```

Naším prvním cílem je zobrazit v 3D-grafickém okně dané mimoběžky spolu s body, které se po nich budou pohybovat pomocí dvou posuvníků pro parametry *s* a *t*. Body budou stále spojeny příčkou. Při zobrazování je třeba dodržet následující postup:

1. Aktivujeme volbu 3D-grafického okna **Možnosti** → **Zjednodušit před vykreslením**
2. Zobrazíme výrazy `#5`, `#6`. Tak dostaneme ty přímky.
3. Do 3D-grafického okna vložíme posuvníky pro ovládání parametrů *s*, *t* a zobrazíme výraz `#7`. Tak dostaneme příčku, která bude prostřednictvím posuvníků pohyblivá.
4. Zobrazíme znovu výrazy `#5`, `#6`. Tentokrát se zobrazí jako krajní body příčky. Zobrazení provedeme pomocí posloupnosti příkazů 3D grafického okna **Vložit** → **Graf...** a nastavíme největší velikost bodů, případně změníme jejich barvu.



Pokračujeme nalezením příčky, která je kolmá k oběma mimoběžkám. Její směr je dán vektorem $\vec{w}$:

```
#8:   w ≔ u × v
```

Potom je tato příčka určena body $P \in p$, $Q \in q$, které splňují vektorovou rovnici:

```
#9:   Q - P = r·w
```

Rovnici #9 zjednodušíme

```
#10:
 [-s - 7·t + 4, - 2·s + 2·t - 2, s + 3·t - 12] = [8·r, 4·r, 16·r]
```

a poté vyřešíme vzhledem k neznámým *r*, *s*, *t*:

```
#11:
SOLVE([-s - 7·t + 4, - 2·s + 2·t - 2, s + 3·t - 12] =
                                  [8·r, 4·r, 16·r],[r, s, t], Real)
```

#12: $$r = -\frac{1}{2} \wedge s = 1 \wedge t = 1$$

Dosazením příslušných výsledků z #12 do *P* (za s dosadíme 1) a do *Q* (za t dosadíme 1) dostaneme krajní body *P0*, *Q0* hledané kolmé příčky.

```
#13:  P0 ≔ P
```

Výraz #13 nejprve zjednodušíme ($\boxed{=}$):

```
#14:                        P0 ≔ [s + 6, 2·s + 1, 10 - s]
```

Potom do #14 dosadíme (užitím tlačítka) za *s* hodnotu 1.

```
#15:                               P0 ≔ [7, 3, 9]
```

Analogicky postupujeme v případě *Q0*:

```
#16:  Q0 ≔ Q

#17:                      Q0 ≔ [10 - 7·t, 2·t - 1, 3·t - 2]

#18:                               Q0 ≔ [3, 1, 1]

#19:  [P0, Q0]
```

Příčku *P0Q0* nakreslíme následujícím postupem:

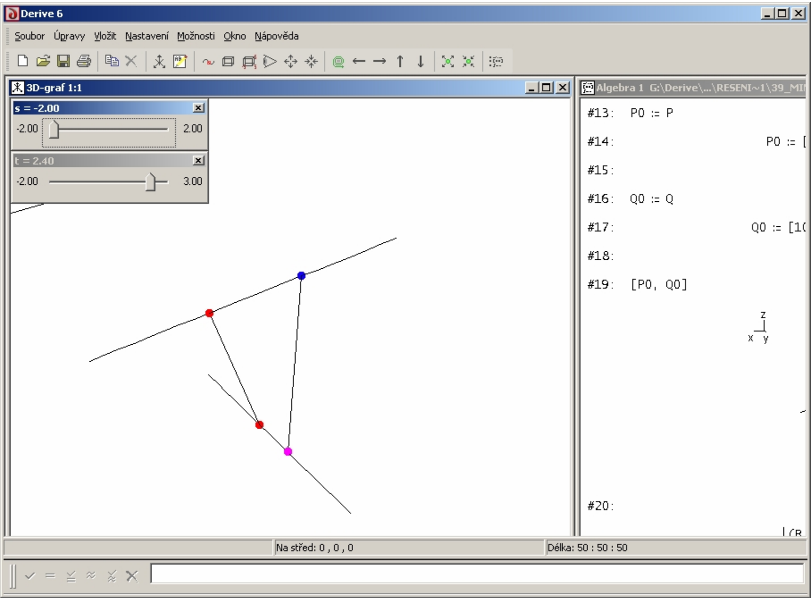
1. Zobrazíme #19. Tak dostaneme onu nejkratší příčku.
2. Pomocí posloupnosti příkazů **Vložit** → **Graf...** zobrazíme #15 a #18. Tak dostaneme krajní body příčky.

Výsledek vidíme na Obr. 35. Potom už je snadné pomocí posuvníků manipulovat s příčkou *PQ* a porovnáním s *P0Q0* ukázat, že příčka kolmá k oběma mimoběžkám je skutečně nejkratší.
Vzdálenost mimoběžek je tedy rovna vzdálenosti bodů *P0*, *Q0*:

#20: $$|Q0 - P0| = 2\cdot\sqrt{21}$$

Což odpovídá průmětu libovolné příčky (tedy i *AB*) do směru kolmého na obě mimoběžky:

#21: $$\frac{|(B - A)\cdot(u \times v)|}{|u \times v|} = 2\cdot\sqrt{21}$$



Obr. 35: Vzdálenost mimoběžek

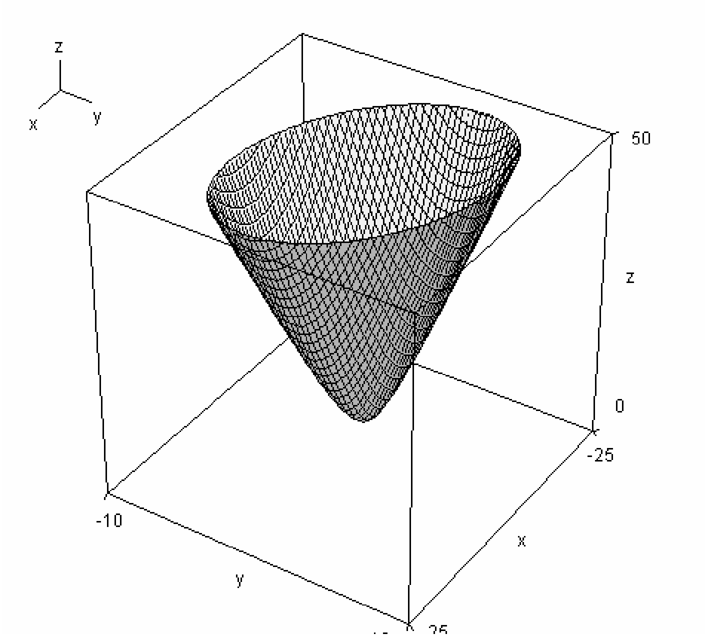
Komu nestačí vizuální průzkum, ten může pokračovat úpravou vztahu |Q-P| pro délku obecné příčky daných mimoběžek a řešit odpovídající úlohu na extrém funkce dvou proměnných, jak je naznačeno na následujících řádcích.

#22: $F(s, t) := |Q - P|$

#23: $F(s, t) := \sqrt{2}\cdot\sqrt{(3\cdot s^2 + 6\cdot s\cdot(t - 2) + 31\cdot t^2 - 68\cdot t + 82)}$

#24: $\text{SOLVE}\left(\frac{d}{ds} F(s, t) = 0 \wedge \frac{d}{dt} F(s, t) = 0, [s, t]\right)$

#25: $s = 1 \wedge t = 1$



Obr. 36: Graf funkce F(s,t)

Příklad 20.6: Určete průmět bodu $A = [3, 4, 4]$ do roviny $r : x + 2y + z = 0$.

Řešení:

```
#1:   A ≔ [3, 4, 4]

#2:   x + 2·y + z = 0
```

Z rovnice `#2` určíme směrový vektor kolmice vedené k rovině z bodu *A*. Je to normálový vektor roviny:

```
#3:   n ≔ [1, 2, 1]
```

Zapíšeme parametrické vyjádření této kolmice:

```
#4:   P ≔ A + t·n

#5:                               P ≔ [t + 3, 2·t + 4, t + 4]
```

Znázornění v 3D grafickém okně provedeme následujícím způsobem:

1. Zobrazíme rovnici roviny (výraz #2).
2. Zobrazíme bod *A* (#1). Použijeme **Vložit** → **Graf...** , nastavíme velikost a barvu bodu.
3. Zobrazíme výraz `#5`. Tak dostaneme kolmici z bodu *A* na rovinu. Pozor, použijeme opět **Vložit** → **Graf...** a pohlídáme si, aby na kartě **Barva grafu** bylo nastaveno **Síť** → **Zapnuto**. Jinak přímku neuvidíme!
4. Do 3D grafického okna vložíme posuvník pro parametr *t* a znovu zobrazíme výraz `#5`. Tak dostaneme bod, který je posuvníkem pohyblivý podél té kolmice.

Potom už můžeme posuvníkem pohybovat bodem podél kolmice a zkoumat, pro jakou hodnotu *t* splyne s rovinou. Z obrázku 37. je patrné, že se takto dostaneme k hodnotě $t = -2.5$. Výsledek výzkumu nakonec konfrontujeme s výpočtem:

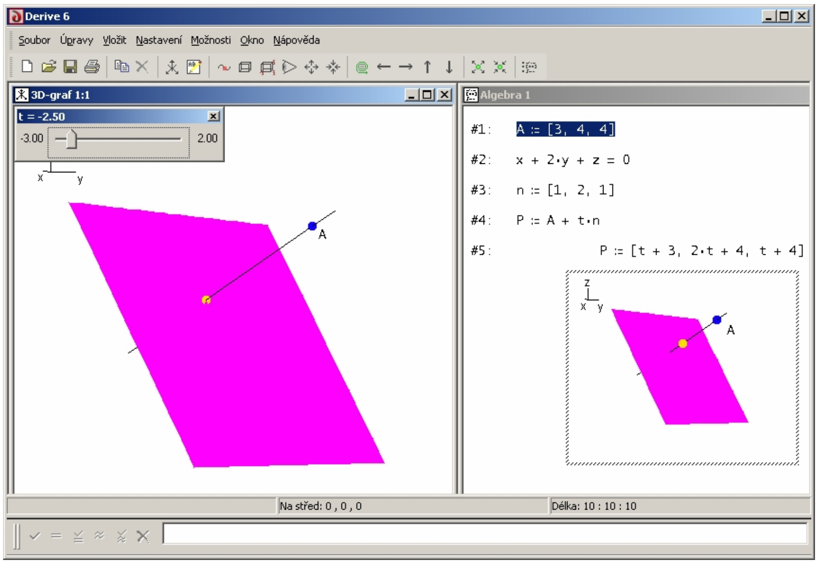
#6: $P_1 + 2\cdot P_2 + P_3 = 0$

#7: $6\cdot t + 15 = 0$

```
#8:   SOLVE(6·t + 15 = 0, t, Real)
```

#9: $t = -\dfrac{5}{2}$

#10: $P := \left[\dfrac{1}{2}, -1, \dfrac{3}{2}\right]$



Obr. 37: Průmět bodu do roviny

20.3.4. Důkaz geometrické vlastnosti

Možnost symbolické manipulace s výrazy nám dovoluje provádět v Derive důkazy některých geometrických vlastností. Slůvko „některých“ znamená takových, jejichž složitost nepřesáhne možnosti programu nebo počítače.

Příklad 20.7: Dokažte, že osy stran trojúhelníka se protínají v jednom bodě.

Řešení:

Trojúhelník vhodně umístíme vzhledem k souřadnicové soustavě:

```
#1:    A := [0, 0]

#2:    B := [k, 0]

