

Matematické výpočty se systémem Maple

autor: Ing. Vladimír Žák

email: zakyn@centrum.cz

web: www.vladimirzak.com/maple

recenzent: Prof. RNDr. Jiří Hřebíček, CSc.

Institut biostatistiky a analýz

Masarykova universita

hrebicek@iba.muni.cz

Tento dokument je určen jen pro výukové účely a pro použití jen osobami, kterým byl tento dokument poskytnut přímo autorem, popř. si ho stáhli z webového sídla autora. Jakékoliv části dokumentu není možné použít bez citace zdroje, tedy názvu dokumentu a jména autora s kontaktními informacemi (jak je zobrazeno v hlavičce), která musí být autorovi sdělena. Pro jiné použití než pro osobní potřeby osob, tedy pro distribuci či kopírování, je nutné obdržet písemný souhlas autora.

>

>

> **restart:**

1. Úvod

Systém Maple byl vytvořen pro symbolické výpočty v matematice. Jde o otevřený systém, kdy do něj může jednotlivý uživatel téměř libovolně zasahovat. V této otevřenosti je schována jeho síla, ale i zárodky mnoha problémů. Tato práce vznikla na základě "poptávky" po uceleném materiálu o práci v tomto programu. Cílem není srovnání s jinými obdobnými systémy, ale návod k užití systému Maple v oblasti diferenciálního a integrálního počtu. Jako mnoho podobných programů, prochází i Maple svým vývojem. Tato práce se snaží upozorňovat na jemné rozdíly v jednotlivých verzích, ale základem se stala verze Maple 8, která byla rozšířena o další nové příkazy a rozšiřující knihovny.

>

2. Úvod do systému Maple

Maple je počítačové prostředí, které bylo vyvinuto na univerzitě Waterloo v Kanadě, pro zjednodušení a zrychlení výpočtů v matematice. Na rozdíl od klasických programů pro numerické výpočty (např. *MATLAB* - také obsahuje nástroj pro symbolické výpočty) modeluje matematické operace se symbolickými výrazy. Maple umožňuje provádět jak symbolické a numerické výpočty, tak vytvářet grafy funkcí, programovat vlastní funkce či procedury, ukládat data v několika formátech (např. *LaTeX*, *HTML*, *RTF*, *MATHML*...) a dokonce provádět export do programovacích jazyků (např. C, Fortran 77, ...).

Funkce implementované v Maplu pokrývají širokou oblast matematiky od základů lineární algebry, diferenciálního a integrálního počtu, přes diferenciální rovnice, geometrii až k logice. Základem práce jsou symbolické operace, které využívají výhody uchovávání čísla v přesném

tvaru (např. $1/6$, ne jako $0,1666\dots$), a proto MAPLE dává výsledky s mnohem větší přesností než při běžných numerických výpočtech v pohyblivé řadové čárce. Výsledek ale může být vyjádřen pomocí desetinného čísla, což vyžaduje zaokrouhlování, tedy chyby ve výpočtu.

Je-li nutné řešit problém numericky, např. při pomalém symbolické výpočtu, Maple využívá přesnosti uložení čísel pro vyjádření výsledku na libovolný počet číslic mantisy.

Jak už bylo uvedeno, je možné také vykreslovat grafy funkcí jedné (dvou) proměnných, funkcí určených parametrickými rovnicemi. Vizualizace funkce v Maplu je dána desítkami předdefinovanými grafickými funkcemi s proměnlivým počtem parametrů.

Základy práce v programu Maplu

Po spuštění programu se otevře nový dokument, který začíná znakem ">" (tzv. *prompt*) a za ním je umístěn kurzor.

Napišeme mapleovský příkaz (výraz), který má být zpracován a řádek ukončíme středníkem. Toto ukončení je nutné, protože jinak Maple očekává pokračování předchozího příkazu. V případě, že zapomenete, program vypíše varování a příkaz neprovede. Stiskneme *ENTER* a příkaz bude vykonán, tzn. program vypíše provedenou operaci na další řádek a kurzor se přesune za následující *prompt*. Pokud chceme následné vypsání potlačit, ukončíme příkaz dvojtečkou (místo středníkem). V případě, že chceme příkaz napsat na více řádků, stačí stisknout *ENTER* a středník napsat až na konci příkazu. Další možností je užití kombinace kláves SHIFT+ENTER. Po tomto odřádkování nenásleduje další *prompt*, a proto je z hlediska přehlednosti zápisu vhodnější.

2.1 Struktura systému Maple

Po spuštění systému Maple, je inicializováno pouze jeho jádro zvané *kernel*, které je rozděleno na 3 základní části:

- interpret jazyka systému Maple - převádí zadané příkazy do strojového kódu
- algoritmy pro numerické výpočty
- funkce a procedury pro zobrazení výsledků a vstupní a výstupní operace

Kernel je napsán v jazyce C, jehož zdrojový kód je vysoce optimalizován, což se odráží na malé velikosti jádra a jeho velké rychlosti. Implementuje nejfrekventovanější funkce pro výpočty aritmetických a jednoduchých polynomiálních výrazů v oboru celých a racionálních čísel.

Přes 90% matematických operací je psáno přímo v Maplu a ty jsou pak soustředěny do knihoven, které dělíme na 3 základní skupiny:

- **hlavní**
 - obsahuje nejčastější příkazy, které nejsou obsaženy přímo v jádře a není ji nutno nahrávat
- **uživatelské**
 - tvořeny méně frekventovanými matematickými příkazy a je nutné ji nahrát pomocí následujícího příkazu: *readlib(cmd)*, kde *cmd* je příkaz, který chceme nahrát
- **packages** (balíčky)
 - každý z "balíčků" obsahuje skupinu příkazů pro danou část matematiky (např. *linalg* - pro výpočty a operace s maticemi)
 - je nutné nahrát je do paměti, před použitím daných příkazů, pomocí jednoho z následujících příkazů:

- **package[cmd](...)** - nahraje příkaz *cmd* z balíčku *package* do paměti
- **with(package)** - nahraje celý balíček do paměti
- **with(package,cmd)** - nahraje příkaz *cmd* z *package* do paměti

První inicializaci používáme až v místě potřeby. Druhou hned na začátku nového dokumentu (doporučuji pro přehlednost), protože Maple přepíše funkce se stejným jménem funkcemi z daného balíčku. Nemusíte se bát, že nezjistíte, které z funkcí vlastně přepsal, protože budete upozorněni varováním, které vypíše seznam redefinovaných funkcí.

Vraťme se nyní k poslední z jmenovaných knihoven. Jak později uvidíte, je nutné mít alespoň základní přehled o možnostech jejich využití. Proto následuje stručný popis těch nejzajímavějších balíčků. Seznam není úplný, ale po napsání příkazu

> **?index,package;**

se na obrazovku vypíše veškeré dostupné balíčky.

Základní rozšiřující knihovny:

- **combinat**
 - kombinatorické funkce pro výpočty permutací, variací a kombinací
- **combstruct**
 - příkazy pro generování a počítání kombinatorických struktur
- **DEtools**
 - nástroje pro manipulaci a vykreslování systémů diferenciálních rovnic, fázové portréty
- **diffforms**
 - příkazy pro řešení problémů v diferenciální geometrii
- **finance**
 - finanční matematika
- **GaussInt**
 - pro práci s čísly tvaru: $a + b*I$, kde a, b jsou celá čísla
- **geometry**
 - dvoudimenzionální Euklidovská geometrie, definice bodů, křivek, přímek, trojúhelníků a kruhů ve dvou dimenzích
- **inttrans**
 - integrální transformace a jejich inverze
- **linalg**
 - manipulace a výpočty s maticemi a vektory
- **LRtools**
 - manipulace, vykreslování a řešení lineárních rekurentních rovnic
- **numapprox**
 - výpočty polynomiální aproximace funkce na daném intervalu
- **plots**
 - specifické funkce pro vykreslování v 2D a 3D, včetně textu a souřadnicového systému
- **simplex**
 - lineární optimalizace, simplexová metoda

- ***stats***
 - statistická manipulace s daty, průměr, variance, korelační koeficient, regresní analýza
- ***student***
 - jednoduché výpočty, integrace Per Partes, Simpsonova metoda, maximalizace funkce, hledání extrémů
- ***sumtools***
 - konečné a nekonečné součty
- ***tensor***
 - počítání s tenzory a jejich aplikace v teorii relativity
- ***VectorCalculus***
 - vektorový počet, obsahuje příkazy pro výpočet plošných a křivkových integrálů (až od Maple 8.0)

V dalším textu naleznete více o *student* package, *plots* package, *VectorCalculus* package.

POZOR

Maple rozlišuje velká a malá písmena, proto je nutné zapsat názvy balíčků přesně podle tabulky, popř. nápovědy systému Maple.

>

3. Základní konstrukce a operace systému Maple

Identifikátor je libovolná posloupnost malých a velkých písmen popř. číslic. Maximální délka je 499 znaků a identifikátor nesmí začínat číslicí a nesmí obsahovat mezery. Některé identifikátory jsou předdefinované autory Maplu, jde o tzv. klíčová slova (např. return, break, int, exp, ...). Tato slova nelze použít pro jiné účely a v případě nevhodného použití Maple ohlásí chybu.

Identifikátor se může stát jménem výrazu a to na základě přiřazovacího příkazu tvaru:

```
> identifikator:=vyraz;
```

identifikator := vyraz

V tomto případě dojde k vyhodnocení výrazu a výsledek bude označen (pojmenován) daným identifikátorem.

Poznámka:

- *komentář* - veškeré znaky na daném řádku, které následují po "#", jsou Maplem ignorovány
- *nápověda* - v systému Windows lze nápovědu vyvolat klávesou *F1*, dále z hlavního menu programu, popř. užít klíčové slovo *help(předmět_dotazu)* nebo *?předmět_dotazu*. V případě, že neznáte přesně předmět dotazu, je vhodné použít následující konstrukci: *?index, předmět_dotazu*

Matematické funkce, konstanty a početní operace Maplu

Maple obsahuje přes 2500 vnitřních funkcí a konstant (např. e, π , ...). Patří mezi ně trigonometrické, hyperbolické, Besselovy funkce a další (např. Diracova, Gaussova error funkce, ...). Následující dva odstavce uvádí nejdůležitější předdefinované konstanty, základní operace a funkce.

Předdefinované konstanty

```
> true<>>false; # logické pravda / nepravda
```

true ≠ false

```
> infinity;
```

∞

```
> Pi;
```

π

```
> exp(1);
```

e

```
> (-1)^(1/2)=I; # imaginární jednotka
```

$I = I$

Funkce a operace

```
> exp(x);
```

e^x

```
> ln(x);
```

$\ln(x)$

```
> log10(x); # dekadický logaritmus
```

$\frac{\ln(x)}{\ln(10)}$

```

> log[b] (x) ;          # logaritmus o daném základu
                        
$$\frac{\ln(x)}{\ln(b)}$$

> sqrt (x) ;
                        
$$\sqrt{x}$$

> abs (x) ;
                        
$$|x|$$

> signum (x) ;
                        
$$\text{signum}(x)$$

> x! ;                 # faktoriál
                        
$$x!$$

> x^2=x**2;           # umocňování
                        
$$x^2=x^2$$

> numer (1/2) ;       # čitatel zlomku
                        
$$1$$

> denom (1/2) ;      # jmenovatel zlomku
                        
$$2$$

> lhs (x+1=0) ;      # levá strana rovnice
                        
$$x + 1$$

> rhs (x+1=0) ;      # pravá strana rovnice
                        
$$0$$

> iquo (7,3) ;        # celočíselné dělení
                        
$$2$$

> irem (7,3) ;        # zbytek po celočíselném dělení
                        
$$1$$

> min (3,5,exp(1)) ; # minimum ze zadaných čísel
                        
$$e$$

> max (3,5,Pi) ;      # maximum ze zadaných čísel
                        
$$5$$

> gcd (15,3) ;        # největší společný dělitel
                        
$$3$$

> lcm (5,3) ;         # nejmenší společný násobek
                        
$$15$$

> round (1.5268) ;    # zaokrouhlení na celá čísla
                        
$$2$$

> trunc (1.5268) ;    # vynechání desetinné části čísla
                        
$$1$$

> frac (1.5268) ;     # vynechání celočíselné části
                        
$$0.5268$$


```

Poznámka:

- goniometrické funkce užívají radiány
- při vyhodnocování výrazu se zachovává běžná priorita vyhodnocování aritmetických operací

- zlomky v exponentech je nutné psát v závorkách
- pro určení pořadí operací se užívají závorky (...)
- násobení se zapisuje pomocí "*", tzn. $2x = 2*x$
- pro získání výsledku předchozí početní operace užíváme %

3.1 Posloupnosti, seznamy, množiny, pole

Základní datovou strukturou systému Maple jsou *posloupnosti*. Jde o skupinu výrazů (přesněji čísel, znaků) oddělených čárkami. Data v posloupnosti mají svoje speciální vlastnosti. Jednou z nich je možnost zřetězení, která se provádí pomocí operátoru "||". Ukažme si to na příkladu:

```
> S:=1,2,3,4; # posloupnost S
S := 1, 2, 3, 4
> a||S; # vytvoření posloupnosti zřetězením
a1, a2, a3, a4
```

Pro generování posloupností existují v Maplu dvě možnosti:

- použití operátoru "\$" jako opakovače

```
> a$5;
a, a, a, a, a
> $1..4;
1, 2, 3, 4
> posl:=x^3$x=1..3;
posl := 1, 8, 27
```

- použití příkazu *seq*

seq(f(i), i = dm..hm); $f(i)$ se mění od dolní do horní meze
seq(f(i), i = vyraz); posloupnost aplikací $f(i)$ na každý člen výrazu

```
> seq(i+3,i=0..3);
3, 4, 5, 6
> seq(i^3,i=2*x+y);
8x^3, y^3
```

Na jednotlivé členy posloupnosti se odkazujeme tzv. *indexovým zápisem*, tzn. použijeme následující syntaxi:

název_posloupnosti[index] , kde *index* je celé číslo (např. S[2] určuje druhý člen posloupnosti S).

```
> S[2];
2
```

Dále poznamenejme, že tento zápis se dá použít také pro zápis indexu u identifikátoru, např.

```
> p[tau];
pτ
```

Další ze struktur systému Maple nazýváme *seznam*. Na rozdíl od posloupností se prvky seznamu zapisují do hranatých závorek, tj. [...]. V této struktuře může být zapsáno několik stejných prvků a ani pořadí prvků se nijak nemění (na rozdíl od množin). To tedy znamená, že následující tři seznamy jsou různé

```
> sez:=[a,b,c]; [b,c,a]; [a,a,b,c,a];
sez := [a, b, c]
[b, c, a]
[a, a, b, c, a]
```

Pro práci se seznamy obsahuje Maple několik příkazů

nops(seznam) - vypíše počet prvků (počítá i opakující se prvky)

op(seznam) - převádí seznam na posloupnost prvků
member(prvek,seznam) - hledá "prvek" v "seznam" - je-li obsažen, výsledkem je *true*, jinak *false*

> *nops (sez) ;*

3

> *op (sez) ;*

a, b, c

> *member (c, sez) ;* # prvek byl v seznamu nalezen

true

Pro nalezení pozice prvku v seznamu se užívá následující konstrukce

> *member (c, sez, 'kde') ;*

kde ;

true

3

> *member (s, sez) ;* # prvek nebyl v seznamu nalezen

false

Stejně jako v posloupnostech se odkazujeme na jednotlivé prvky pomocí indexů.

> *sez [1] ;*

a

Dále se budeme zabývat *množinami*, které se vytvoří uzavřením jednotlivých prvků posloupnosti do složených závorek, tj. "{ }". Maple pracuje s množinami stejně jako my v matematice. To znamená, že pořadí prvků v množině si uspořádá podle určitého klíče a opakující se prvky vypouští. Vraťme se zpět ke třem příkladům, které jsme uvedli v části věnované seznamům. Zapišme je však jako množiny. Výsledkem budou tři shodné množiny.

> *{a,b,c} ; {b,c,a} ; {a,a,b,c,a} ;*

{a, b, c}

{a, b, c}

{a, b, c}

Maple samozřejmě zavádí také operace mezi množinami

- *sjednocení* - *{...} union {...}*

- *průnik* - *{...} intersect {...}*

- *množinový rozdíl* - *{...} minus {...}*

> *A:={a,b,c,d} ;*

B:={a,c,f} ;

> *A union B ;*

{f, a, b, c, d}

> *A intersect B ;*

{a, c}

> *A minus B ;*

{b, d}

System Maple umožňuje provádět s množinami podobné operace jako se seznamy

nops(množina) - výsledkem je počet prvků v množině (srovnej se seznamy)

op(množina) - převádí množinu na posloupnost

member(prvek,mnozina) - hledá daný prvek v dané množině, pokud jde o prvek této množiny, výsledkem je *true*, jinak *false*

V Maplu je možné definovat i prázdné množiny a prázdné seznamy

```
> prazdna_mnozina := {};
nops (%); # vypíše počet prvků množiny
prazdna_mnozina := { }
0
> prazdny_seznam := [];
nops (%); # vypíše počet prvků seznamu
prazdny_seznam := [ ]
0
```

Další ze struktur systému Maplu je **vektor**, který vytvoříme pomocí klíčového slova **array**.
identifikátor := array(rozsah);

Takto bude vytvořen vektor o daném rozsahu, ale jeho složky nebudou inicializovány. Je nutné poznamenat, že k vypsání vektoru musíme užít příkaz pro standardní výpis na obrazovku, např. **print**.

```
> vekt := array(1..3); # vytvoření vektoru o třech složkách
vekt := array(1..3, [ ])
> print(vekt); # výpis vektoru, složky ještě nebyly
inicializovány
[vekt1, vekt2, vekt3]
```

Pro inicializaci složek užijeme známého indexového zápisu

```
> vekt[1] := 1; # inicializace složek vektoru vekt
vekt[2] := 3;
vekt[3] := 5;
> print(vekt); # výpis vektoru
[1, 3, 5]
```

Maple umožňuje vytvořit a inicializovat vektor pomocí jednoho příkazu

```
identifikátor := array(rozsah, [složky]);
> vekt1 := array(1..3, [1, 3, 5]);
```

```
vekt1 := [1, 3, 5]
```

Další požadovanou strukturou je **matice**, kterou vytvoříme opět pomocí klíčového slova **array**.

```
identifikátor := array(řádky, sloupce, [[řádek1],[řádek2],..., [řádekN]]);
```

```
> mat := array(1..3, 1..3, [ [1, 1, 1], [2, 3, 4], [3, 5, 4] ] );
```

```
mat :=  $\begin{bmatrix} 1 & 1 & 1 \\ 2 & 3 & 4 \\ 3 & 5 & 4 \end{bmatrix}$ 
```

Systém Maple verze 8, zavádí ještě jinou možnost vytvoření vektoru či matice. Bohužel, tento způsob není kompatibilní s předchozím zápisem. Je však využíván v nových rozšiřujících knihovnách, např. *LinearAlgebra*, *VectorCalculus*.

Sloupcový vektor

```
> < 1, 2, 3 >;
```

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Řádkový vektor

```
> < 1 | 2 | 3 >;
```

$$[1, 2, 3]$$

Vytvoření matice po sloupcích

```
> < < 1, 2, 3 > | < 4, 5, 6 > >;
```

$$\begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

Rozšíření matice o sloupcový vektor

```
> < % | < x, y, z > >;
```

$$\begin{bmatrix} 1 & 4 & x \\ 2 & 5 & y \\ 3 & 6 & z \end{bmatrix}$$

Vytvoření matice po řádcích

```
> < < 1 | 2 | 3 >, < 4 | 5 | 6 > >;
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

Vytvoření matice 1 x n ze dvou řádkových vektorů

```
> < < 1 | 2 | 3 > | < 4 | 5 | 6 > >;
```

$$[1 \ 2 \ 3 \ 4 \ 5 \ 6]$$

Maple umožňuje vytvářet i vícerozměrné struktury, tzv. *pole*. Opět se zde užije klíčového slova *array*. Problémem těchto struktur je jejich názorné zobrazení. Jako příklad vytvoříme jednoduché trojrozměrné pole.

```
> pole:=array(1..2,1..2,1..2,[[[1,2],[3,4]],[[1,3],[6,7]]]);
```

```
pole := array(1 .. 2, 1 .. 2, 1 .. 2, [
```

```
(1, 1, 1) = 1
```

```
(1, 1, 2) = 2
```

```
(1, 2, 1) = 3
```

```
(1, 2, 2) = 4
```

```
(2, 1, 1) = 1
```

```
(2, 1, 2) = 3
```

```
(2, 2, 1) = 6
```

```
(2, 2, 2) = 7
```

```
]);
```

Pro úplnost ještě uvedme jednu strukturu, která je velmi podobná polím. Jde o *tabulky*. Při jejich konstrukci použijeme klíčového slova *table*.

```
identifikátor := table([pojmenování1 = hodnota, pojmenování2 = hodnota, ...]);
```

Rozdíl mezi poli a tabulkami je v tom, že na jednotlivé prvky pole se odkazujeme pomocí přirozených čísel indexů, zatím co na prvky tabulky se můžeme odkazovat jakýmkoliv řetězcem, který je pojmenováním daného prvku.

```
> preloz:=table([one=jedna,two=dva,three=tri]);
      preloz:=table([one=jedna,two=dva,three=tri])
> preloz[two];
      dva
> zeme:=table([Mz=[5.976*10^24,kg], Rz=9.378164*10^6]);
      zeme:=table([Mz=[0.5976000000 1025,kg], Rz=0.9378164000 107])
> zeme[Mz];
      [0.5976000000 1025,kg]
>
>
```

3.2 Definice funkcí

Jak již bylo řečeno úvodem k tomuto paragrafu, Maple obsahuje velké množství předdefinovaných funkcí. Vyskytnou se však situace, kdy sám uživatel potřebuje vytvořit vlastní funkci. Samozřejmě, že tato možnost existuje.

Základní syntaxe pro definici funkce je následující:

jméno_fce := proměnná -> předpis;

jméno_fce := (prom1,...,promN) -> předpis;

První řádek je určen pro definici funkce jedné proměnné a druhý pro N proměnných. Tento způsob definice se s výhodou používá pro funkce s jednodušším předpisem. Vysvětlení pojmu vyplyne z celého tohoto odstavce.

Několik příkladů využití tohoto způsobu definice funkce

> ***f:=x->x^2+2*x+1;*** # definování funkce f jedné
neznámé x

$$f := x \rightarrow x^2 + 2x + 1$$

> ***f1:=(x,y,z)->x^2+y^2+z^2;*** # definování funkce se třemi
neznámými x, y, z

$$f1 := (x, y, z) \rightarrow x^2 + y^2 + z^2$$

Hodnotu funkce pak vypočítáme stejně jako u předdefinovaných funkcí příkazem

fce(bod);

> ***f(0);f(1);f(w);*** # hodnota funkce v bode 0, 1 a w

1

4

$$w^2 + 2w + 1$$

> ***f1(0,0,0);f1(1,1,1);f1(a,b,c);*** # hodnota fce v bode
(0,0,0), (1,1,1) a (a,b,c)

0

3

$$a^2 + b^2 + c^2$$

Existují však funkce, které bychom tímto způsobem jen těžko definovali. Jde o funkce, které jsou definovány po částech. Ukažme si to na jednoduchém příkladu. Mějme funkci definovanou po částech:

$$f(x) = \begin{cases} x & \text{pro } 2 \leq x < 10 \\ -x & \text{pro } 0 < x < 2 \\ 0 & \text{jinde} \end{cases}$$

Tato funkce nemůže být zapsána pomocí předchozích předpisů. Musíme tedy využít jiného zápisu, tedy pomocí klíčového slova *piecewise*.

jméno_fce := (prom_1,...,prom_n)

->piecewise(nerovnost_1,hodnota_1,...,nerovnost_n,hodnota_n);

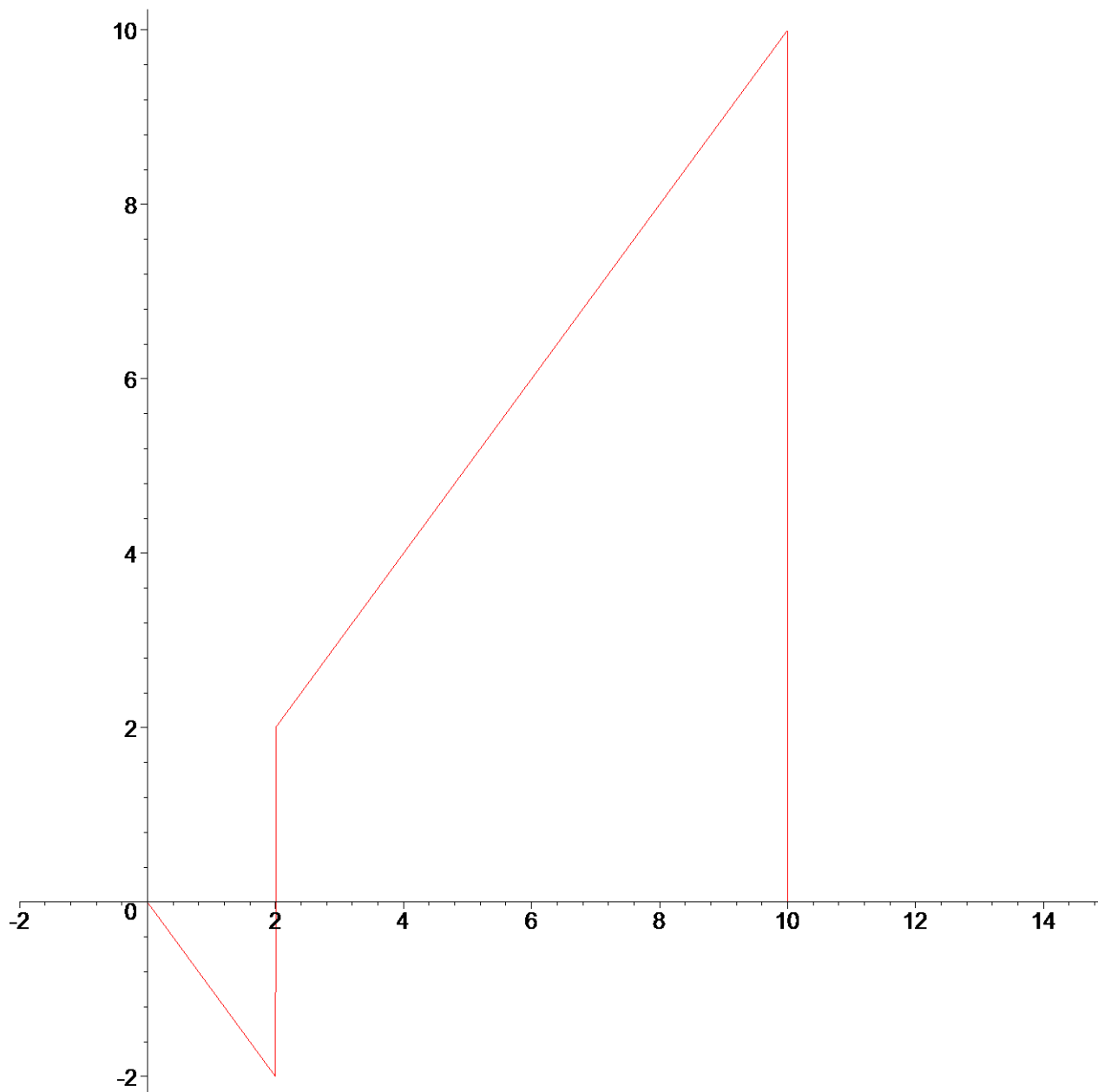
Zapišme výše uvedenou funkci danou po částech

```
> flp:=x->piecewise(2<=x and x<10,x,0<x and x<2,-x,0);
```

```
flp := x → piecewise(2 ≤ x and x < 10, x, 0 < x and x < 2, -x, 0)
```

Všimněte si, že poslednímu parametru (tedy 0) není určen interval, to tedy znamená, že zastupuje hodnotu funkce v nedefinovaných intervalech.

```
> plot(flp,-2..15); # vykreslení funkce - více v kapitole  
Základy grafického zpracování
```



Za nejobecnější definici funkce (používá se při programování v Maplu) však bývá často považována definice pomocí klíčového slova *proc*.

```
jmeno_fce := proc(prom_1,...,prom_n) tělo funkce end proc;
```

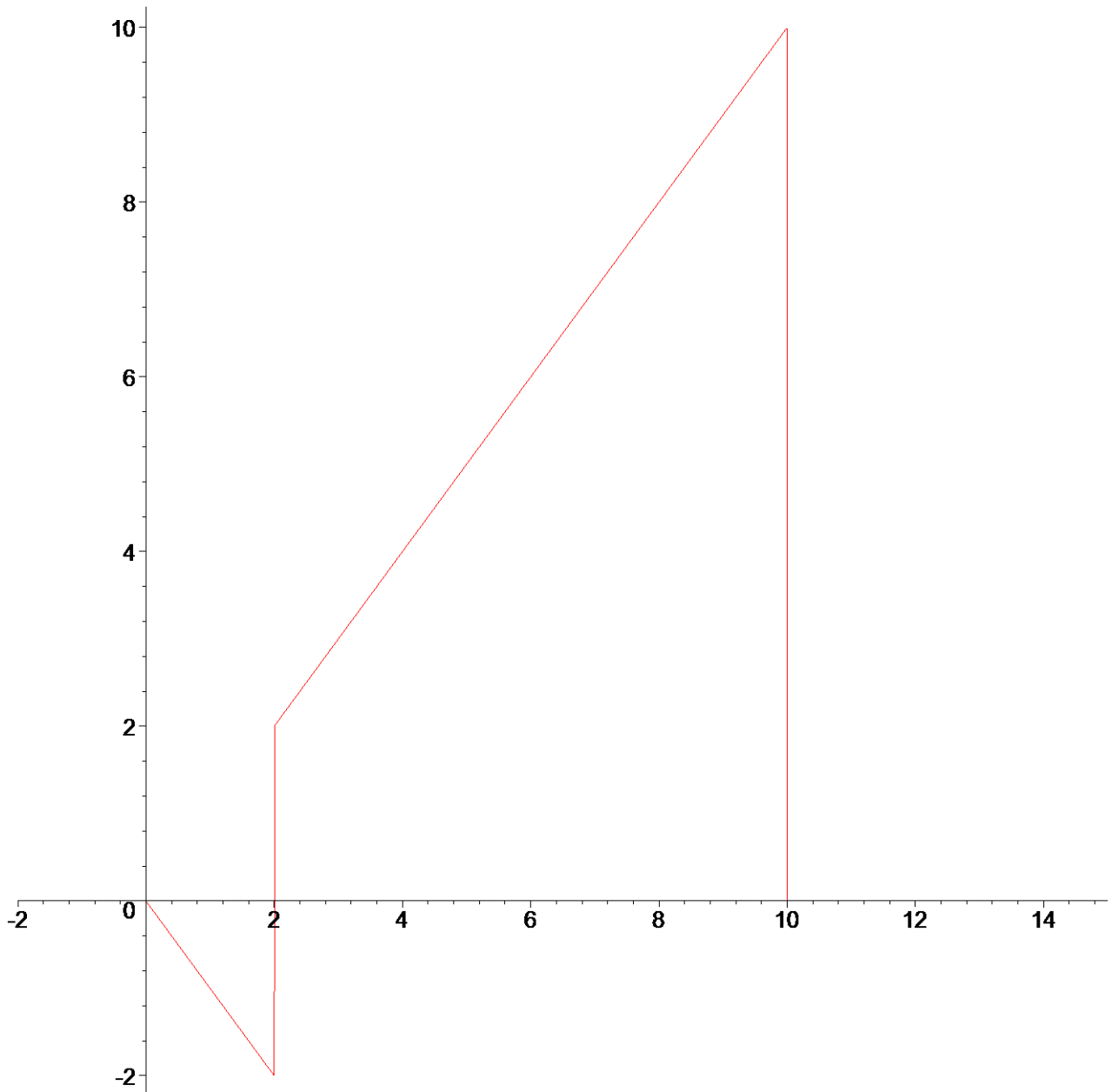
V tomto případě lze v definici funkce využít programovacího jazyka zabudovaného v Maplu.

```
> flp_:=proc(x) # více o programování v Maple  
viz. Příloha 3  
if 2<=x and x<10 then x;
```

```

elif 0<x and x<2 then -x;
else 0
end if;
end proc;
flp_ := proc(x) if 2 ≤ x and x < 10 then x elif 0 < x and x < 2 then -x else 0 end if end proc
> plot(flp_,-2..15); # vykreslení funkce

```



Další možná definice funkce vychází z definovaného výrazu. Využívá příkazu **unapply(výraz,proměnná);**

Ukažme si to na příkladu:

```
> a := x^2+3*x+4;
```

$$a := x^2 + 3x + 4$$

```
> f := unapply(a, x);
```

$$f := x \rightarrow x^2 + 3x + 4$$

S definicí funkce je spojena i definice funkce inverzní. Systém Maple je v tomto směru "nedokonalý", protože pro uživatele nemá "slušný" příkaz pro její výpočet. Obsahuje pouze příkaz

invfunc[f] popř. jiný zápis stejného *f@@(-1)*. Bohužel, tento příkaz není vhodný pro jiné než předdefinované funkce Maplu. Proto je nutné vypočítávat inverzní funkce jinak. Například s využitím příkazu *solve* a definice $f(\text{invf}(x)) = x$. V Maplu tedy užitíme konstrukce *invfce := prom -> solve(f(invfce) = prom, invfce);*

```
> invfunc[sin];          # výsledek hledá v tabulce inverzních funkcí
```

arcsin

```
> invfunc[f];          # neumí nalézt
```

invfunc_f

```
> fce:=x->x^2-3;
```

fce := x → x² - 3

```
> solve(fce(invfce)=x, invfce);
```

```
          # více o příkazu solve v kapitole Úvod do  
řešení algebraických rovnic
```

$\sqrt{3+x}, -\sqrt{3+x}$

```
>
```

```
>
```


3.3 Zjednodušování výrazů

System Maple je zejména určen pro symbolické výpočty, a proto některé výrazy automaticky zjednodušuje

- součet, součin, mocniny čísel
- krácení ve zlomku (i v případě polynomů), zlomky však nerozšiřuje
- základní funkce, např. vyčíslí funkční hodnotu

Jsou však situace, kdy by bylo vhodné daný výraz zjednodušit a Maple to automaticky neudělá. Proto jsou k dispozici příkazy, pomocí kterých lze většinu zjednodušení provést. Nejprve uvedeme jejich ucelený, ale neúplný, přehled a teprve pak se podrobněji zaměříme na příkaz *simplify* a *convert*. Poznámky ke zbylým příkazům naleznete na doprovodném CD.

- *simplify* - základní zjednodušení
- *factor* - rozklad polynomu na součin kořenových činitelů
- *expand* - rozvíjení výrazů (opačný příkaz k *factor*)
- *convert* - převádí výraz na určitý tvar
- *normal* - převádí výraz na jednoduché zlomky
- *combine* - seskupuje výrazy
- *collect* - vytýkání ve výrazech
- *sort* - seřazení

Nyní se vraťme k nejčastěji užívanému příkazu pro zjednodušování. Jde o příkazy *simplify* s možnými syntaxemi

simplify(výraz)

simplify(výraz, pravidlo)

simplify(výraz, *assume* = vlastnosti)

simplify(výraz, *symbolic*)

Prvním parametrem je výraz, který chceme zjednodušit. Druhým parametrem je pravidlo, či soubor pravidel, které se na daný výraz aplikuje. Uveďme přehled předdefinovaných pravidel a jejich použití

- *power* - mocniny, exponenciály, logaritmy
- *radical* - výrazy se zlomkem v exponentu
- *sqrt* - druhé mocniny nebo mocniny druhých odmocnin (v Maple8 nemá velké opodstatnění)
- *trig* - mocniny trigonometrických funkcí

```
> (x^a)^b+4^(1/2);  
simplify((x^a)^b+4^(1/2),power); # stejný výsledek i bez  
vedení druhého parametru
```

$$(x^{(x^2+3x+4)})^b + \sqrt{4}$$

$$(x^{(x^2+3x+4)})^b + 2$$

- > **simplify(sin(x)^2+cos(x)^2, trig);** # parametr trig nemusí být uveden
- 1
- > **simplify(16^(3/2), sqrt);** # parametr sqrt nemusí být uveden
- 64
- > **g:=sqrt(x^2);**
- $g := \sqrt{x^2}$
- > **simplify(g);** # implicitní zjednodušení
- $\text{csgn}(x) x$
- > **simplify(g, assume=real);** # zjednodušení na množině reálných čísel
- $|x|$
- > **simplify(g, assume=positive);** # zjednodušení pro nezáporná x
- x
- > **simplify(g, symbolic);** # symbolické zjednodušení
- x

Největší výhodou tohoto příkazu je možnost zadání vlastních pravidel pro zjednodušení. Uživatel tedy má přímý vliv na budoucí výsledek.

simplify(výraz, { vlastní pravidla });

- > **m_pravidlo:={x^2=1, y^2=1};** # vlastní pravidlo
- $m_pravidlo := \{x^2 = 1, y^2 = 1\}$
- > **v1:=ln(x^2)+y^2+1;** # upravovaný výraz
- $v1 := \ln(x^2) + y^2 + 1$
- > **simplify(v1, m_pravidlo);** # použití vlastního pravidla
- 2
- > **simplify(v1, symbolic);** # symbolické zjednodušení
- $2 \ln(x) + y^2 + 1$

Systém Maple dále nabízí příkaz, pomocí kterého lze převést daný výraz na požadovaný tvar. Jde o příkaz **convert**.

convert(výraz, forma, argument);

Možné hodnoty druhého parametru a jeho vhodného použití uvádí následující (neúplná) tabulka	
<i>polynom</i>	řady a polynomy
<i>exp, expln, expsincos</i>	trigonometrické výrazy do tvaru exponenciálního
<i>parfrac</i>	zlomek na parciální zlomky v dané proměnné
<i>rational</i>	desetinné číslo na zlomek
<i>radians, degrees</i>	stupně, radiány

3.4 Úvod do řešení algebraických rovnic

Základem řešení rovnice a jejich systémů je příkaz *solve*. Pro řešení rovnice o jedné neznámé použijeme předpisu

```
solve(rovnice )
```

V tomto případě není neznámá specifikována, a proto nelze tohoto volání použít při řešení rovnice závislé na parametru. Řešením tohoto problému je specifikace neznámé následujícím voláním

```
solve(rovnice,neznámá )
```

Rozdíl ukáže následující příklad

```
> r0:=x+t-3;
```

```
r0 := x + t - 3
```

```
> solve(r0=0); # nepřesné určení proměnné a parametru, tj.  
nesprávný výsledek
```

```
{x = x, t = -x + 3}
```

```
> solve(r0); # ekvivalentní zápis předchozího, implicitní  
rovnost rce=0
```

```
{x = x, t = -x + 3}
```

```
> solve(r0=0,{x}); # nutné určení neznámé, zbylé neznámé jsou  
parametrem
```

```
{x = -t + 3}
```

Pro řešení soustavy rovnic je určena třetí syntaxe příkazu *solve*, a to

```
solve({rovnice},{neznámé});
```

```
> r:=solve({x-2*y=3,x+y=1},{x,y}); # řešení soustavy rovnic
```

```
r := {y = -2/3, x = 5/3}
```

```
> r[1];r[2]; # výpis řešení
```

```
y = -2/3
```

```
x = 5/3
```

Pro přiřazení kořenů do proměnných uijeme příkaz *assign*

```
> assign(r); # řešení soustavy r přiřadíme do  
proměnných x,y
```

```
> x;y; # kontrolní výpis
```

```
5  
3
```

```
-2  
3
```

Tento příkaz se dá s výhodou využít při vícenásobné inicializaci proměnných.

```
> assign('a','b','c') = (3,2,1); # vícenásobná inicializace
```

```
> [a,b,c]; # kontrolní výpis
```

[3, 2, 1]

Dále je nutné umět tuto inicializaci zrušit. Obecně lze použít příkaz

```
> # restart:
```

, který však není vhodný, protože vymaže všechny dosavadní výsledky. Existuje však několik jiných možností.

Ukažme si to na výše inicializovaných proměnných. Elegantním způsobem zrušení je následující konstrukce

```
identifikátor := 'identifikátor';
```

```
> x:='x'; # zrušení inicializace proměnné x
```

```
x:=x
```

```
> x;y;y:='y': # kontrolní výpis a zrušení
```

```
inicializace y
```

```
x
```

```
-2
```

```
3
```

Pro vícenásobné zrušení hodnot v proměnných se dá užít příkazu

```
unassign('prom', 'prom', 'prom', ...);
```

```
> [a,b,c];
```

```
[3, 2, 1]
```

```
> unassign('a', 'b', 'c'); # odstranění inicializace
```

```
> [a,b,c]; # kontrolní výpis
```

```
[a, b, c]
```

Nyní se vraťme k řešení rovnic, popř. jejich systémů. Po použití příkazu *solve* se doporučuje provést zkoušku. Následující příklad ukáže nutnost tohoto kroku.

Nalezneme tedy řešení rovnice

```
> rce:=(x-1)^3/(x^2-1)=0;
```

$$rce := \frac{(x-1)^3}{x^2-1} = 0$$

Řešení budeme hledat užitím příkazu *solve* v následujícím tvaru

```
> solve(rce, {x});
```

```
{x = 1}, {x = 1}
```

Systém Maple našel řešení, ale tato rovnice řešení nemá. Důvodem je odstranitelná singularita zlomku v bodě $x=1$. Ukažme, že jde o odstranitelnou singularitu a tedy právo dodefinovat hodnotu v tomto bodě.

Spočtěme tedy limitu

```
> Limit(lhs(rce), x=1)=limit(lhs(rce), x=1);
```

```
# příkaz lhs byl užit pro získání levé strany rovnice
```

```
rce
```

$$\lim_{x \rightarrow 1} \frac{(x-1)^3}{x^2-1} = 0$$

Vidíme, že systém Maple odstranil singularitu v bodě $x=1$ ještě před řešením dané rovnice. A to je důvodem k provádění zkoušky po použití příkazu *solve*.

Na dalším příkladu si ukážeme, jak efektivně provést dosazení výsledků do původní rovnice. Pro

tuto operaci lze použít příkaz *subs*. Příkaz *subs(co, kam)* provádí obecně substituce v Maple.

Vraťme se ke kontrole výsledků pomocí příkazu *subs*. Řešme soustavu rovnic

```
> r:={x-y=3,x^2-16*y=0}; # soustava rovnic
      r := {x - y = 3, x2 - 16 y = 0}
> res:=solve(r,{x,y}); # řešení soustavy rovnic
      res := {x = 4, y = 1}, {x = 12, y = 9}
```

Nyní provedeme zkoušku dosazením výsledků do soustavy rovnic

```
> subs(res[1],r); # dosazení výsledků do soustavy
      {3 = 3, 0 = 0}
> subs(res[2],r);
      {3 = 3, 0 = 0}
```

Bylo nutné užít indexového zápisu, protože řešením dané soustavy rovnic v Maplu jsou dvě množiny, které z pochopitelných důvodů musíme dosazovat postupně, tj. přistupujeme k nim pomocí již zmíněného indexového zápisu.

Dále je někdy nutné pojmenovat daný výsledek nějakým identifikátorem. Vzhledem k tomu, že pořadí prvků v množině interně určuje Maple, je nutné umět vypsát výslednou hodnotu požadované proměnné. K tomu opět použije příkaz *subs*.