#3:    C := [l, m]
```

V dalším postupu využijeme k určení obecné rovnice osy strany trojúhelníku funkci `OsaUs` z balíčku „Geometrie.mth“, který byl představen v podkapitole 20.2.2 (str. 146). Balíček tedy nahrajeme do paměti:

```
#4:    LOAD(G:\Derive\Derive_Packages\Geometrie.mth)
```

Potom definujeme užitím funkce `OsaUs` rovnice os všech tří stran:

```
#5:   oA := OsaUs(B, C)

#6:   oB := OsaUs(A, C)

#7:   oC := OsaUs(A, B)
```

Postupně určíme souřadnice společného bodu dvojice os *oA*, *oB*:

```
#8:   SOLVE(oA ∧ oB, [x, y])
```

#9: $$x = \frac{k}{2} \wedge y = -\frac{k\cdot l - l^2 - m^2}{2\cdot m}$$

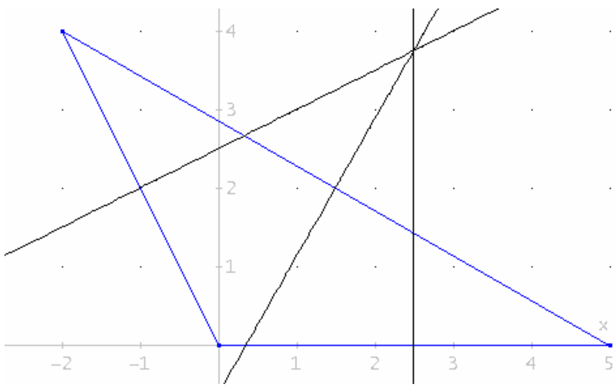
a souřadnice společného bodu dvojice os *oA*, *oC*:

```
#10:  SOLVE(oA ∧ oC, [x, y])
```

#11: $$x = \frac{k}{2} \wedge y = -\frac{k\cdot l - l^2 - m^2}{2\cdot m}$$

Vidíme, že oba body mají stejné souřadnice. Všechny tři osy *oA*, *oB* a *oC* mají tedy společný průsečík. Tím je důkaz proveden.

Po dosazení konkrétních hodnot za parametry *k*, *l*, *m* snadno zobrazíme trojúhelník *ABC* (stačí zobrazit výraz `[A,B,C,A]` při režimu spojování bodů, viz řešení Příkladu 20.3) spolu s osami jeho stran (stačí zobrazit výrazy `#5`, `#6`, `#7`):



Příklad 20.8: Simsonova přímka
Je dán trojúhelník *ABC*. Uvažujme libovolný bod *P* roviny trojúhelníka, jehož kolmé průměty na prodloužené strany daného trojúhelníka *BC*, *CA* a *AB* označíme po řadě *Fa*, *Fb* a *Fc*. Dokažte, že body *Fa*, *Fb* a *Fc* leží na přímce (jsou kolineární) právě tehdy, když bod *P* leží na kružnici opsané trojúhelníku *ABC*. Taková přímka potom existuje ke každému bodu *P* opsané kružnice a nazývá se Simsonova přímka bodu *P*.

Řešení:
Bodům *A*, *B*, *C*, *P* přiřadíme obecné souřadnice:

```
#1:   A := [0, 0]

#2:   B := [k, 0]

#3:   C := [l, m]

#4:   P := [r, s]
```

Do paměti nahrajeme balíček „Geometrie“ (viz str. 146):

```
#5:   LOAD(G:\Derive\Derive_Packages\Geometrie.mth)
```

Užitím funkcí z tohoto balíčku určíme obecné rovnice přímek, v nichž leží strany trojúhelníka *ABC* spolu s kolmicemi na ně spuštěnými z bodu *P*.

```
#6:   a := ObRovP(B, C)

#7:                       a := m·x + y·(k - l) - k·m = 0

#8:   b := ObRovP(A, C)

#9:                             b := m·x - l·y = 0

#10:  c := ObRovP(A, B)

#11:                              c := k·y = 0

#12:  pa := KolmiceVBode(C - B, P)

#13:               pa := x·(k - l) - m·y - k·r + l·r + m·s = 0

#14:  pb := KolmiceVBode(A - C, P)

#15:                     pb := l·x + m·y - l·r - m·s = 0

#16:  pc := KolmiceVBode(B - A, P)

#17:                           pc := k·(x - r) = 0
```

Spočítáme souřadnice pat *Fa*, *Fb* a *Fc*. Použijeme funkci SOLUTIONS, jejímž výsledkem je vektor řešení. V tomto konkrétním případě tedy vektor, jehož jedinou složkou je vektor souřadnic příslušné paty. Abychom se vyhnuli dvojitým závorkám, použijeme na funkci SOLUTIONS operátor ↓1 (nebo SUB 1), kterým ukážeme na ten vnitřní vektor souřadnic.

#18: $Fa := (\text{SOLUTIONS}([a, pa], [x, y]))_1$

#19: $$Fa := \left[\frac{k^2 \cdot r + k\cdot(m\cdot(m - s) - 2\cdot l\cdot r) + l\cdot(l\cdot r + m\cdot s)}{k^2 - 2\cdot k\cdot l + l^2 + m^2}, \frac{m\cdot(k^2 - k\cdot(l + r) + l\cdot r + m\cdot s)}{k^2 - 2\cdot k\cdot l + l^2 + m^2}\right]$$

#20: $Fb := (\text{SOLUTIONS}([b, pb], [x, y]))_1$

#21: $$Fb := \left[\frac{l\cdot(l\cdot r + m\cdot s)}{l^2 + m^2}, \frac{m\cdot(l\cdot r + m\cdot s)}{l^2 + m^2}\right]$$

#22: $Fc := (\text{SOLUTIONS}([c, pc], [x, y]))_1$

#23: $Fc := [r, 0]$

Tři body leží v přímce, jestliže libovolné dva jimi určené vektory jsou závislé:

#24: $Fb - Fa = i\cdot(Fc - Fa)$

Pro naše účely je lepší závislost vektorů vyjádřit následující rovnicí:

#25: $\text{DET}([Fb - Fa, Fc - Fa]) = 0$

Po základním zjednodušení

#26: $$-\frac{k\cdot m^2\cdot(k\cdot(l\cdot s - m\cdot r) - l^2\cdot s - m\cdot(m\cdot s - r^2 - s^2))}{(k^2 - 2\cdot k\cdot l + l^2 + m^2)\cdot(l^2 + m^2)} = 0$$

odstranění jmenovatele

#27: $k\cdot(l\cdot s - m\cdot r) - l^2\cdot s - m\cdot(m\cdot s - r^2 - s^2) = 0$

a roznásobení dle proměnných *r*, *s* dostaneme ekvivalentní podmínku pro souřadnice bodu *P*

#28: $$m\cdot r^2 - k\cdot m\cdot r + m\cdot s^2 + s\cdot(k\cdot l - l^2 - m) = 0$$

Pokud v rovnici #28 nahradíme proměnné *r*, *s* proměnnými *x*, *y*, v tomto pořadí, dostaneme rovnici

#29: $$m\cdot x^2 - k\cdot m\cdot x + m\cdot y^2 + y\cdot(k\cdot l - l^2 - m) = 0$$

Snadno se přesvědčíme, že #29 je rovnice kružnice opsané trojúhelníku *ABC*. Stačí, když ji porovnáme s výsledkem #31 funkce `KrOpsana(A, B, C)` z balíčku `Geometrie`. Na první pohled je zřejmé, že se jedná o identické rovnice.

#30: `KrOpsana(A, B, C)`

#31: $$\frac{m\cdot x^2 - k\cdot m\cdot x + m\cdot y^2 + y\cdot(k\cdot l - l^2 - m^2)}{m} = 0$$

Nástroj „posuvník“ nám umožňuje provést **dynamické znázornění** úlohy.

Nejprve dosadíme konkrétní hodnoty za parametry *k*, *l*, *m*:

#32: $k := 5$

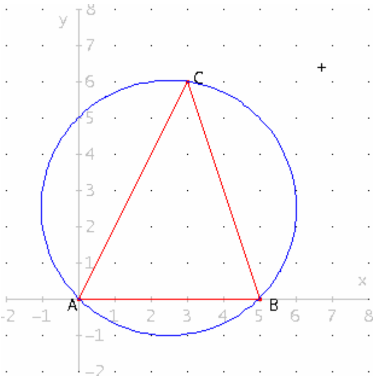
#33: $l := 3$

#34: $m := 6$

Potom zobrazíme trojúhelník *ABC* spolu s opsanou kružnicí (při aktivních volbách **Možnosti → Zobrazení… → Body → Spojovat → ANO**, **Možnosti → Zjednodušit před vykreslením**). Rovnici opsané kružnice dostaneme tak, že po zadání hodnot *k*, *l*, *m* provedeme základní zjednodušení výrazu #28.

#35: `[A, B, C, A]`

#36: $$6\cdot r^2 - 30\cdot r + 6\cdot s^2 - 30\cdot s = 0$$



Abychom mohli pohybovat s bodem *P* podél opsané kružnice, vyjádříme jeho souřadnici *s* jako funkci souřadnice *r* (nebo naopak):

#37: $\text{SOLVE}(6\cdot r^2 - 30\cdot r + 6\cdot s^2 - 30\cdot s = 0, s, \text{Real})$

#38:

$$s = \frac{5 - \sqrt{(-4\cdot r^2 + 20\cdot r + 25)}}{2} \lor s = \frac{\sqrt{(-4\cdot r^2 + 20\cdot r + 25)} + 5}{2}$$

Jeden z výsledků přeneseme klávesou F3 na příkazový řádek a přepíšeme do tvaru přiřazovacího příkazu:

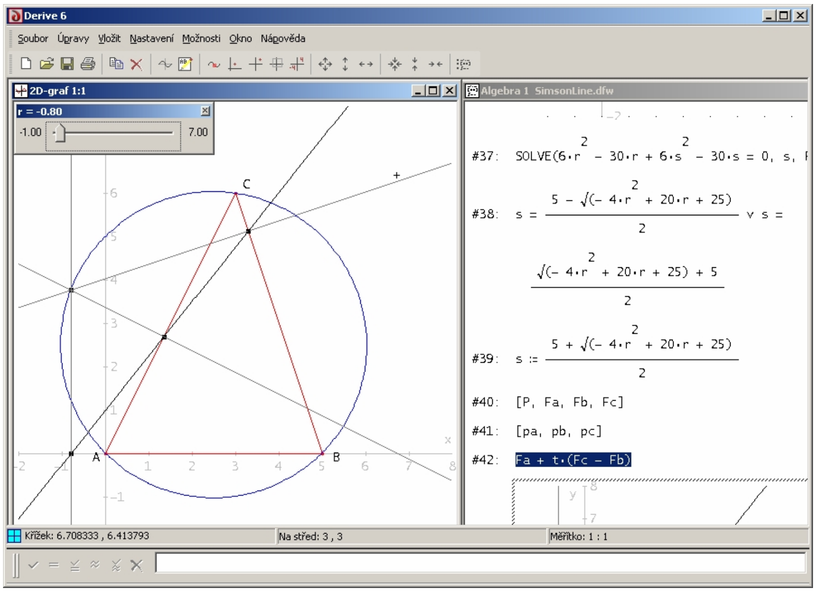
#39: $s := \frac{5 + \sqrt{(-4\cdot r^2 + 20\cdot r + 25)}}{2}$

Potom vložíme do 2D grafického okna posuvník pro parametr *r* a zobrazíme bod *P*, kolmice *pa*, *pb*, *pc*, jejich paty *Fa*, *Fb*, *Fc* a jimi určenou Simsonovu přímku. Pro snazší zobrazení můžeme tyto objekty sdružit do následujících vektorů:

#40: [P, Fa, Fb, Fc]

#41: [pa, pb, pc]

#42: $Fa + t\cdot(Fc - Fb)$



Obr. 38: Simsonova přímka

Příklad 20.9: Descartův list
Je dán trojúhelník *ABC*, na jehož stranách *BC*, *CA* a *AB* jsou dány postupně body *D*, *E* a *F* tak, že každý z nich rozděluje odpovídající stranu na dvě části, jejichž délky jsou v poměru *t*. Jinak řečeno, platí následující rovnost dělících poměrů: $t = (BCD) = (CAE) = (ABF)$. Určete množinu středů kružnic opsaných trojúhelníkům *DEF* při proměnném *t*.

Řešení:
Ukážeme si pouze dynamickou konstrukci hledané množiny. Úplné řešení, s odvozením rovnice odpovídající křivky, najde zájemce v Přílohách.

```
#1:    A := [0, 0]

#2:    B := [k, 0]

#3:    C := [1, m]
```

Body *D*, *E*, *F* definujeme parametricky:

```
#4:    D := B + t·(C - B)
```

```
#5:    E ≔ C + t·(A - C)

#6:    F ≔ A + t·(B - A)
```

Abychom mohli řešení úlohy zobrazit, musíme zadat konkrétní hodnoty souřadnic *k*, *l*, *m* bodů *B*, *C*:

```
#7:    k ≔ 2

#8:    l ≔ 0

#9:    m ≔ 2
```

V 2D grafickém okně pak nejprve zobrazíme základní trojúhelník *ABC*:

```
#10:  [A, B, C, A]
```

Potom do grafického okna vložíme posuvník pro parametr *t* a zobrazíme proměnný trojúhelník *DEF* (aktivujeme volbu **Možnosti → Zjednodušit před vykreslením**):

```
#11:  [D, E, F, D]
```

Pro výpočet souřadnic středu kružnice opsané trojúhelníku *DEF* použijeme funkci `StredKrOps` z balíčku `Geometrie.mth` (str. 146):

```
#12:  LOAD(G:\Derive\Derive_Packages\Geometrie.mth)

#13:  StredKrOps(D, E, F)
```

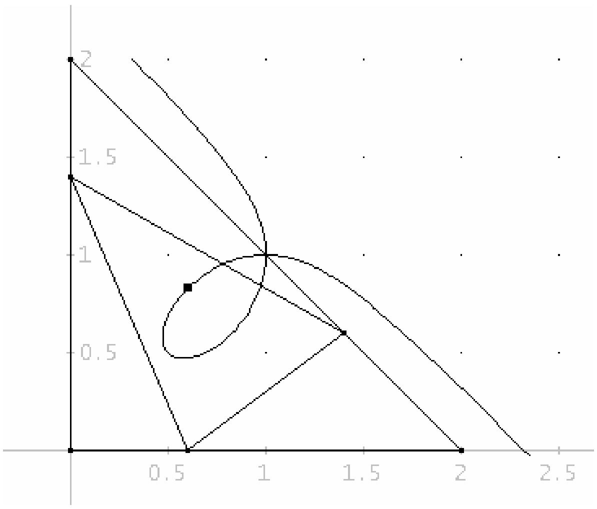
Obrazem následujícího výrazu `#14`, který je výsledkem zjednodušení `#13`, je bod – střed kružnice opsané *DEF*. Protože jeho souřadnice závisí na *t*, je pohyblivý spolu s posuvníkem.

$$\#14: \quad S := \left[- \frac{t^3 - 5\cdot t^2 + 4\cdot t - 1}{3\cdot t^2 - 3\cdot t + 1}, \frac{t^3 + 2\cdot t^2 - 3\cdot t + 1}{3\cdot t^2 - 3\cdot t + 1} \right]$$

Výraz `#14` může být chápán i jako parametrické vyjádření trajektorie onoho středu, tj. parametrické vyjádření hledané křivky. Abychom ji snadno zobrazili, je vhodné nahradit parametr *t* jiným písmenkem, například *p* (protože *t* je pod vlivem posuvníku). Z výrazu `#14` vybereme vektor souřadnic (tedy část napravo od přiřazovacího příkazu) a provedeme substituci proměnné *t* proměnnou *p*. Dostaneme výraz `#15`, jehož zobrazením je hledaná křivka:

$$\#15: \quad \left[-\frac{p^3 - 5\cdot p^2 + 4\cdot p - 1}{3\cdot p^2 - 3\cdot p + 1}, \frac{p^3 + 2\cdot p^2 - 3\cdot p + 1}{3\cdot p^2 - 3\cdot p + 1} \right]$$

Výsledkem uvedeného postupu je, díky užití posuvníku, dynamický obrázek, který názorně ilustruje zadání i řešení daného problému (Obr. 39). V případě námi vyšetřovaného rovnoramenného trojúhelníku je množinou středů kružnic opsaných trojúhelníku *DEF* křivka zvaná Descartův list (Folium of Descartes). Důkaz odvozením rovnice křivky je podán v příloze.



Obr. 39: Descartův list

Příklad 20.10: Strofoida
Uvažujme svazek kružnic jejichž společnou tečnou je osa x a společným bodem dotyku je počátek. Zvolíme-li na ose x bod $A = [a,0]$ a vedeme-li jím průměry ke všem kružnicím svazku, potom krajní body těchto průměrů jsou body křivky, kterou nazýváme *strofoida*.

Naformátováno: Příklad, Řádkování: jednoduché, Ohraničení: Pole: (stínované jednoduché, Automatická, 1 b. šířka čáry)

Řešení:
Úlohu budeme řešit dynamicky, s využitím posuvníku. Parametrem k bude y-ová souřadnice středu kružnice svazku (k je tedy zároveň i poloměr této kružnice).
Rovnicí ve středovém tvaru definujeme kružnici svazku:

#1: $x^2 + (y - k)^2 = k^2$

Zadáme konkrétní souřadnice bodu *A*:

#2: A ≔ [10, 0]

#3: S ≔ [0, k]

Přímku definující svými průsečíky s kružnicí body strofoidy, která je určena bodem *A* a středem *S* kružnice zadáme parametricky:

#4: P ≔ A + t·(S − A)

Potom průsečíky určíme dosazením souřadnic P_1, P_2 z tohoto parametrického vyjádření do rovnice kružnice a jejím řešením vzhledem k neznámé *t*:

#5: $P_1^2 + (P_2 - k)^2 = k^2$

#6: $k^2 \cdot (t - 1)^2 + 100 \cdot (t - 1)^2 = k^2$

Použijeme příkaz `SOLUTIONS` a řešení označíme `tt`, aby nedošlo k případné kolizi s obecným parametrem `t`:

#7: $tt ≔ \mathrm{SOLUTIONS}(k^2 \cdot (t - 1)^2 + 100 \cdot (t - 1)^2 = k^2, t, \mathrm{Real})$

#8: $$tt ≔ \left[\frac{\sqrt{(k^2 + 100)} + k}{\sqrt{(k^2 + 100)}}, \frac{\sqrt{(k^2 + 100)} - k}{\sqrt{(k^2 + 100)}}\right]$$

Výsledné hodnoty parametru `tt1`, `112` dosadíme do parametrického vyjádření přímky `#4` a dostaneme tak průsečíky B1 a B2:

#9: $B1 ≔ A + tt_1 \cdot (S - A)$

#10: $$B1 ≔ \left[- \frac{10 \cdot k}{\sqrt{(k^2 + 100)}}, \frac{k \cdot (\sqrt{(k^2 + 100)} + k)}{\sqrt{(k^2 + 100)}}\right]$$

#11: $B2 ≔ A + tt_2 \cdot (S - A)$

$$\#12: \quad B2 := \left[\frac{10\cdot k}{\sqrt{(k^2 + 100)}}, \frac{k\cdot(\sqrt{(k^2 + 100)} - k)}{\sqrt{(k^2 + 100)}}\right]$$

Trajektorii těchto průsečíků při proměnném parametru k, tedy onu strofoidu, dostaneme zobrazením následující matice #15, která vznikne z vektoru #13 záměnou proměnné k, která je pod vlivem posuvníku, jiným, dosud nepoužitým, písmenkem, např. u.

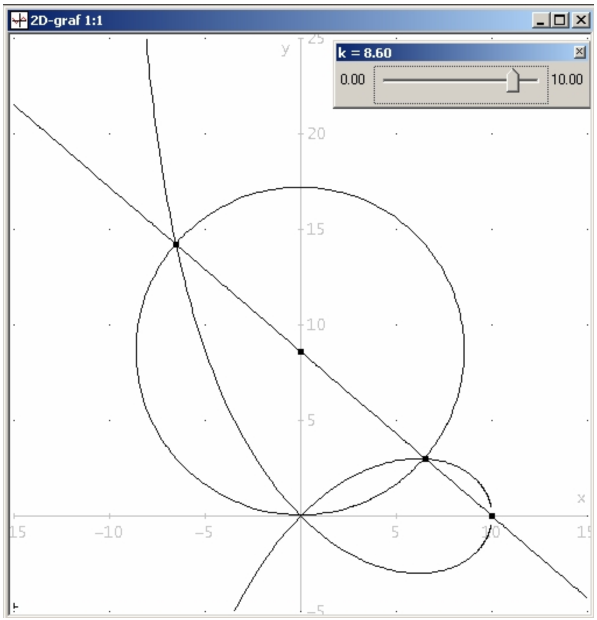
#13: [B1, B2]

$$\#14: \quad \begin{bmatrix} -\frac{10\cdot k}{\sqrt{(k^2 + 100)}} & \frac{k\cdot(\sqrt{(k^2 + 100)} + k)}{\sqrt{(k^2 + 100)}} \\ \frac{10\cdot k}{\sqrt{(k^2 + 100)}} & \frac{k\cdot(\sqrt{(k^2 + 100)} - k)}{\sqrt{(k^2 + 100)}} \end{bmatrix}$$

$$\#15: \quad \begin{bmatrix} -\frac{10\cdot u}{\sqrt{(u^2 + 100)}} & \frac{u\cdot(\sqrt{(u^2 + 100)} + u)}{\sqrt{(u^2 + 100)}} \\ \frac{10\cdot u}{\sqrt{(u^2 + 100)}} & \frac{u\cdot(\sqrt{(u^2 + 100)} - u)}{\sqrt{(u^2 + 100)}} \end{bmatrix}$$

Dynamický obrázek dostaneme takto:
Otevřeme 2D grafické okno, vložíme do něho posuvník pro proměnnou *k* a potvrdíme volby **Možnosti** → **Zjednodušit před vykreslením** a **Možnosti** → **Zobrazení...** → **Body** → **Velikost** → **Velká**. Poté v okně zobrazíme kružnici #1, body *A*, *S* (#2, #3), přímku #4 a průsečíky *B1*, *B2*. Všechny tyto objekty (s výjimkou bodu A) jsou pohyblivé spolu s pohybem posuvníku. Křivku strofoidu, jakožto trajektorii bodů *B1*, *B2*, dostaneme zobrazením výrazu #15. Při pohybu posuvníku opisují průsečíky kružnice a přímky skutečně tuto křivku. Výsledná dynamická

konstrukce křivky v 2D grafickém okně je zachycena na následujícím obrázku 40.



Obr. 40: Dynamická konstukce strofoidy

20.4. Polární souřadnice

Možnost zobrazovat grafy v polárních souřadnicích jsme využili již při zkoumání spirál na straně 85. V této části si uvedeme další dva příklady využití těchto souřadnic. Nejprve pokračujeme vyšetřováním strofoidy, poté se budeme zabývat další křivkou, zvanou kardioida.

Příklad 20.11: Na základě zadání příkladu 20.10 se pokuste odvodit rovnici *strofoidy* v polárních souřadnicích a poté tuto rovnici zobrazte.

Řešení:

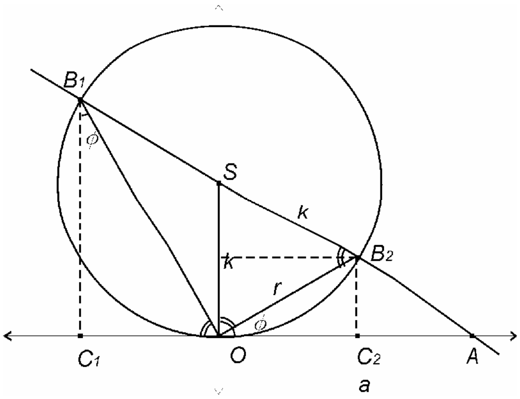
Z obrázku 41, který odpovídá zadání předchozího příkladu 20.10, je zřejmá platnost těchto dvou vztahů: $\cos f = \frac{|OC_2|}{r}$, $\frac{|OC_2|}{k} = \frac{a}{\sqrt{a^2+k^2}}$, kde $r = |OB_2|$, $k = |SO|$

a $A = [0, a]$. Jejich zjednodušením pak dostaneme rovnost $r \cos f = a \dfrac{k}{\sqrt{a^2 + k^2}}$.

Při bližším průzkumu obrázku 41 dále zjistíme, že $|\,!\ OSB_2| = 2f$. Potom ale

můžeme psát, že $\dfrac{k}{\sqrt{a^2 + k^2}} = \cos 2f$. Předchozí rovnost tak lze vyjádřit ve tvaru

$r = a \dfrac{\cos(2j)}{\cos j}$, což je hledaná rovnice strofoidy v polárních souřadnicích (r, f).



Obr. 41: Odvození rovnice strofoidy v polárních souřadnicích

Nyní je naším cílem zobrazit odvozenou rovnici v polárních souřadnicích.