```
> x1:=subs(res[1],x); # pojmenování výsledné hodnoty x z
prvního řešení
      x1 := 4
> y2:=subs(res[2],y); # pojmenování výsledné hodnoty y z
druhého řešení
      y2 := 9
```

Další možností pro určení přesného pořadí neznámých ve výsledku je změna výsledné struktury výsledku, tzn. výsledek nebude zapsán jako množina.

```
> xy:=subs(res[1],[x,y]); # zápis výsledné hodnoty x i y z
prvního řešení do seznamu
      xy := [4, 1]
```

Při řešení by bylo vhodné vyloučit určité, většinou triviální řešení, tzn. bude splněna nějaká dodatečná podmínka. Řešme tedy následující soustavu rovnic

```
> r1:=x+2*y+z+t; # zadání soustavy rovnic -
dvojtečka pro zrušení výpisu
> r2:=3*x-y-4*z-t;
> r3:=3*x+2*y+t;
> r4:=6*x^2-3*z-t;
> solve({r1,r2,r3,r4});
      {x = 0, y = 0, z = 0, t = 0}, {y = -1/3, z = -1/3, t = 7/6, x = -1/6}
> solve({r1,r2,r3,r4,x<>0}); # vyloučení určitého řešení
      {y = -1/3, z = -1/3, t = 7/6, x = -1/6}
```

Nyní se podívejme na řešení soustavy rovnic s parametrem. Z předchozího příkladu vezměme první tři rovnice a hledejme výsledné řešení.

```
> solve({r1, r2, r3});
```

$$\{x = x, t = -7x, z = 2x, y = 2x\}$$

Vidíme, že Maple za parametr zvolil proměnnou x . Určení parametru docílíme volbou nezávislých proměnných následujícím způsobem

```
> solve({r1, r2, r3}, {x, y, z});
```

$$\left\{x = -\frac{t}{7}, z = -\frac{2t}{7}, y = -\frac{2t}{7}\right\}$$

Pro získání výsledku při dané hodnotě parametru uijeme opět příkaz

```
> subs({t=7}, %);
```

$$\{x = -1, z = -2, y = -2\}$$

Příkaz *solve* má samozřejmě určitá omezení. Máme tím na mysli např. neuvádění periodicity u goniometrických funkcí

```
> solve(sin(x), x);
```

0

Pro výpis řešení bez jakýchkoliv omezení je nutné zadat následující příkaz

```
> _EnvAllSolutions := true;
```

```
> solve(sin(x), x);
```

π $_Z1\sim$

```
> _EnvAllSolutions := false: # návrat k původnímu  
nastavení
```

System Maple také nabízí možnost numerického řešení pomocí příkazu *fsolve*. Použití a volání je analogické s příkazem *solve*.

```
> solve({x^4-6*x^2-3*x-5}, {x}); # řešení symbolicky
```

```
{x = RootOf(_Z^4 - 6 _Z^2 - 3 _Z - 5, index = 1)},
```

```
{x = RootOf(_Z^4 - 6 _Z^2 - 3 _Z - 5, index = 2)},
```

```
{x = RootOf(_Z^4 - 6 _Z^2 - 3 _Z - 5, index = 3)},
```

```
{x = RootOf(_Z^4 - 6 _Z^2 - 3 _Z - 5, index = 4)}
```

```
> fsolve({x^4-6*x^2-3*x-5}, {x});
```

$$\{x = -2.371502141\}, \{x = 2.779641553\}$$

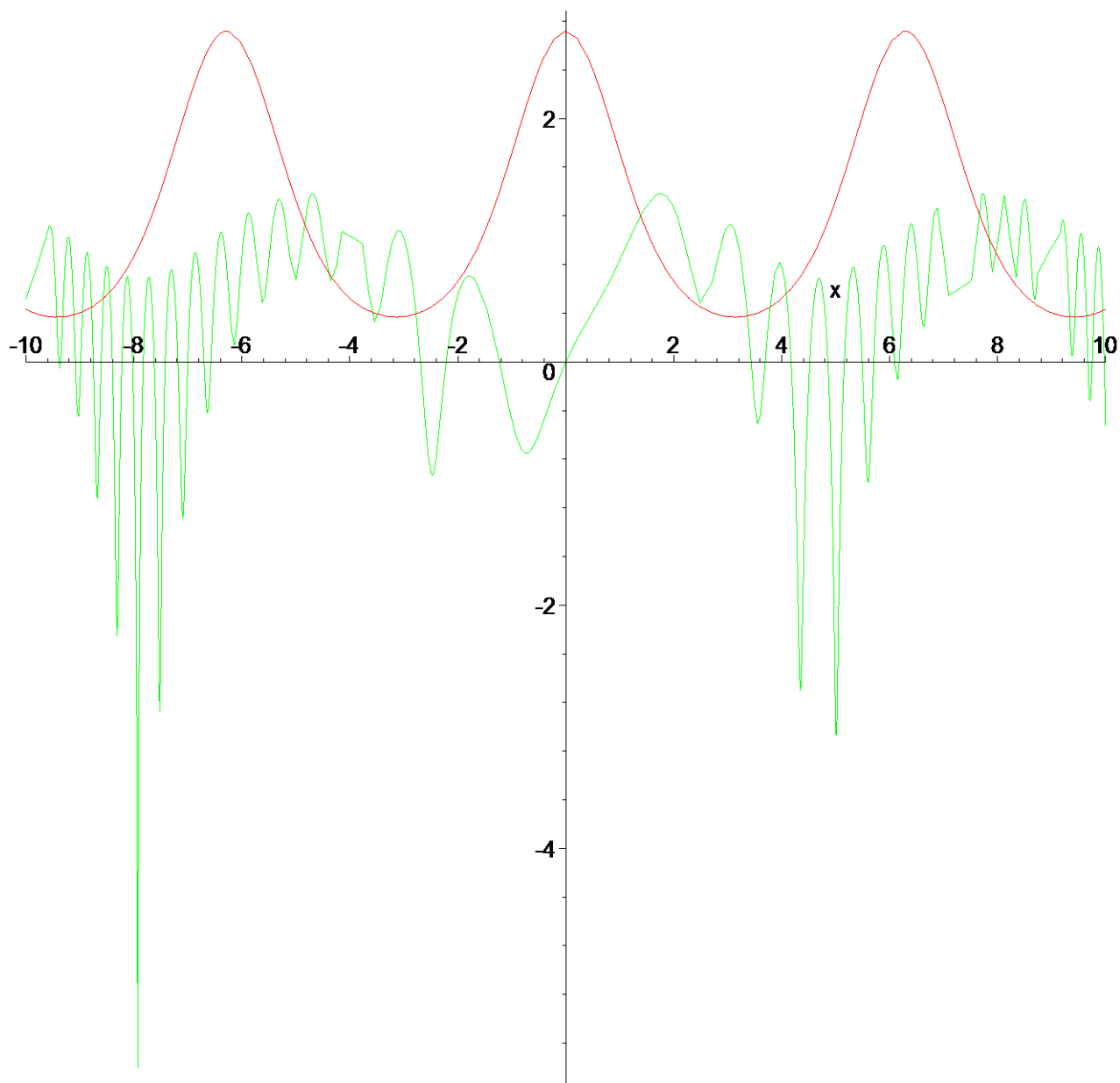
V případě, že Maple neumí nalézt řešení symbolicky, nevypíše žádný výsledek. To samozřejmě neznamená, že řešení neexistuje. Ukažme si to na příkladu.

```
> solve({exp(cos(x))=ln(2+sin(x))-cos(x^2)}), {x}); # symbolické  
řešení neumí nalézt
```

```
> p1:=plot(exp(cos(x)), x=-10..10): # vykreslení  
grafů funkcí
```

```
p2:=plot(ln(2+sin(x))-cos(x^2)), x=-10..10, color=green):
```

```
> plots[display]({p1, p2});
```



```
> fsolve({exp(cos(x))=ln(2+sin(x)-cos(x^2))},{x}); #
numerické řešení
```

```
{x = 4.085543004}
```

V případě, že chceme, aby Maple vypsal numerické řešení v zadaném intervalu, můžeme tento interval určit pomocí třetího parametru

```
> fsolve({exp(cos(x))=ln(2+sin(x)-cos(x^2))},{x},0..3); #
numerické řešení na <0,3>
```

```
{x = 1.385961775}
```

V případě, že numerické řešení na daném intervalu nelze nalézt, je vypsaný zadaný příkaz

```
> fsolve({exp(cos(x))=ln(2+sin(x)-cos(x^2))},{x},-1..1); #
numerické řešení na <-1,1>
```

```
fsolve({ecos(x) = ln(2 + sin(x) - cos(x2))}, {x}, -1 .. 1)
```

```
>
>
```


3.5 Základy grafického zpracování

Tato kapitola si neklade za cíl být referenční částí k zobrazování grafiky v Maplu, ale má za úkol nastínit danou problematiku. Vykreslování grafů bude často součástí jednotlivých kapitol, a proto je tato část probrána samostatně.

Pro zobrazení grafů lze použít příkaz *plot*, který je nahrán již při startu programu. Jde o nejjednodušší příkaz, ale pro vykreslování složitější grafiky je v Maplu speciální knihovna (tzv. package) *plots*.

Grafy ve dvou dimenzích

Jak již bylo řečeno, nejjednodušším příkazem je příkaz *plot*.

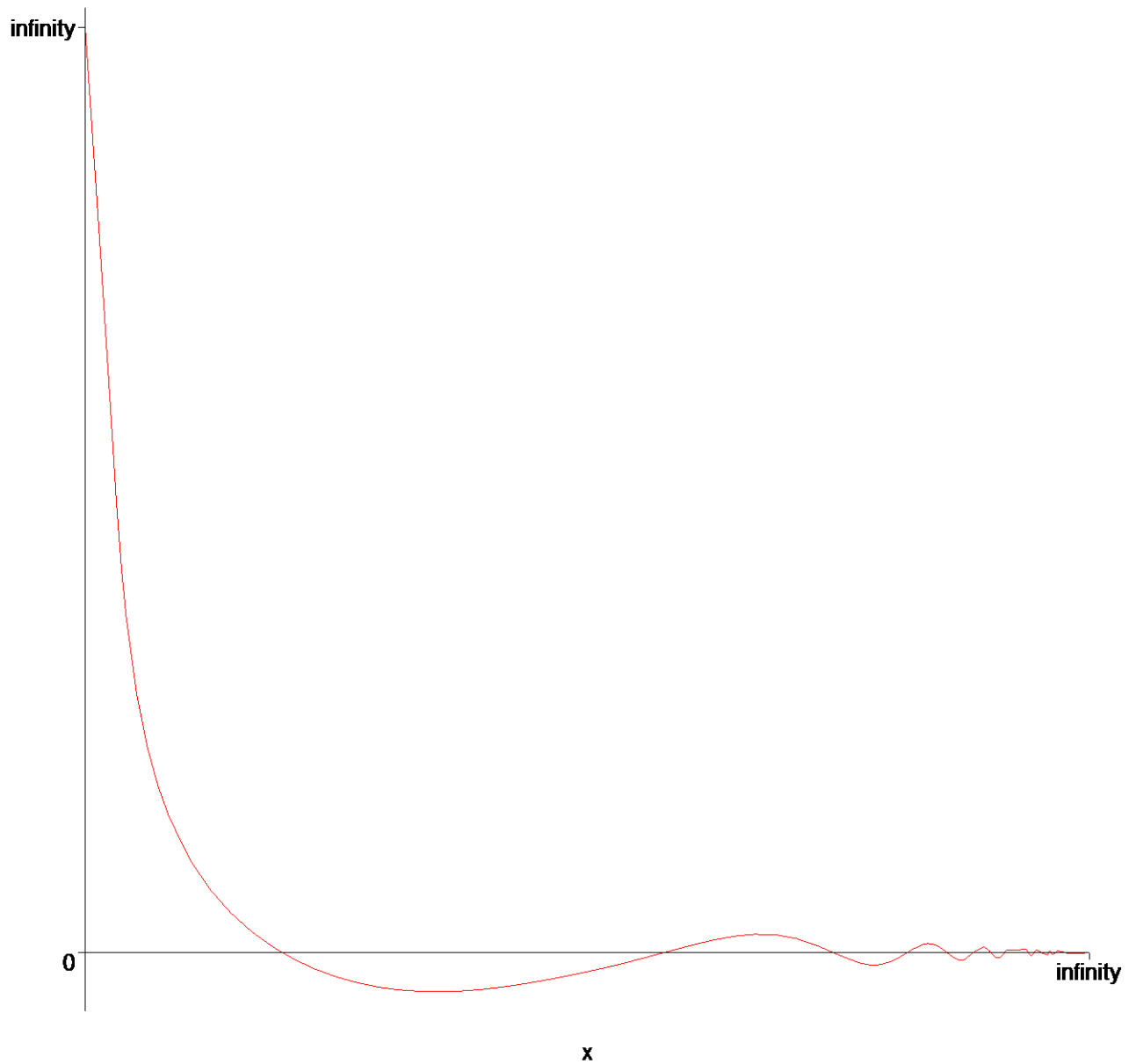
plot(výraz, proměnná = a..b);

plot(funkce, a..b);

plot(funkce, a..b,c..d);

První příkaz vykreslí daný výraz (jde o předpis funkce) na daném intervalu. Druhý se používá pro vykreslení funkce $f(x)$ na daném intervalu a třetí vykresluje funkce na kartézském čtverci $\langle a, b \rangle \times \langle c, d \rangle$.

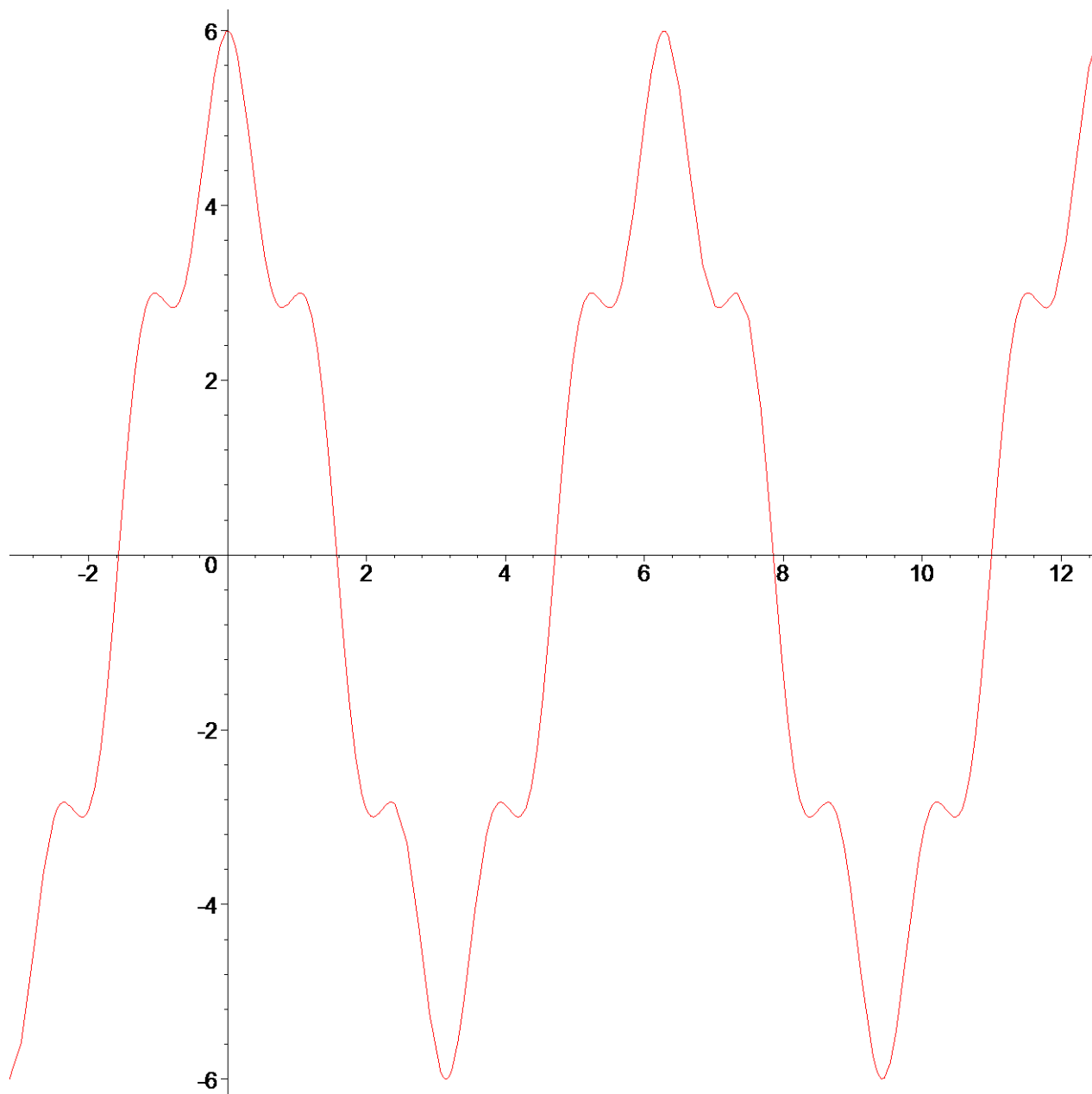
```
> plot(cos(x)/x, x=0..infinity);      # vykreslení grafu funkce dané  
výrazem
```



```

> f:=x->5*cos(x)+cos(5*x) :
> plot(f,-Pi..4*Pi);           # vykreslení grafu funkce
  plot(f(x),x=-Pi..4*Pi):     # ekvivalentní zápis pomocí
  výrazu

```

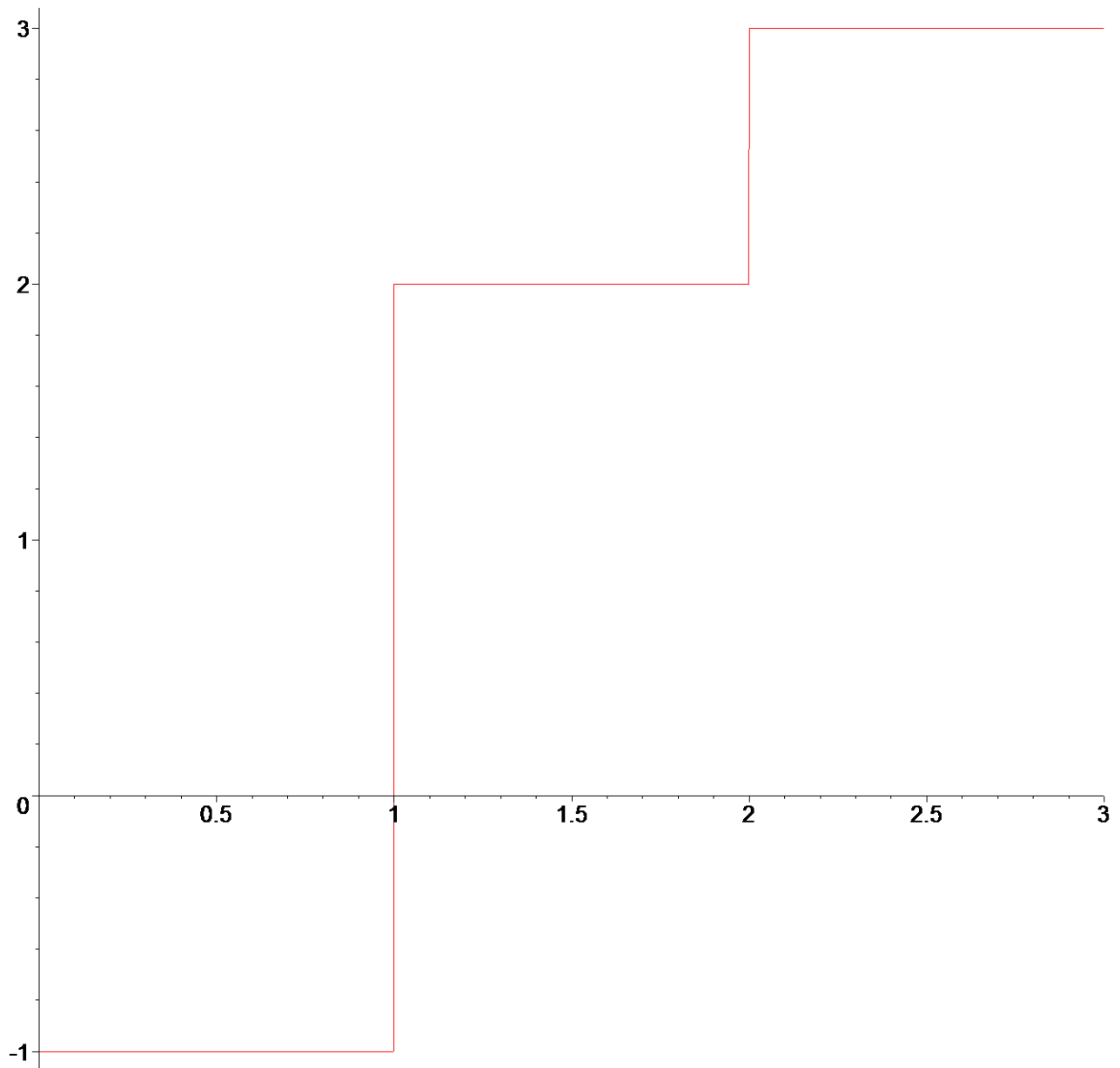


System Maple kreslí grafy funkcí jako spojitou křivku, což v konečném důsledku může znehodnotit graf funkce dané po částech. Vysvětlení ukážeme na následujícím příkladu.

```
> f1:=x->piecewise(x<1,-1,x<2,2,3); # funkce daná po částech
```

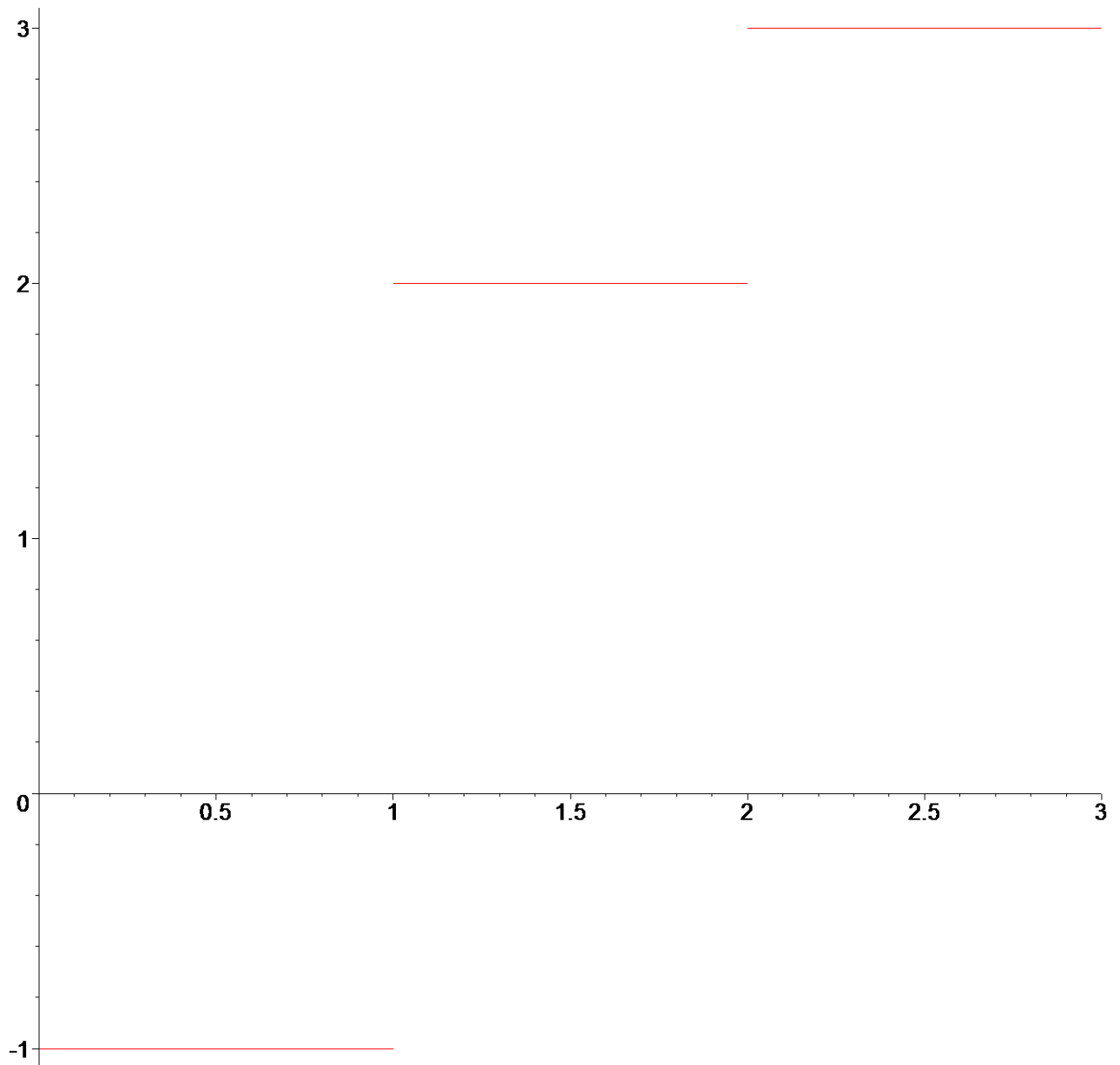
```
fl := x → piecewise(x < 1, -1, x < 2, 2, 3)
```

```
> plot(f1,0..3); # vykreslení funkce
```



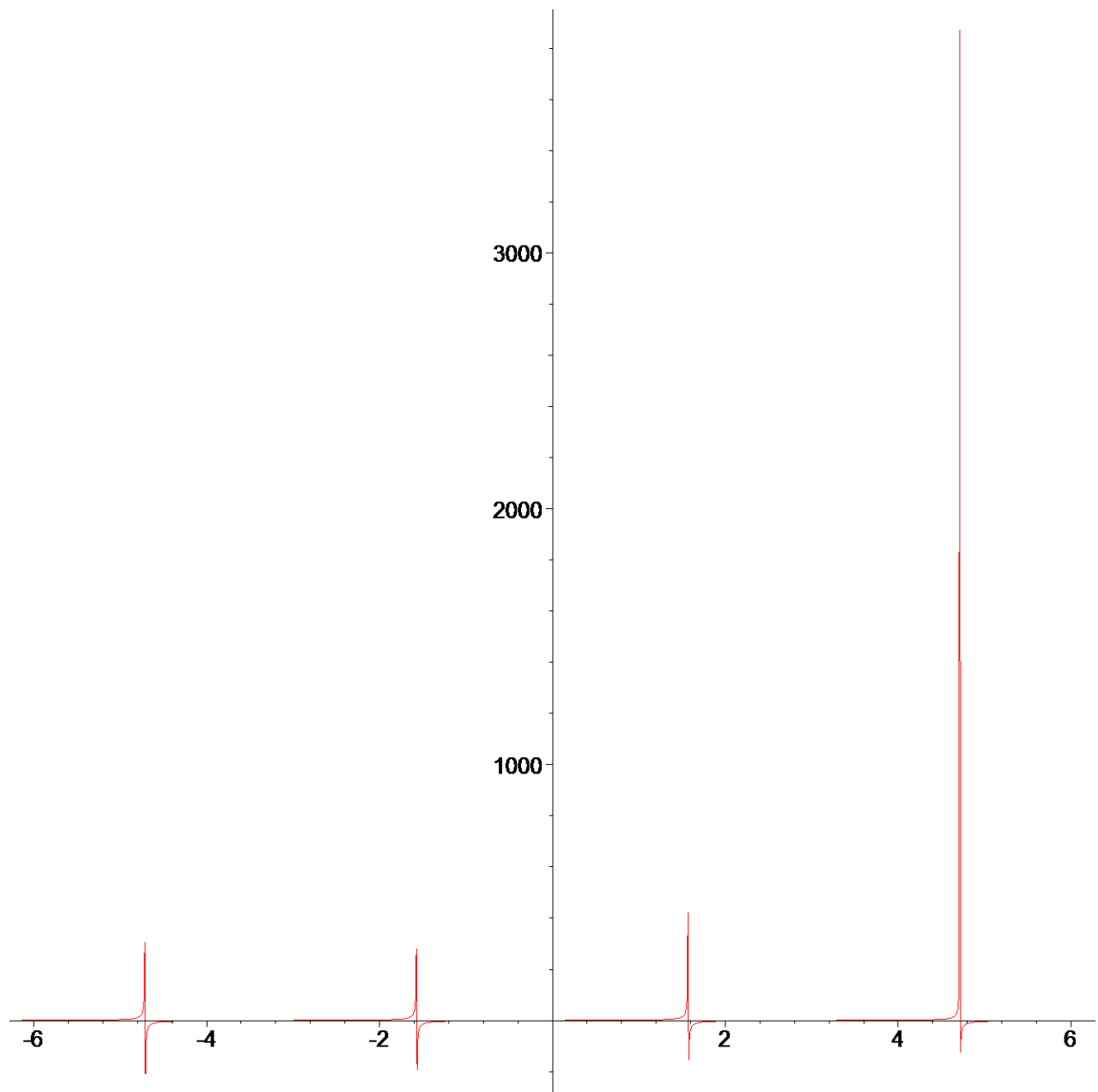
Předchozí obrázek nelze považovat za graf funkce kvůli svislým úsečkám. Je nutné tyto úsečky odstranit, tzn. povolit nespojitosti. Toho docílíme přidáním třetího parametru do příkazu *plot*, a to *discont=true/false*. Hodnota *false* je implicitní.

```
> plot(f1,0..3,discont=true); # povolení nespojitostí
```



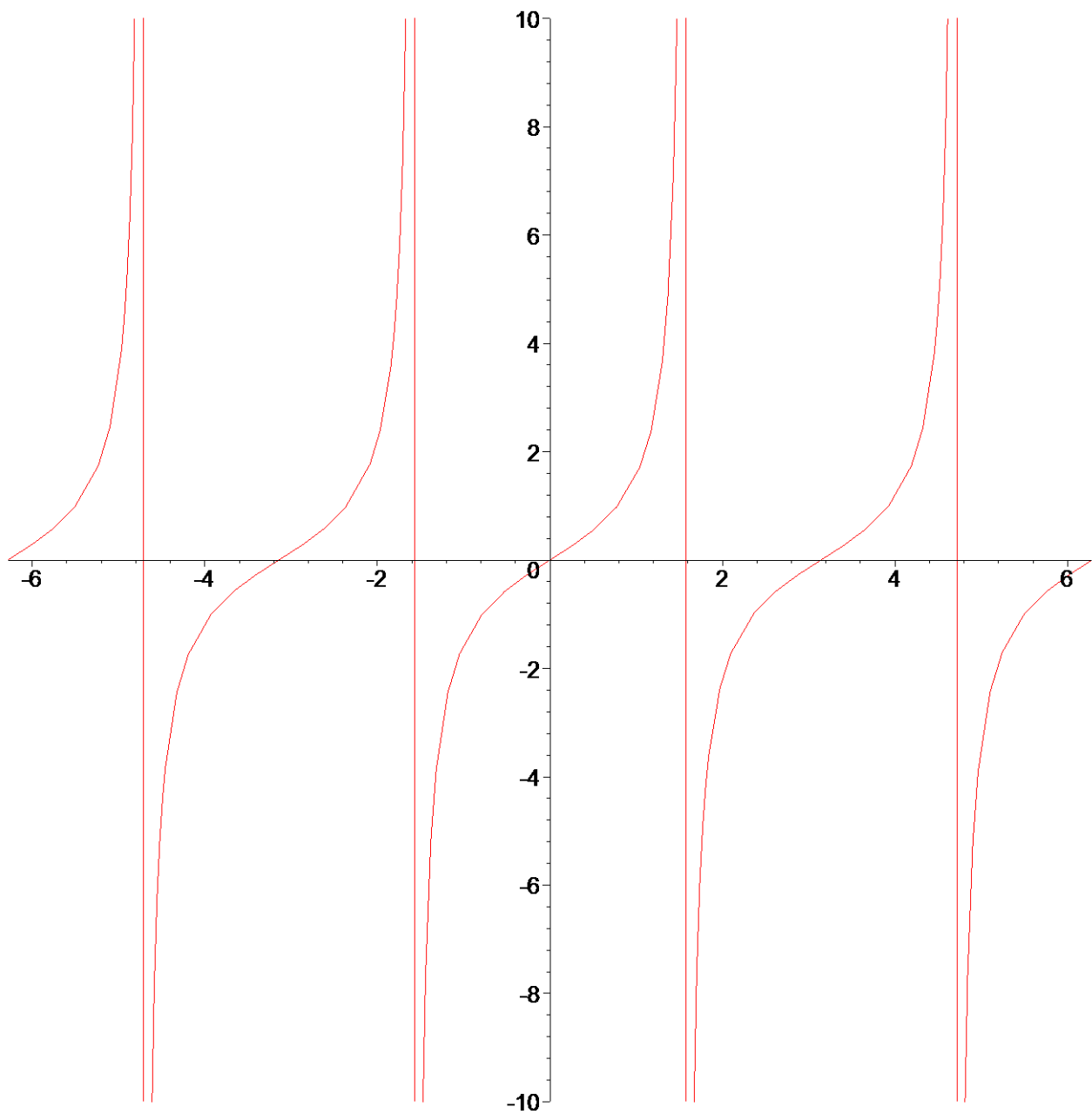
Další příklad shrne dosavadní poznatky o zobrazování funkcí. Nakresleme funkci tangens.

```
> plot(tan,-2*Pi..2*Pi); # funkce tangens
```



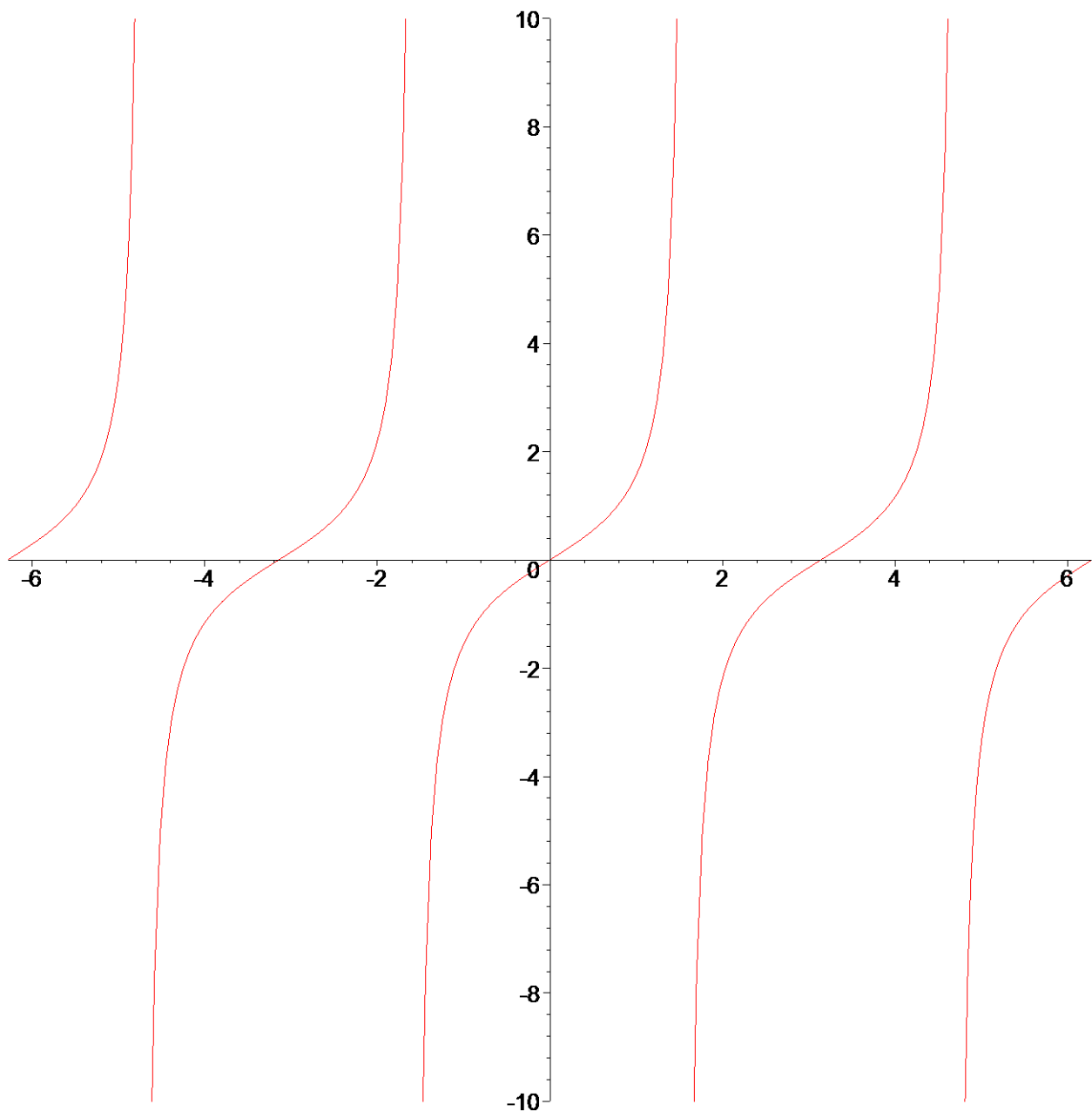
Předchozí obrázek nepřipomíná graf funkce tangens, ale opravdu je jejím grafem. Problémem je, že měřítko na ose y je velmi malé. K odstranění problému využijeme třetí možnou syntaxi příkazu *plot*.

```
> plot(tan,-2*Pi..2*Pi,-10..10);      # omezení hodnot na ose y
```



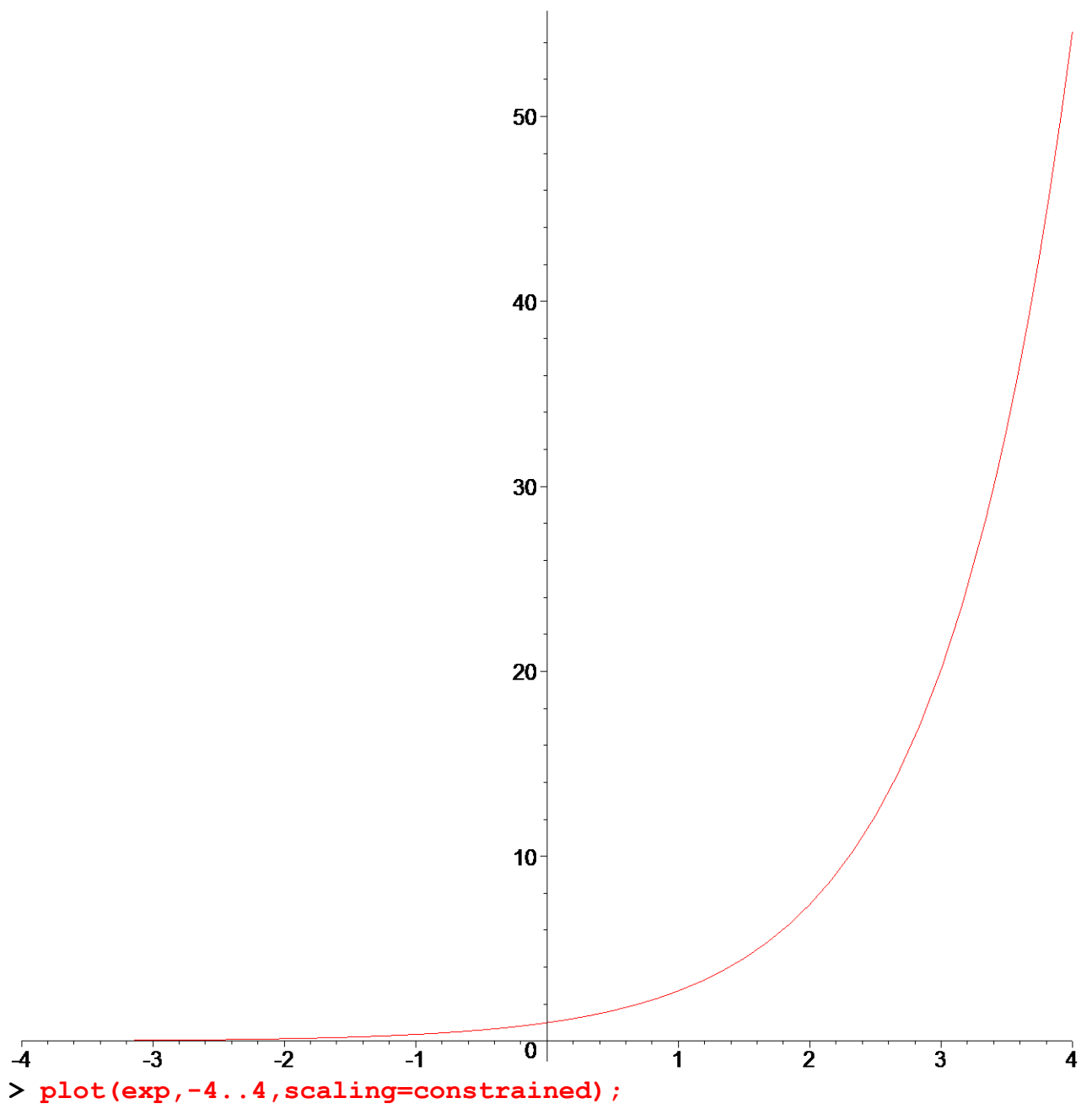
Při použití parametru *discont* dostaneme.

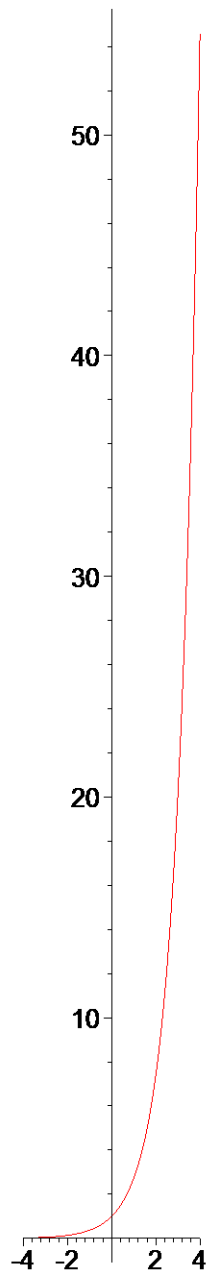
```
> plot(tan,-2*Pi..2*Pi,-10..10, discont=true);
```



System Maple implicitně nevykresluje souřadnicové osy ve stejném měřítku, a tím někdy dochází ke zkreslení informace o průběhu funkce. Pro vykreslení stejného měřítka na obou osách je určen parametr *scaling = constrained / unconstrained*. Hodnota *unconstrained* je implicitní. Následující dva obrázky zdůrazní důležitost tohoto parametru.