```
              COS(2·φ)
#1:    r = a·————————
               COS(φ)
```

Před zobrazením #1 nezapomeneme při aktivním 2D grafickém okně nastavit soustavu polárních souřadnic: **Nastavení → Souřadnicový systém... → Polární**
Aby křivka nebyla zobrazena zkresleně (s rozdílným vertikálním a horizontálním měřítkem), provedeme: **Nastavení → Poměr stran... → Obnovit → OK**

Naformátováno: Derive Menu Char Char, Písmo: (výchozí) Times New Roman, 10 b., vzorek: Žádný

Pokud nechceme zadat konkrétní hodnotu parametru *a* (zajímá nás, jak ovlivňuje podobu grafu), vložíme do grafického okna posuvník pro hodnoty této proměnné. Potom výraz zvýrazníme a stiskneme tlačítko ⊞. Program sám identifikuje výraz #1 jako parametrický, proto se před zobrazením grafu ptá na rozsah hodnot jeho parametru. Po vyplnění dostaneme dynamický graf křivky známé z obrázku 40.
Poznámka:

Při zápisu výrazu v polárních souřadnicích můžeme pro označení těchto souřadnic používat libovolnou dvojici symbolů. Tedy místo dvojice (r, f) jsme klidně mohli použít dvojici (r, a) apod.

Naformátováno: Písmo: 11 b., není Tučné

Naformátováno: Písmo: 11 b., není Tučné

Příklad 20.12: Nechť *P* je pevný bod na kružnici *K*. Uvažujme množinu všech kruhů D_i, jejichž střed leží na *K* a hraniční kružnice prochází bodem *P*. Jaký je tvar plochy, kterou všechny tyto disky pokrývají? Jinak řečeno, jaká je jejich obálka.

Naformátováno: Zarovnat do bloku, Řádkování: jednoduché, Ohraničení: Pole: (stínované jednoduché, Automatická, 1 b. šířka čáry)

Řešení:

Pro reprezentaci problému zvolíme polární souřadnice. Rovnice kružnice v obecné poloze má v polárních souřadnicích tvar

$$r^2 - 2rr_0 \cos(a - a_0) + r_0^2 - R^2 = 0, \tag{1}$$

kde (r_0, a_0) jsou souřadnice středu kružnice a R je její poloměr (je to vlastně kosinová věta pro trojúhelník složený z uvedených prvků). Pokud pro kružnici *K* zvolíme polohu, ve které bod *P* splývá s pólem soustavy souřadnic a pro souřadnice středu platí $r_0^K = R^K = \frac{1}{2}, a_0^K = \frac{p}{2}$, její rovnice je oproti uvedenému obecnému tvaru značně jednodušší a vypadá takto

$$r^K = \sin a^K.$$

To je ale, dle zadání, rovnice množiny poloh středů všech kruhů D_i, jejichž obálku hledáme. Protože jejich hranice procházejí bodem *P* (pólem), pro poloměr každého z nich platí $R_i = r_{0i} = r^K = \sin a^K$. Po dosazení do rovnice (1) dostaneme

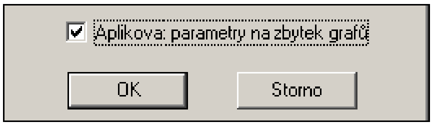
$$r - 2\sin a^K \cos(a - a^K) = 0. \tag{2}$$

Po dosazení konkrétní hodnoty za a^K do (2) tak dostaneme rovnici hraniční kružnice jednoho z uvažovaných kruhů D_i. Úlohu vyřešíme graficky tak, že do jedné soustavy souřadnic zobrazíme dostatečné množství těchto kružnic.

Využijeme příkaz `VECTOR(f(k),k,d,h,krok)` který je obdobou cyklu FOR známého z většiny programovacích jazyků. Příkaz postupně dosazuje do výrazu `f(k)` za proměnnou `k` hodnoty od `d` do `h` s krokem `krok` (parametr `krok` může být ve tvaru zlomku, může být i záporný a když ho neuvedeme, použije se implicitní hodnota 1). Výsledky dosazení (číselné hodnoty, výrazy apod.) se ukládají jako složky do vektoru, jehož dimenze je tedy rovna počtu dosazení. Pokud jsou složkami onoho vektoru předpisy funkcí (což je náš případ), můžeme všechny tyto funkce najednou zobrazit obvyklým způsobem, pokud předtím celý vektor zvýrazníme. Jenom jednu složku (funkci) zobrazíme poté, co z celého vektoru zvýrazníme právě jenom ji.

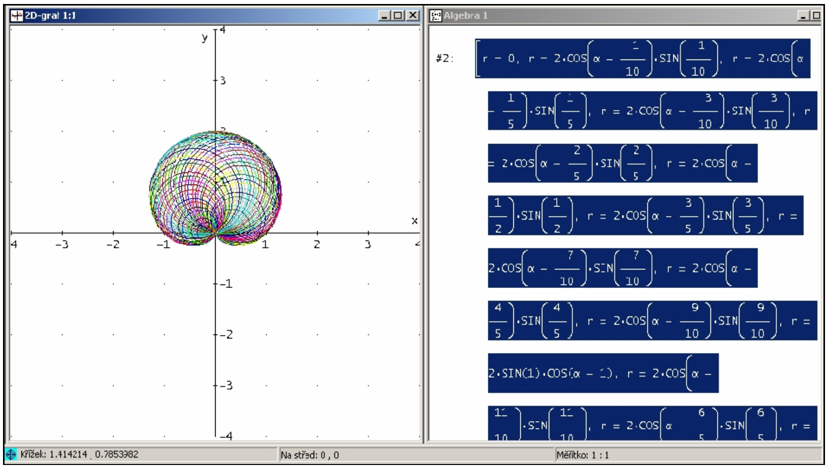
Postup řešení:

1) Do okna Algebra vložíme příkaz
`VECTOR(r=2sin(k)cos(α–k), k, –π, π, 1/10)`
2) Příkaz zjednodušíme ($\boxed{=}$, **Zjednodušit → Základní zjednodušení…**)
3) Na následujícím řádku se objeví vektor rovnic jednotlivých kružnic. Necháme ho zvýrazněný a zobrazíme ho v 2D grafickém okně v polárních souřadnicích.
Všimneme si, že v okně pro zadání parametrů grafu se objevila volba **Aplikovat parametry na zbytek grafů**:



Volbu zatrhneme. Tak zabráníme tomu, že se toto okno zobrazí pro každou kružnici zvlášť.

Výsledek vidíme na Obr. 42. Hraniční křivkou zkoumané plochy je křivka zvaná *kardioida* (*srdcovka*).



Obr. 42: Kardioida

Naformátováno: zarovnání na střed, Řádkování:

Naformátováno: Písmo: 11 b.

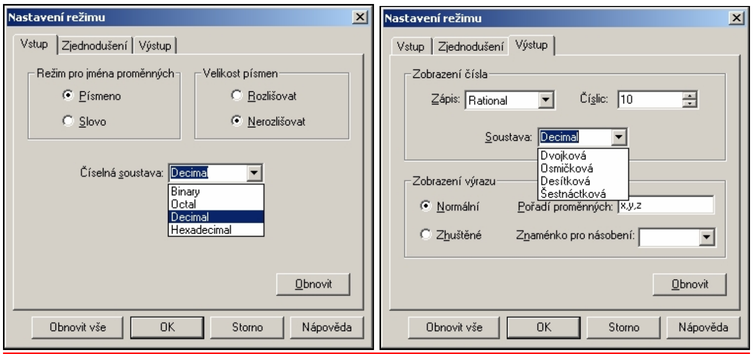
21. Teorie čísel

21.1. Číselné soustavy

Pro převod zápisu čísla do jiné číselné soustavy nebo přímo pro počítání v jiné číselné soustavě využijeme dialogové okno příkazu **Možnosti** → **Nastavení...** → **Vstup** (Obr. 43), konkrétně volbu **Číselná soustava**, která nám umožní nastavit základ číselné soustavy, v níž je reprezentováno vkládané číslo. Program nám přímo nabízí čtyři základy (2…Binary, 8…Octal, 10…Decimal, 16…Hexadecimal), ale jinak je možné do příslušného bílého políčka vepsat libovolné číslo (základ) od 2 do 36. Pro označení číslic větších než 10 se použije odpovídající písmeno abecedy. Chceme-li psát číslice samostatně, píšeme před písmena nulu, např. 0A pro deset, 0C pro dvanáct. Analogicky, pro nastavení reprezentace číselné hodnoty výstupu, použijeme volbu **Možnosti** → **Nastavení...** → **Výstup** → **Soustava** (Obr. 44).

Naformátováno: Písmo: 11

Naformátováno: Písmo: 11 b., vzorek: Žádný (Automatická)



Obr. 43, 44: Nastavení vstupních a výstupních podmínek

Naformátováno: Písmo: 11

Naformátováno: Písmo: 11

Naformátováno: Písmo: 11 b., není Tučné

Naformátováno: Písmo: 11 b., není Tučné

Naformátováno: Písmo: 11

Naformátováno: Řádkování: jednoduché, Ohraničení: Pole: (stínované jednoduché, 1 b. šířka čáry)

Příklad 21.1 : Převeďte číslo 2748 z desítkové soustavy do soustavy o základu 32.

Řešení:

Na vstupu nastavíme desítkovou soustavu a na výstupu soustavu o základu 32:
Možnosti → **Nastavení...** → **Vstup** → **Číselná soustava** → **Decimal**,
Možnosti → **Nastavení...** → **Výstup** → **Soustava** → **32**

Na příkazový řádek píšeme číslo
2748,
ale na pracovní ploše se po jeho odeslání (Enter nebo ✓) objeví odpovídající zápis v soustavě o základu 32:

2LS

Příklad 21.2: Ve dvojkové soustavě proveďte výpočty 10010 + 111, 10010*111, 10010/111

Naformátováno: Řádkování: jednoduché, Ohraničení: Pole: (stínované jednoduché, 1 b. šířka čáry)

Řešení:
Možnosti → Nastavení... → Vstup → Číselná soustava → Binary,
Možnosti → Nastavení... → Výstup → Soustava → Binary

```
#1:   InputBase := Binary

#2:   OutputBase := Binary

#3:                         10010 + 111 = 11001

#4:                         10010·111 = 1111110
```

#5: $$\frac{10010}{111} = \frac{10010}{111}$$

#6: $$\frac{10010}{111} = 10.10010010$$

Vyjádření výsledku desetinným číslem na řádku #6 dostaneme zvýrazněním pravé strany rovnosti #5 a stisknutím tlačítka $\boxed{\approx}$ pro aproximaci vybraného výrazu. K původnímu nastavení programu se nejrychleji vrátíme volbou
Možnosti → Nastavení... → Obnovit vše.

Příklad 21.3: Některá čísla zobrazená na displeji kalkulačky dávají smysl i když se na ně díváme obráceně, tj. když kalkulačku otočíme „nohama vzhůru“. Například z čísla 1995 se stane 5661. Páté číslo, které je čitelné i nohama vzhůru je 8 a patnácté číslo s touto vlastností je 21, které se otočením změní na 12. Určete miliónté číslo, které je čitelné i nohama vzhůru.

Řešení:
Myšlenka řešení je založena na předpokladu, že číslice, které jsou čitelné i „nohama vzhůru“ (tj. 1, 2, 5, 6, 8, 9, 0) představují úplnou sadu číslic číselné soustavy o základu 7; 0=0, 1=1, 2=2, 3=5, 4=6, 5=8, 6=9. Potom je naším úkolem jenom nalezení zápisu čísla 1 000 000 v této soustavě. Nastavíme-li výstup hodnot v programu Derive v sedmičkové soustavě (tj, do políčka na příslušné kartě napíšeme číslo 7), stačí napsat na příkazový řádek číslo 1000000 a po odeslání se v okně Algebra objeví jeho zápis v soustavě sedmičkové.

Poznámka: Nenechme se splést výstupem příkazu nastavení základu (viz řádek #1 níže). Výstup je přece v sedmičkové soustavě.

```
#1:   OutputBase := 10

#2:   11333311
```

Zápis hledaného čísla na řádku #2 ještě vyžaduje náležitou interpretaci vzhledem k „našim“ symbolům (tj. 1, 2, 5, 6, 8, 9, 0) pro číslice v sedmičkové soustavě. Číslici

1 v zápisu čísla odpovídá opět 1, ale číslici 3 v sedmičkové soustavě jsme přidělili symbol 5. Řešením úlohy je tedy číslo 11555511.

21.2. Funkce teorie čísel

21.2.1. Rozklad na prvočinitele

Příklad 21.4: Proveďte rozklad čísla 4956 na prvočinitele.

Naformátováno: Řádkování: jednoduché, Ohraničení: Pole: (stínované jednoduché, 1 b. šířka čáry)

Řešení:
Použijeme příkaz
Zjednodušit → Rozložit na činitele... (za kterým je skryta funkce FACTOR).
V příslušném dialogovém okně můžeme zvolit typ rozkladu. Prvočíselný rozklad ale dostaneme pro jakýkoliv z uvedených typů.

#1: 4956

#2: FACTOR(4956, Number)

#3: $2^2 \cdot 3 \cdot 7 \cdot 59$

Nebo můžeme nastavit režim „Zobrazit kroky“

#1: DisplaySteps := true

#2: 4956

#3: FACTOR(4956, Number)

If n^p divides m,

$$m \rightarrow n^p \cdot \frac{m}{n^p}$$

#4: $\mathrm{FACTOR}(2^2 \cdot 1239)$

If n divides m,

$$m \rightarrow n \cdot \frac{m}{n}$$

#5: $\mathrm{FACTOR}(2^2 \cdot 3 \cdot 413)$

If n divides m,

$$m \rightarrow n \cdot \frac{m}{}$$

n

#6: $\text{FACTOR}(2^2\cdot 3\cdot 7\cdot 59)$

If n is fully factored,

FACTOR(n) → n

#7: $2^2\cdot 3\cdot 7\cdot 59$

Naformátováno: Písmo: 12 b.

21.2.2. Největší společný dělitel

Víme, že Derive má mnoho funkcí, které nejsou „pokryty“ grafickým rozhraním. Přesto mohou být pro nás užitečné. V souvislosti s teorií čísel stojí za zmínku například funkce:

GCD(n1,n2,…,nn) … největší společný dělitel *(the greatest common divisor)*,
LCM(n1,n2,…,nn) … nejmenší společný násobek *(the least common multiple)*,
PRIME?(n) … test prvočísla,
NEXT_PRIME(n) (PREVIOUS_PRIME(n)) … následující (předchozí) prvočíslo.
MOD(m,n) ... nezáporný zbytek při celočíselném dělení m/n, tj. kongruence m modulo n.

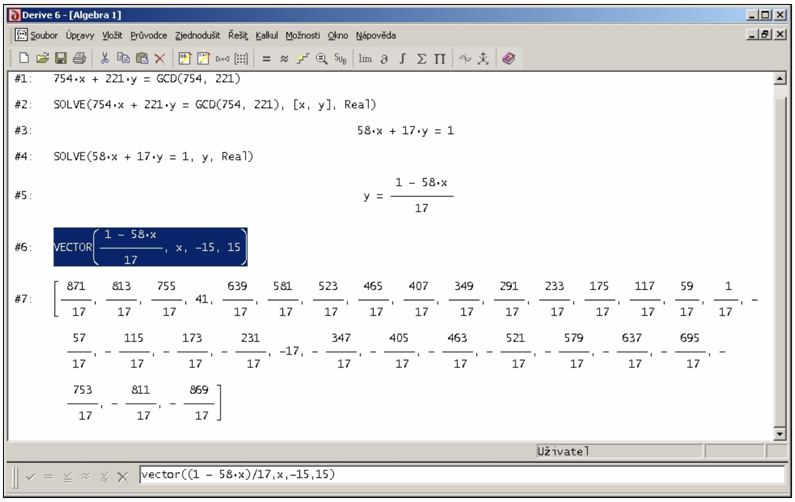
Kompletní informace o všech funkcích získáme v nápovědě programu (**Nápověda → Témata nápovědy → Vestavěné funkce a konstanty → Funkce teorie čísel**).
Další vhodné funkce najdeme v balíčku NUMBER.MTH (**Nápověda → Témata nápovědy → Uživatelské balíčky matematických funkcí**).

Příklad 21.5: Najděte taková celá čísla *x, y*, pro která platí
$$754x + 221y = D(754{,}221)$$

Naformátováno: Řádkování: jednoduché, Ohraničení: Pole: (stínované jednoduché, Automatická, 1 b. šířka čáry)

Řešení:
Kromě výše zmíněné funkce GCD použijeme příkaz SOLVE (**Řešit → Výraz**) a příkaz VECTOR(f(n),n,d,h,k). Postup řešení i výsledek vidíme na Obr. 45.

Poznámka: Místo funkce VECTOR by možná byla vhodnější funkce TABLE. Vyzkoušejte.



Obr. 45: Největší společný dělitel GCD.

Naformátováno: Písmo: 11

Naformátováno: Písmo: 11

Naformátováno: Písmo: 11 b., není Tučné

21.2.3. Celočíselné dělení

Příklad 21.6: Určete koncové trojčíslí čísla $2007^{2007^{2007}}$.

Řešení:

Koncové trojčíslí je rovno zbytku při dělení daného čísla číslem 1000.

Z příkazového řádku tedy odešleme výraz `MOD(2007^2007^2007, 1000) =`

#1: $$\mathrm{MOD}\left(2007^{2007^{2007}}, 1000\right) = 343$$

Koncové trojčíslí je 343.

Příklad 21.7: Rozhodněte o dělitelnosti výrazu $13^{13^{13}} + 14^{14^{14}}$ číslem 11.

Řešení:

#1: $$\mathrm{MOD}\left(13^{13^{13}}, 11\right) = 8$$

#2: $$\mathrm{MOD}\left(14^{14^{14}}, 11\right) = 3$$

Po sečtení zbytků dostáváme 11. Odpověď tedy zní, že daný výraz je dělitelný 11.

21.3. Diofantická rovnice

Počítačový program může žákům pomoci dotáhnout do konce řešení úlohy, jejíž zadání je snadno pochopitelné, matematické vyjádření nepřesahuje rámec učiva, ale úpravy vedoucí k řešení jsou natolik komplikované nebo speciální, že není nijak přínosné s nimi všechny žáky zatěžovat. Jak ukazuje následující příklad, program v takovém případě rozhodně nemusí hrát roli nějaké černé skříňky, do níž strčíme zadání a vypadne správné řešení. Naopak, postup řešení takovéto úlohy se zápisem v programu stane zřejmý a nijak obtížný.

Příklad 21.6: Chytali spolu čtyři poctiví rybáři. Skončili za tmy, úlovek nepočítali a šli spát. V noci jeden vstal, rozpočítal chycené ryby na čtyři stejné díly, jedna ryba mu zbyla a tak ji hodil do moře. Vzal si svou čtvrtinu úlovku a odešel. Totéž udělal po chvíli druhý rybář, potom třetí a čtvrtý. Každý z nich si myslel, že vstal první a ostatní ještě spí. Jaký je nejmenší počet ryb, které rybáři ulovili?

Řešení:

Hledaný nejmenší počet ryb označíme proměnnou *r*. Podíly jednotlivých rybářů potom proměnnými *x*, *y*, *z*, *v*.

Pomocí přiřazovacích příkazů popíšeme zmiňovaná čtyři noční dělení:

#1: $x := \frac{r - 1}{4}$

#2: $y := \frac{3 \cdot x - 1}{4}$

#3: $z := \frac{3 \cdot y - 1}{4}$

#4: $v := \frac{3 \cdot z - 1}{4}$

Zjednodušíme v (v= (¢))

#5: $v = \frac{27 \cdot r - 175}{256}$

Lepší je vyjádřit r:

#6: $\text{SOLVE}\left(v = \frac{27 \cdot r - 175}{256}, r, \text{Real}\right)$

„Vyčistíme“ proměnné `v, r`:

#7: `v :=`

#8: `r :=`

#9: $$r = \frac{256 \cdot v + 175}{27}$$

Použijeme přiřazovací příkaz k definici `r` jako proměnné:

#10: $$r := \frac{256 \cdot v + 175}{27}$$

A vypíšeme tabulku prvních padesáti hodnot `r`:

#11: `TABLE(r, v, 1, 50)`

#12:
$$\begin{bmatrix} 1 & \frac{431}{27} \\ 2 & \frac{229}{9} \\ \cdots & \cdots \\ 25 & \frac{6575}{27} \\ \mathbf{26} & \mathbf{253} \\ 27 & \frac{7087}{27} \\ \cdots & \cdots \\ 50 & \frac{4325}{9} \end{bmatrix}$$

První celočíselná hodnota $r = 253$ odpovídá hodnotě $v = 26$. Nejmenší počet ulovených ryb je tedy 253.

22. Kombinatorika

K dispozici jsou nám mimo jiné tyto funkce:
`PERM(m, n)` … počet variací *n*-té třídy z *m* prvků
`PERM(n, n) = n!.`
`COMB(m, n)` … počet kombinací *n*-té třídy z *m* prvků (kombinační číslo)
`RANDOM(n)` … generátor náhodných čísel

Příklad 22.1: Zobrazte prvních 5 řádků Pascalova trojúhelníku

Řešení:
Na příkazový řádek napíšeme a na pracovní plochu odešleme výraz:
`VECTOR(VECTOR(COMB(m, n), n, 0, 5), m, 0, 5)`

#1: `VECTOR(VECTOR(COMB(m, n), n, 0, 5), m, 0, 5)`

Po zjednodušení #1 dostaneme:

#2: $$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 \\ 1 & 3 & 3 & 1 & 0 & 0 \\ 1 & 4 & 6 & 4 & 1 & 0 \\ 1 & 5 & 10 & 10 & 5 & 1 \end{bmatrix}$$

Příklad 22.2: Řešte v N rovnici $\dfrac{(n+2)!}{n!} = 2\dfrac{n!}{(n-2)!} + 3!$.

Řešení:
Řešíme přímo (tj. na #1 použijeme **Řešit → Výraz…**):

#1: $$\frac{(n + 2)!}{n!} = \frac{2\cdot n!}{(n - 2)!} + 3!$$

#2: $$\mathrm{SOLVE}\left(\frac{(n + 2)!}{n!} = \frac{2\cdot n!}{(n - 2)!} + 3!,\ n,\ \mathrm{Real}\right)$$

#3: $$n = 4 \vee n = 1$$

Nebo řešíme postupnými úpravami obou stran rovnice:

#4: $$\frac{(n+2)!}{n!} = \frac{2\cdot n!}{(n-2)!} + 3!$$

Provedeme základní zjednodušení:

#5: $$(n+1)\cdot(n+2) = 2\cdot(n^2 - n + 3)$$

Roznásobíme:

#6: $$n^2 + 3\cdot n = 2\cdot n^2 - 2\cdot n + 4$$

Převedeme vše na jednu stranu

#7: $$(n^2 + 3\cdot n = 2\cdot n^2 - 2\cdot n + 4) - n^2 - 3\cdot n$$

#8: $$0 = n^2 - 5\cdot n + 4$$

a rozložíme na součin kořenových činitelů:

#9: $$0 = (n-1)\cdot(n-4)$$

Řešením jsou hodnoty $n = 1$, $n = 4$.

23. Komplexní čísla

Komplexní čísla můžeme v Derive zapisovat v algebraickém tvaru $a + b \cdot i$ a, jak je vidět z následující ukázky, běžně s nimi počítat. K dispozici jsou též funkce pro rozbor komplexních čísel a manipulaci s nimi, např. ABS, RE, IM , CONJ atd. Protože vše potřebné najde zájemce v Nápovědě programu, zaměříme se v této kapitole jenom na seznámení s možnostmi grafického znázornění komplexních čísel.

Základní početní operace s komplexními čísly v algebraickém tvaru:

#1: $$(3 + 4 \cdot i) \cdot (6 - i) = 22 + 21 \cdot i$$

#2: $$\frac{3 + 4 \cdot i}{6 - i} = \frac{14}{37} + \frac{27 \cdot i}{37}$$

#3: $$(3 + 4 \cdot i) + (6 - i) = 9 + 3 \cdot i$$

#4: $$|3 + 4 \cdot i| = 5$$

#5: $$\mathrm{CONJ}(3 + 4 \cdot i) = 3 - 4 \cdot i$$

Připomínám, že symbol i imaginární jednotky **nelze zaměňovat s běžným písmenem** i. Tento symbol buď získáme výběrem z panelu matematických symbolů, nebo ho napíšeme pomocí kombinace kláves Ctrl + i, případně dvojicí znaků #i.

V Derive není funkce pro znázornění komplexního čísla v Gaussově rovině. V této kapitole si předvedeme balíček funkcí, který by mohl tento nedostatek řešit.

Pro správné fungování uvedených procedur je třeba provést tato nastavení 2D grafického okna:
Možnosti → Zobrazení → Body → Spojovat ANO
Možnosti → Zjednodušit před vykreslením
Doporučuji ještě vypnout režim automatického střídání barev grafů:
Možnosti → Zobrazení → Barva → Automaticky měnit barvu nových grafů NE

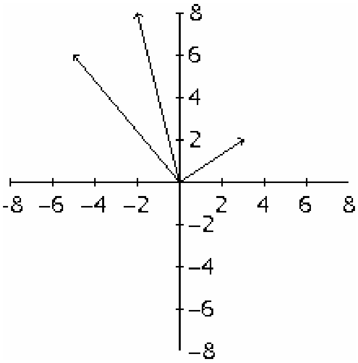
Procedury pro znázornění komplexních čísel:

1. KC(z) ... znázornění komplexního čísla z jako úsečky v Gaussově rovině.

#1: $$\mathrm{KC}(z) := \begin{bmatrix} 0 & 0 \\ \mathrm{RE}(z) & \mathrm{IM}(z) \end{bmatrix}$$

2. `KC_r(z)` ... znázornění komplexního čísla z pomocí jeho průvodiče r v Gaussově rovině (Rozdíl oproti předchozímu znázornění úsečkou je v tom, že průvodič je zakončen šipkou.).

$$\#1:\mathrm{KC\_r}(z) := \left[\begin{bmatrix} 0 & 0 \\ \mathrm{RE}(z) & \mathrm{IM}(z) \end{bmatrix}, \left[[\mathrm{RE}(z), \mathrm{IM}(z)], [\mathrm{RE}(z), \mathrm{IM}(z)] + \frac{1}{4\cdot|z|}\cdot\left[-\frac{\sqrt{2}}{2}\cdot\mathrm{RE}(z) + \frac{\sqrt{2}}{2}\cdot\mathrm{IM}(z), -\frac{\sqrt{2}}{2}\cdot\mathrm{RE}(z) - \frac{\sqrt{2}}{2}\cdot\mathrm{IM}(z)\right]\right], \left[[\mathrm{RE}(z), \mathrm{IM}(z)], [\mathrm{RE}(z), \mathrm{IM}(z)] + \frac{1}{4\cdot|z|}\cdot\left[-\frac{\sqrt{2}}{2}\cdot\mathrm{RE}(z) - \frac{\sqrt{2}}{2}\cdot\mathrm{IM}(z), +\frac{\sqrt{2}}{2}\cdot\mathrm{RE}(z) - \frac{\sqrt{2}}{2}\cdot\mathrm{IM}(z)\right]\right]\right]$$



3. `KomplOdmGraf_r(r, n)`, `KomplOdmGraf(r, n)` ... znázornění jednoho řešení nebo vektoru více řešení `r` binomické rovnice, tj. znázornění komplexní odmocniny z daného čísla. Parametr `r` je vektor řešení rovnice, který je výsledkem použití funkce `SOLUTIONS`.

a) Užitím průvodičů