```
> plot(exp, -4..4);
```

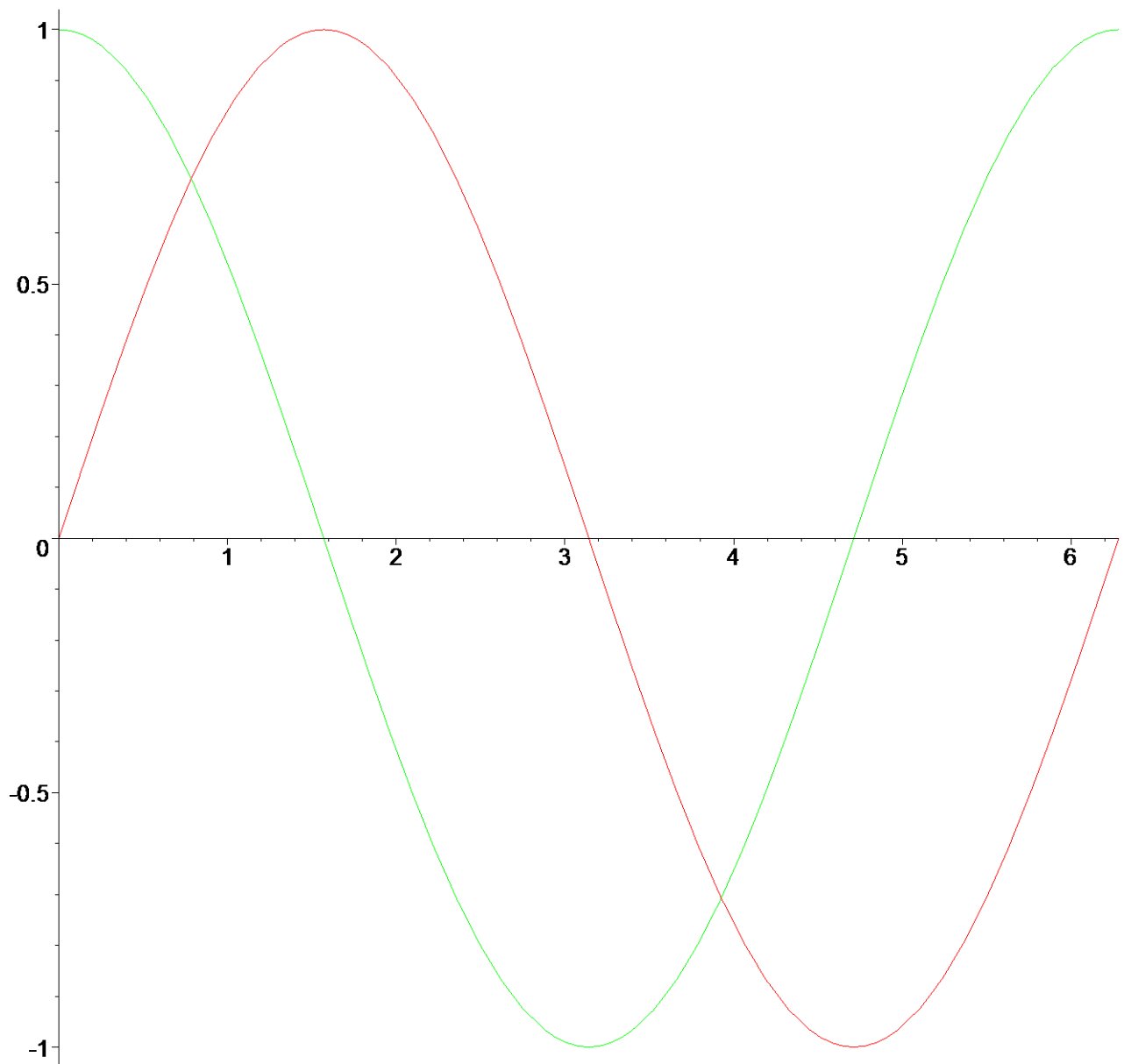





Maple samozřejmě umožňuje vykreslit více grafů do jednoho obrázku, a to pomocí konstrukce ***plot([seznam funkcí], proměnná = rozsah);***

Zde je nutné zdůraznit, že jde o seznam. Je tedy nutné použít [.]

```
> plot([sin,cos],0..2*Pi); # vykreslení funkcí sin, cos
```

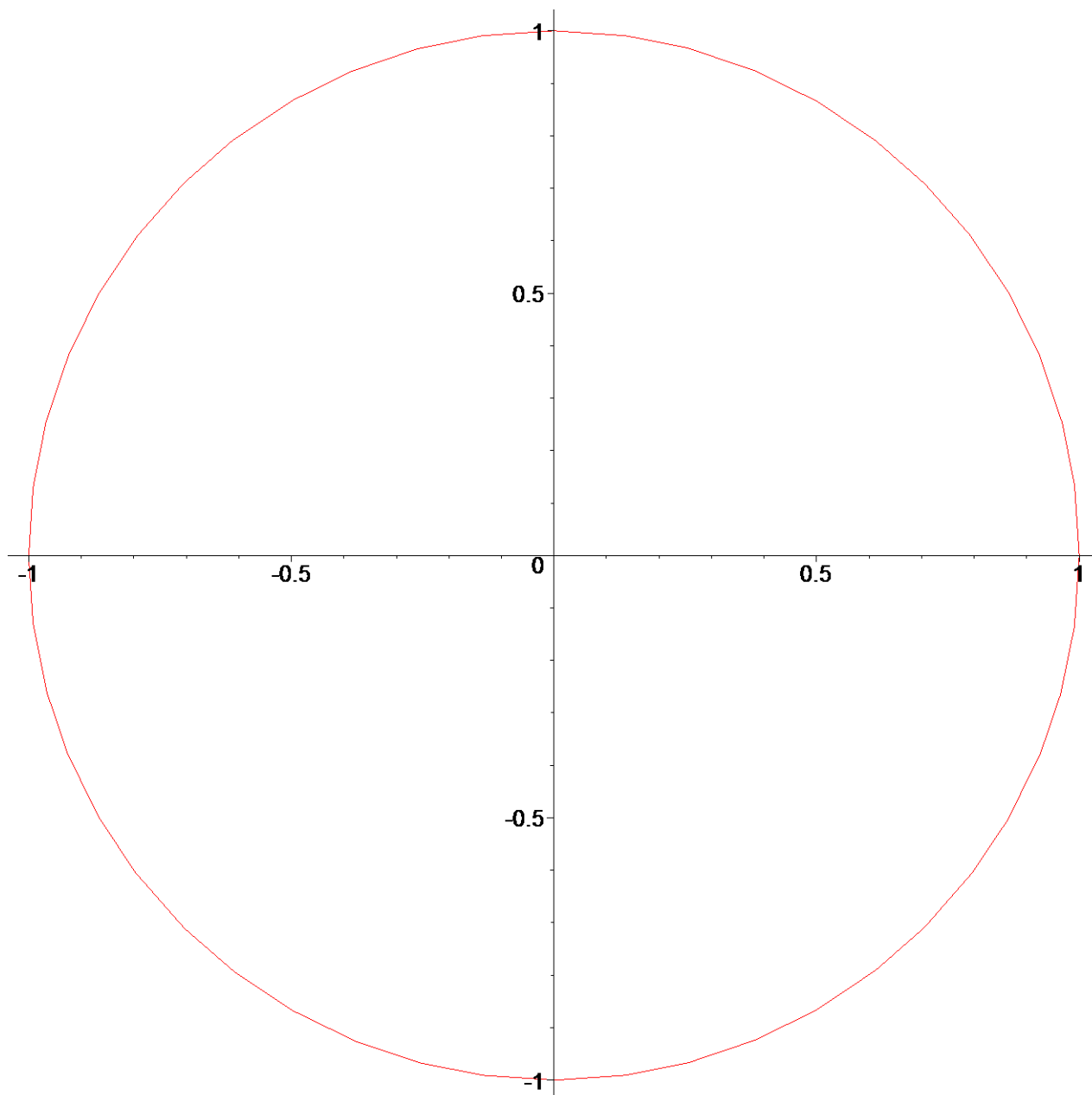


Velmi podobný způsob zápisu se užívá u funkcí zadaných parametrickými rovnicemi. Zde jde o konstrukci

`plot([relace1,relace2 , proměnná = rozsah]);`

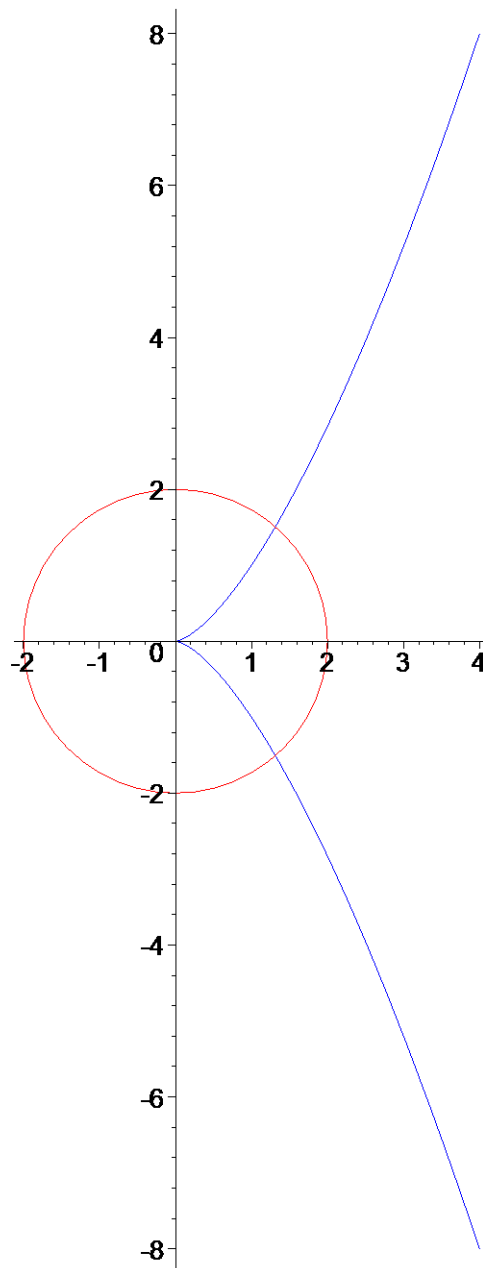
Rozdíl je v použití hranatých závorek.

`> plot([sin,cos,0..2*Pi]);` **`# parametrické zadání kružnice`**



Pro vykreslení více funkcí zadaných parametrickými rovnicemi je nutné zdvojit [.]

```
> plot([ [2*cos(t),2*sin(t),t=0..2*Pi] , [s^2,s^3,s=-2..2] ],  
       scaling=constrained, color=[red, blue]);
```



Příkaz *plot* má mnoho dalších parametrů. Uvedme nejčastěji používané

color = [barva1,barva2, ...] - definice barev funkcí dle pořadí jejich zadání (lze užít i konstrukci RGB)

linestyle = [hodnota, hodnota, ...] - definice stylu čar (každý styl má přiřazenou určitou hodnotu, viz. HELP)

title = `řetězec` - nadpis grafu

legend = [křivka1`, `křivka2`, ...] - definice legendy

labels = [`osa x`, `osa y`, ...] - pojmenování souřadnicových os

Ještě poznamenejme, že většinu parametrů příkazu *plot* lze nastavit z kontextového menu vybráním příslušné položky.

```
> fce:=[x,sin(x),cos(x),tan(x)]:
```

```
> plot(fce, x= -2*Pi..2*Pi, #
```

```
definiční interval
```

```
-5..5,
```

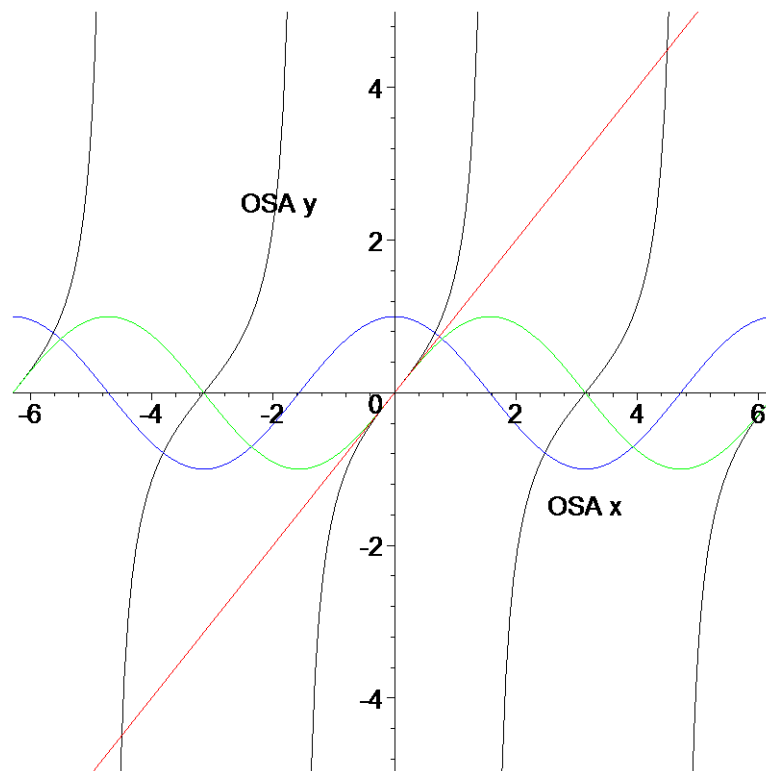
```
# obor
```





```

hodnot na ose y
    color=[red, green, blue, black], #
definice barev
    title=`Ukázka užití parametrů příkazu plot`, #
titulek obrázku
    discontinuity, #
povoleny singularity
    legend=[`x`,`sin`,`cos`,`tan`], #
definice legendy
    labels=[`OSA x`,`OSA y`] #
pojmenování souřadnicových os
);

```

Ukázka užití parametrů příkazu plot



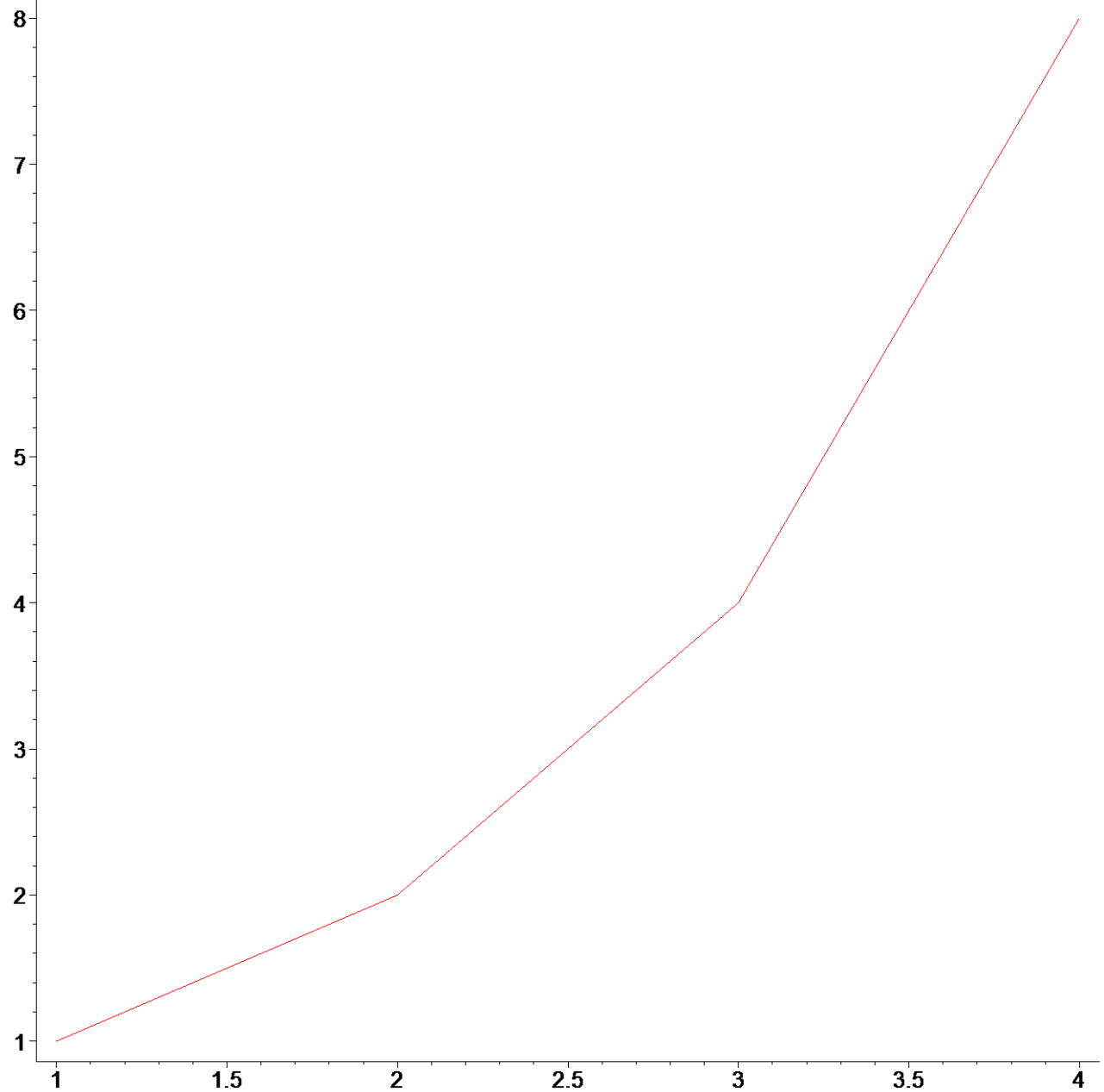
	x
	sin
	cos
	tan

Ještě se krátce zastavme u vykreslování bodů. Jednotlivé body uzavřeme jako seznam do [.]

```
> body:=[[1,1],[2,2],[3,4],[4,8]];
```

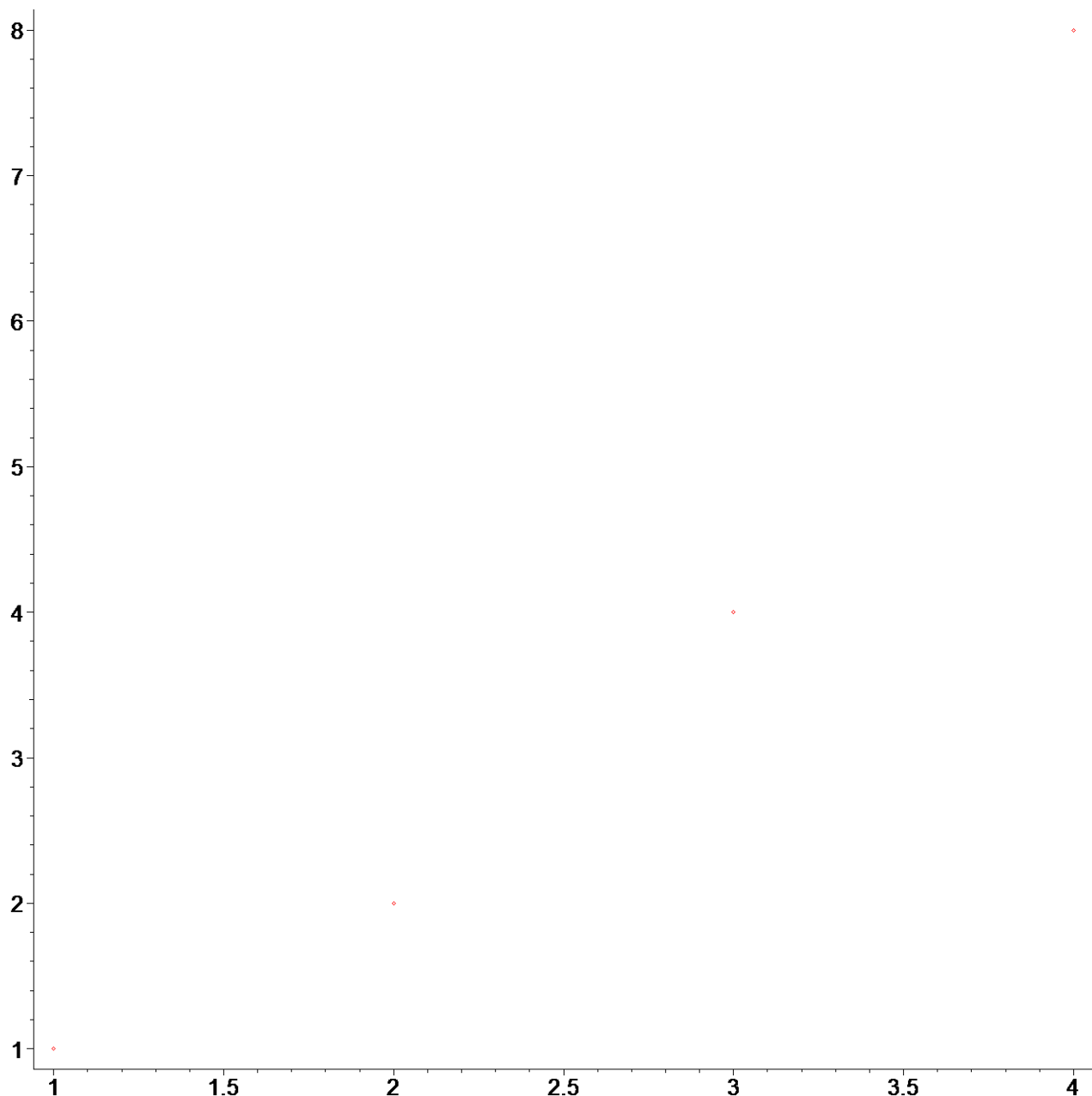
```
body := [[1, 1], [2, 2], [3, 4], [4, 8]]
```

```
> plot(body); # implicitně kreslí spojitou křivku
```



Pro vykreslení pouze jednotlivých bodů musíme užít parametru *style = point*.

```
> plot(body, style=point); # zobrazení pouze bodů
```

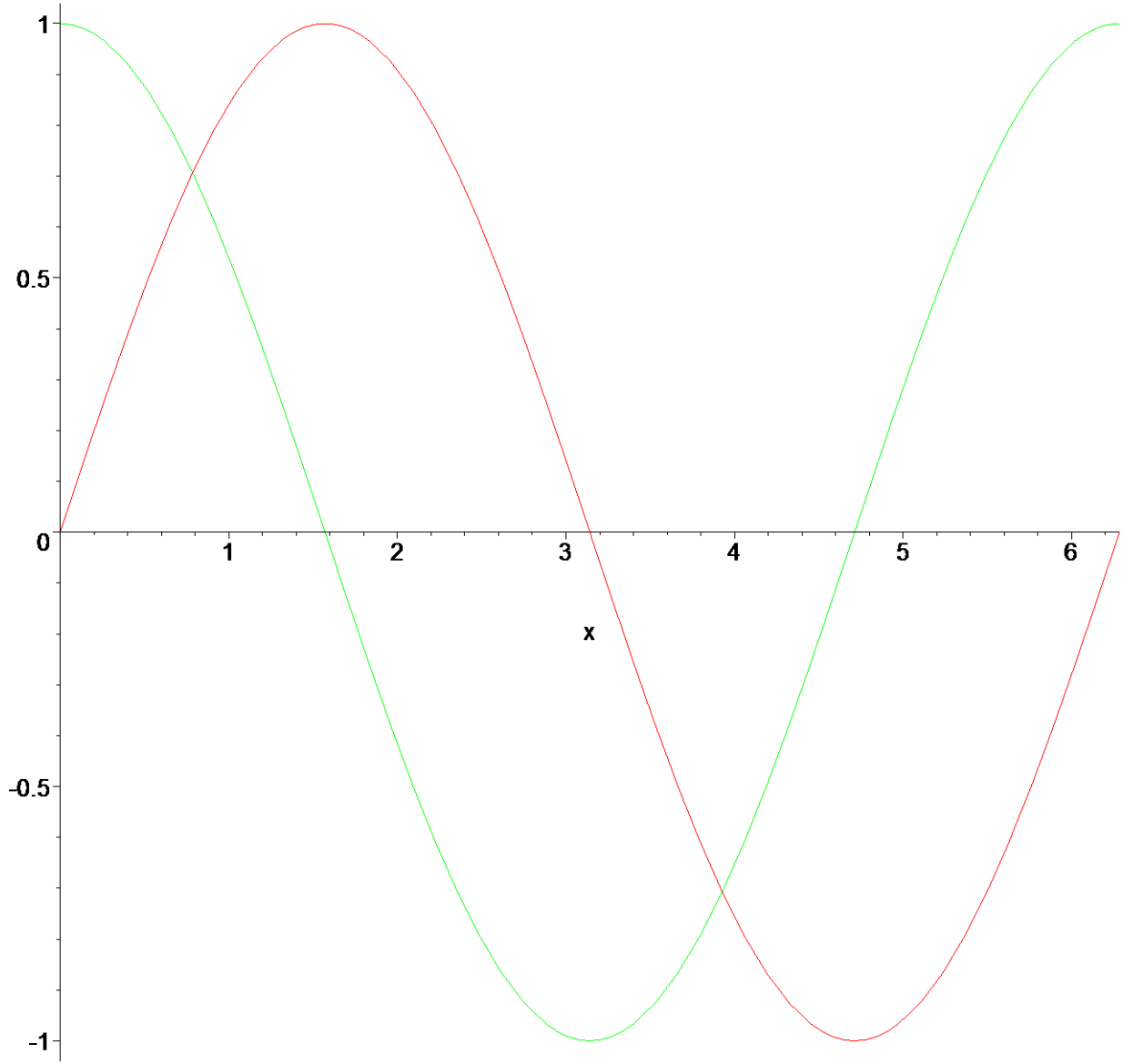


Nyní přejdeme k rozšiřující knihovně *plots*. Jak již bylo řečeno, rozšiřující knihovny (tzv. *packages*) se nahrávají pomocí příkazu *with(plots)*; Po jeho uvedení a stisknutí klávesy ENTER se zobrazí seznam příkazů, které jsou nyní dostupné. Ještě poznamenejme, že většina příkazů z této knihovny se chová podobně jako příkaz *plot* (akceptuje parametry, rozsah na osy *y*, apod.).

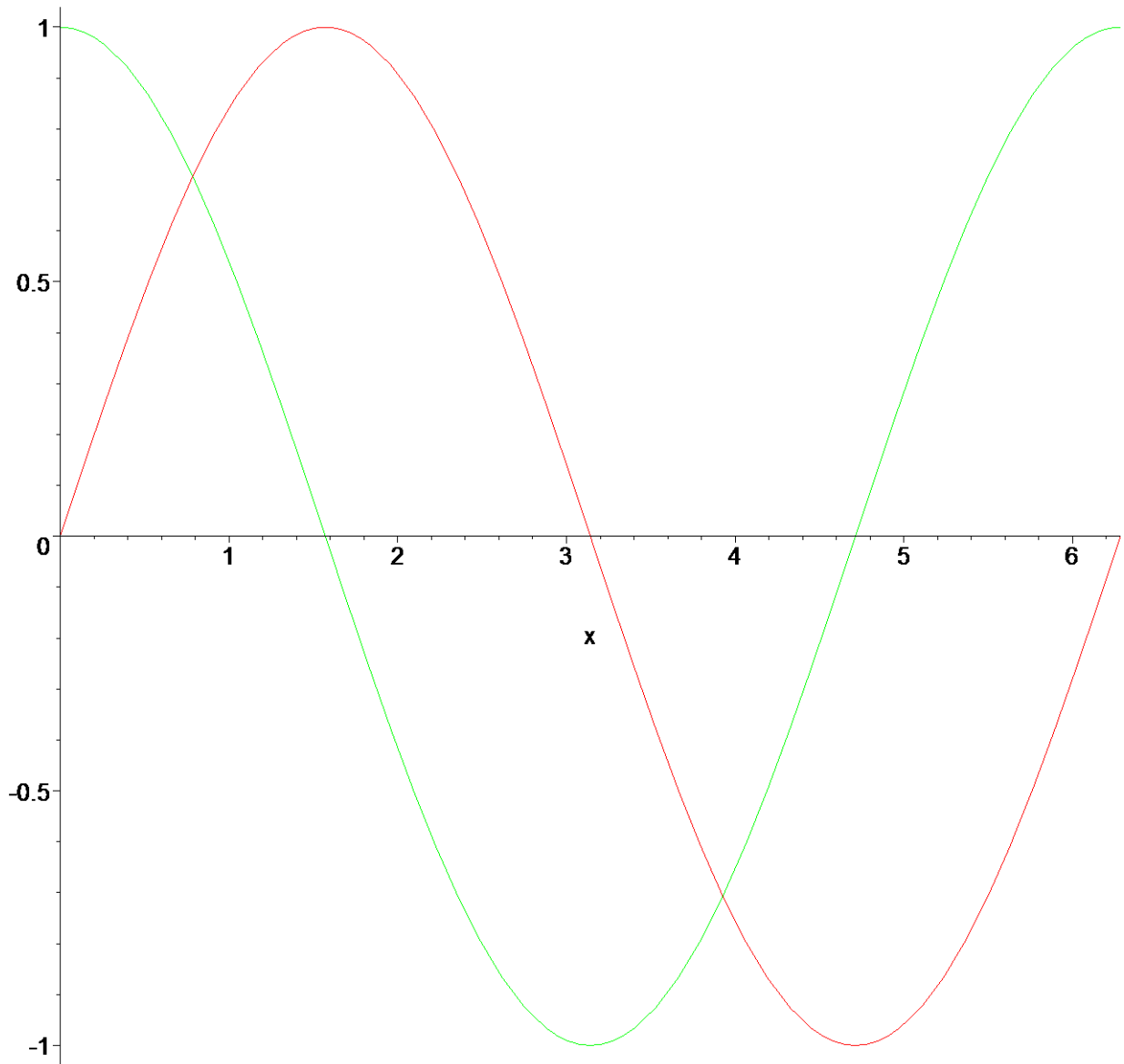
Následující dva obrázky ukazují, že pro vykreslení více funkcí do jednoho lze použít i množinu místo seznamu, nicméně v tom případě není zaručeno, že je pořadí zadaných funkcí takové, jak ukazuje zápis.

Nemusí se tedy např. s pořadím v barvách nebo třeba v legendě.

```
> plot([sin(x), cos(x)], x=0..2*Pi); # spravne
```

```
> plot({sin(x),cos(x)},x=0..2*Pi); # spatne
```



```

>
>
>
>
> restart;with(plots);

```

Warning, the name `changecoords` has been redefined

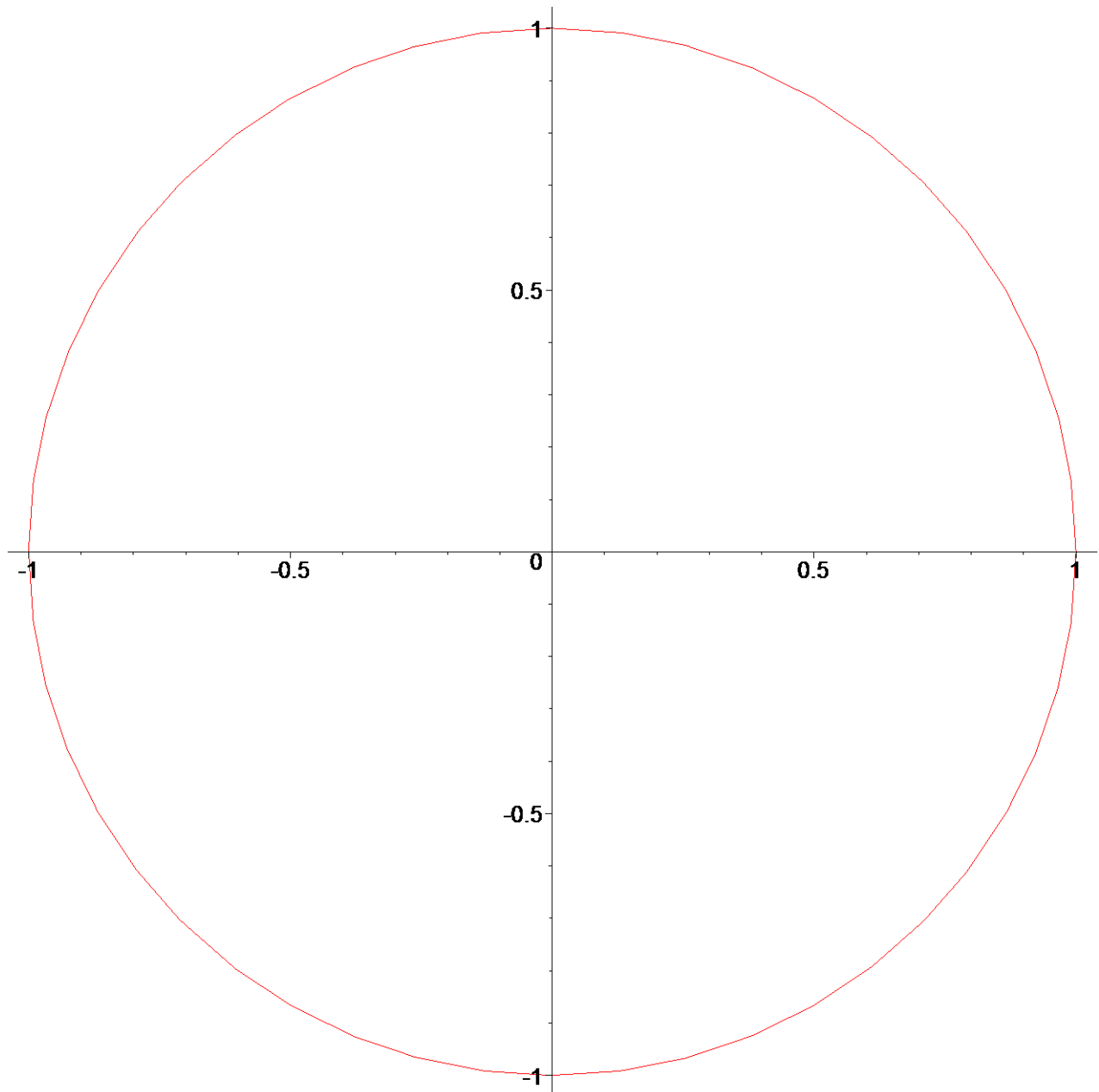
[*animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d, conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, cylinderplot, densityplot, display, display3d, fieldplot, fieldplot3d, gradplot, gradplot3d, graphplot3d, implicitplot, implicitplot3d, inequal, interactive, listcontplot, listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d, polyhedra_supported, polyhedraplot, replot, rootlocus, semilogplot, setoptions, setoptions3d, spacecurve, sparsematrixplot,*

sphereplot, surfdata, textplot, textplot3d, tubeplot]

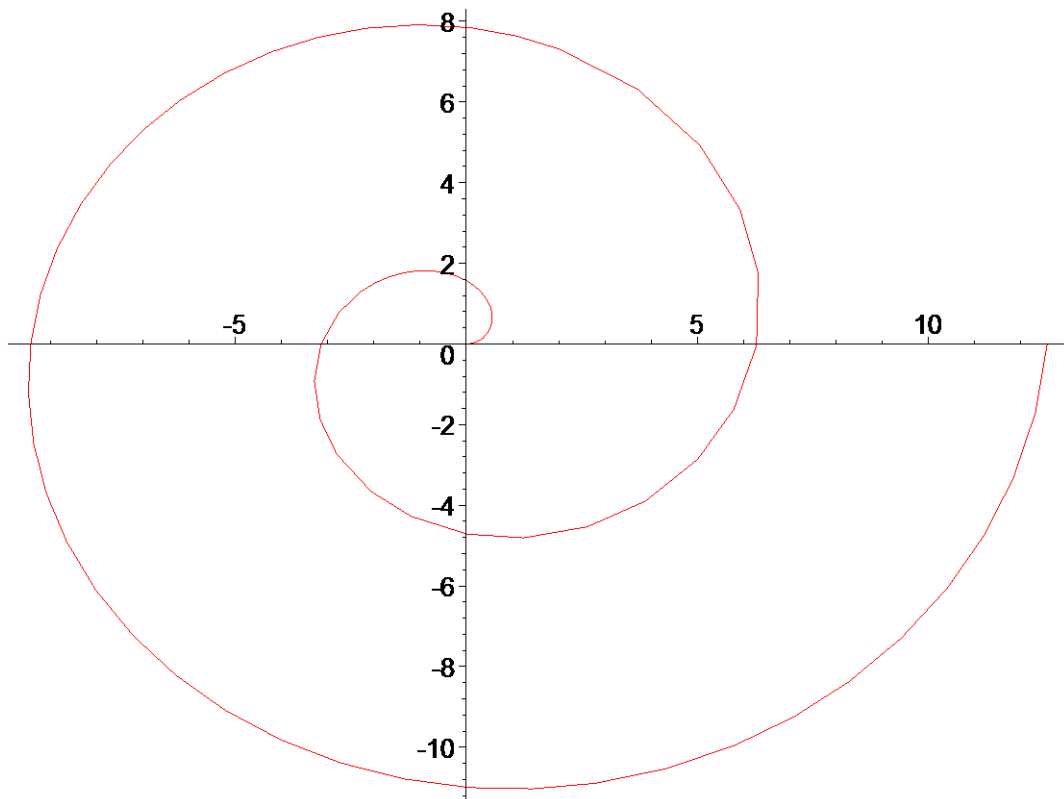
Následuje přehled nejpoužívanějších příkazů a jejich použití.

• **polarplot(výraz, úhel = rozsah)** - grafy funkcí v polárních souřadnicích

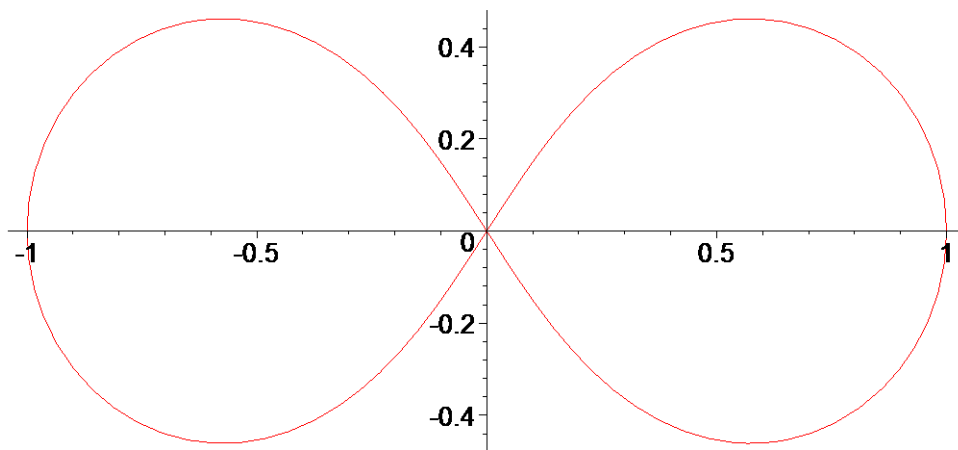
> **polarplot(1, theta=0..2*Pi, scaling=constrained);** # kružnice



> **polarplot(theta, theta=0..4*Pi);** # spirála

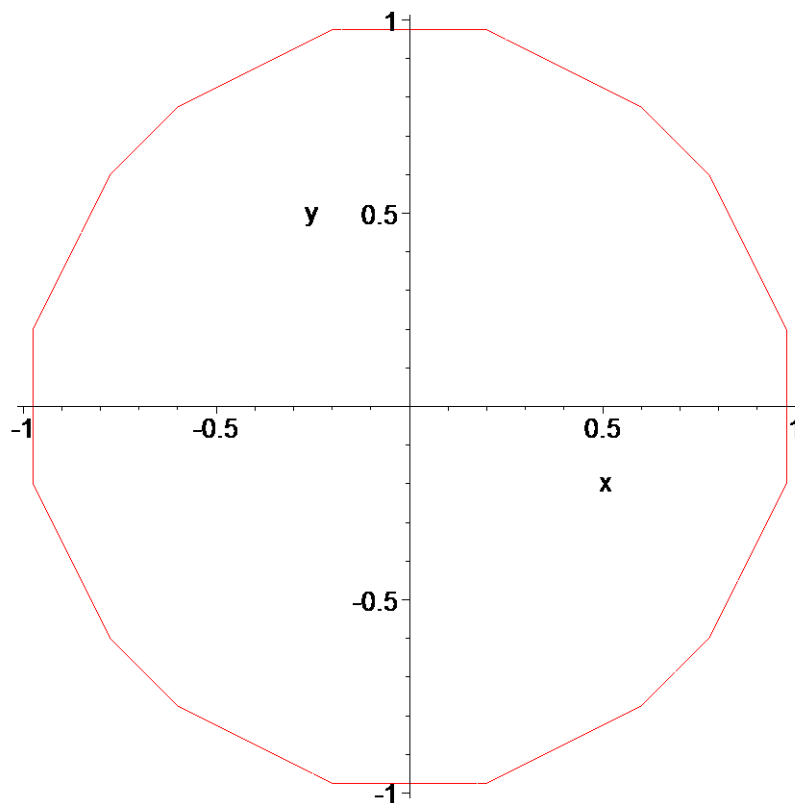


> `polarplot([sin(t), cos(t), t=0..2*Pi], scaling=constrained);`



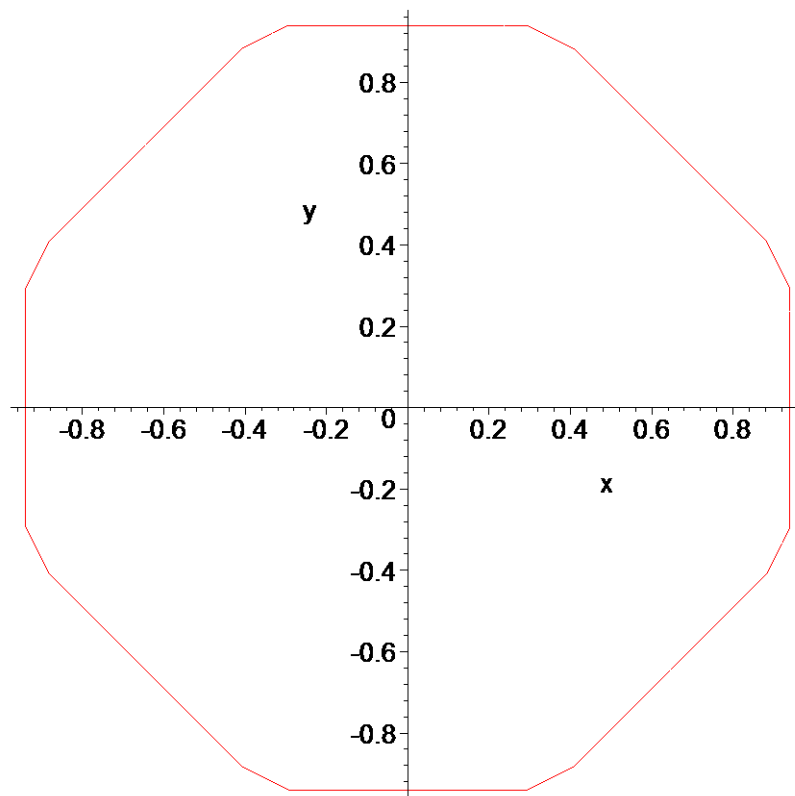
• `implicitplot(F(x,y)=0, proměnná = rozsah)` - grafy funkcí daných implicitně, tj. ve tvaru $F(x,y) = 0$.

> `implicitplot(x^2+y^2-1, x=-5..5, y=-5..5, scaling=constrained); # lze vypustit =0`

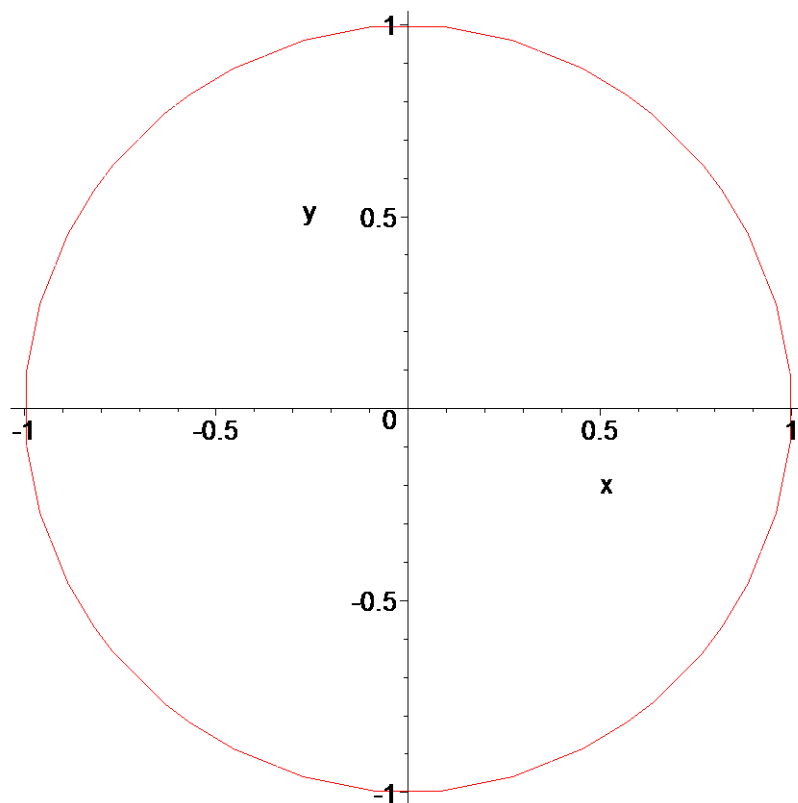


Podle předpisu funkce jde o kružnici, ale předchozí obrázek tomu nenasvědčuje. Problém je v počtu bodů, které Maple vykresluje. Lze jej nastavit pomocí parametru *numpoints*. Na dalších dvou obrázcích ukážeme jeho použití.