```
      KomplOdmGraf_r(r, n) :=
        Prog
#1:       n := DIM(r)
          VECTOR(KC_r(r↓k), k, 1, n)
```

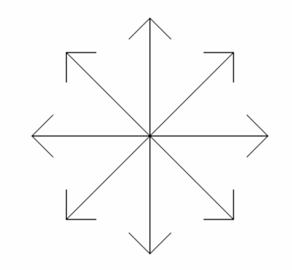
Příklad užití:

```
#1: Res := SOLUTIONS(x^8 - 1 = 0, x)
```

$$\#2:\ Res := \left[1,\ -1,\ i,\ -i,\ \frac{\sqrt{2}}{2} + \frac{\sqrt{2}\cdot i}{2},\ \frac{\sqrt{2}}{2} - \frac{\sqrt{2}\cdot i}{2},\ -\frac{\sqrt{2}}{2} + \frac{\sqrt{2}\cdot i}{2},\ -\frac{\sqrt{2}}{2} - \frac{\sqrt{2}\cdot i}{2}\right]$$

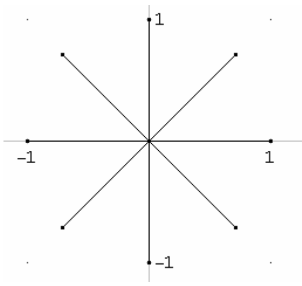
```
#3: KomplOdmGraf_r(Res)
```

Výraz #3 zobrazíme v 2D grafickém okně:



b) Jako úsečky v Gaussově rovině