```
> implicitplot(x^2+y^2-1,x=-5..5,y=-5..5,scaling=constrained,numpoints=300);
```

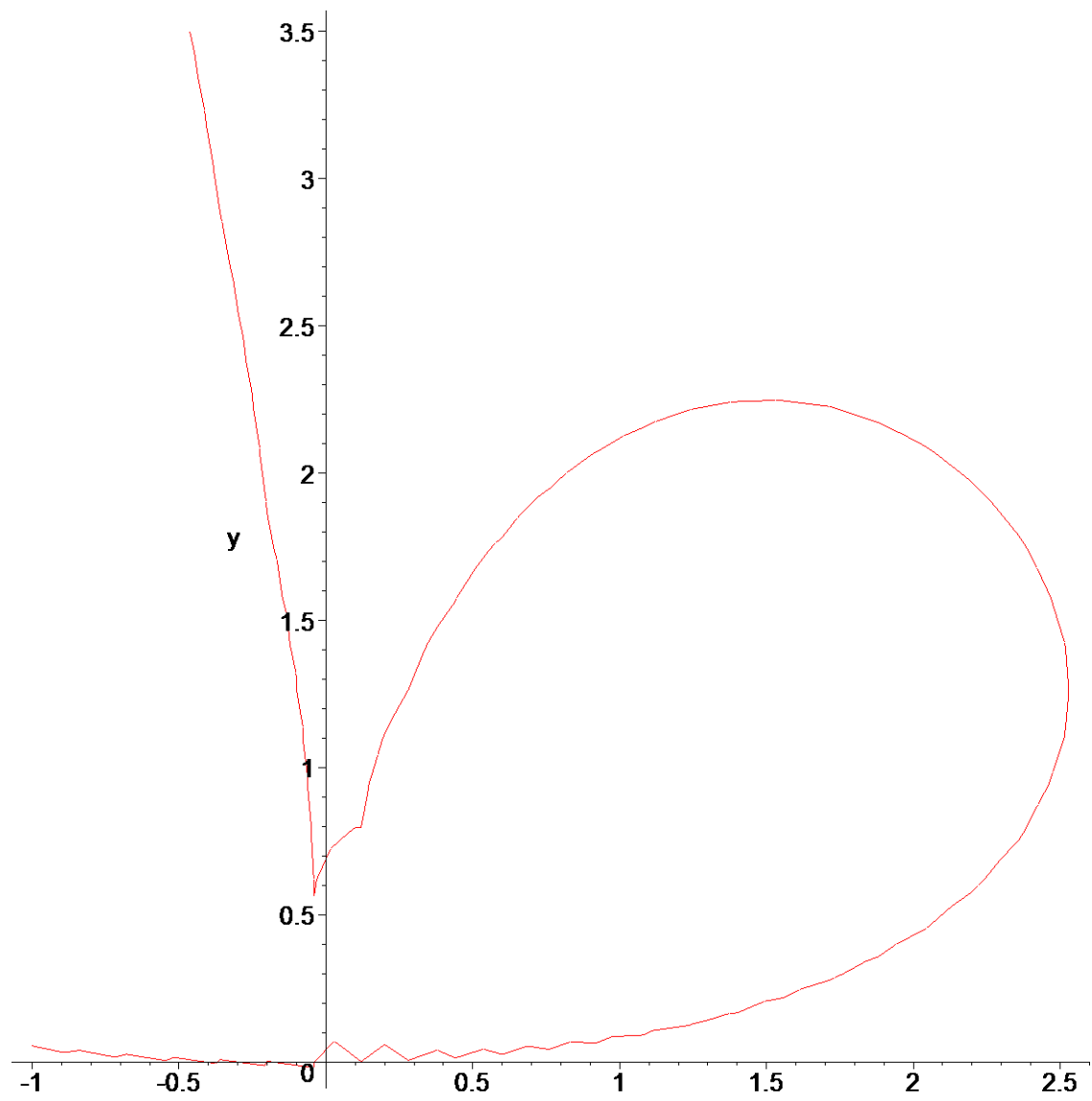


```
> implicitplot(x^2+y^2-1,x=-5..5,y=-5..5,scaling=constrained,numpoints=3000);
```

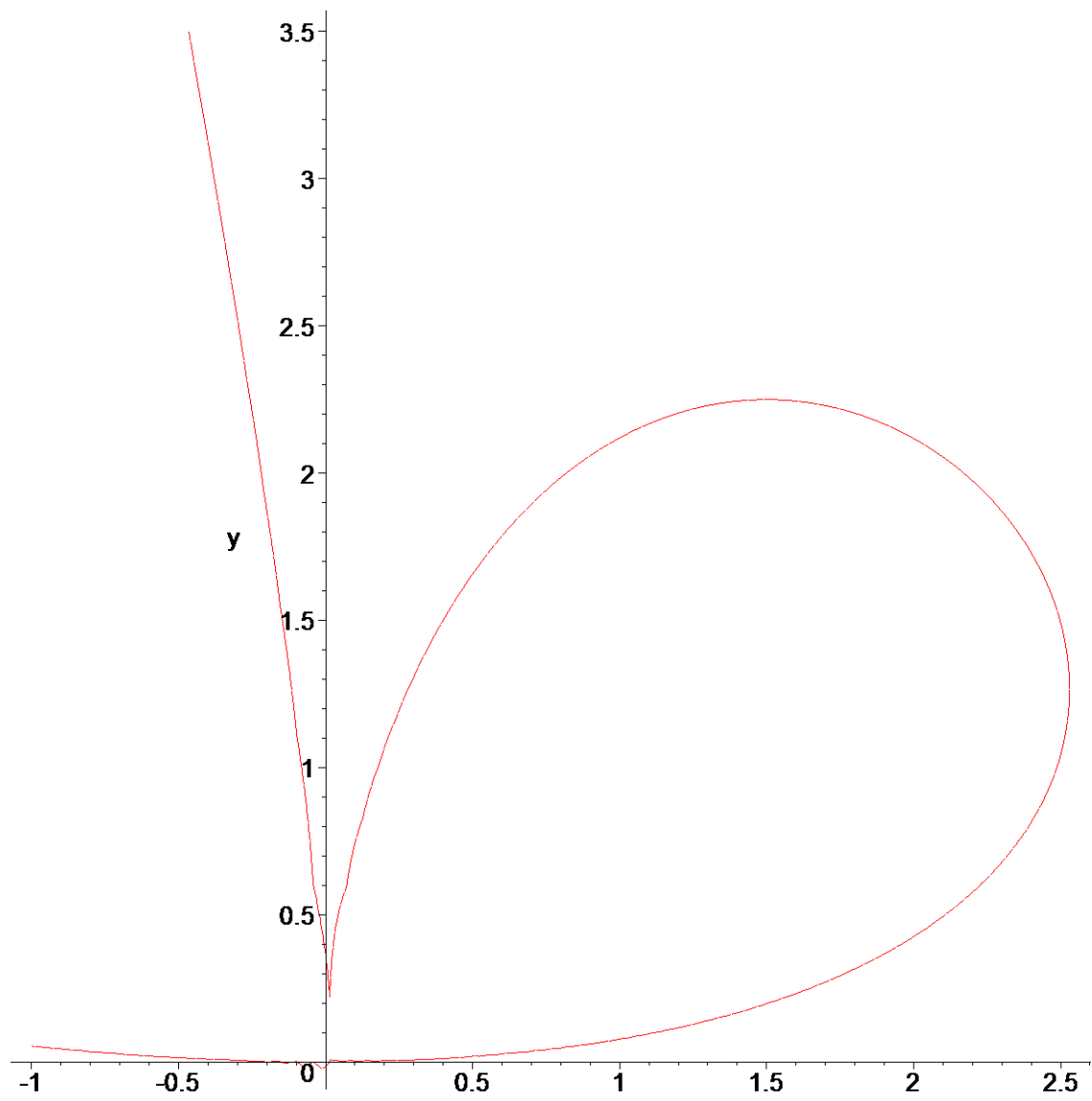


Následující příklad ukazuje problémy při zobrazování implicitně zadaných funkcí

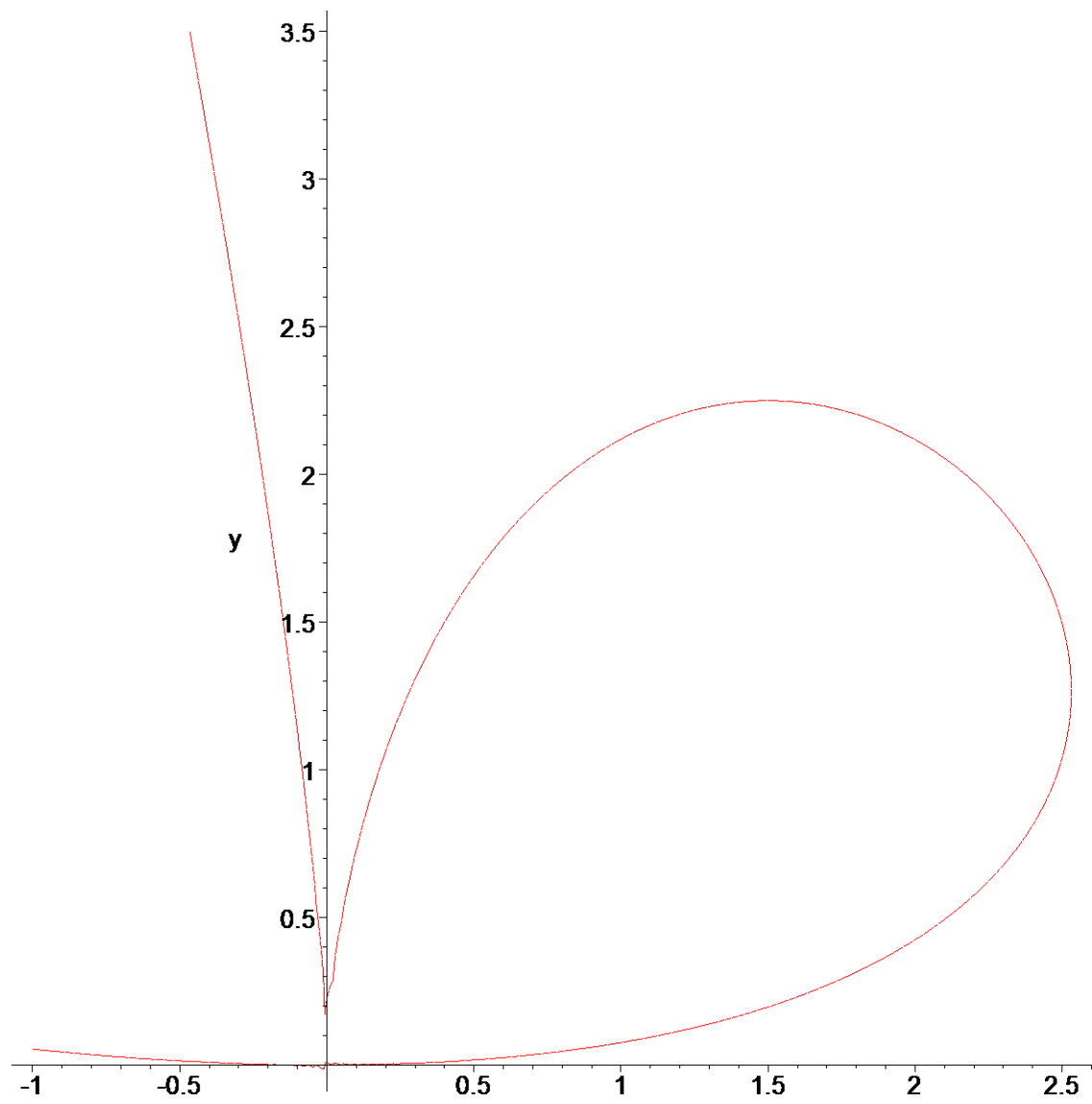
```
> implicitplot( (x/2+y/3)^4 - x^2*y=0, x=-1..3, y=-0.5..3.5);
```



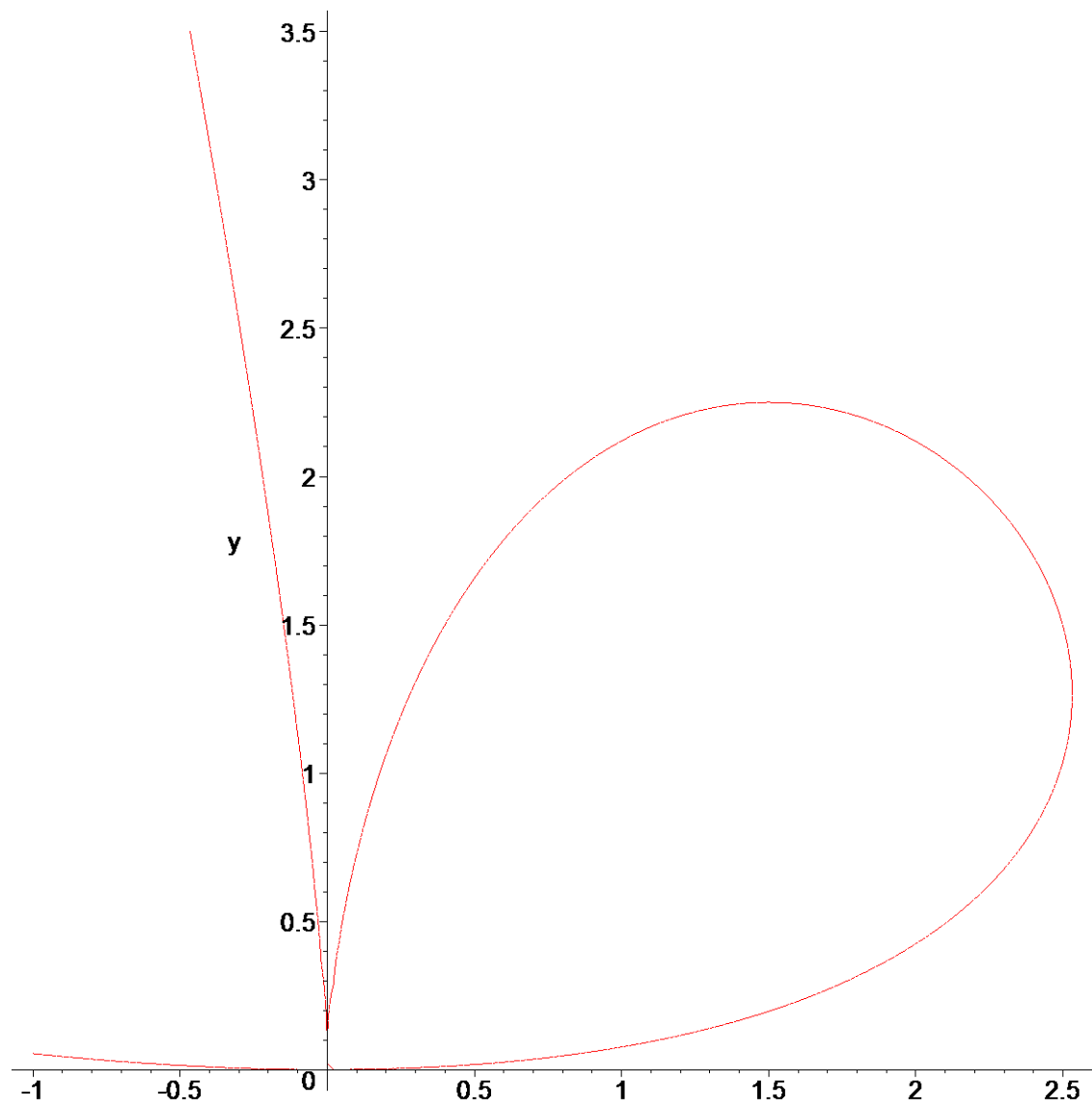
```
> implicitplot( (x/2+y/3)^4 - x^2*y=0, x=-1..3,  
y=-0.5..3.5,numpoints=5000);
```



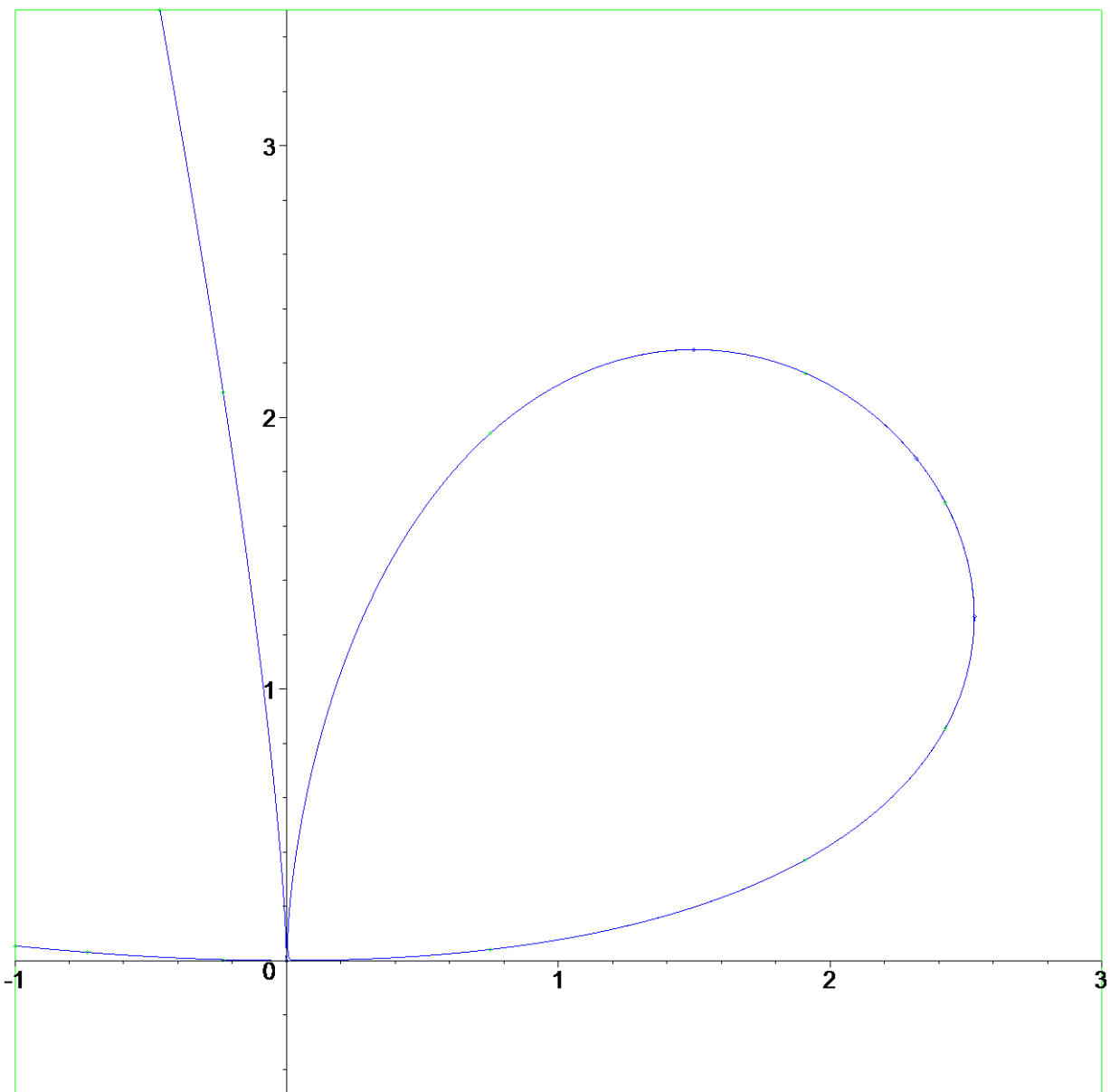
```
> implicitplot( (x/2+y/3)^4 - x^2*y=0, x=-1..3,  
y=-0.5..3.5,numpoints=20000);
```

```
> implicitplot( (x/2+y/3)^4 - x^2*y=0, x=-1..3,  
y=-0.5..3.5,numpoints=36670);
```



```
> algcurves[plot_real_curve]((x/2+y/3)^4 - x^2*y, x,  
y,view=[-1..3,-0.5..3.5]);
```

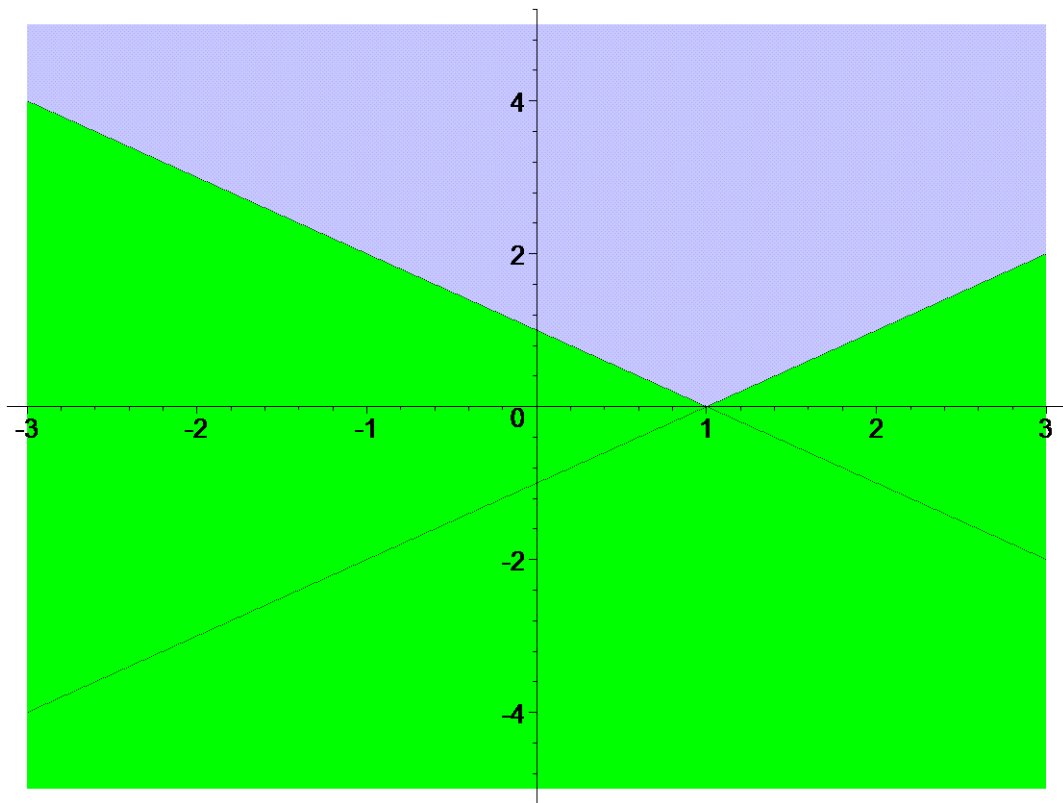


>

- *inequal*({omezení}, proměnná = rozsah)

- vykreslení množiny bodů dané lineárními nerovnostmi (vhodné v optimalizaci k zobrazení množiny přípustných řešení)

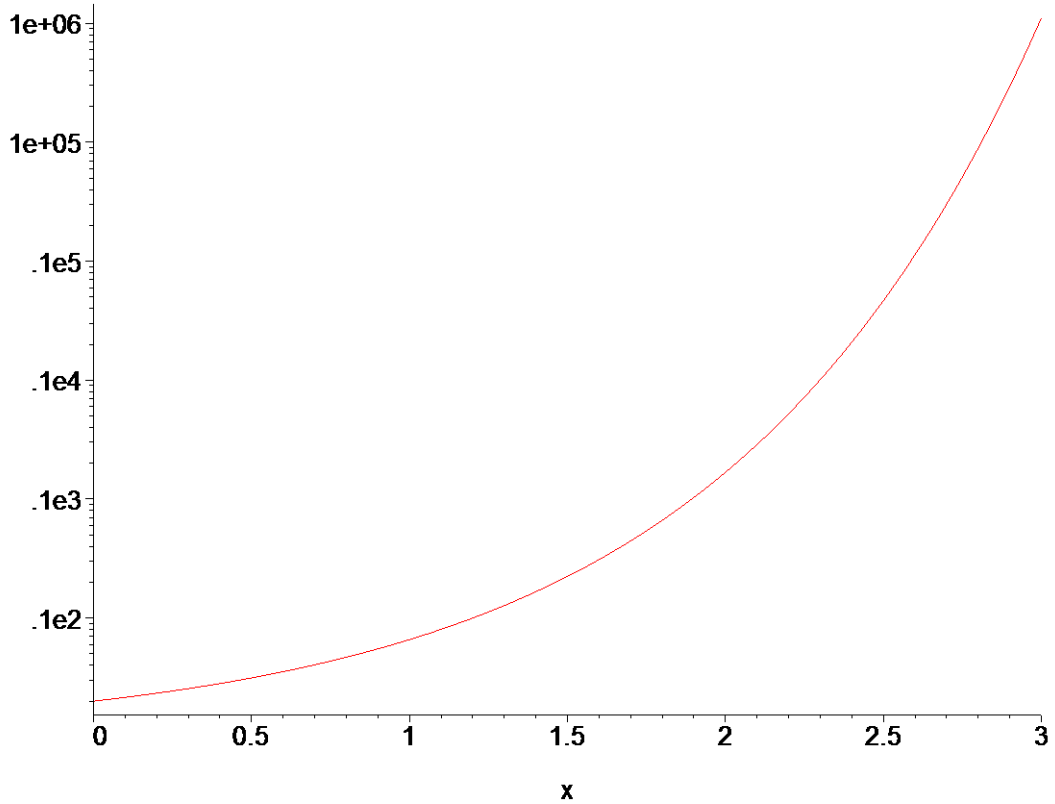
```
> inequal ({x-y<1,x+y>1},x=-3..3,y=-5..5,  
           optionsexcluded=(color=green,thickness=2));  
           # parametr optionsexcluded určuje zobrazení doplňku,  
viz. HELP
```



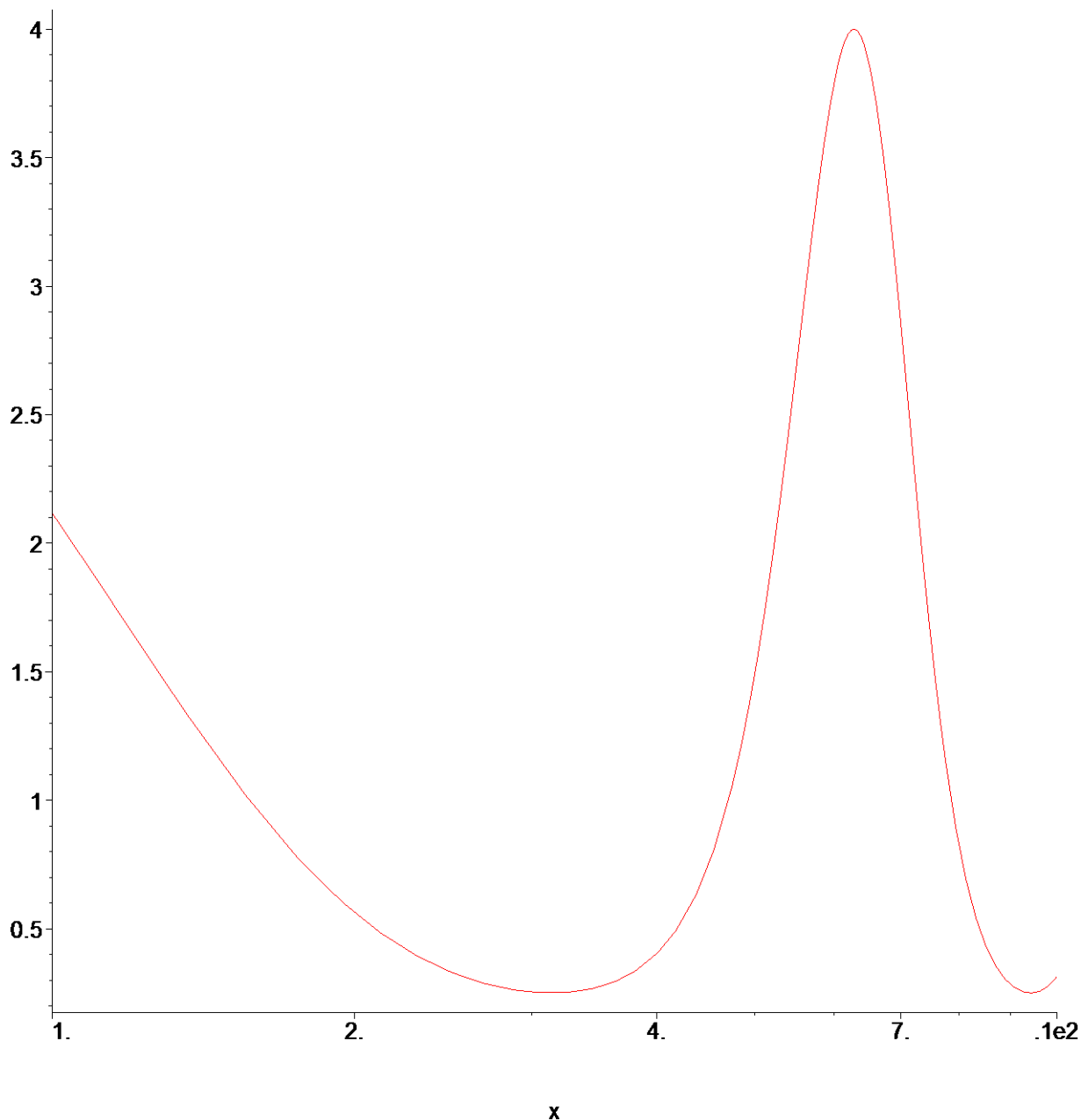
- *logplot* , popř. *semilogplot* stupnice

- graf funkce $y=f(x)$, kde na ose y (popř. x) je logaritmická stupnice

> `logplot(2^exp(x), x=0..3);`



> `semilogplot(4^cos(x), x=1..10);`

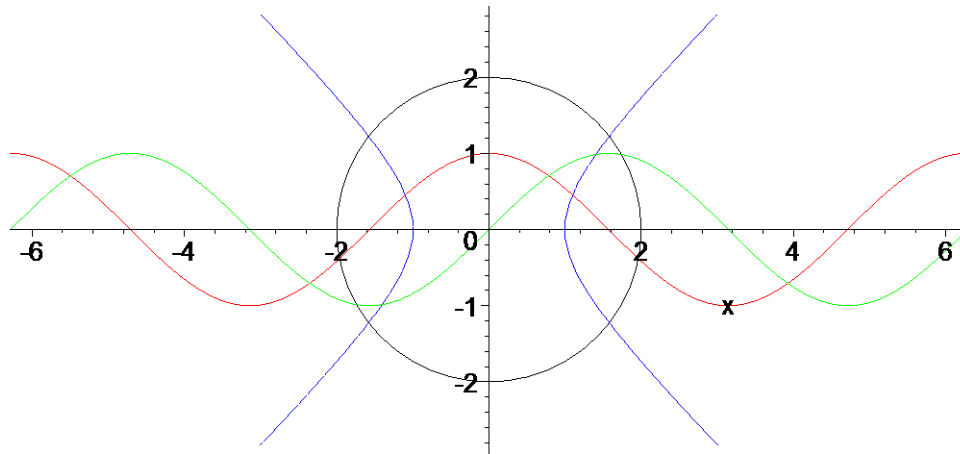


- *display({fce1, fce2, ...}, parametry)* - pro vykreslení více grafů v jednom obrázku, jsou-li jednotlivé obrázky pojmenovány

```

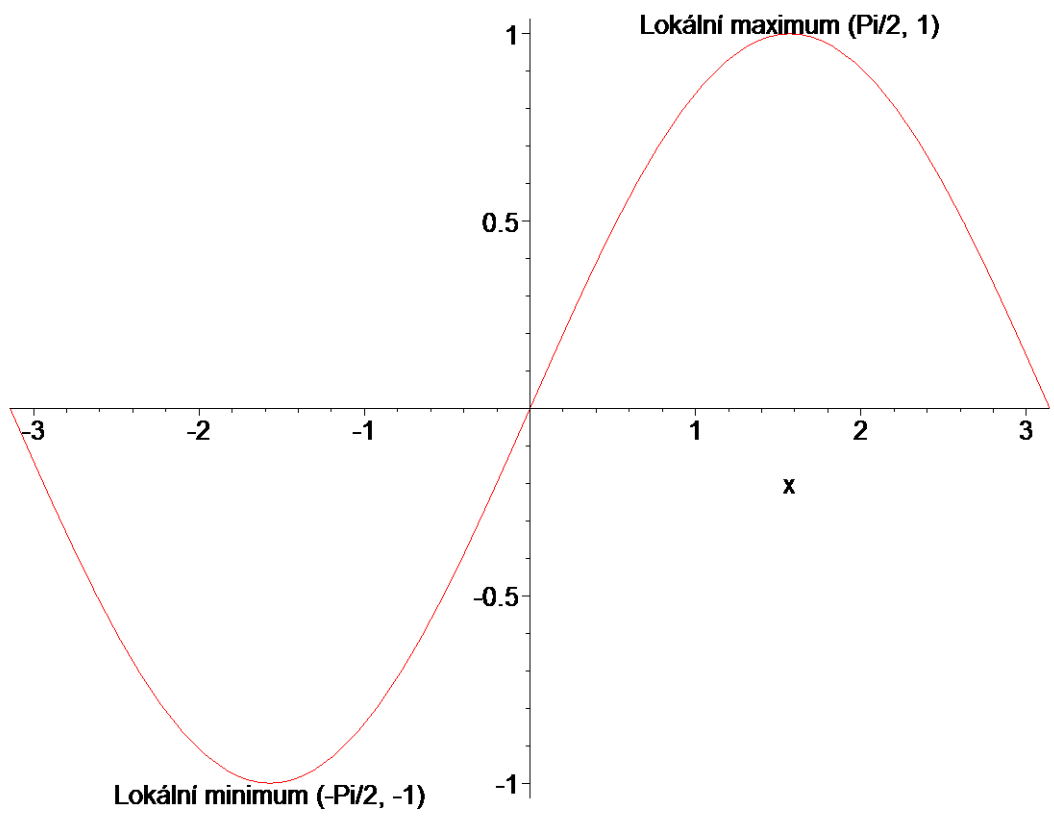
> numpoints=500: #
počet vykreslovaných bodů
a:=plot(sin(x), x=-2*Pi..2*Pi, color=green): #
sin(x)
b:=plot(cos(x), x=-2*Pi..2*Pi, color=red): #
cos(x)
c:=polarplot(2, theta=0..2*Pi, color=black): #
kružnice
d:=implicitplot(x^2-y^2-1, x=-3..3, y=-3..3, color=blue): #
hyperbola
> display([a,b,c,d], scaling=constrained);

```



- `textplot([souřadnice x, souřadnice y, `popis`, parametry])` - umožňuje psát texty do grafu na určená místa

```
> obrazek := plot(sin(x), x=-Pi..Pi): # sin(x)
t1 := textplot([Pi/2, 1, `Lokální maximum (Pi/2,
1)`], align=ABOVE):
t2 := textplot([-Pi/2, -1, `Lokální minimum (-Pi/2,
-1)`], align=BELOW):
display({obrazek, t1, t2});
```



Grafy ve třech dimenzích

Maple umožňuje vykreslovat graf funkce dvou proměnných jako plochu ve třech dimenzích. K tomu je určen příkaz `plot3d`, který je svou syntaxí velmi podobný příkazu `plot`. Pouze je nutné přidat informaci o rozsahu druhé nezávislé proměnné.

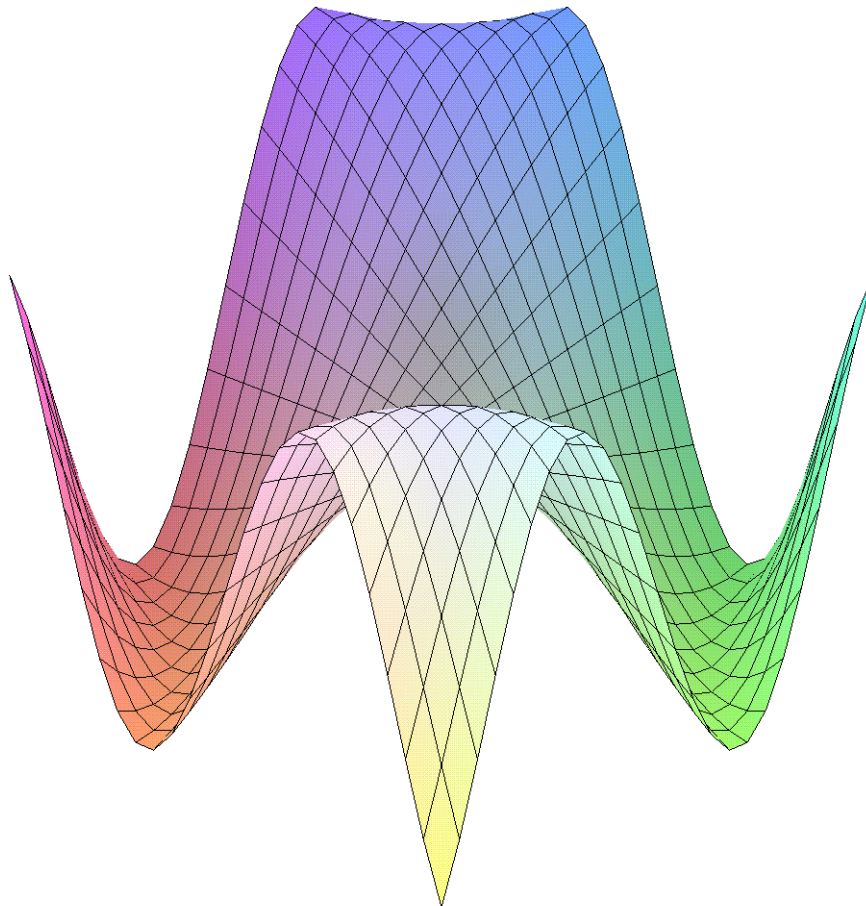
```
plot3d(výraz, proměnná = a..b, proměnná = c..d);
```

```
plot3d(funkce, a..b,c..d);
```

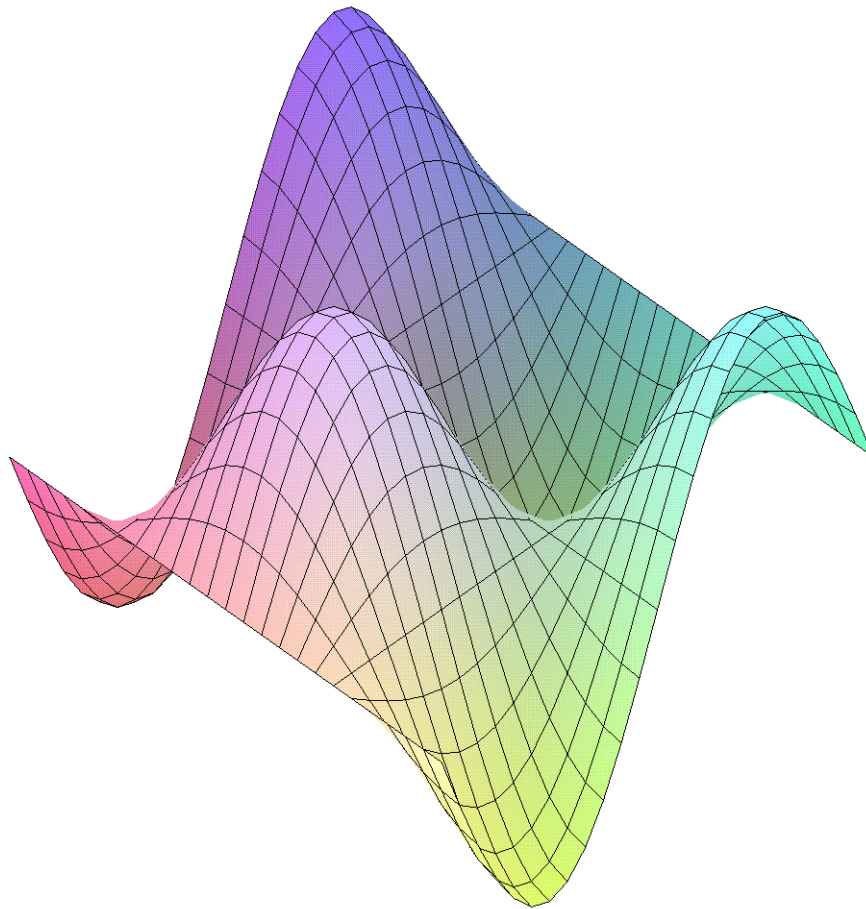
```
plot3d(funkce, a..b,c..d,e..f);
```

První příkaz vykreslí daný výraz (jde o předpis funkce) na daném kartézském čtverci. Druhý se používá pro vykreslení funkce $f(x,y)$ a třetí vykresluje funkce na kartézském čtverci s mezením rozsahu funkčních hodnot.

```
> plot3d(sin(x*y), x=-2..2, y=-2..2);
```

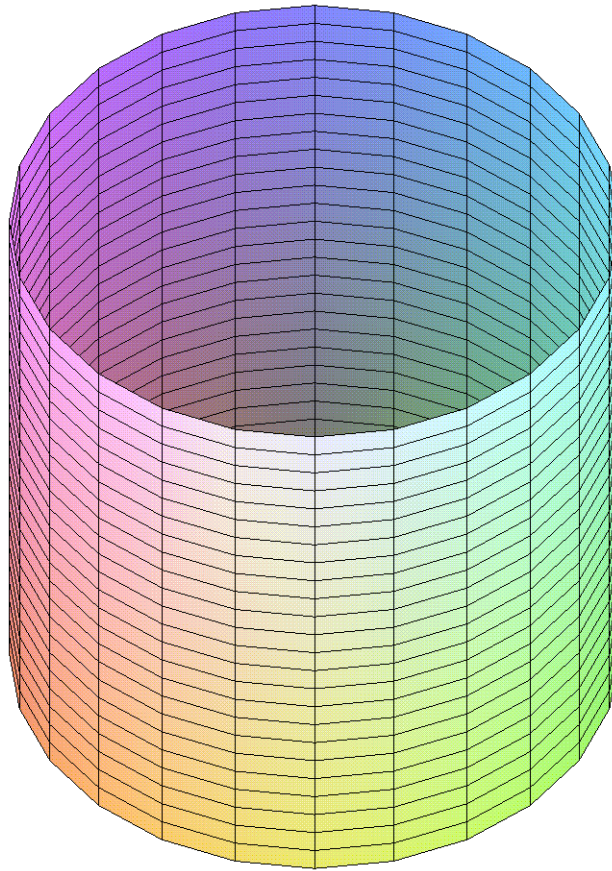


```
> f:=(x,y)->sin(x)*cos(y); # definice funkce  
> plot3d(f,-Pi..Pi,-Pi..Pi);
```

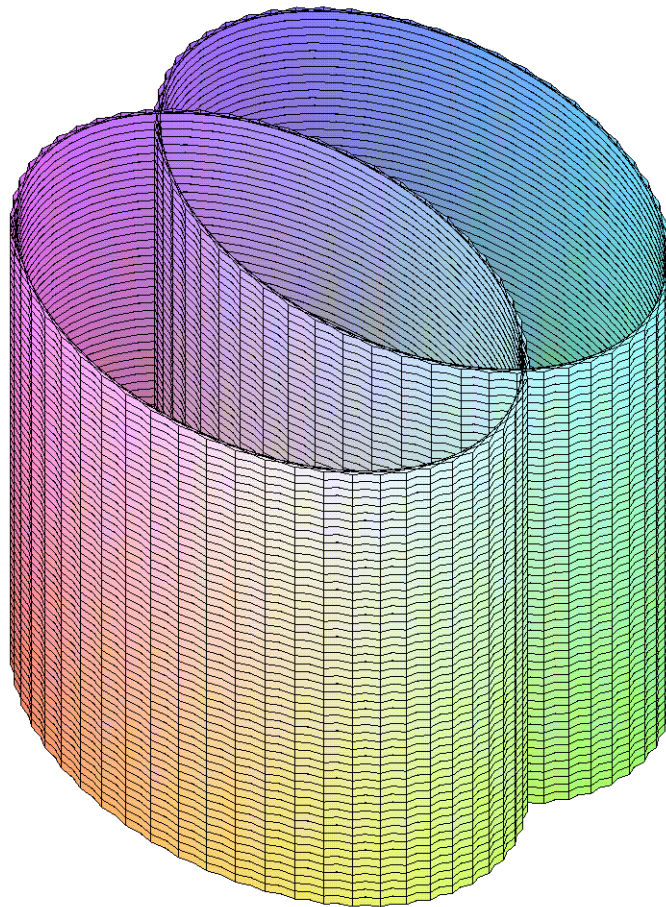
Pro parametrické vyjádření je zápis příkazu *plot3d* analogický jako při použití *plot*, ukažme si to na příkladu válce.

```
> plot3d( [sin(s), cos(s), t], s=-Pi..Pi, t=-1..2);
```



Pro zobrazení více ploch v jednom obrázku postupujeme analogicky jako při zobrazování pomocí příkazu *plot*. Je tu jedno zásadní omezení. Všechny funkce musí být buď zadány pomocí parametrických rovnic a nebo ve standardním tvaru. Kombinace zadání není přípustná. Dalším rozdílem je uzavření do množinové závorky, tj. `{.}`.

```
> plot3d( { [sin(x), cos(x), y], [1+cos(y), sin(y), x] },  
          x=-4*Pi..4*Pi, y=-4*Pi..4*Pi, numpoints=3000 );
```



System Maple umožňuje po kliknutí na obrázek nastavovat určité vlastnosti a to pomocí tzv. kontextové nabídky, kterou vyvoláme kliknutím pravého tlačítka na daný obrázek. Ale nejen to, Maple také umožňuje rotaci obrázku kolem jednotlivých souřadnicových os (klikneme levým tlačítkem a držíme ho stisknuté, posouváním kurzoru dosáhneme rotace). Veškeré tyto operace jsou dosažitelné i z hlavního menu programu.

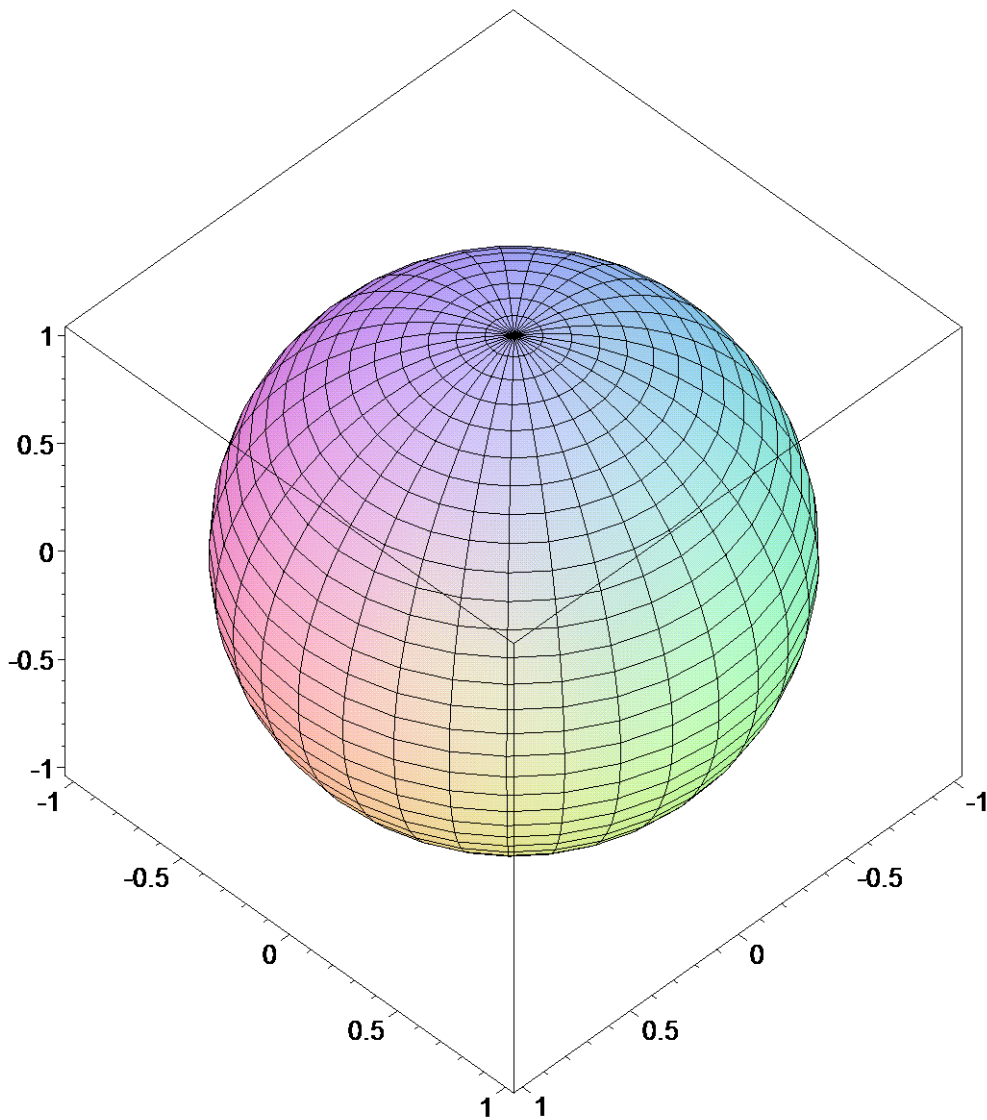
Jak víme, kartézský systém souřadnic je jen jedním z několika souřadnicových systémů. Maple nabízí několik příkazů, které vykreslují grafy v jiných souřadnicových systémech. Tyto příkazy jsou součástí tzv. *plots package*. Lze však docílit vykreslení funkcí zadaných v jiných souřadnicových systémech i pomocí příkazu *plot3d*, kde uvedeme parametr *coords=soustava*. Tento parametr má několik předdefinovaných hodnot, které lze nalézt v nápovědě. Ukažme si to na kouli ve sférických souřadnicích.

```
> plot3d(1, theta=0..2*Pi, phi=0..Pi,
          axes=boxed, # zobrazení
          souřadnicových os
```

```

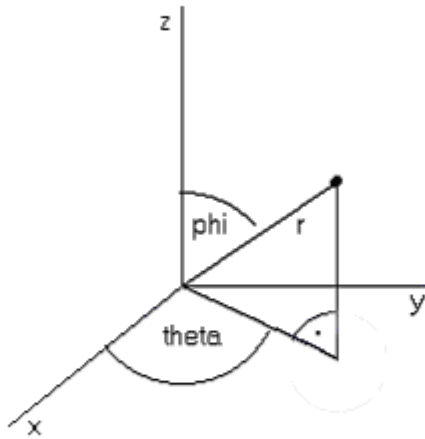
souřadnice      coords=spherical,      # sférické
osách           scaling=constrained,  # stejná měřítka na
mřížka         style=PATCH,        # bude zobrazena
               numpoints=1055);

```

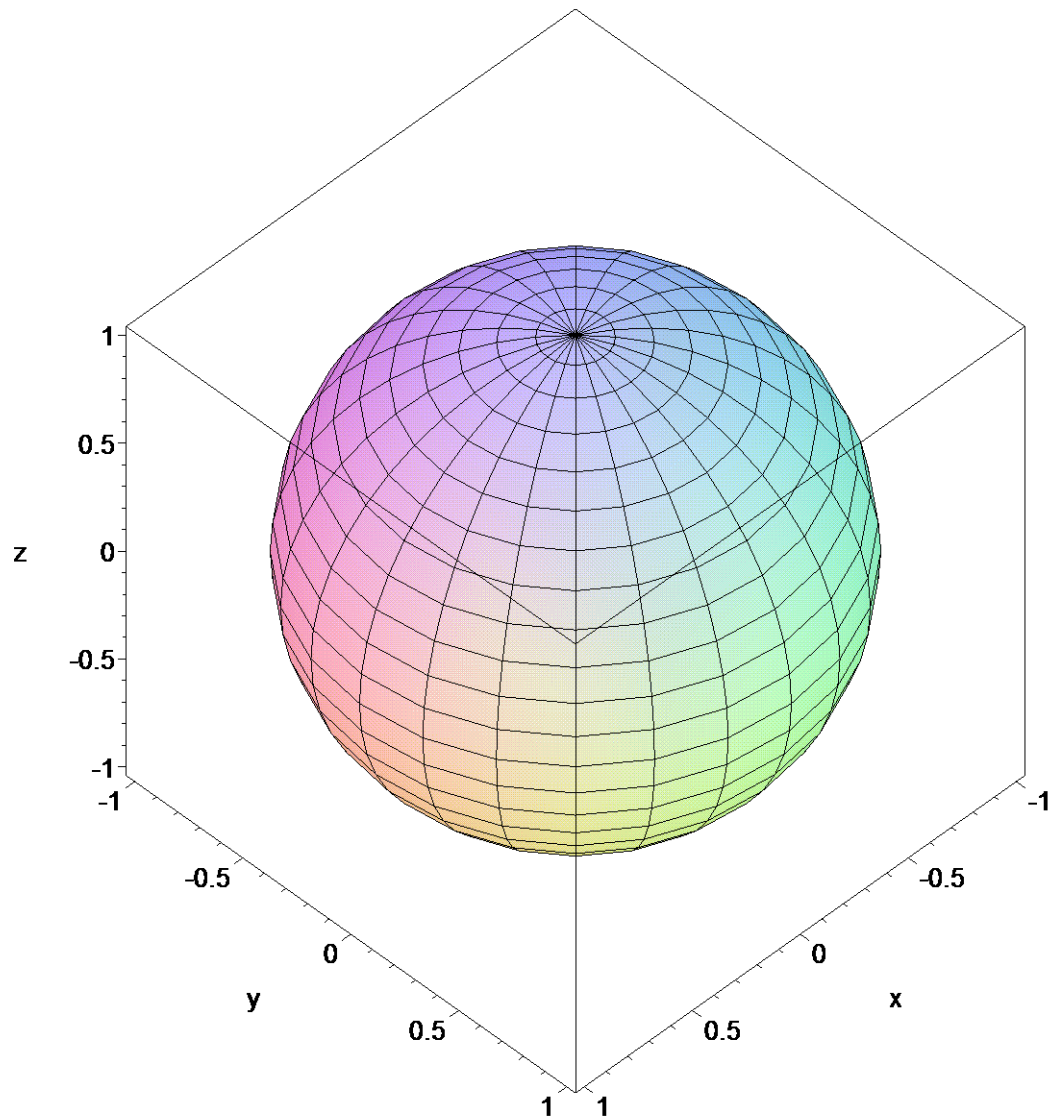


Dále uvedeme některé příkazy z rošiřující knihovny *plots*.

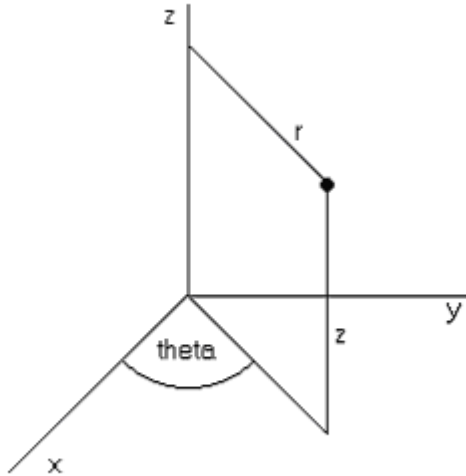
- *sphereplot*(*r*, *theta* = *rozsah*, *phi* = *rozsah*)
 - vykreslení funkcí ve sférických souřadnicích (význam jednotlivých parametrů vysvětluje následující obrázek)



```
> sphereplot(1,theta=0..2*Pi,phi=0..Pi,scaling=constrained,axes=boxed);
```



- ***cylinderplot(r,theta = rozsah, z = rozsah)***
- vykreslení funkcí v cylindrických souřadnicích (význam jednotlivých parametrů vysvětluje následující obrázek)

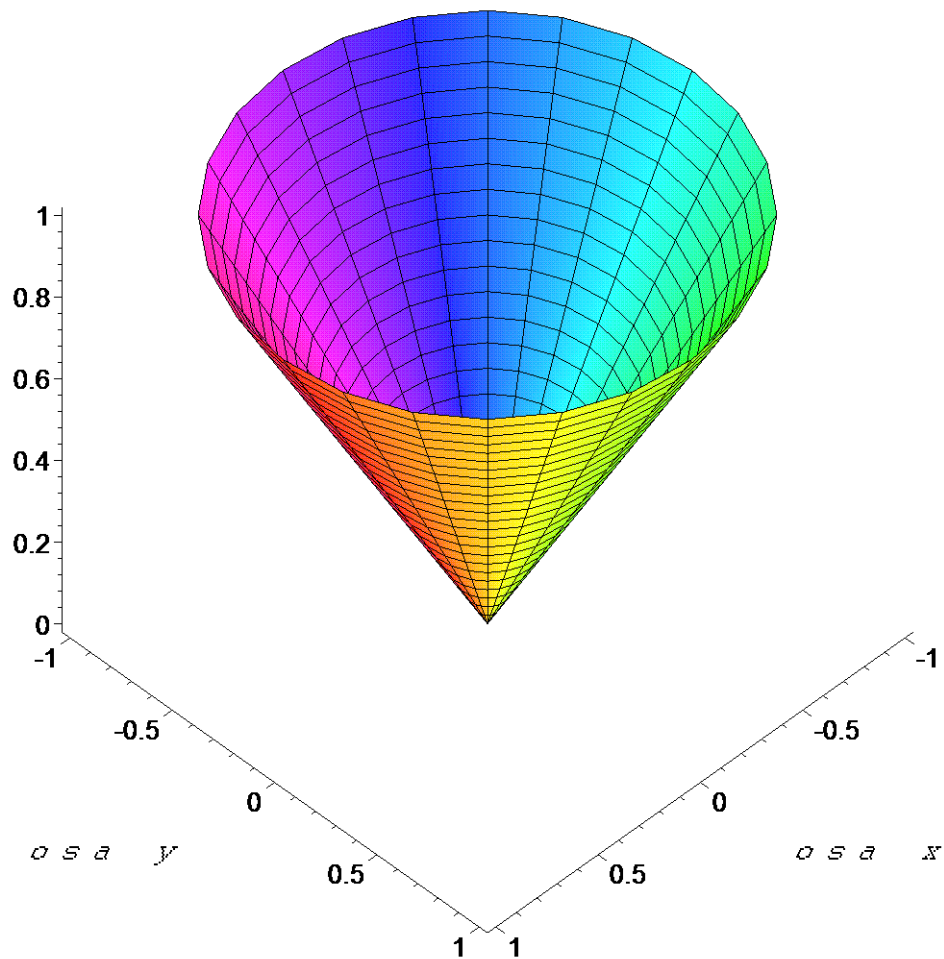


```

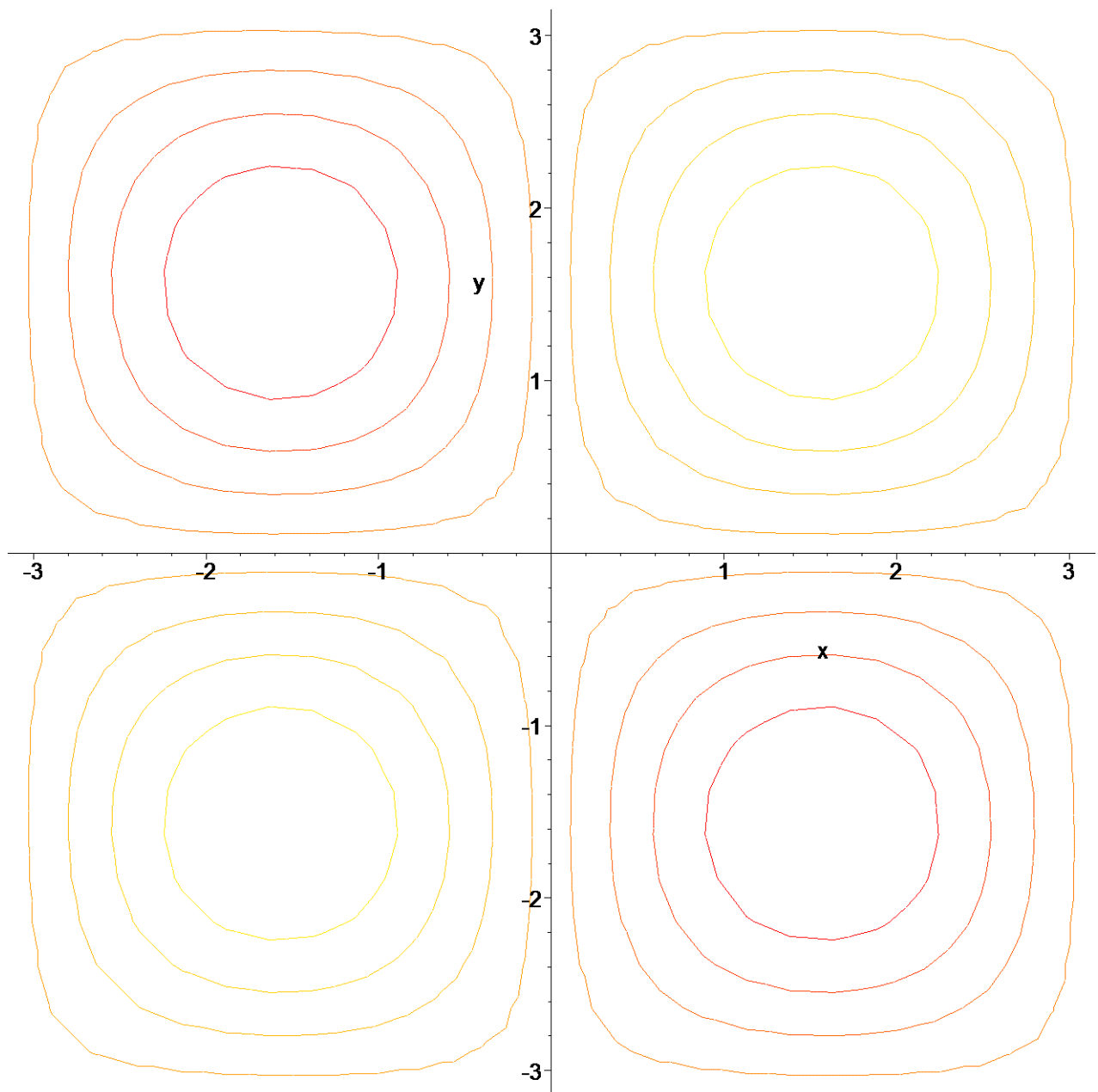
> cylinderplot(z, theta = 0..2*Pi, z=0..1, #
  zadání plochy
  title = `Ukázka použití parametrů pro vykreslování`, #
  titulek obrázku
  titlefont = [TIMES,BOLD,16], #
  formát titulku
  labels = [`osa x`,`osa y`,`osa z`], #
  pojmenování os
  labelfont = [COURIER,OBLIQUE,12], #
  styl popisu os
  axes = framed, #
  styl zobrazení os
  color = theta #
  barevné odlišení parametru
  );

```

Ukázka použití parametrů pro vykreslování

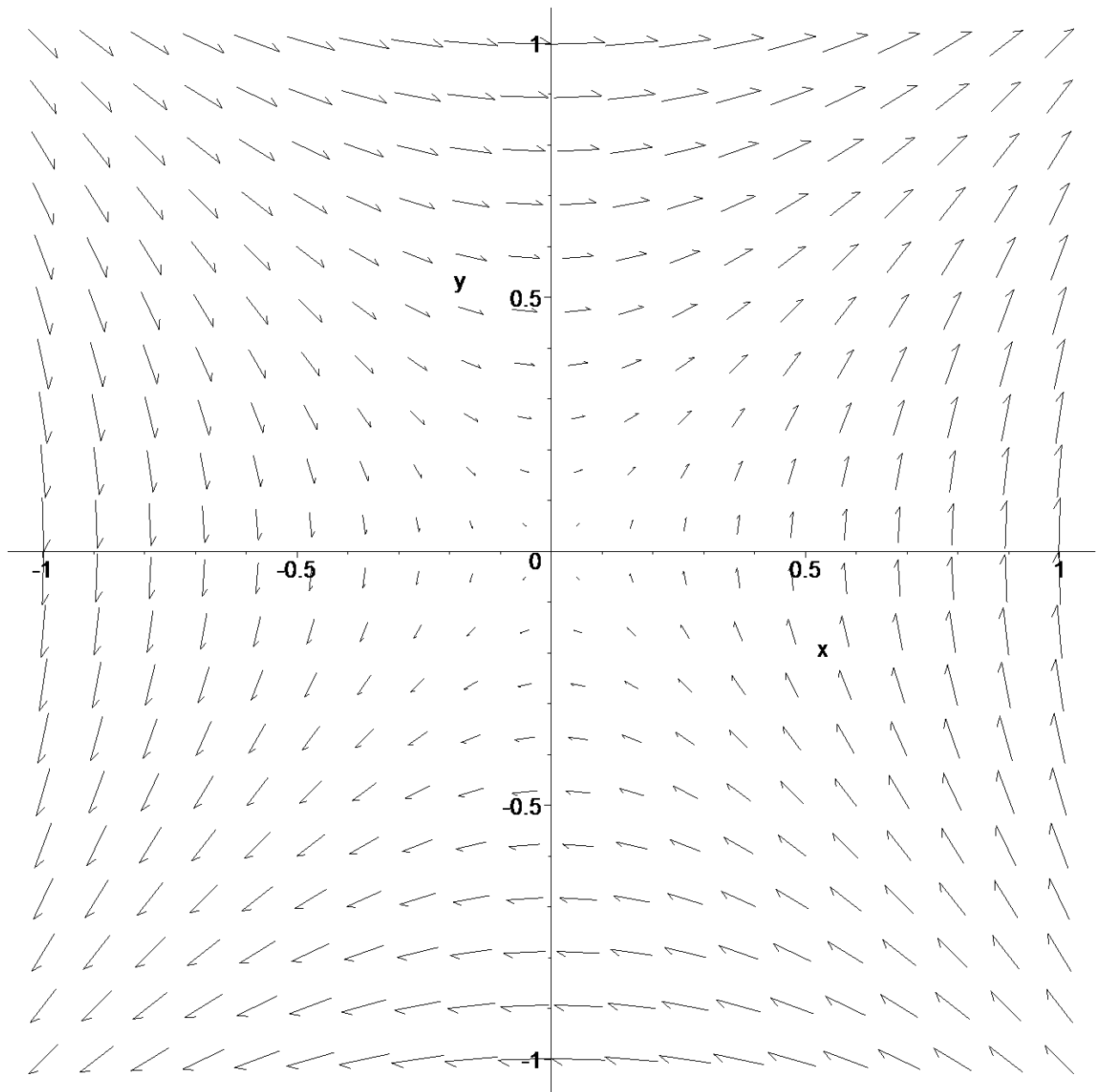


- `contourplot` , popř. `contourplot3d` - topografická mapa ve 2D, popř. 3D
- > `contourplot(sin(x)*sin(y), x=-Pi..Pi, y=-Pi..Pi);`



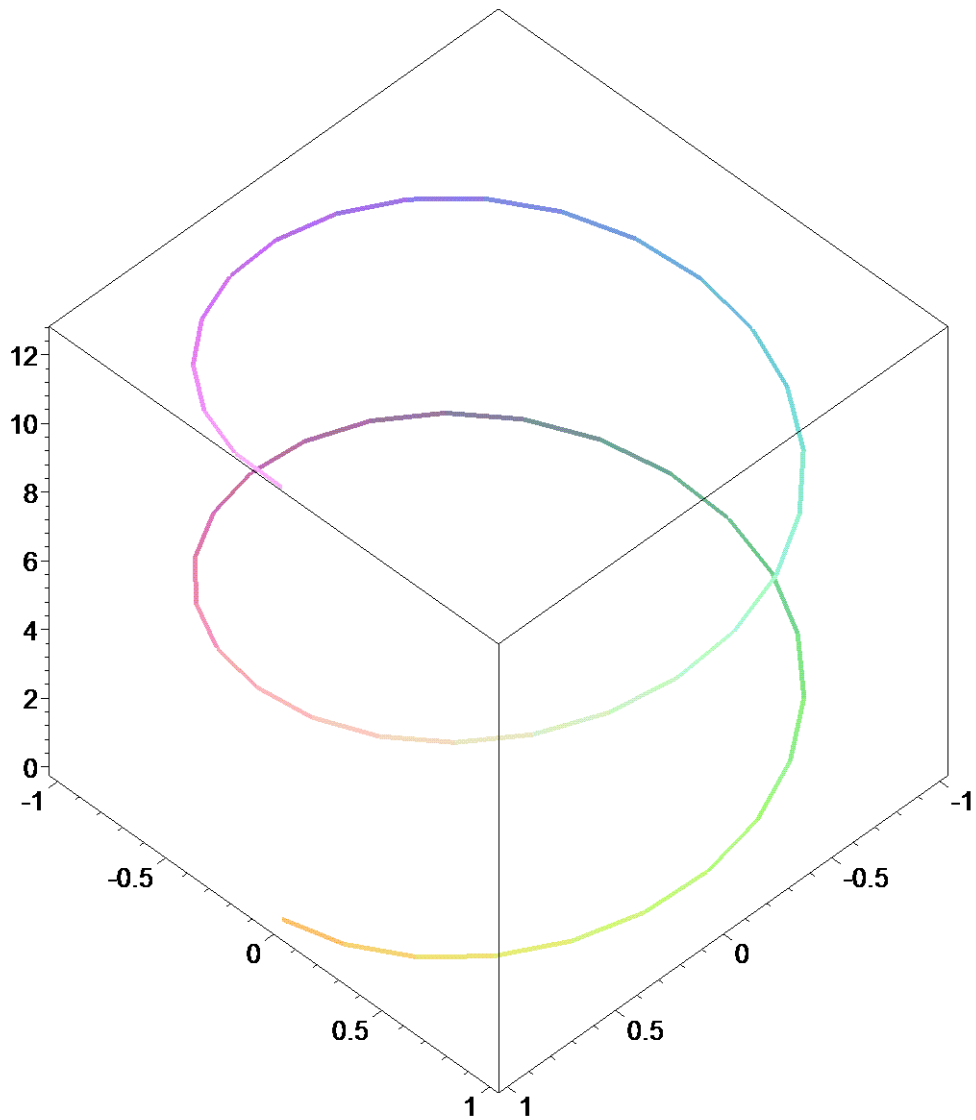
- ***fieldplot***, popř. ***fieldplot3d*** - vektorové pole ve 2D, popř. 3D (použití v kapitole Křivkový a plošný integrál)

> **`fieldplot([y*cos(x), x*cos(y)], x=-1..1, y=-1..1);`**

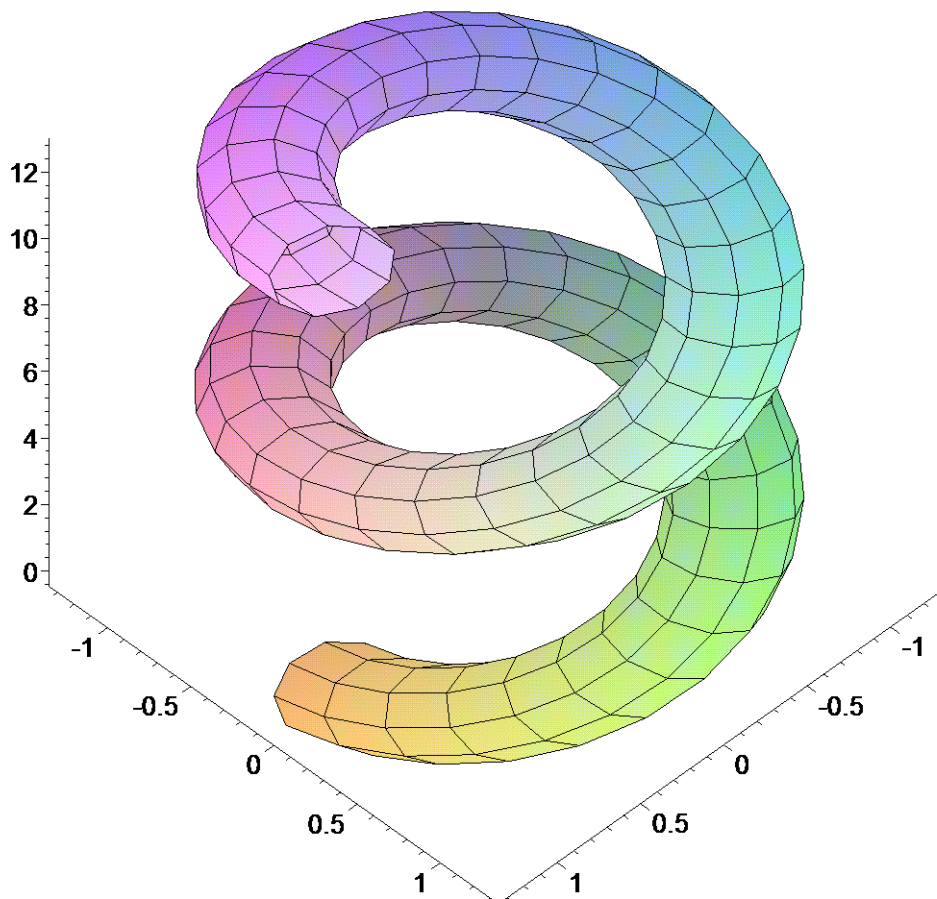


• *spacecurve* - vykreslení prostorových křivek

> `spacecurve([cos(t), sin(t), t], t=0..4*Pi, axes=boxed, thickness=5);`

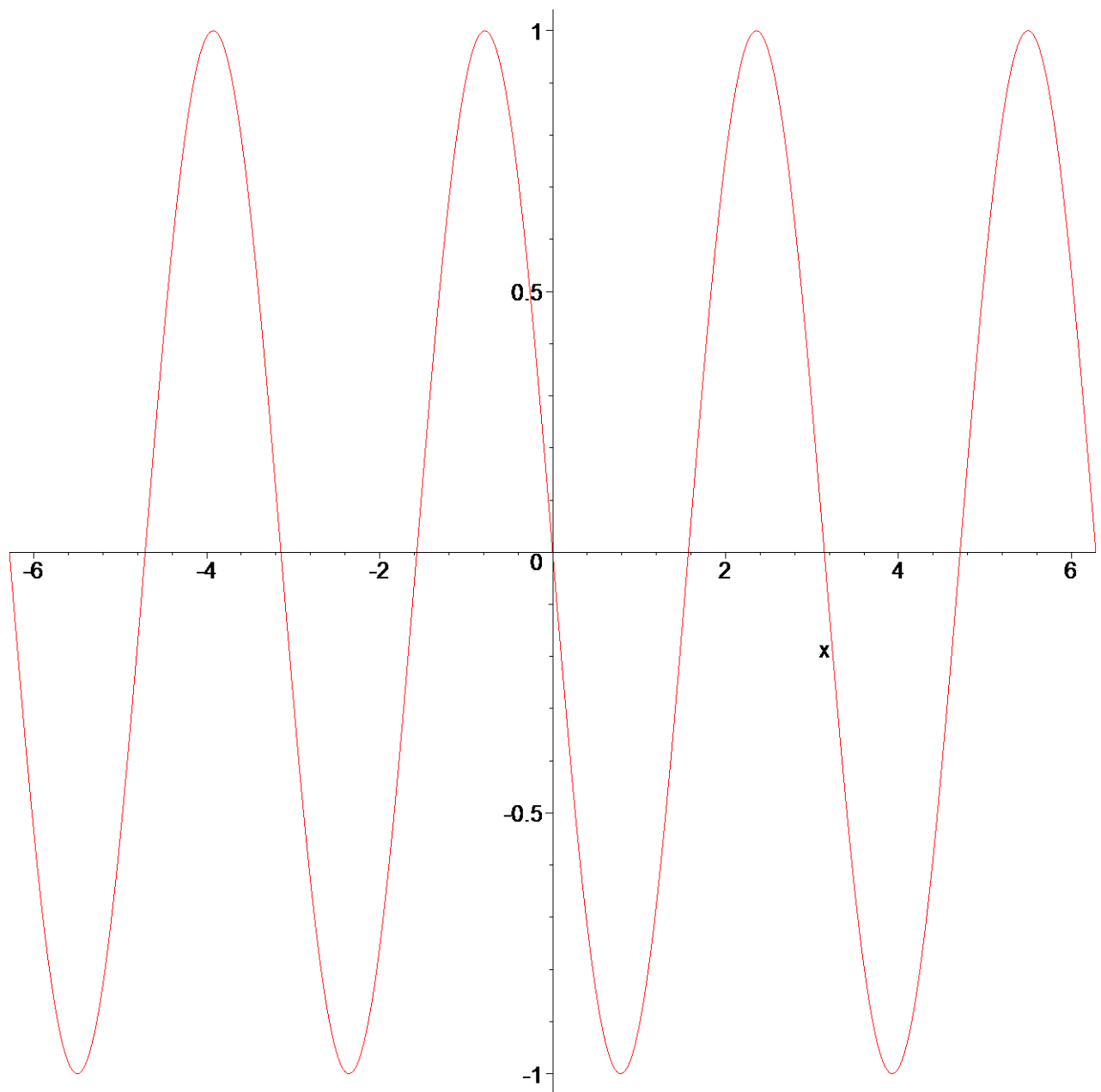


- *tubeplot* - vykreslení křivky v prostoru jako trubici o daném poloměru
- ```
> tubeplot([cos(t), sin(t), t], t=0..4*Pi, radius=0.3, axes=framed);
```

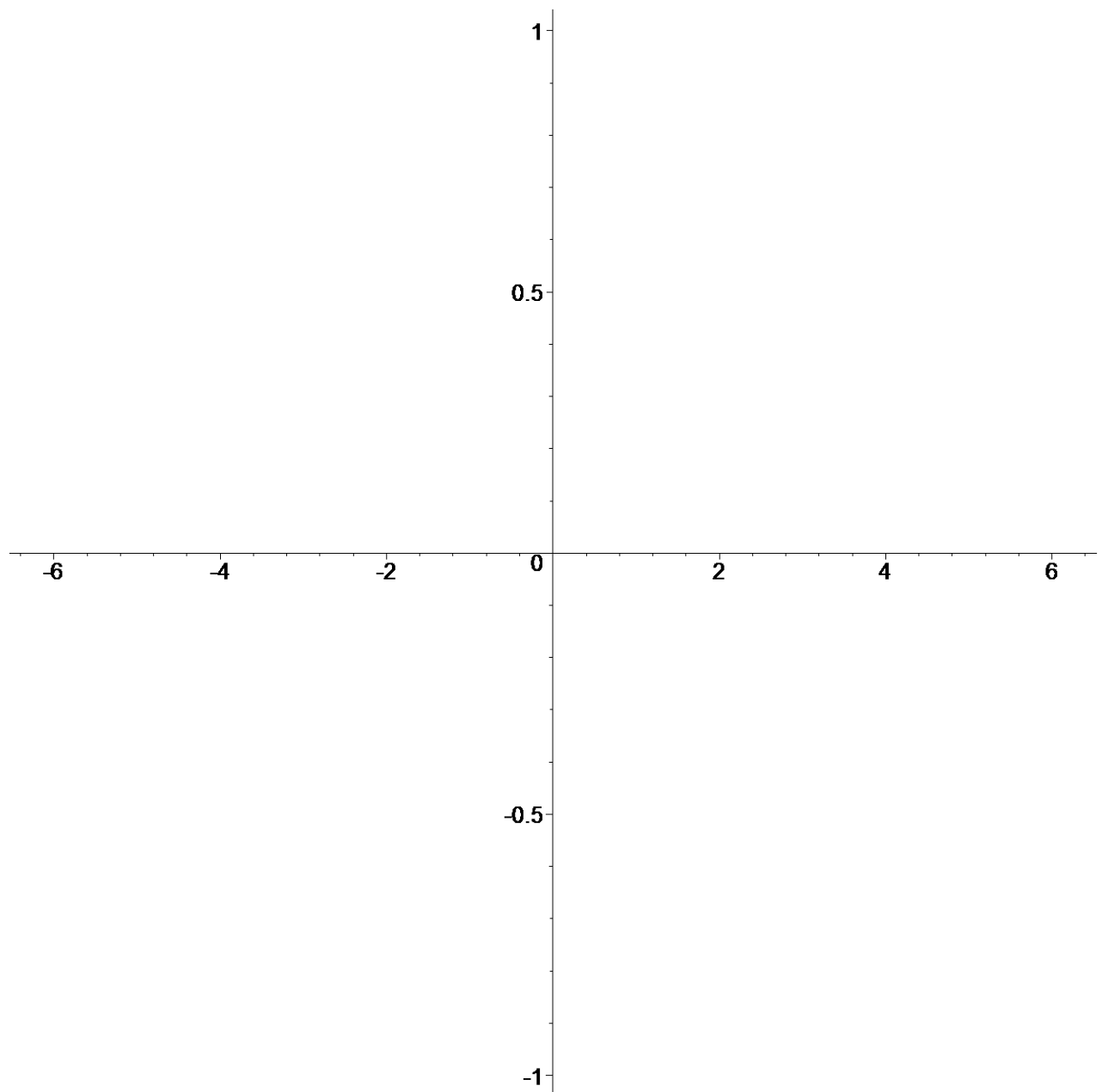


Grafy poměrně dobře zobrazují informace o daných funkcích, ale tato zobrazení jsou statická a nevystihují vše přesně, např. deformace, inflexní body, apod. Systém Maple nabízí řešení tohoto problému pomocí animací. K dispozici jsou příkazy pro jejich vytvoření, *animate* a *animate3d*, *animatecurve*. Animace získáme vykreslováním grafu funkce dvou nezávislých proměnných, kdy jednu z nich považujeme za čas. Tuto proměnnou zafixujeme a volíme časový krok. Vykreslíme graf funkce jedné proměnné s daným pevným časem. Animace tedy vznikne složením jednotlivých obrázků, tzv. framů (*frames*). Po vytvoření animace je ještě třeba spustit ji pomocí položky *Play* z *Animation menu*, která se zobrazí po kliknutí na obrázek animace, popř. vybráním z kontextového menu.

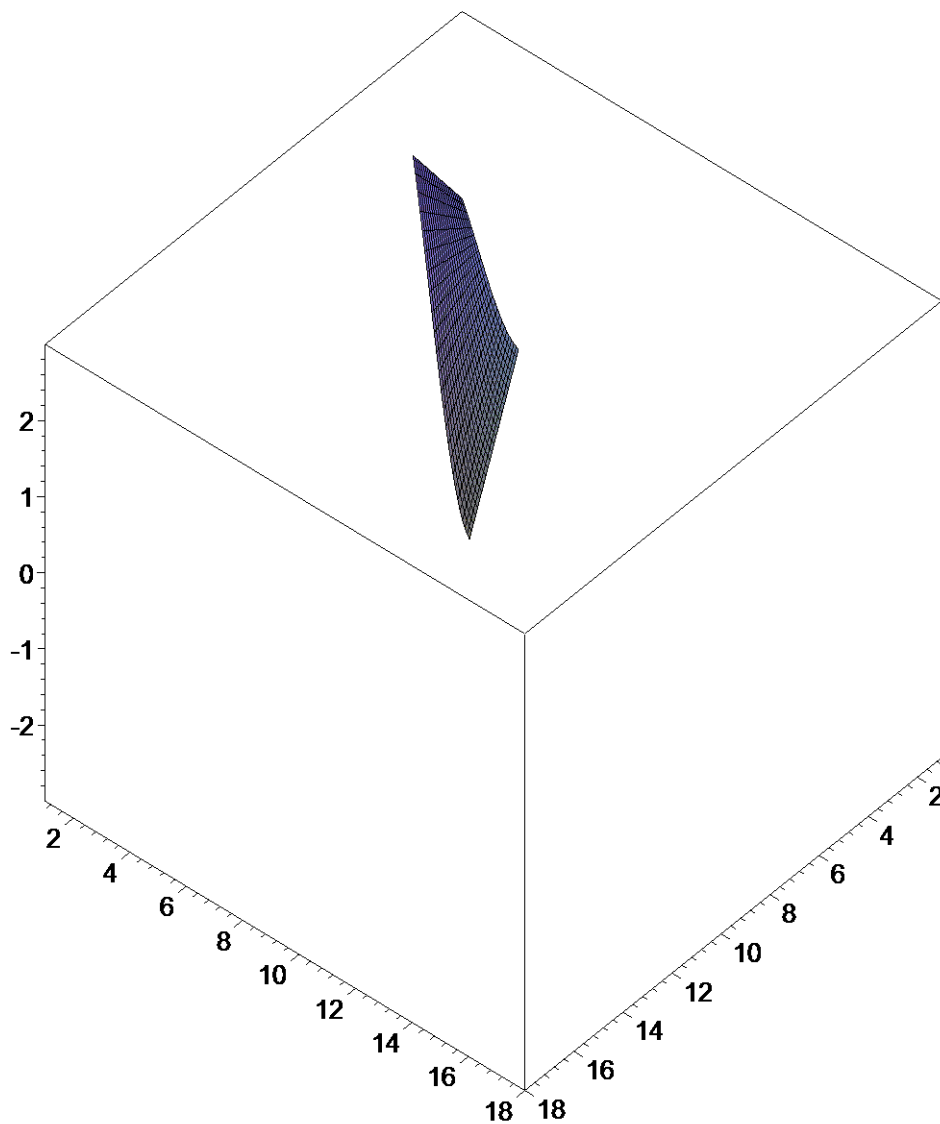
```
> animate(sin(x*t), x=-2*Pi..2*Pi, t=-2..2, frames=18, numpoints=500);
```