```
      KomplOdmGraf(r, n) :=
        Prog
#1:       n := DIM(r)
          VECTOR(KC(r↓k), k, 1, n)
```



24. Finanční funkce

Příklad 24.1: V bance jsme dostali půjčku 100 000 Kč, kterou budeme splácet měsíčními splátkami při roční nominální úrokové míře 8 % s měsíčním úrokováním (Složená úroková míra pro jedno úrokovací období, kterým je měsíc, tak činí jednu dvanáctinu roční úrokové míry).
a) Kolik měsíců budeme půjčku splácet při výši (měsíční) splátky 2000 Kč?
b) Půjčku potřebujeme splatit do tří let. Jaká musí být výše měsíční splátky?

Řešení:
Využijeme finanční funkce `NPER` a `PMT`. Informace o přesném významu a syntaxi těchto a ještě několika dalších funkcí finanční matematiky najde zájemce v nápovědě.

a) Jako parametry funkce `NPER` zadáme postupně úrokovou míru pro jedno úrokovací období, výši splátky (splátka se od vkladu odlišuje záporným znaménkem) a výši půjčky:

#1: $\mathrm{PocetSplatek} := \mathrm{NPER}\left(\frac{1}{12}\cdot 0.08,\ -2000,\ 100000\right)$

#2: $\mathrm{PocetSplatek} := \dfrac{\mathrm{LN}\left(\frac{3}{2}\right)}{\mathrm{LN}\left(\frac{151}{150}\right)}$

#3: $\mathrm{PocetSplatek} := 61.02227425$

Půjčku budeme splácet 61 měsíců.

b) Jako parametry funkce `PMT` zadáme postupně úrokovou míru pro jedno úrokovací období, počet období a výši půjčky:

#4: $\mathrm{VyseSplatky} := \mathrm{PMT}\left(\frac{1}{12}\cdot 0.08,\ 3\cdot 12,\ 100000\right)$

#6: $\mathrm{VyseSplatky} := -3133.636546$

Výše měsíční splátky (splátka se od vkladu odlišuje záporným znaménkem) musí činit 3134 Kč.

25. Logické operace

Připomeňme stručně, že v Derive lze upravovat logické či množinové výrazy. V případě logických výrazů – složených výroků, můžeme programu zadat, prostřednictvím jeho funkce `TRUTH_TABLE`, vytvoření tabulky pravdivostních hodnot těchto výroků.

Příklad 25.1: Zjednodušte logický výraz: $(A \wedge B) \vee (A \wedge \neg B)$

Řešení:

`#1:   (a ∧ b) ∨ (a ∧ ¬ b)`

Provedeme základní zjednodušení `#1`:

`#2:   a`

Příklad 25.2: Rozhodněte, zda je logický výraz $(A \vee \neg A) \vee B$ tautologií nebo kontradikcí.

Řešení:

`#1:   TRUTH_TABLE(a, b, (a ∨ ¬ a) ∨ b)`

Po základním zjednodušení (=) výrazu #1 dostaneme tabulku pravdivostních hodnot daného složeného výroku:

#2:

a	b	(a ∨ ¬ a) ∨ b
true	true	true
true	false	true
false	true	true
false	false	true

Protože je podle tabulky výrok vždy pravdivý (tj. má na všech jejích řádcích hodnotu `true`), jedná se o tautologii.

26. Fraktály

Běžné nastavení kreslení 2-D grafů, v němž se pro každý následující graf použije jiná barva, spolu s příkazy ITERATE a VECTOR nám dovoluje poměrně jednoduše zobrazit některé fraktální jevy v Gaussově rovině komplexních čísel. Samozřejmě, nemůžeme čekat zázraky. Velice rychle narazíme na meze těchto možností. Uvedené příklady rozhodně nijak nezastíní speciální programy, které najdeme na internetu nebo si je napíšeme sami. Můžeme je však brát jako impuls pro pochopení a případné další zkoumání krásného světa fraktálů.

26.1. ITERATE, ITERATES

Hodnotou funkce ITERATE(f(x),x,p,n) je výsledek n opakování přiřazení f(x) → **x** (tj. do výrazu f(x) dosazuji za x opět výraz f(x)) s počáteční hodnotou x = p. Posloupnost všech n+1 postupných výsledků (i s počáteční hodnotou) tohoto přiřazení (tedy iterací funkce f(x)) dostaneme funkcí ITERATES(f(x),x,p,n).

Několik příkladů použití funkcí ITERATE, ITERATES :

#1: ITERATE(1 + x, x, a, 5) = a + 5

#2:
ITERATES(1 + x, x, a, 5) = [a, a + 1, a + 2, a + 3, a + 4, a + 5]

#3: $\mathrm{ITERATES}\left(\frac{1}{x}, x, a, 5\right) = \left[a, \frac{1}{a}, a, \frac{1}{a}, a, \frac{1}{a}\right]$

#4: ITERATES(x·i, x, i, 5) = [i, −1, −i, 1, i, −1]

26.2. VECTOR

Příkaz VECTOR(f(k),k,d,h,krok) je obdobou cyklu **FOR** známého z většiny programovacích jazyků. Příkaz postupně dosazuje do výrazu f(k) za proměnnou k hodnoty od d do h s krokem krok (parametr krok může být ve tvaru zlomku, může být i záporný a když ho neuvedeme, použije se implicitní hodnota 1). Výsledky dosazení (číselné hodnoty, výrazy apod.) se ukládají jako složky do vektoru, jehož dimenze je tedy rovna počtu dosazení.

Několik příkladů použití funkce VECTOR:

$$\#1: \quad \mathrm{VECTOR}\left(\frac{1}{n^2}, n, 1, 10, 2\right) = \left[1, \frac{1}{9}, \frac{1}{25}, \frac{1}{49}, \frac{1}{81}\right]$$

$$\#2: \quad f(x) := x + \frac{a}{x - 1}$$

$$\#3: \quad \mathrm{VECTOR}(f(x), a, 1, 3) = \left[\frac{1}{x - 1} + x, \frac{2}{x - 1} + x, \frac{3}{x - 1} + x\right]$$

26.3. Fraktální jevy v rovině komplexních čísel

Jevy popíši jen na úrovni uvedených příkladů. Uvažujme funkci komplexní proměnné *f(z)*. Zvolme nějaký bod $z = x + yi$ komplexní roviny a provádějme iterace funkce *f(z)* s počáteční hodnotou v tomto bodě. Zajímá nás jak rychle se vzdalují hodnoty těchto iterací od onoho výchozího bodu. Zvolme určitý počet iterací **n** (vzhledem k možnostem programu bude poměrně malý) a rozdělme body komplexní roviny (přesněji, nějaké její části) do skupin (lišících se obarvením) podle toho, jak daleko se "dostanou" po tomto počtu iterací od své výchozí polohy. Výsledkem budou fraktální obrazce v Gaussově rovině. Pro iterace funkce $f(z) = z^2 + c$, kde *c* je komplexní parametr, dostáváme tzv. **Juliovy množiny**. Pro konkrétní Juliovu množinu je tak charakteristická určitá hodnota parametru *c*, zatímco výchozí hodnota $z = x + yi$ probíhá nějakou oblastí komplexní roviny. Pokud výchozí bod iterací necháme pevný, třeba $z = 0$, a určitou oblastí komplexní roviny necháme probíhat naopak parametr *c*, dostaneme tzv. **Mandelbrotovu množinu**. I jiné funkce komplexní proměnné mohou přinést pozoruhodné výsledky. Zajímavé je například zkoumání iterací funkce $f(z) = \dfrac{2z^3 + 1}{3z^2}$. To je funkce, jejímiž iteracemi získáme řešení rovnice $z^3 - 1 = 0$ tzv. **Newtonovou iterační metodou**. Je zajímavé sledovat, s jakou ochotou se body roviny přibližují k některému z kořenů této rovnice. Tato "ochota" je vyjádřena hodnotou derivace iterační funkce $f'(x)$. Spádové oblasti Gaussovy roviny příslušející jednotlivým kořenům rovnice a navíc rozdělené podle rychlosti konvergence tak dostaneme zobrazením oblastí roviny komplexních čísel, lišících se hodnotou této derivace.

Naformátováno: Písmo: 11 b., není Tučné

Příklad 26.1: Juliova množina.
Zobrazte Juliovu množinu pro $c = 0.2 + 0.4i$ a oblast komplexní roviny se středem v počátku a poloměrem 2.

Naformátováno: Řádkování: jednoduché, Ohraničení: Pole: (stínované jednoduché, Automatická, 1 b. šířka čáry)

Odstraněno: ¶

Řešení: (**Obr. 46**)

1. Deklarujeme komplexní proměnné *z*, *k*

```
#1:   z :∈ Complex
#2:   k :∈ Complex
```

2. Na příkazový řádek zadáme a odešleme výraz (**Pozor** na správný zápis $\imath$) :

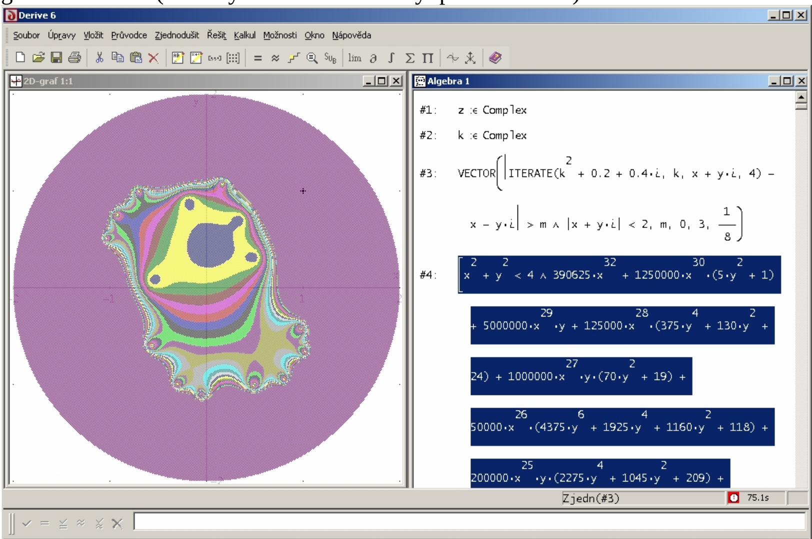
```
VECTOR(ABS(ITERATE(k^2+0.2+0.4ί,k,x+yί,4)–x–yί)>m AND ABS(x+yί)<2,
m, 0, 3, 1/8)
```

#3:

$$\mathrm{VECTOR}\left(\left|\mathrm{ITERATE}(k^2 + 0.2 + \imath\cdot 0.4,\ k,\ x + \imath\cdot y,\ 4) - x - \imath\cdot y\right| > m \wedge \left|x + \imath\cdot y\right| < 2,\ m,\ 0,\ 3,\ \frac{1}{8}\right)$$

3. Výraz zjednodušíme (můžeme volit i numerický výpočet, tj. aproximaci)
Výsledkem je vektor. Jeho složkami jsou předpisy relací definujících vždy ty body uvedené oblasti komplexní roviny, které se po čtyřech iteracích funkce $f(z) = z^2 + c$ dostaly od své výchozí polohy dále než je hodnota parametru **m**.

4. Výsledný vektor necháme zvýrazněný a obvyklým způsobem zobrazíme v 2D-grafickém okně (doba vykreslování může být poněkud delší).



Obr. 46: Juliova množina

Příklad 26.2: Newtonova iterační metoda
Zobrazte oblasti přitažlivosti jednotlivých kořenů v Gaussově rovině komplexních čísel při řešení rovnice $z^3 - 1 = 0$ Newtonovou iterační metodou.

Naformátováno: Řádkování: jednoduché, Ohraničení: Pole: (stínované jednoduché, Automatická, 1 b. šířka čáry)

Řešení: (**Obr. 47**)

Odstraněno: ¶

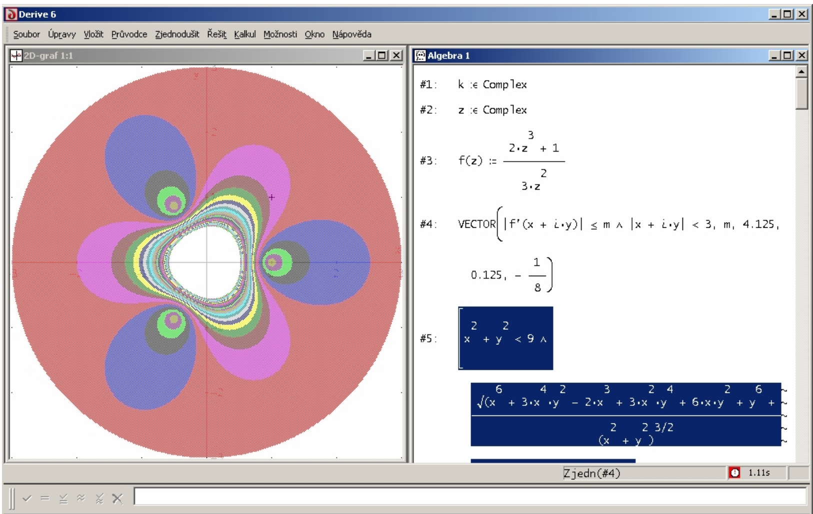
Posloupnost řešení rovnice je dána rekurentním vztahem $x_{n+1} = x_n - \dfrac{g(x_n)}{g'(x_n)}$, kde $g(x) = x^3 - 1$. Ten můžeme přepsat do tvaru $x_{n+1} = f(x_n) = \dfrac{2x_n^{\ 3} + 1}{3x_n^{\ 2}}$. Nejprve tedy definujeme tuto funkci (viz #3 na **Obr. 47**).

Potom užijeme funkci VECTOR, jejíž hodnotou bude vektor výsledků porovnání hodnoty $|f'(z)|$ s postupně se měnícími hodnotami parametru m. Omezujeme se na oblast vymezenou podmínkou $|z| < 3$.

#4:

$$\mathrm{VECTOR}\left(|f'(x + i\cdot y)| \le m \wedge |x + i\cdot y| < 3,\ m,\ 4.125,\ 0.125,\ -\frac{1}{8}\right)$$

Složkami výsledného vektoru jsou vlastně předpisy relací proměnných *x*, *y*. Při zobrazení jim na obrázku odpovídají oblasti vybarvené stejnou barvou (ne tak docela, když je oblastí více než barev, posloupnost barev se začne opakovat). Oblasti téže barvy v rovině tak odpovídají bodům, které se kořenům rovnice přibližují s přibližně stejnou ochotou.



Obr. 47: Newtonova iterační metoda, $\mathbf{z^3 - 1 = 0}$

Přílohy

1. Balíček funkcí „Integraly.mth”

Balíček funkcí pro výpočet a znázornění dolních a horních integrálních součtů **spojitých a hladkých** funkcí. Pro správnou činnost grafických funkcí je třeba nastavit tyto parametry 2D grafického okna:
Možnosti - Zjednodušit před vykreslením
Možnosti - Zobrazení - Barva - vypnout automatickou změnu barvy
Možnosti - Zobrazení - Body - Spojovat Ano

1. Pomocné procedury pro výpočet nejmenší a největší hodnoty funkce na intervalu. Předpokládá se, že funkce je na intervalu **spojitá a má derivaci v každém jeho vnitřním bodě**.