```
> animatecurve(sin(x), x=-2*Pi..2*Pi, frames=20);
```

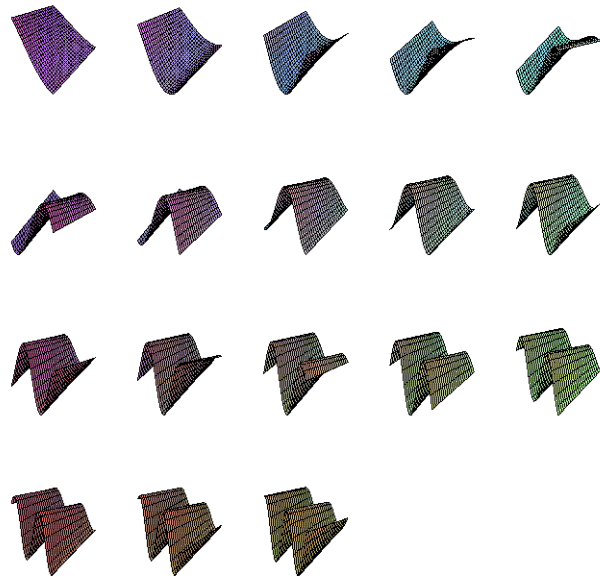


```
> animate3d([u*t,v*t,u*cos(v*t)],u=1..3,v=1..3,t=1..6,style=patch,
axes=boxed,frames=8);
```



Pro zobrazení všech snímků animace se použije následující konstrukce.

- > `p:=animate3d([u*t,v*t,u*cos(v*t)],u=1..3,v=1..3,t=1..6,style=patchnograd,axes=boxed,frames=18) :`
- > `display(op(1,p)) ;`



>



## 4. Diferenciální počet funkcí jedné a více reálných proměnných

V této kapitole se postupně seznámíme s příkazy *limit*, *diff*, *series*, *taylor*, *mtaylor*, *extrema*, *minimize*, *maximize*, *implicitdiff* a dalšími.

### 4.1 Limita funkce jedné reálné proměnné

Limita funkce je jedna ze základních a nejdůležitějších vlastností funkce.

Příkaz, kterým se počítá limita funkce  $f(x)$  v bodě  $x=x_0$  má tvar

***limit(f(x),x=x0)***

Pro matematický zápis limity se užívá příkazu ***Limit*** se stejnými parametry.

Pro výpočty limity zleva či zprava se užívají následující konstrukce

***limit(f(x),x=x0,left)*** - limita zleva

***limit(f(x),x=x0,right)*** - limita zprava

Místo funkce  $f(x)$  lze přímo použít výraz pro výpočet její hodnoty.

V případě, že systém Maple neumí spočítat hodnotu limity, přejde na nový *prompt*.

Hodnota  $x_0$ , ke kterému se blíží proměnná  $x$ , může být libovolné reálné číslo, výraz,  $+\infty$  nebo  $-\infty$ . Pokud je  $x_0 = \pm\infty$ , jde o limitu v nevlastním bodě. Samozřejmě se zde počítá limita zleva popř. zprava. Výsledná hodnota může být reálné číslo, symbol,  $\pm\infty$  nebo nemusí být definována.

Při použití příkazu *limit* však může být výsledkem výpočtu i interval. Znamená to, že funkce nemá v daném bodě limitu nebo ji nelze určit, ale je známo, ve kterém intervalu leží funkční hodnoty pro  $x$  blízkí se k  $x_0$ .

> ***Limit((x^3+7)^(1/3)-sqrt(x+3))/(x-1),x=1):***  
***%=value(%) ;***

$$\lim_{x \rightarrow 1} \frac{(x^3 + 7)^{(1/3)} - \sqrt{x + 3}}{x - 1} = 0$$

> ***Limit(ln(2+exp(3\*x))/ln(3+exp(2\*x)),x=infinity):***  
***%=value(%) ;***

$$\lim_{x \rightarrow \infty} \frac{\ln(2 + e^{(3x)})}{\ln(3 + e^{(2x)})} = \frac{3}{2}$$

> ***Limit((x-1)/(x+1))^x,x=infinity):***  
***%=value(%) ;***

$$\lim_{x \rightarrow \infty} \left( \frac{x-1}{x+1} \right)^x = e^{(-2)}$$

> ***Limit(cos(1/x)^2,x=0):***  
***%=value(%) ; # limita neexistuje, funkce osciluje***

$$\lim_{x \rightarrow 0} \cos\left(\frac{1}{x}\right)^2 = 0 .. 1$$

K důkazu neexistence limity můžeme použít jednostranných limit

> **Limit (tan (x) , x=Pi/2 , left) :**  
**%=value (%) ;**

$$\lim_{x \rightarrow \left(\frac{\pi}{2}\right)^-} \tan(x) = \infty$$

> **Limit (tan (x) , x=Pi/2 , right) :**  
**%=value (%) ;**

$$\lim_{x \rightarrow \left(\frac{\pi}{2}\right)^+} \tan(x) = -\infty$$

> **Limit (tan (x) , x=Pi/2) :**  
**%=value (%) ;**

$$\lim_{x \rightarrow \left(\frac{\pi}{2}\right)} \tan(x) = \text{undefined}$$

> **f:=x->sin(1/x)/x;**

**# definice funkce**

$$f := x \rightarrow \frac{\sin\left(\frac{1}{x}\right)}{x}$$

> **Limit (f (x) , x=0) :**  
**%=value (%) ;**

$$\lim_{x \rightarrow 0} \frac{\sin\left(\frac{1}{x}\right)}{x} = \text{undefined}$$

>

>

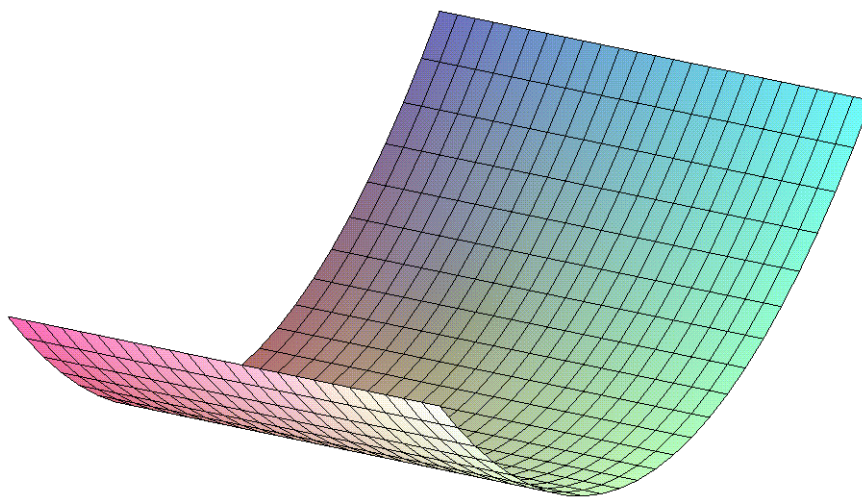
## 4.2 Limita funkce více reálných proměnných

Nejprve zopakujme práci s funkcemi více proměnných v systému Maple.

```
> f := (x, y) -> x^2 + y; # definice funkce dvou proměnných
```

$$f := (x, y) \rightarrow x^2 + y$$

```
> plot3d(f, -2..2, -2..2); # nakreslení grafu
```



```
> f1 := (x, y, z) -> 1/sqrt(1-x^2-y^2-z^2);
```

$$f1 := (x, y, z) \rightarrow \frac{1}{\sqrt{1-x^2-y^2-z^2}}$$

```
> singular(f1(x, y, z), {x, y, z}); # body nespojitosti, více viz. spojitost funkce
```

$$\{y = y, z = z, x = -\sqrt{1-y^2-z^2}\}, \{x = \sqrt{1-y^2-z^2}, y = y, z = z\}$$

Výpočet limity funkce více proměnných je obecně velmi problematická. Výpočet limity funkce

dvou proměnných se provádí pomocí příkazu

***limit(f(x,y),{x=x0,y=y0})***

Vše, co bylo v předchozím odstavci řečeno o příkazu *limit*, platí i zde.

Pro výpočet dvojnásobné limity uijeme konstrukce

***limit(limit(f(x,y),x=x0),y=y0)***

> **Limit((x^3+y^3)/(x^2+y^2),{x=1,y=2}):**

**%=value(%); # Maple limitu spočítá**

$$\text{Limit}\left(\frac{x^3+y^3}{x^2+y^2}, \{x=1, y=2\}\right) = \frac{9}{5}$$

> **Limit((x^3+y^3)/(x^2+y^2),{x=0,y=0}):**

**%=value(%); # Maple neumí limitu spočítat**

$$\text{Limit}\left(\frac{x^3+y^3}{x^2+y^2}, \{y=0, x=0\}\right) = \text{limit}\left(\frac{x^3+y^3}{x^2+y^2}, \{y=0, x=0\}\right)$$

Následující příklad ukazuje využití dvojnásobných limit při důkazu neexistence limity

> **Limit((x^2-y^2)/(x^2+y^2),{x=0,y=0}):**

**%=value(%);**

$$\text{Limit}\left(\frac{x^2-y^2}{x^2+y^2}, \{y=0, x=0\}\right) = \text{undefined}$$

> **Limit(Limit((x^2-y^2)/(x^2+y^2),y=0),x=0):**

**%=value(%);**

$$\lim_{x \rightarrow 0} \lim_{y \rightarrow 0} \frac{x^2-y^2}{x^2+y^2} = 1$$

> **Limit(Limit((x^2-y^2)/(x^2+y^2),x=0),y=0):**

**%=value(%);**

$$\lim_{y \rightarrow 0} \lim_{x \rightarrow 0} \frac{x^2-y^2}{x^2+y^2} = -1$$

Další příklad ukazuje, že rovnost dvojnásobných limit nezaručuje existenci limity

> **limit(x\*y/(x^2-y^2),{x=0,y=0});**

$$\text{limit}\left(\frac{x y}{x^2-y^2}, \{x=0, y=0\}\right)$$

> **limit(limit(x\*y/(x^2-y^2),x=0),y=0);**

**0**

> **limit(limit(x\*y/(x^2-y^2),y=0),x=0);**

**0**

Nyní se k počátku soustavy souřadnic budeme blížit po přímkách

> **y:=k\*x;**

$$y := k x$$

> **Limit(x\*y/(x^2-y^2),x=0)=limit(x\*y/(x^2-y^2),x=0);**

$$\lim_{x \rightarrow 0} \frac{x^2 k}{x^2 - k^2 x^2} = -\frac{k}{-1 + k^2}$$

Vidíme, že výsledná limita je závislá na volbě konstanty  $k$ , proto limita původní funkce neexistuje.

>

>

### 4.3 Spojitost funkce

V tomto odstavci se budeme zabývat singularitami funkce. Důležitost vyplývá z celého odstavce. Nejprve uveďme definici jednotlivých typů bodů nespojitosti.

#### **Definice:**

Nechť funkce  $f$  je definována v jistém okolí  $O(x_0) - \{x_0\}$  bodu  $x_0$ . V samotném okolí bodu  $x_0$  může a nemusí být definována. Řekneme, že tato funkce má v bodě  $x_0$

- **odstranitelnou singularitu**, jestliže existuje  $\lim_{x \rightarrow x_0} f(x) \neq f(x_0)$ , popř.  $f(x_0)$  není definováno
- **nespojitosť 1. druhu** (typu skoku), jestliže existují vlastní limity  $\lim_{x \rightarrow x_0^+} f(x)$ ,  $\lim_{x \rightarrow x_0^-} f(x)$  a jsou různé
- **nespojitosť 2. druhu**, jestliže alespoň jedna z jednostranných limit neexistuje nebo je nevlastní

Pro hledání bodů nespojitosti lze v Maplu užít několik příkazů

- **discont(fce(x),x)**
  - výsledkem je množina bodů v nichž může nespojitost nastat
- **fdiscont(fce(x), x=interval, přesnost, použitá metoda )**
  - numerické hledání nespojitostí funkce, má velká omezení -více v HELPU
- **iscont( f(x), x=a..b, 'interval' )**
  - tento příkaz vypíše *true*, v případě, že v daném intervalu nenalezl žádný "podezřelý" bod v opačném případě vypíše *false*. Třetí parametr je nepovinný, nabývá hodnoty *open* / *closed*, pro otevřený / uzavřený interval, implicitně *open*.

V případě, hledání singulárních bodů funkcí více proměnných nemůžeme užít ani jeden z předchozích příkazů. Je nutné použít některý z příkazů

- **singular( f(x) , x )**  
**singular( f(x,y), {x,y} )**  
**singular( f(x,y,z), {x,y,z} )**
  - je založen na výpočtu pomocí příkazu *solve*, tj. má určité omezení (viz. Příklad 1)

Místo funkce  $f(x)$  lze použít i výraz pro výpočet hodnoty této funkce.

Podrobnosti a příklady k těmto příkazům naleznete v nápovědě systému Maple.

> **restart:**

#### **Příklad 1**

**Odstranitelná singularita** - problém při řešení rovnice

> **(x-1)^3 / (x^2-1) = 0;**

$$\frac{(x-1)^3}{x^2-1} = 0$$

> **f:=lhs(%);**

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

Řešíme rovnici pomocí

```
> solve({f}, x);
```

```
{x = 1}, {x = 1}
```

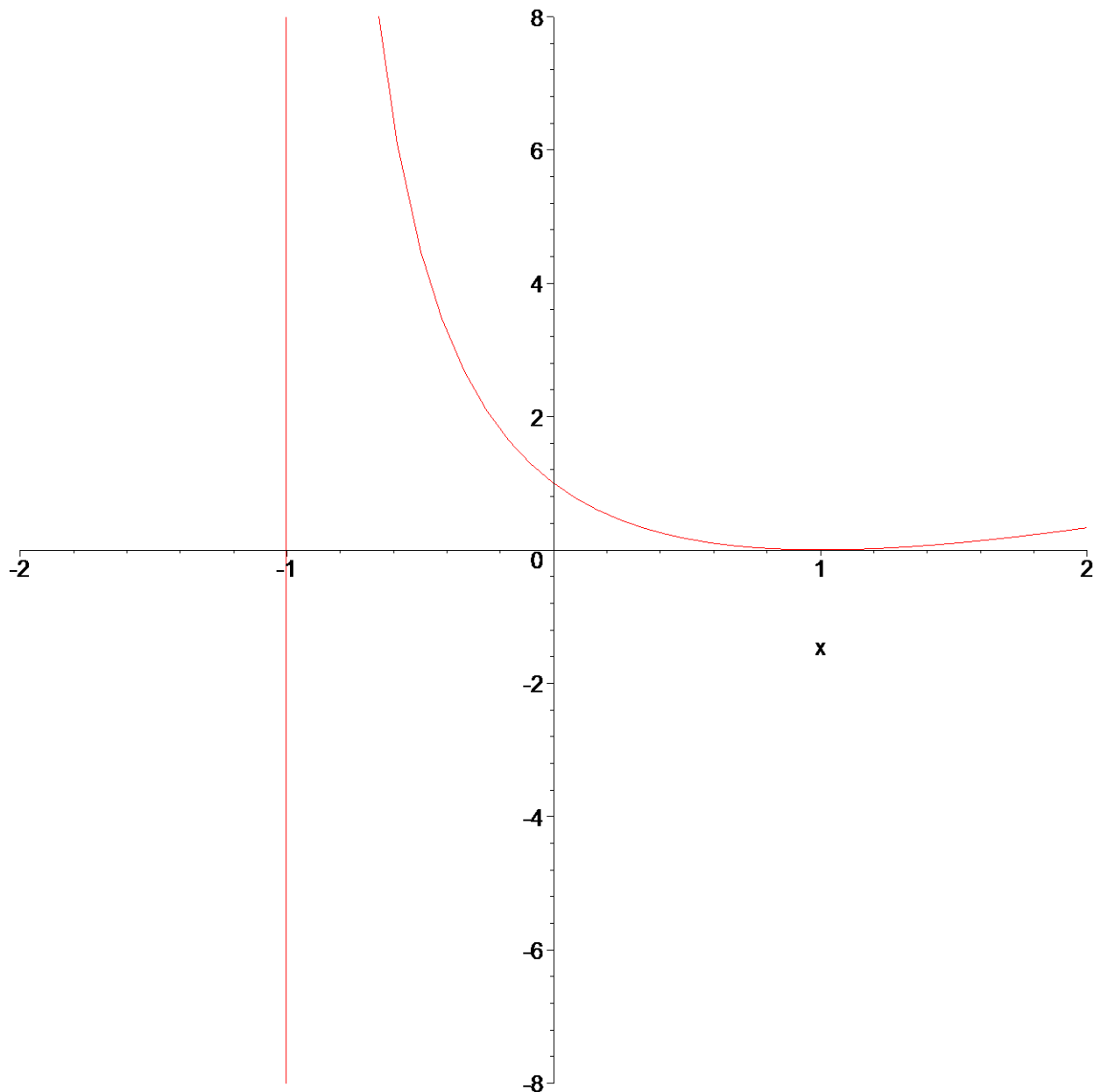
Vidíme, že řešením je bod  $x=1$ , ale pokus o dosažení řešení selže, protože bychom dělili nulou

```
> subs(%, f);
```

```
Error, numeric exception: division by zero
```

Důvodem tohoto chování je odstranitelná singularita v bodě  $x=1$ .

```
> plot(f, x=-2..2, -8..8);
```



Nalezneme singulární body pomocí příkazů

```
> discont(f, x);
```

```
{-1, 1}
```

```
> singular(f);
```

```
{x = -1}, {x = ∞}, {x = -∞}
```

Je vidět, že systém Maple v rámci příkazu *singular* bod  $x=1$  nepovažuje za singulární bod.

Jediným možným vysvětlením je nalezení odstranitelné singularity v tomto bodě. To znamená, že zde existuje vlastní limita, pomocí níž Maple funkci dodefinoval. Právě zde je vidět závislost tohoto příkazu na příkazu *solve*.

> `Limit(f,x=1)=limit(f,x=1);`

$$\lim_{x \rightarrow 1} \frac{(x-1)^3}{x^2-1} = 0$$

Tento příklad ukazuje důležitost znalosti singulárních bodů při práci s Maplem a hlavně uvádí pádny důvod pro provedení zkoušky dosazením při řešení rovnice pomocí příkazu *solve*.

>

### **Příklad 2**

*Nespojitost prvního druhu* ( typ skoku )

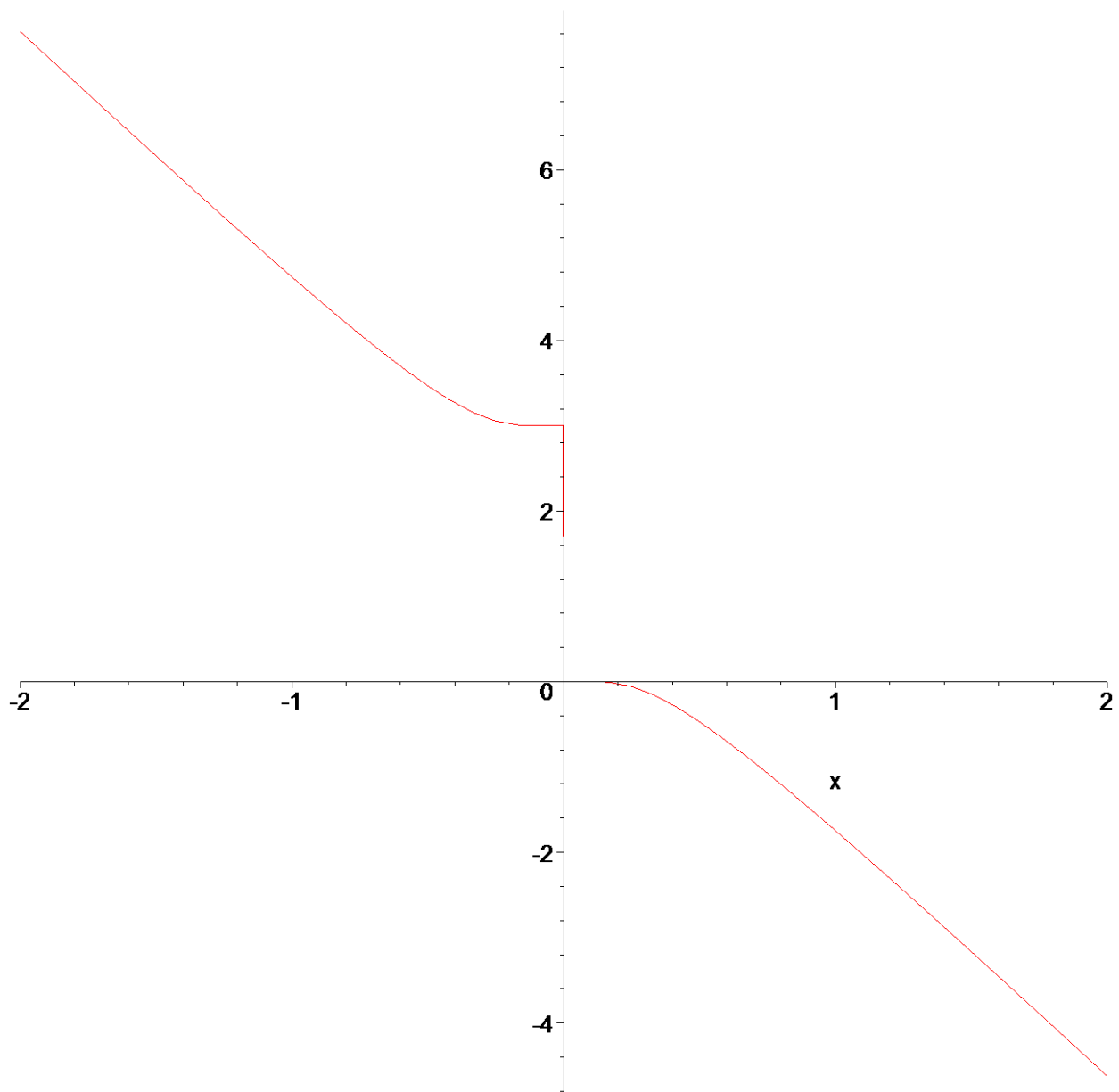
> `f1:=3/(1-exp(1/x));`

$$f1 := \frac{3}{1 - e^{\left(\frac{1}{x}\right)}}$$

Nakresleme obrázek

> `plot(f1,x=-2..2);`





Z obrázku je vidět singularita v bodě  $x=0$ .

Nalezněme body nespojitosti

> `discont(f1, x) ;`

`{0}`

Dokažme výpočtem příslušných limit, že jde o nespojitost prvního druhu

> `Limit(f1, x=0)=limit(f1, x=0) ;`

$$\lim_{x \rightarrow 0} \frac{3}{1 - e^{\left(\frac{1}{x}\right)}} = \text{undefined}$$

> `Limit(f1, x=0, left)=limit(f1, x=0, left) ;`

$$\lim_{x \rightarrow 0^-} \frac{3}{1 - e^{\left(\frac{1}{x}\right)}} = 3$$

> `Limit(f1, x=0, right)=limit(f1, x=0, right) ;`

$$\lim_{x \rightarrow 0^+} \frac{3}{1 - e^{\left(\frac{1}{x}\right)}} = 0$$

>

>

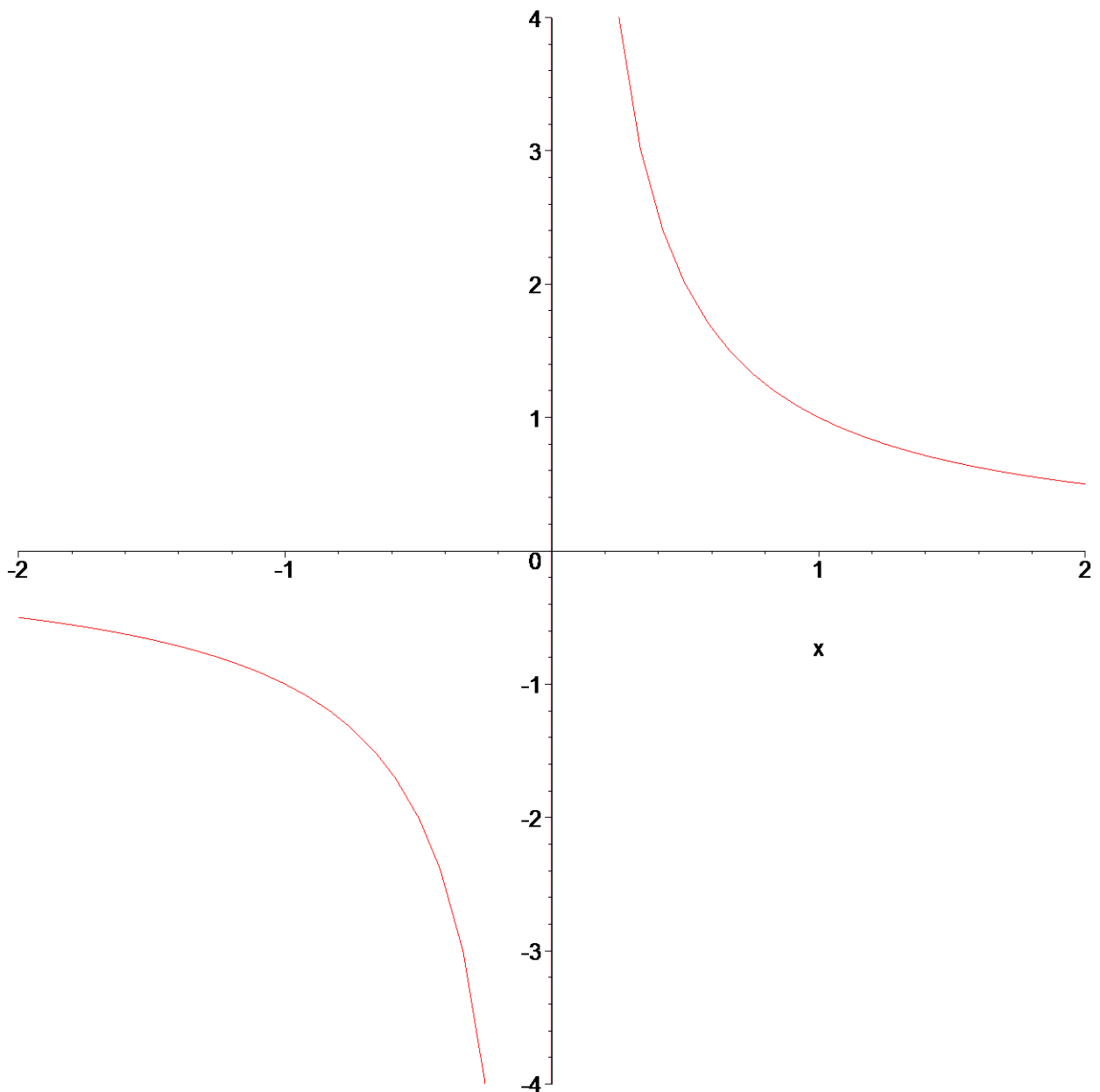
### Příklad 3

*Nespojitost druhého druhu* (tzv. pól)

> **f2:=1/x;**

$$f2 := \frac{1}{x}$$

> **plot(f2,x=-2..2,-4..4);**



Hledejme body nespojitosti

> **discont(f2,x);**

{0}

Spočtěme příslušné limity

> `Limit(f2,x=0)=limit(f2,x=0);`

$$\lim_{x \rightarrow 0} \frac{1}{x} = \text{undefined}$$

> `Limit(f2,x=0,left)=limit(f2,x=0,left);`

$$\lim_{x \rightarrow 0^-} \frac{1}{x} = -\infty$$

> `Limit(f2,x=0,right)=limit(f2,x=0,right);`

$$\lim_{x \rightarrow 0^+} \frac{1}{x} = \infty$$

Z výsledků je patrné, že jde o bod nespojitosti druhého druhu.

>

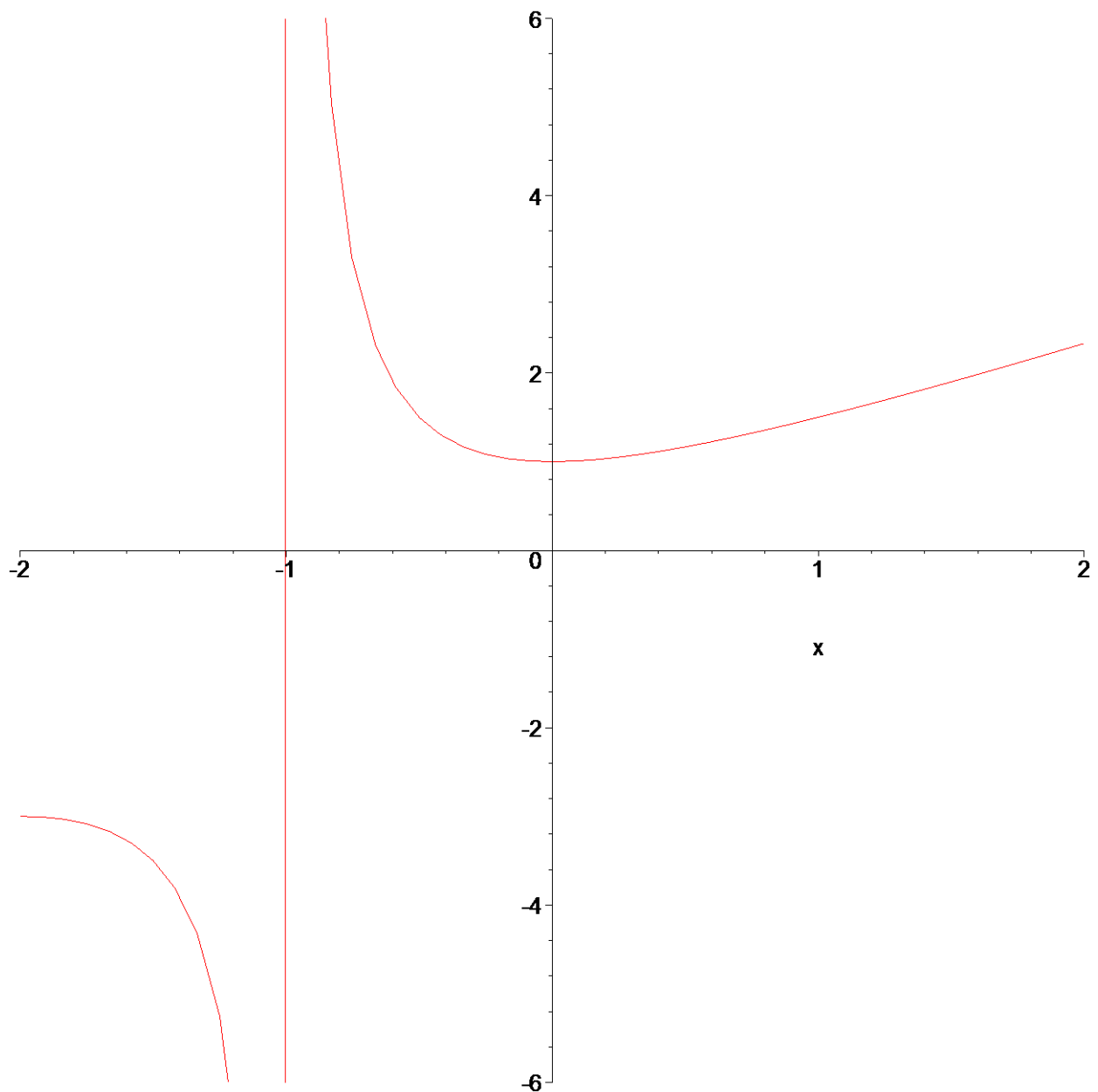
>

#### **Příklad 4**

> `f3:=(x^3-1)/(x^2-1);`

$$f3 := \frac{x^3 - 1}{x^2 - 1}$$

> `plot(f3,x=-2..2,-6..6);`



Body nespojitosti

> `discont(f3, x);`

$\{-1, 1\}$

Vyšetřeme o jaký typ nespojitosti v jednotlivých bodech jde.

Bod  $x = -1$

> `Limit(f3, x=-1) = limit(f3, x=-1);`

$$\lim_{x \rightarrow (-1)} \frac{x^3 - 1}{x^2 - 1} = \text{undefined}$$

> `Limit(f3, x=-1, left) = limit(f3, x=-1, left);`

$$\lim_{x \rightarrow (-1)^-} \frac{x^3 - 1}{x^2 - 1} = -\infty$$

> `Limit(f3, x=-1, right) = limit(f3, x=-1, right);`

$$\lim_{x \rightarrow (-1)^+} \frac{x^3 - 1}{x^2 - 1} = \infty$$

V bodě  $x=-1$  je nespojitost druhého druhu.

Bod  $x=1$

> **Limit(f3,x=1)=limit(f3,x=1);**

$$\lim_{x \rightarrow 1} \frac{x^3 - 1}{x^2 - 1} = \frac{3}{2}$$

> **Limit(f3,x=1,left)=limit(f3,x=1,left);**

$$\lim_{x \rightarrow 1^-} \frac{x^3 - 1}{x^2 - 1} = \frac{3}{2}$$

> **Limit(f3,x=1,right)=limit(f3,x=1,right);**

$$\lim_{x \rightarrow 1^+} \frac{x^3 - 1}{x^2 - 1} = \frac{3}{2}$$

V bodě  $x=1$  je odstranitelná singularita.

>

>

>

## 4.4 Derivace funkce

Dalším základním pojmem diferenciálního počtu je derivace funkce.

K výpočtu první derivace funkce jedné proměnné jsou určeny příkazy

**Diff(f(x), x)** - matematický zápis

**diff(f(x), x)** - výpočet

Místo funkce  $f(x)$  lze v těchto příkazech použít i výraz pro výpočet hodnoty funkce. Pro určení hodnoty derivace funkce zapsané pomocí příkazu *Diff* lze užít příkaz **value**.

Pro výpočet druhé, popř. třetí, derivace funkce  $f(x)$  použijeme tento příkaz ve tvaru

**diff(f(x),x,x)** , popř. **diff(f(x), x,x,x)**.

Vyšší stupeň derivace se tvoří přidáním dalšího parametru do příkazu *diff*.

> **Diff(ln(x<sup>2</sup>-3\*x+5), x) :**

**%=value(%);**

$$\frac{d}{dx} \ln(x^2 - 3x + 5) = \frac{2x - 3}{x^2 - 3x + 5}$$

> **f:=x->ln(x<sup>2</sup>-3\*x+5);**

$$f := x \rightarrow \ln(x^2 - 3x + 5)$$

> **Diff(f(x), x, x) :**

**%=value(%);**

$$\frac{d^2}{dx^2} \ln(x^2 - 3x + 5) = \frac{2}{x^2 - 3x + 5} - \frac{(2x - 3)^2}{(x^2 - 3x + 5)^2}$$

> **Diff(f(x), x\$3) :**

**%=value(%); # užití znaku \$ jako opakovače**

$$\frac{d^3}{dx^3} \ln(x^2 - 3x + 5) = -\frac{6(2x - 3)}{(x^2 - 3x + 5)^2} + \frac{2(2x - 3)^3}{(x^2 - 3x + 5)^3}$$

Příkaz *diff* lze také užít pro výpočet parciální derivace funkce více proměnných

**diff(f(x,y,z), x)** - parciální derivace podle proměnné  $x$

> **Diff(sin(t)\*ln(s), t, s) :**

**%=value(%); # smíšená parciální derivace**

$$\frac{\partial^2}{\partial s \partial t} (\sin(t) \ln(s)) = \frac{\cos(t)}{s}$$

Další možností, jak v Maplu vypočítat derivaci funkce, je operátor derivace  $D$ .

**D(jméno funkce)** - výpočet první derivace

**(D@@n)(jméno funkce)** - výpočet  $n$ -té derivace

Ekvivalentní zápis pomocí *Diff* je **Diff(jméno funkce, proměnná)**.

Pomocí něj lze vypočítat derivace jak standardních funkcí, tak i funkcí definovaných uživatelem. Výsledkem provedení operace je opět funkce.

Operátor  $D$  je také obsažen v rozšiřující knihovně *student*. Proto další příklady užití tohoto

operátoru naleznete v příloze *student package*.

> **restart:**

**D(cos);**

$-\sin$

> **f:=x->cos(x)+7\*x^2-2\*x;**

$f := x \rightarrow \cos(x) + 7x^2 - 2x$

> **df:=D(f);**

$df := x \rightarrow -\sin(x) + 14x - 2$

> **df(0);**

$-2$

> **ddf:=(D@@2)(df);**

$ddf := \sin$

> **dddh:=(D@@3)(h);**  
**zápis třetí derivace**

**# h není definováno, proto symbolický**

$dddh := (D^{(3)})(h)$

> **h:=x->x^4-3\*x^3;**

$h := x \rightarrow x^4 - 3x^3$

> **dddh(h);**

$24h - 18$

Následují příklady, které jsou podrobně vysvětleny v příloze *student package*.

> **D(h);**

**# derivace podle první proměnné**

$x \rightarrow 4x^3 - 9x^2$

> **D[1,1](h);**

**# druhá derivace podle první proměnné**

$x \rightarrow 12x^2 - 18x$

> **f2:=(x,y,z)->x^3\*y^2\*z^2;**

$f2 := (x, y, z) \rightarrow x^3 y^2 z^2$

> **D[1\$3](f2);**  
**proměnné x**

**# parciální derivace třetího řádu podle**

$(x, y, z) \rightarrow 6y^2 z^2$

> **D[1,1,1](f2)(0,1,1);**  
**x v bodě [0,1,1]**

**# parciální derivace třetího řádu podle**

$6$

> **D[1,2,3](f2);**  
**x, y, z**

**# parciální derivace třetího řádu podle**

$(x, y, z) \rightarrow 12x^2 y z$

>

>

## 4.5 Aproximace funkce

Příkazy, pomocí kterých se v systému Maple počítá Taylorův rozvoj funkce  $f$  v okolí bodu  $x = x_0$ , mají tvar

$series(f(x), x=x_0, n)$   
 $taylor(f(x), x=x_0, n)$

Místo funkce  $f(x)$  můžeme přímo uvést výraz pro výpočet hodnoty  $f(x)$ . Stupeň  $n$  je nepovinným parametrem, který je implicitně nastaven na hodnotu 6 v proměnné *Order* (analogie s *Digits*). Dále systém Maple obsahuje příkaz pro potlačení zbytku, protože jinak by nešlo např. vykreslit daný rozvoj, popř. dosadit za proměnnou. Jde o příkaz ***convert(taylor, polynomial)***.

Pro snížení stupně Taylorova polynomu je v systému Maple k dispozici příkaz ***rem(taylor, (x-x0)^n, x)***,

který odstraní všechny členy počínaje členem  $(x - x_0)^n$ .

Pro přímý výpočet Taylorova aproximačního polynomu se užívá příkaz

***mtaylor(f(x), x=x0, n)***,

který se musel v předchozích verzích systému Maple nahrát do paměti pomocí příkazu ***readlib(mtaylor)***. Význam parametrů je shodný s předchozími příkazy.

V případě aproximace funkce dvou či více proměnných se užívají tyto příkazy v následujících tvarech

***series(f(x,y), {x=x0,y=y0}, n)***  
***taylor(f(x,y), {x=x0,y=y0}, n)***  
***mtaylor(f(x,y), {x=x0,y=y0}, n)***

>

Obecný vzorec

> ***series(f(x), x=x0);***

$(\cos(x_0) + 7x_0^2 - 2x_0) + (-\sin(x_0) - 2 + 14x_0)(x - x_0) + \left(7 - \frac{1}{2}\cos(x_0)\right)(x - x_0)^2 +$

$\frac{1}{6}\sin(x_0)(x - x_0)^3 + \frac{1}{24}\cos(x_0)(x - x_0)^4 - \frac{1}{120}\sin(x_0)(x - x_0)^5 + O((x - x_0)^6)$

> ***taylor(f(x), x=x0);***

$f(x_0) + D(f)(x_0)(x - x_0) + \frac{1}{2}(D^{(2)})(f)(x_0)(x - x_0)^2 + \frac{1}{6}(D^{(3)})(f)(x_0)(x - x_0)^3 +$

$\frac{1}{24}(D^{(4)})(f)(x_0)(x - x_0)^4 + \frac{1}{120}(D^{(5)})(f)(x_0)(x - x_0)^5 + O((x - x_0)^6)$

Následující příklad ukazuje, že pomocí *series* získám i MacLaurinovu řadu.

> ***f:=exp(x)/x;***

> ***series(f, x=0);***

$x^{-1} + 1 + \frac{1}{2}x + \frac{1}{6}x^2 + \frac{1}{24}x^3 + \frac{1}{120}x^4 + O(x^5)$

> ***taylor(f, x=0);*** ***#neexistuje taylorova rada***



Error, does not have a taylor expansion, try series()

Použití je následující.

> **t1:=series(cos(x),x=Pi);**

$$t1 := -1 + \frac{1}{2}(x - \pi)^2 - \frac{1}{24}(x - \pi)^4 + O((x - \pi)^6)$$

> **Order;**

**# udává stupeň**

**Digits;**

**# udává počet platných desetinných**

**míst**

6

10

> **t2:=taylor(sin(x),x=0,10);** **# změna stupně aproximace**

$$t2 := x - \frac{1}{6}x^3 + \frac{1}{120}x^5 - \frac{1}{5040}x^7 + \frac{1}{362880}x^9 + O(x^{10})$$

> **plot(t2(x),x=-5..5);**

**# Maple neumí vykreslit kvůli**

**zbytku O**

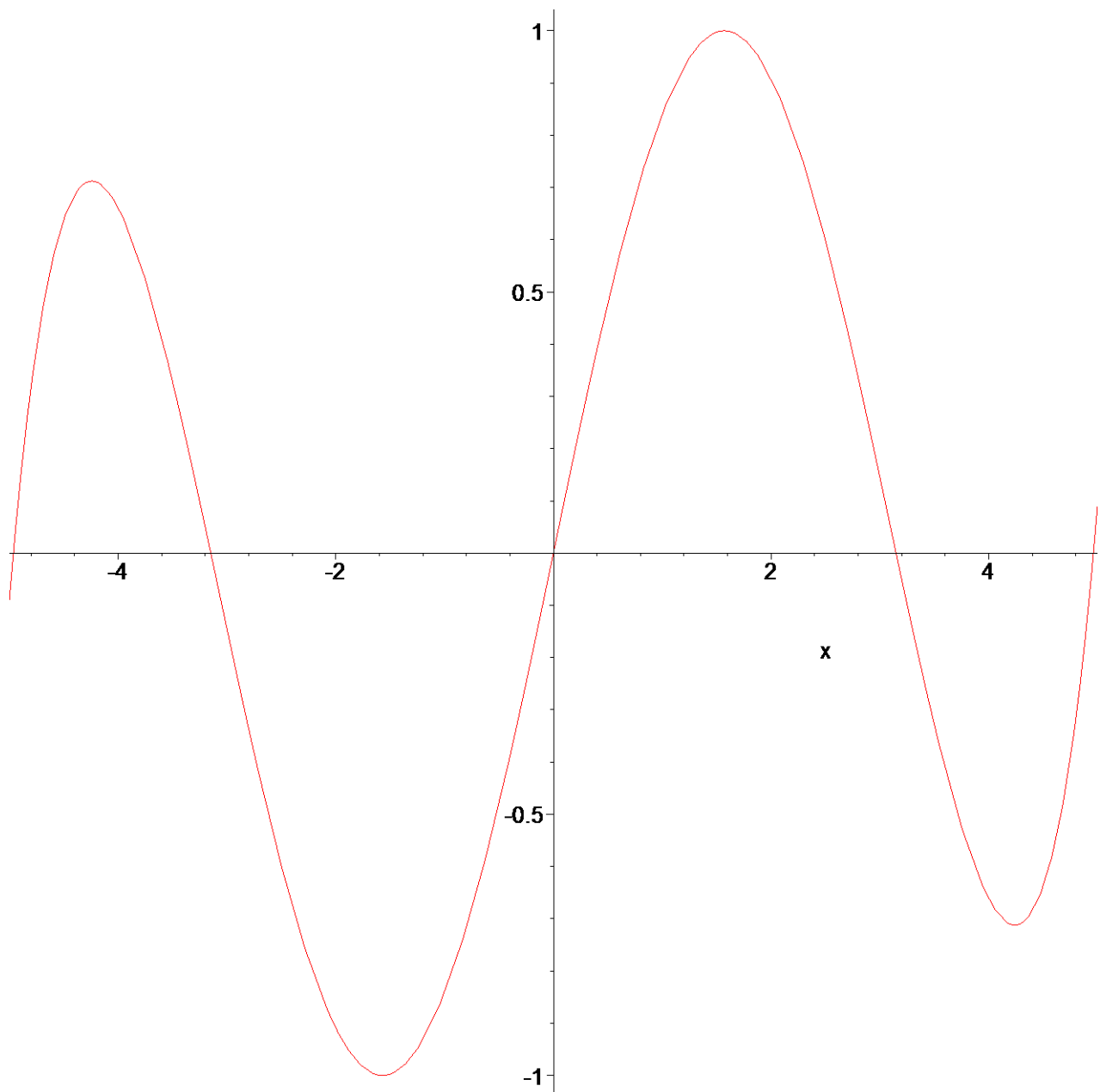
Plotting error, empty plot

> **t3:=convert(t2,polynomial);**

**# odstranění zbytku**

$$t3 := x - \frac{1}{6}x^3 + \frac{1}{120}x^5 - \frac{1}{5040}x^7 + \frac{1}{362880}x^9$$

> **plot(t3(x),x=-5..5);**



```
> t4:=mtaylor(ln(x),x=1,4); # přímý výpočet beze zbytku
```

$$t4 := x - 1 - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3}$$

```
> rem(t4,x-1,x); # odstranění již prvního členu
```

0

```
> rem(t4,(x-1)^2,x); # odstranění všeho od zadaného členu
```

$x - 1$

Funkce více proměnných

```
> f:=mtaylor((x-y)/(x+y),{x=1,y=1},3);
```

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

```
> f:=unapply(%,x,y); # získáme funkci
```

$$f := (x, y) \rightarrow \frac{1}{2}x - \frac{1}{2}y - \frac{1}{4}(x-1)^2 + \frac{1}{4}(y-1)^2$$

>  $f(1+h, 1-k)$  ;  
daném bodě

# výpočet hodnoty funkce v

$$\frac{1}{2}h + \frac{1}{2}k - \frac{1}{4}h^2 + \frac{1}{4}k^2$$

>  
>  
>

## 4.6 Funkce zadané parametrickými rovnicemi

Tento odstavec ukáže možnost vytvoření vlastní procedury. Půjde o příkaz, pomocí kterého spočteme derivaci funkcí daných parametricky až do řádu čtyři včetně.

Nejprve uveďme několik vztahů, pomocí kterých se tyto derivace počítají.

Z věty o derivaci funkce dané parametricky má funkce  $f(x)$  v příslušném intervalu proměnné  $x$

$$\text{derivaci a v odpovídajících si bodech platí } f'(x) = \frac{dy}{dx} = \frac{\frac{dy}{dt}}{\frac{dx}{dt}} = \frac{\psi'(t)}{\phi'(t)} .$$

Pro druhou derivaci platí

$$y'' = \frac{dy'}{dx} = \frac{d \frac{\psi'}{\phi}}{dt \frac{dx}{dt}} = \frac{\psi''(t)\phi'(t) - \psi'(t)\phi''(t)}{\phi'^2(t)} \frac{1}{\phi'(t)} = \frac{\psi''(t)\phi'(t) - \phi''(t)\psi'(t)}{\phi'^3}$$

$$\text{Pro třetí derivaci platí } y''' = \frac{d^3 y}{dx^3} = \frac{\phi'^2 \psi''' - \phi' \psi' \phi''' - 3\phi' \phi'' \psi'' + 3\phi''^2 \psi'}{\phi'^5}$$

Nyní přistupme k vytvoření již zmiňované procedury. Nazveme ji *diffpar*.

|                                                       |               |
|-------------------------------------------------------|---------------|
| <i>diffpar</i> ( [rce1,rce2], prom)                   | - 1.derivace  |
| <i>diffpar</i> ( [rce1,rce2], prom, prom)             | - 2. derivace |
| <i>diffpar</i> ( [rce1,rce2], prom, prom, prom)       | - 3. derivace |
| <i>diffpar</i> ( [rce1,rce2], prom, prom, prom, prom) | - 4.derivace  |

### Procedura *diffpar*

Tato procedura byla z důvodu přehlednosti napsána pomocí základních příkazů programování v systému Maple. Lze ji upravit tak, aby se chovala jako procedura, která je součástí programu. Jde především o typovou kontrolu. Pro pochopení je však nutné přečíst přílohu *Programování - jazyk MPJ*.

```
> diffpar:=proc(rce,prom)
 local vysl,x,y,dx,dy,dxx,dyy,dxxx,dyyy,dxxxx,dyyyy;

 if is(type(rce,'list')) and (nops(rce)=2) then
 # spravny parametr
 else
 return `Chyba - nespravny pocet rovnic`;
 end if;

 x:=rce[1]; # získání zadaných rovnic
 y:=rce[2];
 dx:=diff(x,prom); # výpočet derivace
 dy:=diff(y,prom);

 if nargs=2 then
 lprint(`1.derivace`);
 vysl:=dy/dx ;
```

```

elif nargs=3 then
 lprint(`2.derivace`);
 dxx:=diff(dx,prom);
 dyy:=diff(dy,prom);
 vysl:=simplify((dyy*dx-dxx*dy)/(dx)^3);

elif nargs=4 then
 lprint(`3.derivace`);
 dxx:=diff(dx,prom);
 dyy:=diff(dy,prom);
 dxxx:=diff(dxx,prom);
 dyyy:=diff(dyy,prom);

vysl:=simplify((dx^2*dyyy-dx*dy*dxxx-3*dx*dxx*dyy+3*dxx^2*dy)/(d
x^5));

elif nargs=5 then
 lprint(`4.derivace`);
 dxx:=diff(dx,prom);
 dyy:=diff(dy,prom);
 dxxx:=diff(dxx,prom);
 dyyy:=diff(dyy,prom);
 dxxxx:=diff(dxxx,prom);
 dyyyy:=diff(dyyy,prom);

vysl:=simplify(1/dx^5*(-6*dx^2*dxx*dyyy+dx^3*dyyyy-4*dx^2*dyy*dxx
-dx^2*dy*dxxxx+10*dx*dxx*dxxx*dy+15*dx*dxx^2*dyy-15*dxx^3*dy));

elif nargs>5 then
 return `Tato procedura pocita maximalne 4.derivaci`;
end if;

return vysl;
end proc:

```

Ukázka použití

```

> diffpar([ln(t),1/(1-t)],t);
`1.derivace`

```

$$\frac{t}{(1-t)^2}$$

```

> diffpar([ln(t),1/(1-t),0],t);

```

*Chyba - nespravny pocet rovnic*

```

> diffpar([ln(t),1/(1-t)],t,t);
`2.derivace`

```

$$-\frac{(t+1)t}{(-1+t)^3}$$

```
> diffpar([ln(t), 1/(1-t)], t, t, t);
`3.derivace`
```

$$\frac{(t^2+1+4t)t}{(-1+t)^4}$$

```
> diffpar([ln(t), 1/(1-t)], t$4); # je možné užit operátoru $
jako opakovače
`4.derivace`
```

$$-\frac{11t^2+t^3+11t+1}{t(-1+t)^5}$$

```
> diffpar([ln(t), 1/(1-t)], t, t, t, t, t);
```

*Tato procedura pocita maximalne 4.derivaci*

```
>
>
```

## 4.7 Extrémy funkce

Systém Maple obsahuje příkazy pro výpočet extrémů jak volných tak i vázaných rovnicemi (nerovnicemi) a může velmi pomoci při řešení úloh hledání extrémů funkcí jedné nebo více proměnných.

Pro výpočet minimální hodnoty funkce jedné či více proměnných je určen příkaz *minimize* a pro nalezení maximální hodnoty příkaz *maximize*. Počítáme-li volné globální minimum (maximum) funkce, zadáme pouze funkci, tzn. příkazy mají jeden parametr. V případě omezení pomocí rovnic zadáme tato omezení jako další parametr. Pomocí intervalů lze také specifikovat oblast, ve které extrém hledáme. Oba příkazy vypíší hodnotu minima (maxima) dané funkce. Pomocí parametru *location* bude výpis rozšířen o souřadnice bodů, v nichž extrém nastává.

Následující příklady ukáží použití těchto příkazů

> **f:=x\*y\*exp(-(x^2+y^2)/2);**

$$f := x y e^{\left(-\frac{x^2}{2} - \frac{y^2}{2}\right)}$$

> **maximize(%, location);**

$$e^{(-1)}, \{[\{x = 1, y = 1\}, e^{(-1)}], [\{x = -1, y = -1\}, e^{(-1)}]\}$$

> **minimize(%%, location);**

$$-e^{(-1)}, \{[\{x = 1, y = -1\}, -e^{(-1)}], [\{y = 1, x = -1\}, -e^{(-1)}]\}$$

Na dalším příkladu ukážeme specifikaci intervalů

> **minimize(x^2-3\*x+y^2\*x+3, location);**

$$-\infty, \{[\{x = \text{INTERVAL}(-\infty .. 0), y = -\infty\}, -\infty], [\{x = \text{INTERVAL}(-\infty .. 0), y = \infty\}, -\infty]\}$$

> **minimize(x^2-3\*x+y^2\*x+3, x=0..2, y=0..2, location);**

$$\frac{3}{4}, \left\{ \left[ \left\{ x = \frac{3}{2}, y = 0 \right\}, \frac{3}{4} \right] \right\}$$

Další příklad ukáže omezení těchto příkazů. Maple neumí pomocí těchto příkazů nalézt extrém funkce na dané množině  $M$ .

> **f1:=x^2-y^2+4;**

$$f1 := x^2 - y^2 + 4$$

> **M:=x^2+y^2=1;**

$$M := x^2 + y^2 = 1$$

> **maximize(f1, M);**

$$\text{maximize}(x^2 - y^2 + 4, x^2 + y^2 = 1)$$

> **minimize(f1, M);**

$$\text{minimize}(x^2 - y^2 + 4, x^2 + y^2 = 1)$$

Proto je v Maple k dispozici další příkaz určený pro hledání extrémů funkcí jedné či více proměnných. Jde o příkaz *extrema*, který hledá extrémy metodou Lagrangeových multiplikátorů. *extrema(výraz, {omezení}, {proměnné}, 'souřadnice')*

Poslední parametr je nepovinný, ale specifikuje řetězec, do kterého budou uloženy souřadnice bodů, v nichž nastává maximum a minimum.

Tento příkaz nerozlišuje mezi minimy a maximy, hledá oba extrémy. Díky specifikaci proměnných

můžeme tento příkaz užít také při hledání extrémů funkce s parametry, kdy pro některou hodnotu parametru nabývá funkce maxima a pro jinou minima. V případě hledání volného extrému, tj. nejsou žádné omezující podmínky, je nutné zadat druhý parametr v podobě prázdné množiny. Použití tohoto příkazu ukážeme na již zmíněném příkladu. Hledáme extrémy funkce

> **f1;**

$$x^2 - y^2 + 4$$

na množině

> **M;**

$$x^2 + y^2 = 1$$

> **extrema (f1, {M}, {x, y}, 'body');**

$$\{3, 5\}$$

> **body;**

$$\{\{x = 0, y = 1\}, \{x = 0, y = -1\}, \{y = 0, x = -1\}, \{y = 0, x = 1\}\}$$

Pro úplnost uveďme ještě případ, kdy je extrém vázán omezujícími podmínka ve tvaru nerovnic. V tomto případě systém Maple poskytuje podporu pro lineární funkce více proměnných a pro omezení ve tvaru nerovnic. Řešení však není založeno na prostředcích matematické analýzy a je obsaženo v rozšiřující knihovně *simplex*. Více informací naleznete na doprovodném CD.

>

>



## 4.8 Funkce dané implicitně

Funkce dané implicitně rovnicí  $F(x, y) = 0$  jsou častým problémem při výpočtech. Systém Maple obsahuje příkaz *implicitdiff*, který počítá derivace těchto funkcí.

*implicitdiff*( {f1,...,fm}, {y1,...,yn}, {u1,...,ur}, x1,..., xk ) ,

kde  $f_1, \dots, f_m$  jsou algebraické výrazy, popř. rovnice;  $y_1, \dots, y_n$  jsou výrazy nebo funkce závislých proměnných  $u_1, \dots, u_r$ ;  $x_1, \dots, x_k$  jsou proměnné podle nichž se derivuje.

Nejprve však ukažme postup výpočtu derivace funkce dané implicitně rovnicí  $F(x, y) = 0$  bez užití tohoto příkazu. Tuto rovnici derivujeme podle proměnné  $x$  a na  $y$  se díváme jako na funkci proměnné  $x$ , tj.  $y(x)$ . Derivací obdržíme rovnici  $F_x(x, y) + y'F_y(x, y) = 0$ . Z této rovnice pak vypočteme  $y'$ . Tento postup lze užít i při výpočtu vyšších derivací.

> **Fxy:=x^2+x\*y(x)+y(x)^2-3; # zadání funkce**

$$F_{xy} := x^2 + x y(x) + y(x)^2 - 3$$

> **derFxy:=diff(Fxy,x); # výpočet derivace funkce dle x**

$$\text{derFxy} := 2x + y(x) + x \left( \frac{d}{dx} y(x) \right) + 2y(x) \left( \frac{d}{dx} y(x) \right)$$

> **solve(derFxy,diff(y(x),x)); # vypočtená derivace**

$$-\frac{2x + y(x)}{x + 2y(x)}$$

Nyní provedeme výpočet pomocí příkazu *implicitdiff*

> **F:=x^2+x\*y+y^2-3;**

$$F := x^2 + x y + y^2 - 3$$

> **implicitdiff(F,y,x); # výpočet první derivace**

$$-\frac{2x + y}{x + 2y}$$

> **implicitdiff(F,y,x,x); # výpočet druhé derivace**

$$-\frac{6(x^2 + x y + y^2)}{x^3 + 6x^2 y + 12x y^2 + 8y^3}$$

Více informací o tomto příkazu naleznete v helpu.

>

>

## 5. Integrální počet funkcí jedné a více proměnných

V této kapitole budou probrány příkazy sloužící k nalezení primitivních funkcí, Riemannových integrálů a k jejich aplikacím.

### 5.1 Primitivní funkce

System Maple má příkaz, pomocí kterého se vypočítá primitivní funkce s nulovou integrační konstantou k funkci  $f$ . Jde o příkaz *int*. Dále příkaz *Int*.

*int(f(x),x)*

*Int(f(x),x)*

První příkaz se používá pro výpočet primitivní funkce, druhý pouze pro její matematický zápis a hodnotu pak vypočteme pomocí příkazu *value*. Prvním parametrem obou příkazů může být i výraz pro výpočet hodnoty funkce  $f(x)$ . Druhý parametr je proměnná, podle níž se integruje. Výsledkem příkazu je funkce, ke které není připočtena aditivní konstanta.

> **f:=x->sin(x)+cos(x)-2;**

$$f := x \rightarrow \sin(x) + \cos(x) - 2$$

> **Int(f(x),x);** # matematický zápis

$$\int \sin(x) + \cos(x) - 2 \, dx$$

> **value(%);** # vypočtená hodnota

$$-\cos(x) + \sin(x) - 2x$$

> **diff(%,x)-f(x);** # provedení zkoušky

0

Následující příklad ukazuje výpočet aditivní konstanty. Nalezneme primitivní funkci  $F$  k funkci  $f(x)$  takovou, že  $F(0)=0$ .

> **f:=x->1/2+sin(x);** # zadání funkce f(x)

$$f := x \rightarrow \frac{1}{2} + \sin(x)$$

> **Int(f(x),x);**

$$\int \frac{1}{2} + \sin(x) \, dx$$

> **g:=unapply(int(f(x),x),x);**  
# primitivní funkce k f(x) - zapsána jako funkce pomocí unapply

$$g := x \rightarrow \frac{1}{2}x - \cos(x)$$

Funkce  $g(x)$  však nespĺňuje požadovanou podmínku  $g(0)=0$ . Splnění této podmínky docílíme následujícím postupem.

> **g(x)-g(0);**

$$\frac{x}{2} - \cos(x) + 1$$

> **F:=unapply(%,x);** # výsledná primitivní funkce

$$F := x \rightarrow \frac{1}{2}x - \cos(x) + 1$$

> `Int(x^3*cos(x^2), x) = int(x^3*cos(x^2), x);`

$$\int x^3 \cos(x^2) dx = \frac{1}{2} \cos(x^2) + \frac{1}{2} x^2 \sin(x^2)$$

> `diff(rhs(%), x);` **# zkouška**

$$x^3 \cos(x^2)$$

Předchozí příklad ukazuje použití příkazu *rhs*, popř. *lhs*, pomocí kterých se odkazujeme na pravou, popř. levou stranu rovnice.

> `int(1/(x+exp(x)), x);` **# výpočet primitivní funkce**

$$\int \frac{1}{x + e^x} dx$$

Maple neumí nalézt tuto primitivní funkci pomocí elementárních funkcí. Proto ji označil jako funkci definovanou pomocí integrálu. Poznamenejme, že my tuto primitivní funkci rovněž nedokážeme nalézt.

>

>

## 5.2 Výpočet primitivních funkcí pomocí knihovny "student"

Postup při výpočtu primitivní funkce lze také vysvětlit při použití rozšiřující knihovny *student*. Jsou v ní uvedeny příkazy *intparts*, *changevar*, pro dva základní postupy, metodu per partes a metodu substituční.

### Metoda per partes

Metoda vychází z matematických vzorců

$$\int f(x)g(x)dx = f(x)G(x) - \int f'(x)G(x)dx \quad (1)$$

nebo

$$\int f(x)g(x)dx = F(x)g(x) - \int F(x)g'(x)dx, \quad (2)$$

kde  $F, G$  jsou primitivní funkce k funkcím  $f, g$ . V obou případech se počítá integrál na levé straně a volí se ten ze vzorců, ve kterém výpočet integrálu na pravé straně a primitivní funkce  $G(x)$ , popř.  $F(x)$  není náročný, anebo je méně náročný.

Příkaz určený pro výpočet metodou per partes má tvar

***intparts(Int(h(x),x), k(x))*** ,

kde  $h(x) = f(x)g(x)$  (viz. levé strany vztahů (1), (2)) a  $k(x) = f(x)$  při užití vztahu (1), popř.

$k(x) = g(x)$  při užití druhého vztahu.  $k(x)$  je tedy vždy ta funkce, kterou chceme derivovat.

Příkaz pro výpočet metodou per partes může mít i obecnější tvar

***intparts(Z(x,y),k(x))*** ,

kde  $Z(x,y)$  je výraz pro výpočet hodnoty funkce  $Z$  dvou proměnných a  $y = \int f(x)g(x)dx$ .

Ještě připomeňme, že tento příkaz je součástí rozšiřující knihovny *student*. Proto je nutné knihovnu nahrát do paměti.

```
> restart:with(student) : # nahrání rozšiřující knihovny
```

### Příklad 1

Vypočtěte následující integrál metodou per partes

```
> i0:=Int(x*sin(x), x) ;
```

$$i0 := \int x \sin(x) dx$$

```
> i0=intparts(i0,x) ; # výpočet - za nederivovanou složku volíme x
```

$$\int x \sin(x) dx = -x \cos(x) - \int -\cos(x) dx$$

Použili jsme vzorec (1) pro  $f(x)=x$ ,  $g(x)=\sin(x)$ . Vidíme, že výpočet integrálu na pravé straně je již snadný. Stejně tak výpočet primitivní funkce  $G(x)$  byl snadný.

```
> i0=intparts(i0,sin(x)) ;
```

$$\int x \sin(x) dx = \frac{1}{2} \sin(x) x^2 - \int \frac{1}{2} \cos(x) x^2 dx$$

Použili jsme zde vzorec (2) pro  $f(x)=x$  a  $g(x)=\sin(x)$ . Vidíme, že výpočet integrálu na pravé straně je náročnější než výpočet původního integrálu.

> `i0:=intparts(i0,1);`

$$\int x \sin(x) dx = \sin(x) - x \cos(x) - \int 0 dx$$

Opět jsme použili vzorec (2) pro  $f(x)=x \sin(x)$  a  $g(x)=1$ . Výpočet primitivní funkce  $F(x)$  je náročný, ve skutečnosti se jedná o výpočet původního integrálu. Porovnejme poslední variantu s výsledkem příkazu pro výpočet primitivní funkce bez použití příkazu *intparts*:

> `i0:=int(integrand(i0),x);`

$$\int x \sin(x) dx = \sin(x) - x \cos(x)$$

Pomocí příkazu *integrand* z rozšiřující knihovny *student* získáme integrand z daného integrálu. Dále ukážeme použití příkazu *intparts* v obecnějším tvaru

> `intparts(x^3+2*x+sin(i0^2),x);`

$$x^3 + 2x + \sin\left((-x \cos(x) - \int -\cos(x) dx)^2\right)$$

>

#### Příklad 2

> `i1:=Int(x^3*ln(x),x);`

$$i1 := \int x^3 \ln(x) dx$$

> `intparts(% , ln(x));`

$$\frac{1}{4} \ln(x) x^4 - \int \frac{x^3}{4} dx$$

Výpočet integrálu na pravé straně nám nečiní žádné potíže. Pro výpočet výsledné primitivní funkce po použití příkazu *intparts* se užívá příkazu *value*.

> `value(%);`

$$\frac{1}{4} \ln(x) x^4 - \frac{x^4}{16}$$

>

#### Příklad 3

> `i2:=Int(x^2*exp(x),x);`

$$i2 := \int x^2 e^x dx$$

> `i2:=intparts(% , x^2);`

$$\int x^2 e^x dx = x^2 e^x - \int 2x e^x dx$$

Metodu per partes opakujeme pro pravou stranu, tzn. užitíme příkazu *rhs* pro získání pravé strany předchozí rovnice

> `intparts(rhs(%),x);`

$$x^2 e^x - 2x e^x + \int 2 e^x dx$$

>

Tento příklad slouží také k ukázce jiné knihovny a to **Student**, která je nová od verze Maple 8 a

doplňuje schopnosti knihovny **student**.

> **with(Student[Calculus1]);**

[*AntiderivativePlot, AntiderivativeTutor, ApproximateInt, ApproximateIntTutor, ArcLength, ArcLengthTutor, Asymptotes, Clear, CriticalPoints, CurveAnalysisTutor, DerivativePlot, DerivativeTutor, DiffTutor, ExtremePoints, FunctionAverage, FunctionAverageTutor, FunctionChart, FunctionPlot, GetMessage, GetNumProblems, GetProblem, Hint, InflectionPoints, IntTutor, Integrand, InversePlot, InverseTutor, LimitTutor, MeanValueTheorem, MeanValueTheoremTutor, NewtonQuotient, NewtonsMethod, NewtonsMethodTutor, PointInterpolation, RiemannSum, RollesTheorem, Roots, Rule, Show, ShowIncomplete, ShowSteps, Summand, SurfaceOfRevolution, SurfaceOfRevolutionTutor, Tangent, TangentSecantTutor, TangentTutor, TaylorApproximation, TaylorApproximationTutor, Understand, Undo, VolumeOfRevolution, VolumeOfRevolutionTutor, WhatProblem*]

>

Ukážeme, jak je možné nechat si od Maple poradit při postupu výpočtu.

> **Hint(i2); # napoveda k vypoctu i2**

[*parts, x<sup>2</sup>, e<sup>x</sup>*]

> **Rule[%](i2); # aplikace napovezeneho**

$$\int x^2 e^x dx = x^2 e^x - \int 2x e^x dx$$

> **Hint(%);**

[*constantmultiple*]

> **# Undrestand - prikaz, pomoci ktere lze specifikovat, ktere veci jiz umim... napr. vytknuti konstanty z integralu**

> **Rule[%](%%);**

$$\int x^2 e^x dx = x^2 e^x - 2 \int x e^x dx$$

> **Hint(%);**

**Rule[%](%%);**

[*parts, x, e<sup>x</sup>*]

$$\int x^2 e^x dx = x^2 e^x - 2x e^x + 2 \int e^x dx$$

> **Hint(%);**

**Rule[%](%%);**

[*exp*]

$$\int x^2 e^x dx = x^2 e^x - 2x e^x + 2 e^x$$

> **Hint(%);**

**Rule[%](%%);**

[ ]

Následující funkce vypíše námi provedené nápovědy a jejich aplikace

> **ShowSteps();**

$$\begin{aligned}
 \int x^2 e^x dx &= x^2 e^x - \int 2x e^x dx \\
 &= x^2 e^x - 2 \int x e^x dx \\
 &= x^2 e^x - 2x e^x + 2 \int e^x dx \\
 &= x^2 e^x - 2x e^x + 2 e^x
 \end{aligned}$$

Nyní se podívejme na průvodce knihovny Student - jde o Maplety.

```

> IntTutor(i2); # tutorial pro vypocet integralu
>
> ApproximateIntTutor(); # tutorial pro aproximace fci
>

```

Volání tutorialu pro aproximaci funkcí.

```

> ApproximateInt(sin(x), 0..Pi, 'partition' = 10, 'method' =
 upper, 'partitiontype' = normal, 'iterations' = 6, 'output' =
 'animation', 'showarea' = false);

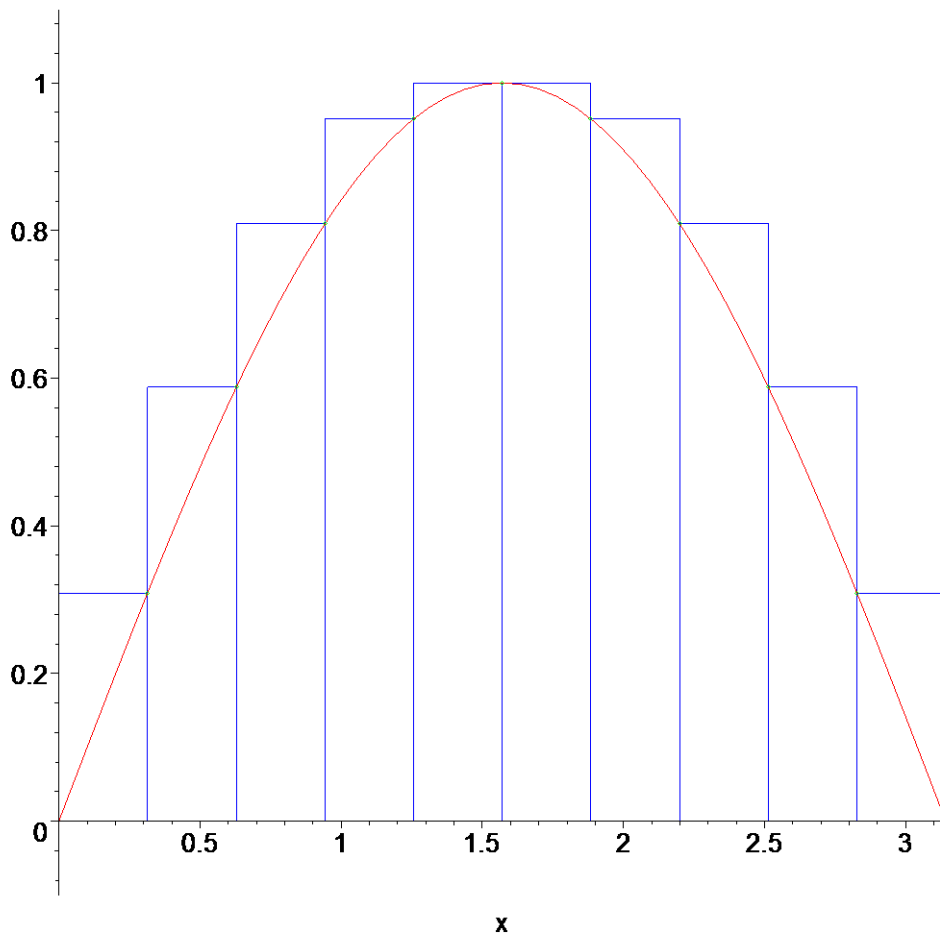
```

An Approximation of the Integral of

$$f(x) = \sin(x)$$

on the Interval [0, Pi]

Using an Upper Riemann Sum



Tutorial pro Newtonovu metodu.

```

> NewtonsMethodTutor();

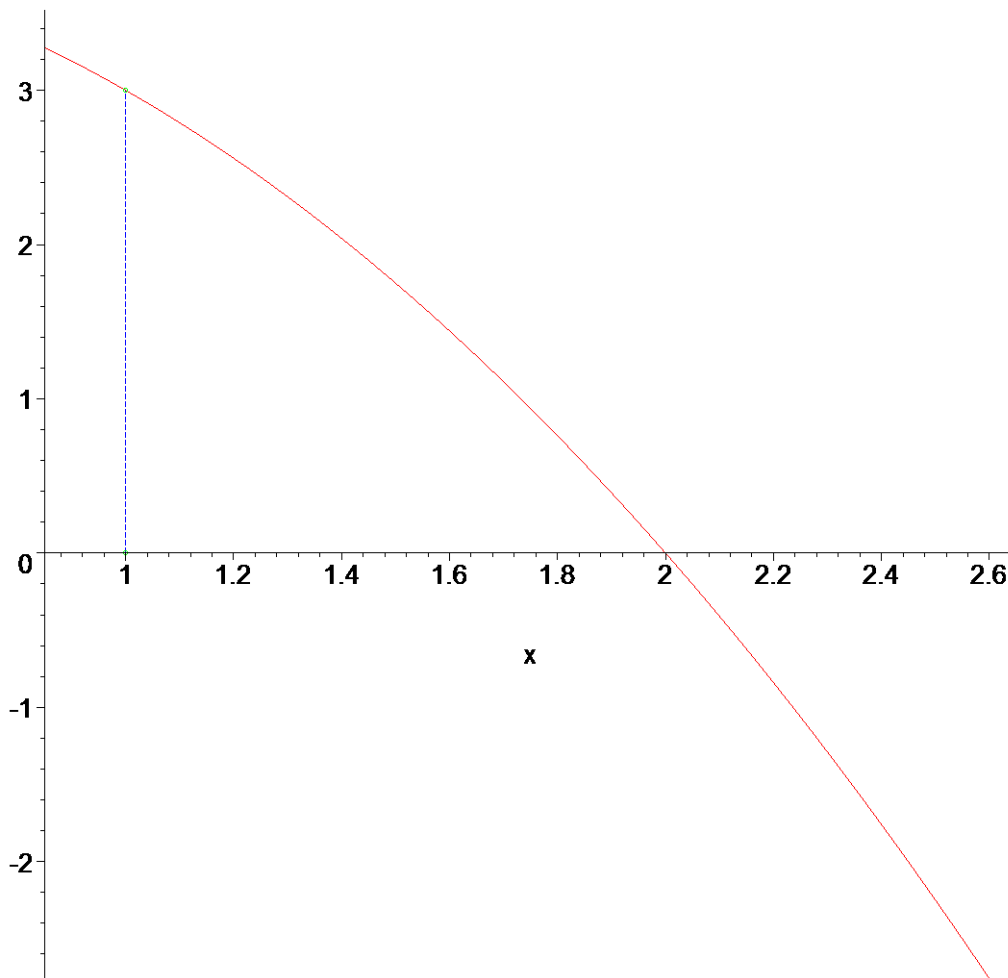
```

```
> NewtonsMethod(4-x^2, 1, iterations=5, output=animation);
```

5 Iterations of Newton's Method Applied to

$$f(x) = 4 - x^2$$

with Initial Point  $x = 1$



>

Nyní naprogramujeme proceduru, která bude automaticky používat nápovědu a aplikovat ji. Výsledkem bude kompletní výpis jednotlivých kroků.