```
      NejM(g, a, b, s := [], d, h, n) :=
        Prog
          s := SOLUTIONS (∂(g, x, 1) = 0 ∧ a < x < b, x, Real)
          d := SUBST(g, x, a)
#1:       h := SUBST(g, x, b)
          If s = []
             n := MIN(d, h)
             n := MIN(d, h, s)
          n

      NejV(g, a, b, s := [], d, h, n) :=
        Prog
          s := SOLUTIONS (∂(g, x, 1) = 0 ∧ a < x < b, x, Real)
          d := SUBST(g, x, a)
#2:       h := SUBST(g, x, b)
          If s = []
             n := MAX(d, h)
             n := MAX(d, h, s)
          n
```

2. Pomocné procedury pro vykreslení obdélníků

#3: $\text{ObdNejM}(f, a, b) := \text{POLYGON\_FILL}\begin{bmatrix} a & 0 \\ b & 0 \\ b & \text{NejM}(f, a, b) \\ a & \text{NejM}(f, a, b) \end{bmatrix}$

$$\#4:\quad \text{ObdNejV}(f, a, b) := \text{POLYGON\_FILL}\begin{bmatrix} a & 0 \\ b & 0 \\ b & \text{NejV}(f, a, b) \\ a & \text{NejV}(f, a, b) \end{bmatrix}$$

3. Funkce pro grafické znázornění součtů (mají příponu _G) a pro jejich výpočet. Parametry: f ... jméno funkce nebo její předpis, a ... dolní mez, b ... horní mez, n ... počet dílčích intervalů rozdělení int. (a;b).

Příklad zadání: `DolniS_G(ln(x),3,5,8), HorniS(ln(x),4,9,15)=`

Poznámka: Zatím není vyřešeno znaménko funkce, tj. na intervalu, kde je funkce záporná se dolní součty mění v horní a naopak.

$$\#5:\quad \text{DolniS\_G}(f, a, b, n) := \text{VECTOR}\left(\text{ObdNejM}\left(f, a + \frac{b - a}{n}\cdot k, a + \frac{b - a}{n}\cdot(k + 1)\right), k, 0, n - 1\right)$$

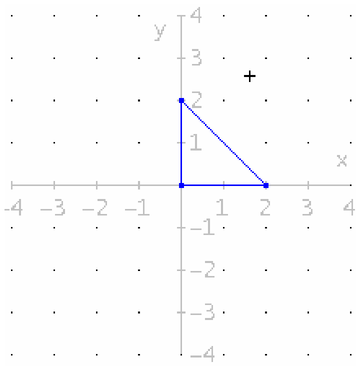
$$\#6:\quad \text{HorniS\_G}(f, a, b, n) := \text{VECTOR}\left(\text{ObdNejV}\left(f, a + \frac{b - a}{n}\cdot k, a + \frac{b - a}{n}\cdot(k + 1)\right), k, 0, n - 1\right)$$

$$\#7:\quad \text{DolniS}(f, a, b, n) := \text{APPROX}\left(\sum_{k=0}^{n-1} \text{NejM}\left(f, a + \frac{b - a}{n}\cdot k, a + \frac{b - a}{n}\cdot(k + 1)\right)\cdot\frac{b - a}{n}\right)$$

$$\#8:\quad \text{HorniS}(f, a, b, n) := \text{APPROX}\left(\sum_{k=0}^{n-1} \text{NejV}\left(f, a + \frac{b - a}{n}\cdot k, a + \frac{b - a}{n}\cdot(k + 1)\right)\cdot\frac{b - a}{n}\right)$$

2. Descartův list – kompletní řešení Příkladu 20.9

```
#1:   LOAD(C:\Derive\Derive_Packages\Geometrie.mth)

#2:   A := [0, 0]

#3:   B := [k, 0]

#4:   C := [l, m]

#5:   A1 := B + t·(C - B)

#6:   B1 := C + t·(A - C)

#7:   C1 := A + t·(B - A)

#8:   k := 2

#9:   l := 0

#10:  m := 2

#11:  [A, B, C, A]
```




```
#12:  [A1, B1, C1, A1]

#13:  StredKrOps(A1, B1, C1)
```

#14: $S := \left[-\frac{t^3 - 5\cdot t^2 + 4\cdot t - 1}{3\cdot t^2 - 3\cdot t + 1}, \frac{t^3 + 2\cdot t^2 - 3\cdot t + 1}{3\cdot t^2 - 3\cdot t + 1}\right]$

#15: $\left[-\frac{u^3 - 5\cdot u^2 + 4\cdot u - 1}{3\cdot u^2 - 3\cdot u + 1}, \frac{u^3 + 2\cdot u^2 - 3\cdot u + 1}{3\cdot u^2 - 3\cdot u + 1}\right]$

Hledáme rovnici křivky

#16: $x = -\frac{u^3 - 5\cdot u^2 + 4\cdot u - 1}{3\cdot u^2 - 3\cdot u + 1}$

#17: $y = \frac{u^3 + 2\cdot u^2 - 3\cdot u + 1}{3\cdot u^2 - 3\cdot u + 1}$

Obě parametrické rovnice nejprve převedeme na polynomické s proměnnou u:

#18: $x + \frac{u^3 - 5\cdot u^2 + 4\cdot u - 1}{3\cdot u^2 - 3\cdot u + 1}$

#19: $\left(x + \frac{u^3 - 5\cdot u^2 + 4\cdot u - 1}{3\cdot u^2 - 3\cdot u + 1}\right)\cdot(3\cdot u^2 - 3\cdot u + 1)$

#20: $x\cdot(3\cdot u^2 - 3\cdot u + 1) + u^3 - 5\cdot u^2 + 4\cdot u - 1$

#21: $3\cdot u^2\cdot x - 3\cdot u\cdot x + x + u^3 - 5\cdot u^2 + 4\cdot u - 1$

První polynom:

#22: $u^3 + u^2\cdot(3\cdot x - 5) + u\cdot(4 - 3\cdot x) + x - 1$

$$\#23:\quad \left(y - \frac{u^3 + 2\cdot u^2 - 3\cdot u + 1}{3\cdot u^2 - 3\cdot u + 1}\right)\cdot(3\cdot u^2 - 3\cdot u + 1)$$

$$\#24:\quad y\cdot(3\cdot u^2 - 3\cdot u + 1) - u^3 - 2\cdot u^2 + 3\cdot u - 1$$

Druhý polynom:

$$\#25:\quad -u^3 + y + u^2\cdot(3\cdot y - 2) + 3\cdot u\cdot(1 - y) - 1$$

Pro eliminaci proměnné u z obou rovnic použijeme metodu resultantů (je možno využít i Gröbnerovu bázi pomocí funkce GROEBNER_BASIS). Konkrétně využijeme determinant Sylvesterovy matice Syl obou polynomů.

$$\#26:\mathrm{Syl} := \begin{bmatrix} 1 & 3\cdot x - 5 & 4 - 3\cdot x & x - 1 & 0 & 0 \\ 0 & 1 & 3\cdot x - 5 & 4 - 3\cdot x & x - 1 & 0 \\ 0 & 0 & 1 & 3\cdot x - 5 & 4 - 3\cdot x & x - 1 \\ -1 & 3\cdot y - 2 & -3\cdot y + 3 & y - 1 & 0 & 0 \\ 0 & -1 & 3\cdot y - 2 & -3\cdot y + 3 & y - 1 & 0 \\ 0 & 0 & -1 & 3\cdot y - 2 & -3\cdot y + 3 & y - 1 \end{bmatrix}$$

$$\#27:\quad \mathrm{DET}(\mathrm{Syl}) = x^3 - 3\cdot x^2 + x\cdot(y + 2) + y^3 - 3\cdot y^2 + 2\cdot y - 1$$

Determinant Sylvesterovy matice je polynomickou rovnicí zkoumané křivky

$$\#28:\quad x^3 - 3\cdot x^2 + x\cdot(y + 2) + y^3 - 3\cdot y^2 + 2\cdot y - 1$$

Křivka má právě jeden singulární bod [1,1].

$$\#29:\quad F(x, y) := x^3 - 3\cdot x^2 + x\cdot(y + 2) + y^3 - 3\cdot y^2 + 2\cdot y - 1$$

$$\#30:\quad \left(\frac{d}{dx}\right)^1 F(x, y) = 3\cdot x^2 - 6\cdot x + y + 2$$

$$\#31:\quad \left(\frac{d}{dy}\right)^1 F(x, y) = x + 3\cdot y^2 - 6\cdot y + 2$$

#32:

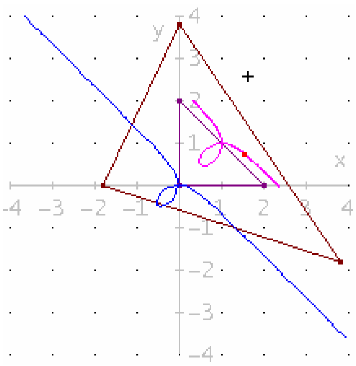
$$\text{SOLVE}\left(\left[x^3 - 3\cdot x^2 + x\cdot(y + 2) + y^3 - 3\cdot y^2 + 2\cdot y - 1,\ 3\cdot x^2 - 6\cdot x + y + 2,\ x + 3\cdot y^2 - 6\cdot y + 2\right], [x, y]\right)$$

#33: $[x = 1 \wedge y = 1]$

Po vhodné transformaci souřadnicového systému (nahradíme x a y výrazy x+1 a y+1, v tomto pořadí) dostaneme dobře známou rovnici Descartesova listu:

#34: $x^3 + x\cdot y + y^3$

#35: $x^3 + x\cdot y + y^3 = 0$



Literatura

[1] Audin, M.: *Geometry*, Springer, 2003.

[2] Burjan, V., Maxian, M.: *Opakování z matematiky pro třídy gymnázií se zaměřením na matematiku*, SPN, Praha, 1991.

[3] Böhm, J.: *Background Pictures as a Stimulating Means for Math Teaching*, Proceedings DES-TIME-2006 (formou CD), bk teachware Schriftenreihe Nr. SR-54, Germany, 2006.

[4] Böhm J.: *Narroving the gap between computer algebra and dynamic geometry*, DES-TIME-2006, workshop, Germany, 2006.

[5] Bušek, I.: *Řešené maturitní úlohy z matematiky*, SPN Praha, 1985.

[6] Calda, E.: *Z umělecké dílny profesora Ypsilona*, Rozhledy M-F, č. 9-10, 1983.

[7] Konhauser, J., Velleman, D., Wagon, S.: *Which way did the bicycle go?...and other intriguing mathematical mysteries*, The Mathematical Association of America, Dolciani Mathematical Expositions-No.18, USA, 1996.

[8] Kubát, J.: *Sbírka úloh z matematiky pro přípravu k maturitní zkoušce a k přijímacím zkouškám na VŠ*, Victoria Publishing, Praha, 1993.

[9] Kutzler, B., Kokol-Voljc, V.: *Derive 5 – The Mathematical Assistant for Your PC*, Soft Warehouse Europe, Austria, 2000.

[10] Kutzler, B., Kokol-Voljc, V.: *Derive 6, Pokročilá matematika pro vaše PC, překlad příručky programu*, České Budějovice, 2004.

[11] Livio, M.: *The Golden Ratio*, Broadway books, USA, 2003.

[12] Maor, E.: *e: The story of a number*; Princeton University Press, USA, 1998.

[13] Odvárko, O., Kadleček, J.: *Matematika pro 9. ročník z.š.*, 2. díl, Prométheus Praha, 2000.

[14] Pech, P.: *Klasické versus počítačové metody při řešení úloh v geometrii*, Jihočeská univerzita v Českých Budějovicích, České Budějovice, 2005.

[15] Šarounová, A. a kol.: *Matematika 9, II. díl*, Prométheus Praha, 2000.

[16] Voráčová, Š.: *Výuka analytické geometrie na střední škole užitím Derive*, Sborník konferencePedagogický software 2006, Scientific Pedagogical Publishing, Č. Budějovice, ISBN:80-85645-56-4, 2006, pp.225-228

[17] Vossler, D. L.: *Exploring Analytic Geometry with Mathematica*, Academic Press, London, UK, 2000.

[18] www.acdca.ac.at, *Austrian Center for Didactics of Computer Algebra.*

[19] www.austromath.at/dug, *Derive Users Group Newsletter.*

[20] www.cms.livjm.ac.uk/deriveprogramming

Rejstřík