```
> ShowAllSteps:=proc(i)
 local iu,h; # lokální proměnné
 iu:=i;
 h:=Hint(iu); # nápověda číslo 1
 while h<>[] do # procházím dokud není nápověda prázdná
 use Student[Calculus] in
 lprint(h);
 iu:=Rule[%](%%);
 h:=Hint(iu);
 end use;
 end do;
 Student[Calculus1][ShowSteps](); # vypisu všechny kroky
end proc;
> ShowAllSteps(i2);
[parts, x^2, exp(x)]
```



```
[constantmultiple]
[parts, x, exp(x)]
[exp]
```

$$\begin{aligned} \int x^2 e^x dx &= x^2 e^x - \int 2x e^x dx \\ &= x^2 e^x - 2 \int x e^x dx \\ &= x^2 e^x - 2x e^x + 2 \int e^x dx \\ &= x^2 e^x - 2x e^x + 2 e^x \end{aligned}$$

Procedura funguje jak pro zobrazení kroků integrálů, limit a derivací.

```
> lim1:=Limit(x*cos(x)*ln(x),x=0);
```

$$lim1 := \lim_{x \rightarrow 0} x \cos(x) \ln(x)$$

```
> ShowAllSteps(lim1);
```

$$\begin{aligned} \lim_{x \rightarrow 0} x \cos(x) \ln(x) &= \left( \lim_{x \rightarrow 0} \ln(x) x \right) \left( \lim_{x \rightarrow 0} \cos(x) \right) \\ &= \left( \lim_{x \rightarrow 0} -x \right) \left( \lim_{x \rightarrow 0} \cos(x) \right) \\ &= - \left( \lim_{x \rightarrow 0} x \right) \left( \lim_{x \rightarrow 0} \cos(x) \right) \\ &= 0 \end{aligned}$$

```
> der1:=Diff(x^2*sin(x),x);
```

$$der1 := \frac{d}{dx} (x^2 \sin(x))$$

```
>
```

```
> ShowAllSteps(der1);
```

$$\begin{aligned} \frac{d}{dx} (x^2 \sin(x)) &= \left( \frac{d}{dx} (x^2) \right) \sin(x) + x^2 \left( \frac{d}{dx} \sin(x) \right) \\ &= 2x \sin(x) + x^2 \left( \frac{d}{dx} \sin(x) \right) \\ &= 2x \sin(x) + x^2 \cos(x) \end{aligned}$$

```
>
```

```
>
```

```
>
```

```
>
```

#### Příklad 4

```
> i3:=Int(x^3*cos(x^2),x);
```

$$i3 := \int x^3 \cos(x^2) dx$$

Za funkci  $k(x)$  volíme  $x^2$ . Jiné volby by nevedly k cíli.

```
> intparts(i3,x); # tato volba nevede k cíli
```

$$x \left( \frac{1}{2} \sin(x^2) x - \frac{1}{4} \sqrt{2} \sqrt{\pi} \operatorname{FresnelS} \left( \frac{\sqrt{2} x}{\sqrt{\pi}} \right) \right) - \int \frac{1}{2} \sin(x^2) x - \frac{1}{4} \sqrt{2} \sqrt{\pi} \operatorname{FresnelS} \left( \frac{\sqrt{2} x}{\sqrt{\pi}} \right) dx$$

> `i3=intparts(i3,x^2);` # jediná volba, která vede k cíli

$$\int x^3 \cos(x^2) dx = \frac{1}{2} x^2 \sin(x^2) - \int \sin(x^2) x dx$$

Výpočet integrálu na pravé straně lze již snadno provést substituční metodou (viz. následující odstavec).

> `intparts(i3,x^3);` # opět nevede k cíli

$$\frac{1}{2} x^3 \sqrt{2} \sqrt{\pi} \operatorname{FresnelC} \left( \frac{\sqrt{2} x}{\sqrt{\pi}} \right) - \int \frac{3}{2} x^2 \sqrt{2} \sqrt{\pi} \operatorname{FresnelC} \left( \frac{\sqrt{2} x}{\sqrt{\pi}} \right) dx$$

>  
>

#### Příklad 5

> `i4:=Int(exp(x)*sin(x),x);`

$$i4 := \int e^x \sin(x) dx$$

> `i4=intparts(i4,sin(x));`

$$\int e^x \sin(x) dx = e^x \sin(x) - \int \cos(x) e^x dx$$

Integrál na pravé straně opět počítáme metodou per partes

> `i4=intparts(rhs(%),cos(x));`

$$\int e^x \sin(x) dx = e^x \sin(x) - \cos(x) e^x + \int -e^x \sin(x) dx$$

> `simplify(%);`

$$\int e^x \sin(x) dx = e^x \sin(x) - \cos(x) e^x - \int e^x \sin(x) dx$$

Vidíme, že na obou stranách rovnice vystupuje hledaný integrál. Rovnici proto řešíme pomocí příkazu *solve* vzhledem k neznámé, kterou je tento integrál.

> `i4=solve(%,i4);`

$$\int e^x \sin(x) dx = \frac{1}{2} e^x \sin(x) - \frac{1}{2} \cos(x) e^x$$

Elegantnější je však použití příkazu *isolate(odkud,co)*, pomocí kterého vyjádříme danou proměnnou (výraz) z rovnice.

> `isolate(%%,i4);`

$$\int e^x \sin(x) dx = \frac{1}{2} e^x \sin(x) - \frac{1}{2} \cos(x) e^x$$

>

#### Příklad 6

> `i5:=Int(arctan(x),x);`

$$i5 := \int \arctan(x) dx$$

> `i5=intparts(i5,arctan(x));`

$$\int \arctan(x) dx = \arctan(x) x - \int \frac{x}{1+x^2} dx$$

Výpočet integrálu na pravé straně lze již snadno provést substituční metodou (viz. následující odstavec).

>

>

### Metoda substituční

Metoda vychází z matematických vzorců

$$\int f(\varphi(x))\varphi'(x)dx = \int f(t)dt \Big|_{t=\varphi(x)} \quad (3)$$

nebo

$$\int f(x)dx = \int f(\varphi(t))\varphi'(t)dt \Big|_{t=\varphi^{-1}(x)} \quad (4)$$

Příkaz pro výpočet substituční metodou má tvar

**changevar(relace, Int(g(x),x), nová proměnná)**,

kde relace je vztah tvaru  $\varphi(x) = \psi(t)$  mezi starou proměnnou  $x$  a novou proměnnou  $t$ . V případě relace, v níž je nová proměnná uvedena v relaci pouze na pravé straně, nemusí být třetí parametr uveden.

#### Příklad 1

```
> restart:with(student):
```

```
> i1:=Int((sin(x))^2/(cos(x))^4,x);
```

$$i1 := \int \frac{\sin(x)^2}{\cos(x)^4} dx$$

Použijeme substituci  $\tan(x) = t$ , neboť  $\frac{1}{\cos^2(x)} dx = dt$

```
> i1=changevar(tan(x)=t,i1);
být uveden
```

```
třetí parametr nemusí
```

$$\int \frac{\sin(x)^2}{\cos(x)^4} dx = \int t^2 dt$$

Integrál na pravé straně již dovedeme snadno spočítat.

```
> value(rhs(%));
```

$$\frac{t^3}{3}$$

Musíme ještě zpět dosadit  $t=\tan(x)$  pomocí příkazu *subs*.

```
> subs(t=tan(x),%);
```

$$\frac{1}{3} \tan(x)^3$$

Výsledek tedy je

```
> i1=%;
```

$$\int \frac{\sin(x)^2}{\cos(x)^4} dx = \frac{1}{3} \tan(x)^3$$

```
> value(i1);
```

```
přímým výpočtem
```

```
kontrola správnosti výsledku
```

$$\frac{1}{3} \frac{\sin(x)^3}{\cos(x)^3}$$

>

### Příklad 2

> `i2:=Int(1/sqrt(1-(sin(x))^4),x);`

$$i2 := \int \frac{1}{\sqrt{1 - \sin(x)^4}} dx$$

Vyjdeme ze vzorce  $1 = (\cos^2 x + \sin^2 x)^2 = \cos^4 x + 2 \sin^2 x \cos^2 x + \sin^4 x$ , tj.

$\sqrt{1 - \sin^4 x} = \sqrt{\cos^4 x + 2 \sin^2 x \cos^2 x} = \cos^2 x \sqrt{1 + 2 \tan^2 x}$ . Tedy označíme

> `i2p:=Int(1/((cos(x))^2*sqrt(1+2*(tan(x))^2)),x);`

$$i2p := \int \frac{1}{\cos(x)^2 \sqrt{1 + 2 \tan(x)^2}} dx$$

a dostáváme

> `i2=i2p;`

$$\int \frac{1}{\sqrt{1 - \sin(x)^4}} dx = \int \frac{1}{\cos(x)^2 \sqrt{1 + 2 \tan(x)^2}} dx$$

Ze stejného důvodu jako v předcházejícím příkladu zavedeme substituci  $\tan(x)=t$ .

> `i2p1:=changevar(t=tan(x),i2p);`

Error, (in changevar) usage: changevar(g(a) = f(b),F(..), b)

Je ohlášena chyba, neboť v relaci příkazu *changevar* nová proměnná *t* nevystupuje pouze na pravé straně. Chybu lze opravit takto: místo  $t=\tan(x)$  napíšeme  $\tan(x)=t$ , nebo do příkazu *changevar* připišeme třetí parametr, tím se vyhneme jakýmkoliv nepříjemnostem.

> `i2p1:=changevar(t=tan(x),i2p,t);`

$$i2p1 := \int \frac{1}{\sqrt{1 + 2 t^2}} dt$$

Tento integrál však ještě neumíme spočítat. Zavedeme v něm novou substituci  $\sqrt{2}t = z$ .

> `i2p2:=changevar(sqrt(2)*t=z,i2p1,z);`

$$i2p2 := \int \frac{\sqrt{2}}{2 \sqrt{1 + z^2}} dz$$

Tento integrál již dovedeme spočítat

> `i2p3:=value(%);`

$$i2p3 := \frac{1}{2} \sqrt{2} \operatorname{arcsinh}(z)$$

Nyní se musíme vrátit zpět k původní proměnné, tj. dosadíme  $\tan(x)=t$  a  $\sqrt{2}t = z$ , tj.

$z = \sqrt{2} \tan(x)$

> `i2=subs(z=sqrt(2)*tan(x),i2p3);`

$$\int \frac{1}{\sqrt{1 - \sin(x)^4}} dx = \frac{1}{2} \sqrt{2} \operatorname{arcsinh}(\sqrt{2} \tan(x))$$

Ještě poznamenejme, že systém Maple sám tento integrál takto elegantně nespočítá, posuďte sami.

> **value(i2);**

$$\begin{aligned} & - \left( \sqrt{2} \operatorname{EllipticF} \left( \frac{(-1 + \cos(x)) (\sqrt{2} - 1) I}{\sin(x)}, 3 + 2\sqrt{2} \right) \sqrt{-3 + 2\sqrt{2}} I \right. \\ & - 4 \operatorname{EllipticPi} \left( \frac{\sqrt{-3 + 2\sqrt{2}} \sin(x)}{1 + \cos(x)}, \frac{1}{-3 + 2\sqrt{2}}, \frac{\sqrt{-3 - 2\sqrt{2}}}{\sqrt{-3 + 2\sqrt{2}}} \right) \\ & \left. + 2 \operatorname{EllipticPi} \left( \frac{\sqrt{-3 + 2\sqrt{2}} \sin(x)}{1 + \cos(x)}, \frac{1}{-3 + 2\sqrt{2}}, \frac{\sqrt{-3 - 2\sqrt{2}}}{\sqrt{-3 + 2\sqrt{2}}} \right) \sqrt{2} \right) \cos(x) (1 + \cos(x)) \\ & \sqrt{-\frac{2 \cos(x) + 2\sqrt{2} \cos(x) - 4 - 2\sqrt{2}}{1 + \cos(x)}} \sqrt{\frac{\sqrt{2} \cos(x) - \cos(x) + 2 - \sqrt{2}}{1 + \cos(x)}} / ( \\ & \sqrt{-(\cos(x)^2 - 2) \cos(x)^2} (\sqrt{2} - 1) \sqrt{-3 + 2\sqrt{2}} ) \end{aligned}$$

>

**Příklad 3**

> **i3:=Int(x\*exp(-x^2),x);**

$$i3 := \int x e^{(-x^2)} dx$$

Zavedeme substituci  $-x^2 = t$

> **changevar(-x^2=t,i3,t);**

$$\int -\frac{1}{2} e^t dt$$

Po výpočtu již známého integrálu se vrátíme k původní proměnné

> **i3=subs(t=-x^2,value(%));**

$$\int x e^{(-x^2)} dx = -\frac{1}{2} e^{(-x^2)}$$

Aplikujme námi naprogramovanou proceduru pro automatický postupný výpočet.

> **ShowAllSteps(i3);**

```
[change, u = x^2, u]
[constantmultiple]
[change, u1 = -u, u1]
[constantmultiple]
[exp]
[revert]
[revert]
```

$$\int x e^{(-x^2)} dx = \int \frac{1}{2} e^{(-u)} du$$

$$\begin{aligned}
&= \frac{1}{2} \int e^{(-u)} du \\
&= \frac{1}{2} \int -e^{ul} du \\
&= -\frac{1}{2} \int e^{ul} du \\
&= -\frac{1}{2} e^{ul} \\
&= -\frac{1}{2} e^{(-u)} \\
&= -\frac{1}{2} e^{(-x^2)}
\end{aligned}$$

>  
>  
>

#### Příklad 4

> **i4:=Int(cos(Pi/x)/x^2,x);**

$$i4 := \int \frac{\cos\left(\frac{\pi}{x}\right)}{x^2} dx$$

Použijeme substituci  $t = \pi/x$

> **changevar(t=Pi/x,i4,t);**

$$\int -\frac{\cos(t)}{\pi} dt$$

Vypočteme a vrátíme se k původní proměnné

> **i4=subs(t=Pi/x,value(%));**

$$\int \frac{\cos\left(\frac{\pi}{x}\right)}{x^2} dx = -\frac{\sin\left(\frac{\pi}{x}\right)}{\pi}$$

>

#### Příklad 5

> **i5:=Int(4\*x/(1-x^4)^(1/2),x);**

$$i5 := \int \frac{4x}{\sqrt{1-x^4}} dx$$

Zavedeme substituci  $x^2 = t$

> **changevar(x^2=t,i5,t);**

$$\int \frac{2}{\sqrt{1-t^2}} dt$$

> `i5:=subs (t=x^2 , value (%) ) ;`

$$\int \frac{4x}{\sqrt{1-x^4}} dx = 2 \arcsin(x^2)$$

>  
>

#### Příklad 6

V minulém odstavci v příkladu 4 jsme metodou per partes vypočítali následující integrál

> `Int (x^3*cos (x^2) , x)=1/2*x^2*sin (x^2) -Int (x*sin (x^2) , x) ;`

$$\int x^3 \cos(x^2) dx = \frac{1}{2} x^2 \sin(x^2) - \int x \sin(x^2) dx$$

Integrál na pravé straně počítáme nyní substitucí  $x^2 = t$

> `i6:=changevar (x^2=t, Int (x*sin (x^2) , x) , t) ;`

$$i6 := \int \frac{1}{2} \sin(t) dt$$

A tedy

> `Int (x^3*cos (x^2) , x)=1/2*x^2*sin (x^2) -subs (t=x^2, value (i6)) ;`

$$\int x^3 \cos(x^2) dx = \frac{1}{2} x^2 \sin(x^2) + \frac{1}{2} \cos(x^2)$$

>  
>

#### Příklad 7

> `restart:with(student) ;`

> `i7:=Int (x^2/sqrt (1-x^2) , x) ;`

$$i7 := \int \frac{x^2}{\sqrt{1-x^2}} dx$$

Integrand upravíme tak, že přičteme a odečteme v čitateli 1 a zlomek rozdělíme na dva

> `p:=(x^2-1)/sqrt (1-x^2)+1/sqrt (1-x^2) ;`

$$p := \frac{-1+x^2}{\sqrt{1-x^2}} + \frac{1}{\sqrt{1-x^2}}$$

Jednotlivé zlomky označíme

> `p1:=(-1+x^2)/(1-x^2)^(1/2)=-sqrt (1-x^2) ;`

`p2:=1/((1-x^2)^(1/2)) ;`

$$p1 := \frac{-1+x^2}{\sqrt{1-x^2}} = -\sqrt{1-x^2}$$

Počítáme tedy integrál



> `i7u:=Int(rhs(p1),x)+Int(p2,x);`

$$i7u := \int -\sqrt{1-x^2} dx + \int \frac{1}{\sqrt{1-x^2}} dx$$

Provedeme kontrolní výpočet, zda jde o stejné integrály

> `is(value(i7)=value(i7u));`

*true*

Druhý integrál z *i7u* nalezneme přímo v tabulkách základních integrálů a označíme jej

> `i7u2:=arcsin(x);`

Při výpočtu prvního integrálu, označíme ho *i7u1*, zavedeme substituci  $x = \sin t$  pro

$x \in \langle -1, 1 \rangle$ , tj. pro  $t \in \left\langle -\frac{\pi}{2}, \frac{\pi}{2} \right\rangle$ .

> `i7u1:=Int(-sqrt(1-x^2),x);`

$$i7u1 := \int -\sqrt{1-x^2} dx$$

> `i7u1_:=changevar(sin(t)=x,i7u1,t);`

$$i7u1_ := \int -\sqrt{1-\sin(t)^2} \cos(t) dt$$

Zjednodušíme pomocí známého vzorce pro součet druhé mocniny *sin* a *cos*. Dostaneme

> `assume(-Pi/2<t,t<Pi/2);` # upraven obor proměnné *t*, dle výše uvedené substituce

> `i7u1_:=simplify(i7u1_);`

$$i7u1_ := -\int \cos(t)^2 dt$$

Dále užijeme vzorce pro výpočet polovičního úhlu

> `i7u1_:=subs(cos(t)^2=1/2*(1+cos(2*t)),i7u1_);`

$$i7u1_ := -\int \frac{1}{2} + \frac{1}{2} \cos(2t) dt$$

Tento integrál už lze snadno spočítat

> `v:=value(i7u1_);`

$$v := -\frac{t}{2} - \frac{1}{4} \sin(2t)$$

Předchozí výsledek zjednodušíme

> `simplify(v,{sin(2*t)=2*cos(t)*sin(t)});`

$$-\frac{t}{2} - \frac{1}{2} \cos(t) \sin(t)$$

Vrátíme se k původní proměnné *x* substitucí  $t = \arcsin x$

> `i7u1:=simplify(subs(t=arcsin(x),v));`

$$i7u1 := -\frac{1}{2} \arcsin(x) - \frac{1}{2} \cos(\arcsin(x)) \sin(\arcsin(x))$$

A tedy výsledný integrál je roven

> `i7=simplify(i7u1+i7u2);`

$$\int \frac{x^2}{\sqrt{1-x^2}} dx = -\frac{x\sqrt{1-x^2}}{2} + \frac{1}{2} \arcsin(x)$$

Provedeme kontrolu výpočtu

> **value(i7)** ;

$$-\frac{x\sqrt{1-x^2}}{2} + \frac{1}{2} \arcsin(x)$$

>

>

### 5.3 Riemannův a nevlastní Riemannův integrál

V předchozích dvou kapitolách byly popsány příkazy  $int(f(x),x)$  a  $Int(f(x),x)$  pro primitivní funkce a příkazy  $intparts(Int(h(x),x),k(x))$  a  $changevar(relace,Int(g(x),x),nová\ proměnná)$  z knihovny *student* pro výpočet primitivní funkce metodou per partes a metodou substituční. Všechno, co bylo uvedeno o těchto příkazech, včetně použití příkazu *value*, se přenáší i na příkazy *int*, *Int*, *intparts* a *changevar* pro výpočet a zápis Riemannova integrálu a nevlastního integrálu. Rozdíl je pouze v tom, že se v příkazech dodá informace o mezích.

Příkazy pak mají tvar

$int(f(x), x=dolní\ mez .. horní\ mez)$

$Int(f(x), x=dolní\ mez .. horní\ mez)$

$intparts(Int(h(x),x= dolní\ mez .. horní\ mez), k(x))$

$changevar(relace, Int(g(x),x= dolní\ mez .. horní\ mez), nová\ proměnná)$

Jako meze mohou být uvedeny libovolné výrazy. Mohou být i nevlastní (*-infinity*, *infinity*) a integrand může mít singularity.

> **restart:with(student) :**

#### Příklad 1

> **f:=x->sin(x)+cos(x)-2;**

> **Int(f(x),x=-Pi..Pi);**  $f:=x \rightarrow \sin(x) + \cos(x) - 2$  **# matematický zápis**

$$\int_{-\pi}^{\pi} \sin(x) + \cos(x) - 2 \, dx$$

> **value(%);** **# vypočtená hodnota**  
-4 π

>

#### Příklad 2

> **Int(abs(x),x=-2..2);**

$$\int_{-2}^2 |x| \, dx$$

> **value(%);**

4

>

#### Příklad 3

> **Int(x^3\*cos(x^2),x=0..1);**

$$\int_0^1 x^3 \cos(x^2) \, dx$$

> **value(%);**

$$\frac{1}{2} \cos(1) + \frac{1}{2} \sin(1) - \frac{1}{2}$$

>

#### Příklad 4

```
> int(1/(x+exp(x)), x=0..1);
```

$$\int_0^1 \frac{1}{x + e^x} dx$$

Maple tento integrál neumí spočítat analyticky. My to také neumíme. Proto místo výsledku výpočtu vypíše pouze matematický zápis tohoto integrálu.

>

#### Příklad 5

```
> restart:with(student):
```

```
> i:=Int(abs(x)/x, x=-1..2);
```

$$i := \int_{-1}^2 \frac{|x|}{x} dx$$

```
> int(integrand(i), x);
```

# výpočet primitivní funkce

$$\begin{cases} -x & x \leq 0 \\ x & 0 < x \end{cases}$$

```
> value(i);
```

1

>

## Metoda per partes

Při výpočtu použijeme příkaz *intparts*

### Příklad 6

Metoda per partes

```
> with(student):
> Int(x*cos(x), x=0..Pi/2);
```

$$\int_0^{\frac{\pi}{2}} x \cos(x) dx$$

Integrál vypočteme metodou per partes

```
> intparts(%, x); # za nederivovanou složku
vezmeme x
```

$$\frac{\pi}{2} - \int_0^{\frac{\pi}{2}} \sin(x) dx$$

Tento integrál již snadno spočítáme

```
> value(%);
```

$$\frac{\pi}{2} - 1$$

>

### Příklad 7

```
> i:=Int(exp(x)*cos(x), x=0..Pi);
```

$$i := \int_0^{\pi} e^x \cos(x) dx$$

Použijeme metodu per partes

```
> intparts(i, cos(x));
```

$$-e^{\pi} - 1 - \int_0^{\pi} -\sin(x) e^x dx$$

Ještě jednou

```
> intparts(%, sin(x));
```

$$-e^{\pi} - 1 + \int_0^{\pi} -e^x \cos(x) dx$$

Zjednodušíme a položíme do rovnice.

```
> i=simplify(%);
```

$$\int_0^{\pi} e^x \cos(x) dx = -e^{\pi} - 1 - \int_0^{\pi} e^x \cos(x) dx$$

Integrál z rovnice vyjádříme

```
> isolate(%, i);
```

$$\int_0^{\pi} e^x \cos(x) dx = -\frac{1}{2} e^{\pi} - \frac{1}{2}$$

> value(i);

# přímý výpočet

$$-\frac{1}{2} e^{\pi} - \frac{1}{2}$$

>

>

### Substituční metoda

Při výpočtu použijeme příkaz *changevar*

#### Příklad 8

- > **restart:with(student) :**
- > **i1:=Int(x^2\*sqrt(1-x^2),x=0..1) ;**

$$i1 := \int_0^1 x^2 \sqrt{1-x^2} dx$$

Použijeme substituci  $x = \sin(t)$ . Jak je vidět z následujícího výsledku operace, systém Maple přepočítá i integrační meze.

- > **i1s:=changevar(x=sin(t),i1,t) ;**

$$i1s := \int_0^{\frac{\pi}{2}} \sin(t)^2 \sqrt{1-\sin(t)^2} \cos(t) dt$$

Počítáme tedy integrál

- > **ilu:=Int(sin(t)^2\*cos(t)^2,t = 0 .. 1/2\*Pi) ;**

$$ilu := \int_0^{\frac{\pi}{2}} \sin(t)^2 \cos(t)^2 dt$$

Použijeme známé vzorce pro *sin* a *cos*

- > **simplify(ilu,{sin(t)^2=1/2\*(1-cos(2\*t)),cos(t)^2=1/2\*(1+cos(2\*t))}) ;**

$$\int_0^{\frac{\pi}{2}} -\frac{1}{4} \cos(2t)^2 + \frac{1}{4} dt$$

Výše uvedený integrál snadno spočítáme pomocí vzorce pro dvojnásobný argument funkce *cos*. Užili jsme ho už v předchozím kroku

- > **value(%);**

$$\frac{\pi}{16}$$

- >
- >

#### Příklad 9

- > **restart:with(student) :**
- > **i2:=Int(x/sqrt(x^2+1),x=1..2) ;**

$$i2 := \int_1^2 \frac{x}{\sqrt{x^2+1}} dx$$

Zavedeme substituci

- > **changevar(t=x^2+1,i2,t) ;**

$$\int_2^5 \frac{1}{2\sqrt{t}} dt$$

Snadno spočteme integrál

> **value (%) ;**

$$\sqrt{5} - \sqrt{2}$$

>

>

**Příklad 10**

> **restart:with(student) ;**

> **i3:=Int(x\*cos(x)^2,x=-Pi..Pi) ;**

$$i3 := \int_{-\pi}^{\pi} x \cos(x)^2 dx$$

Všimněme si, že integrand je funkce lichá

> **f:=unapply(integrand(i3),x) ;**

$$f := x \rightarrow x \cos(x)^2$$

> **is(f(-x)=-f(x)) ;**

**# vlastnost liché funkce**

*true*

Tedy integrál roven 0

> **value(i3) ;**

0

>

>

**Příklad 11**

> **restart:with(student) ;**

> **i4:=Int(x^3/(x^8+1),x=-1..0) ;**

$$i4 := \int_{-1}^0 \frac{x^3}{x^8+1} dx$$

Zavedeme substituci a po dosazení vyměníme integrační meze

> **changevar(x^4=u,i4,u) ;**

$$\int_0^1 -\frac{1}{4u^2+4} du$$

Integrál se snadno spočítá

> **value (%) ;**

$$-\frac{\pi}{16}$$

>

>

>



### Integrály s proměnnými mezemi

Integrály s proměnnými mezemi využijeme později při výpočtu vícenásobných integrálů

> `Int(x^2+1,x=1..y-1)=int(x^2+1,x=1..y-1);`

$$\int_1^{y-1} x^2 + 1 dx = \frac{(y-1)^3}{3} - \frac{7}{3} + y$$

Následující příklad ukáže, že systém Maple se správně vyrovná i s kolizí označení, kdy proměnnou v horní mezi označíme stejným písmenkem  $x$  jako integrační proměnnou.

> `Int(x^2+1,x=1..x-1);`

$$\int_1^{x-1} x^2 + 1 dx$$

> `value(%);`

$$\frac{(x-1)^3}{3} - \frac{7}{3} + x$$

>

>

### Nevlastní integrály

>

#### Příklad 12

> `Int(1/sqrt(x-1),x=1..4);`

$$\int_1^4 \frac{1}{\sqrt{x-1}} dx$$

> `value(%);`

$$2\sqrt{3}$$

Jedná se o nevlastní integrál. Integrand má singularitu v bodě 1.

>

#### Příklad 13

Určíme, zda dané integrály konvergují či divergují

> `Int(1/x^3,x=1..infinity)=int(1/x^3,x=1..infinity);`

$$\int_1^{\infty} \frac{1}{x^3} dx = \frac{1}{2}$$

> `Int(1/sqrt(x),x=1..infinity)=int(1/sqrt(x),x=1..infinity);`

$$\int_1^{\infty} \frac{1}{\sqrt{x}} dx = \infty$$

> `Int(1/(x^2+1),x=-infinity..infinity)=int(1/(x^2+1),x=-infinity..infinity);`

$$\int_{-\infty}^{\infty} \frac{1}{x^2 + 1} dx = \pi$$

> `Int(x/sqrt(x^5+1), x=0..infinity)=int(x/sqrt(x^5+1), x=0..infinity);`

$$\int_0^{\infty} \frac{x}{\sqrt{x^5 + 1}} dx = \frac{1}{5} B\left(\frac{1}{10}, \frac{2}{5}\right)$$

> `Int((x^2+1)/(x^3+1), x=0..infinity)=int((x^2+1)/(x^3+1), x=0..infinity);`

$$\int_0^{\infty} \frac{x^2 + 1}{1 + x^3} dx = \infty$$

>  
>

#### Příklad 14

> `restart:with(student);`  
> `i:=Int(1/(1+x^3), x=0..infinity);`

$$i := \int_0^{\infty} \frac{1}{1 + x^3} dx$$

Rozložíme integrand na parciální zlomky. Pomocí příkazu *integrand* získáme integrand předchozího integrálu.

> `iu:=convert(integrand(i), parfrac, x);`

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

Spočteme primitivní funkci

> `iu1:=int(iu, x);`

$$iu1 := \frac{1}{3} \ln(x+1) - \frac{1}{6} \ln(x^2-x+1) + \frac{1}{3} \sqrt{3} \arctan\left(\frac{(2x-1)\sqrt{3}}{3}\right)$$

Dle Leibnitz-Newtonovy věty dosadíme meze. Pro *infinity* musíme počítat limitu.

> `limit(iu1, x=infinity)-simplify(subs(x=0, %));`

$$\frac{2\sqrt{3}\pi}{9}$$

Kontrola výpočtu

> `value(i);`

$$\frac{2\sqrt{3}\pi}{9}$$

>

#### Příklad 15

> `i := Int(1/x, x=-1..1) ;`

$$i := \int_{-1}^1 \frac{1}{x} dx$$

Problémem při integraci je neodstranitelná singularita v bodě 0. Proto rozdělíme integrál na dva a pokud oba konvergují, pak jejich součet je právě hledaným integrálem.

> `i1 := Int(1/x, x=-1..0) ;`

$$i1 := \int_{-1}^0 \frac{1}{x} dx$$

> `i2 := Int(1/x, x=0..1) ;`

$$i2 := \int_0^1 \frac{1}{x} dx$$

Spočítáme limitu pro  $x$  jdoucí k 0 zleva pro první z dílčích integrálů.

> `limit(value(i1), x=0, left) ;`

$-\infty$

Limita je nevlastní, a proto i původní integrál diverguje.

> `value(i) ;`

*undefined*

>  
>

## 5.4 Aplikace integrálu

Tento odstavce se věnuje aplikacím Riemannova integrálu

### Obsah rovinných obrazců

Nechť  $g(x)$  a  $h(x)$  jsou funkce spojité na intervalu  $\langle a, b \rangle$  a platí  $g(x) < h(x)$  na  $\langle a, b \rangle, a \neq b$ .

Označme  $M = \{(x, y) \in \mathbb{E}_2 \mid (a \leq x \leq b) \wedge (g(x) \leq y \leq h(x))\}$ . Pak obsah obrazce  $M$  je roven

$$P(M) = \int_a^b (h(x) - g(x)) dx.$$

### Poznámky:

- Je-li funkce  $h(x)$  dána parametrickými rovnicemi  $x = \varphi(t), t = \psi(t), t \in \langle \alpha, \beta \rangle$ , které jsou spojité a platí  $\varphi'(t) \neq 0$  na  $\langle \alpha, \beta \rangle$  a funkce  $g(x) = 0$ . Pak substitucí  $x = \varphi(t)$  dostáváme

$$P(M) = \int_a^b h(x) dx = \int_{\alpha}^{\beta} \psi(t) \varphi'(t) dt$$

- Je-li obrazec  $M$  zadán v polárních souřadnicích nerovnicemi  $\alpha \leq \varphi \leq \beta, 0 \leq \rho \leq f(\varphi)$ , kde  $f(\varphi)$  je spojitá na  $\langle \alpha, \beta \rangle$ , pak jeho obsah určíme vzorcem  $P(M) = \frac{1}{2} \int_{\alpha}^{\beta} f^2(\varphi) d\varphi$ .

V této části budeme používat příkazy pro vykreslování grafů funkcí, proto je vhodné nejdříve prostudovat kapitolu *Základy grafického zpracování*.

>

### Příklad 1

Určeme obsah obrazce, který je ohraničen parabolami.

> **restart:with(plots):with(student):**

Warning, the name changecoords has been redefined

Paraboly jsou zadány rovnicemi

> **p1:=3\*y=x^2-2;**

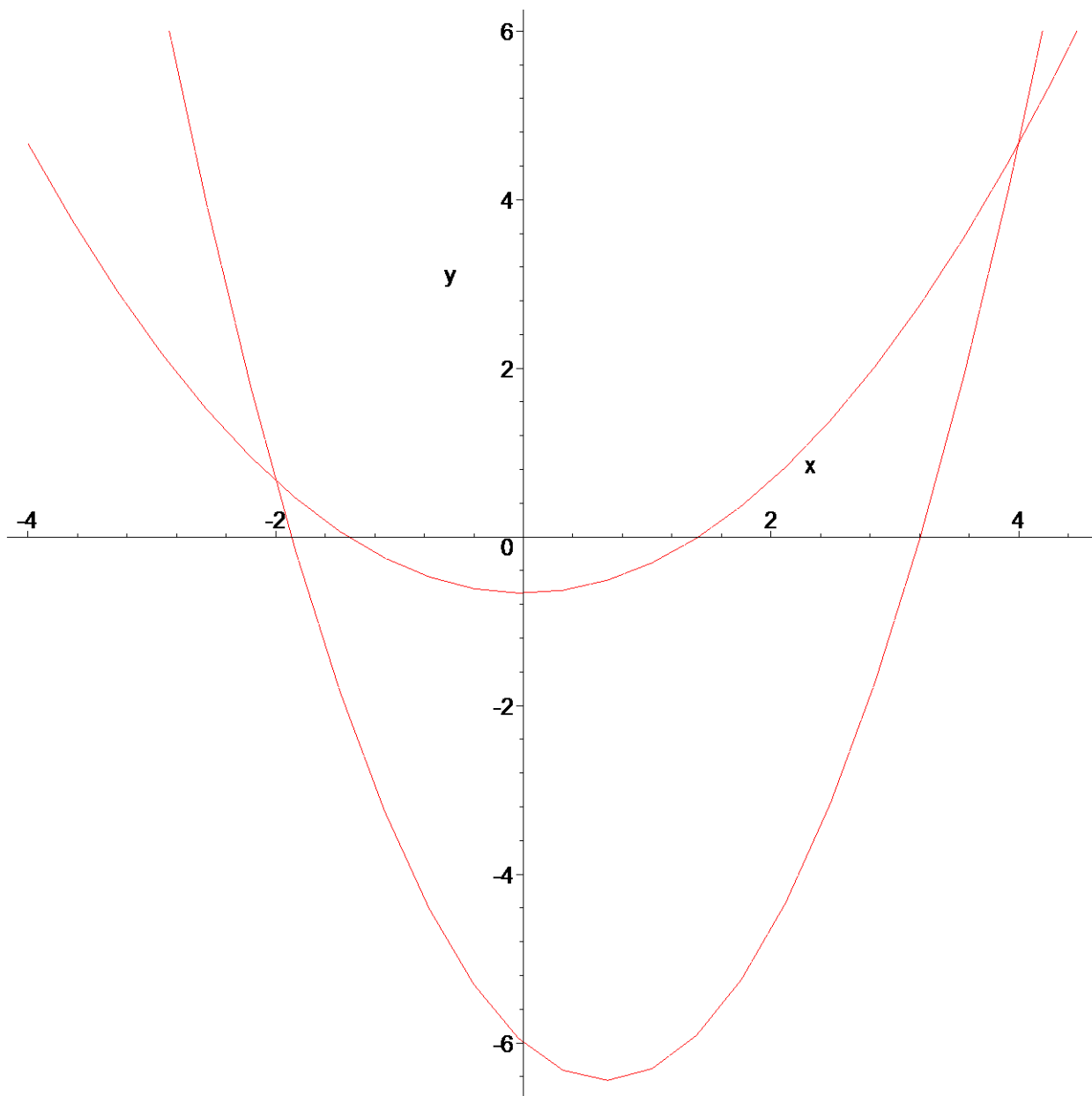
**p2:=y=x^2-4/3\*x-6;**

$$p1 := 3y = x^2 - 2$$

$$p2 := y = x^2 - \frac{4}{3}x - 6$$

Vykreslíme graf funkcí

> **implicitplot({p1,p2},x=-4..5,y=-7..6);**



Určíme průsečíky obou parabol

> **intercept (p1, p2) ;**

$$\{x = -2, y = \frac{2}{3}\}, \{x = 4, y = \frac{14}{3}\}$$

Vyjádříme y z obou rovnic

> **h := isolate (p1, y) ;**

$$h := y = \frac{x^2}{3} - \frac{2}{3}$$

> **g := isolate (p2, y) ;**

$$g := y = x^2 - \frac{4}{3}x - 6$$

Vytvoříme argument pro dosazení do vzorce

> **F := unapply (rhs (h-g), x) ;**

$$F := x \rightarrow -\frac{2}{3}x^2 + \frac{16}{3} + \frac{4}{3}x$$

Počítáme tedy integrál

```
> i1:=Int(F(x),x=-2..4);
```

$$i1 := \int_{-2}^4 -\frac{2}{3}x^2 + \frac{16}{3} + \frac{4}{3}x dx$$

Obsah tedy je

```
> value(%);
```

24

```
>
```

### Příklad 2

Určete obsah obrazce, který je ohraničen osou  $y$  a semikubickou parabolou.

```
> restart:with(plots):with(student):
```

Warning, the name changecoords has been redefined

Rovnice semikubické paraboly

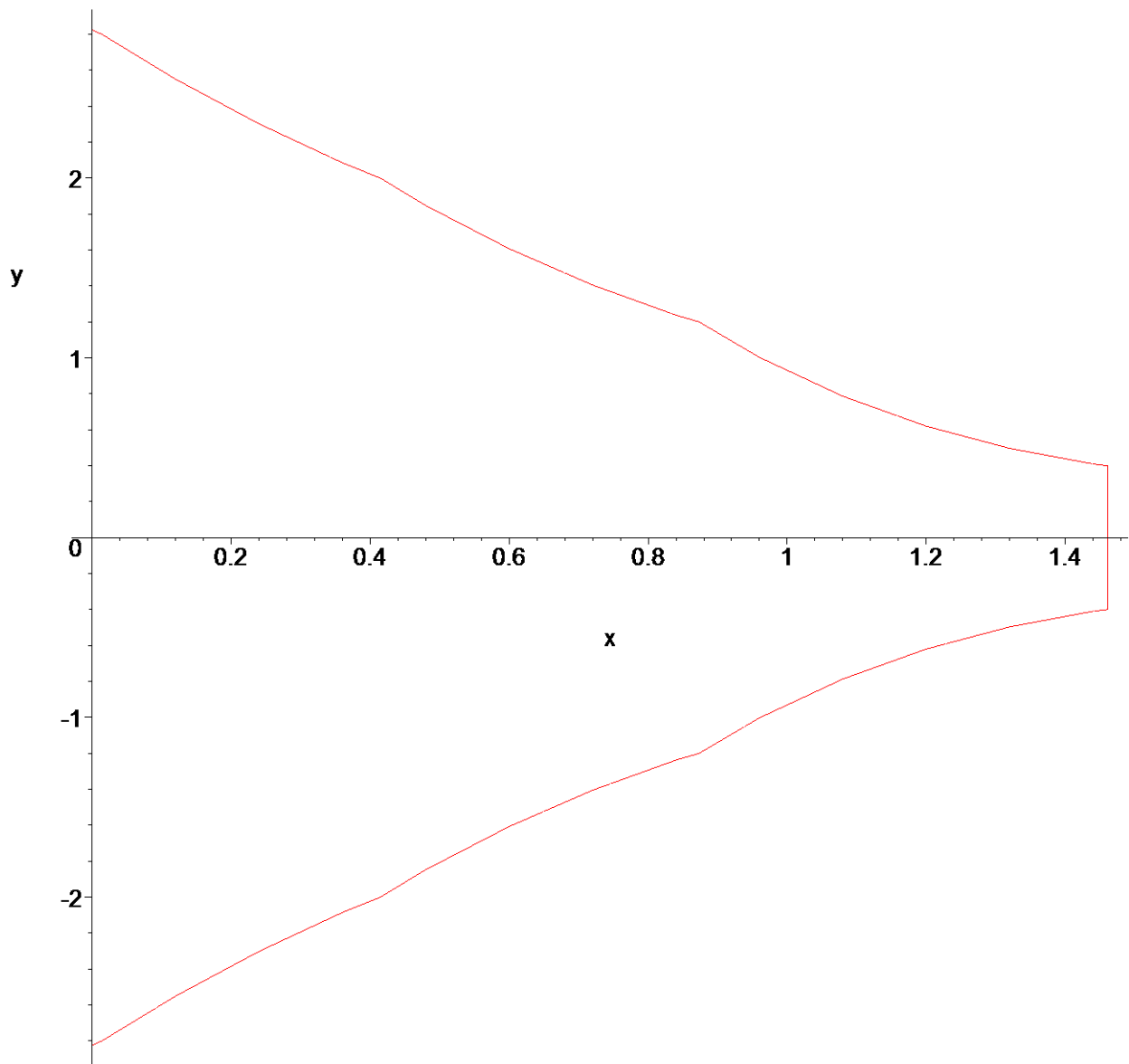
```
> p:=[y^2=(a-x)^3];
```

$$p := [y^2 = (a - x)^3]$$

```
> #assume(a>0);
```

Pokus o vykreslení funkce dané implicitně. Je vidět, že systém Maple má určité problémy. Pro nakreslení grafu je samozřejmě nutné zvolit pevné  $a$ .

```
> implicitplot(y^2=(2-x)^3,x=0..3,y=-10..10);
```



Rozdělíme předepsanou funkci na dvě

```
> h:=sqrt((a-x)^3);
```

$$h := \sqrt{(a-x)^3}$$

```
> g:=-sqrt((a-x)^3);
```

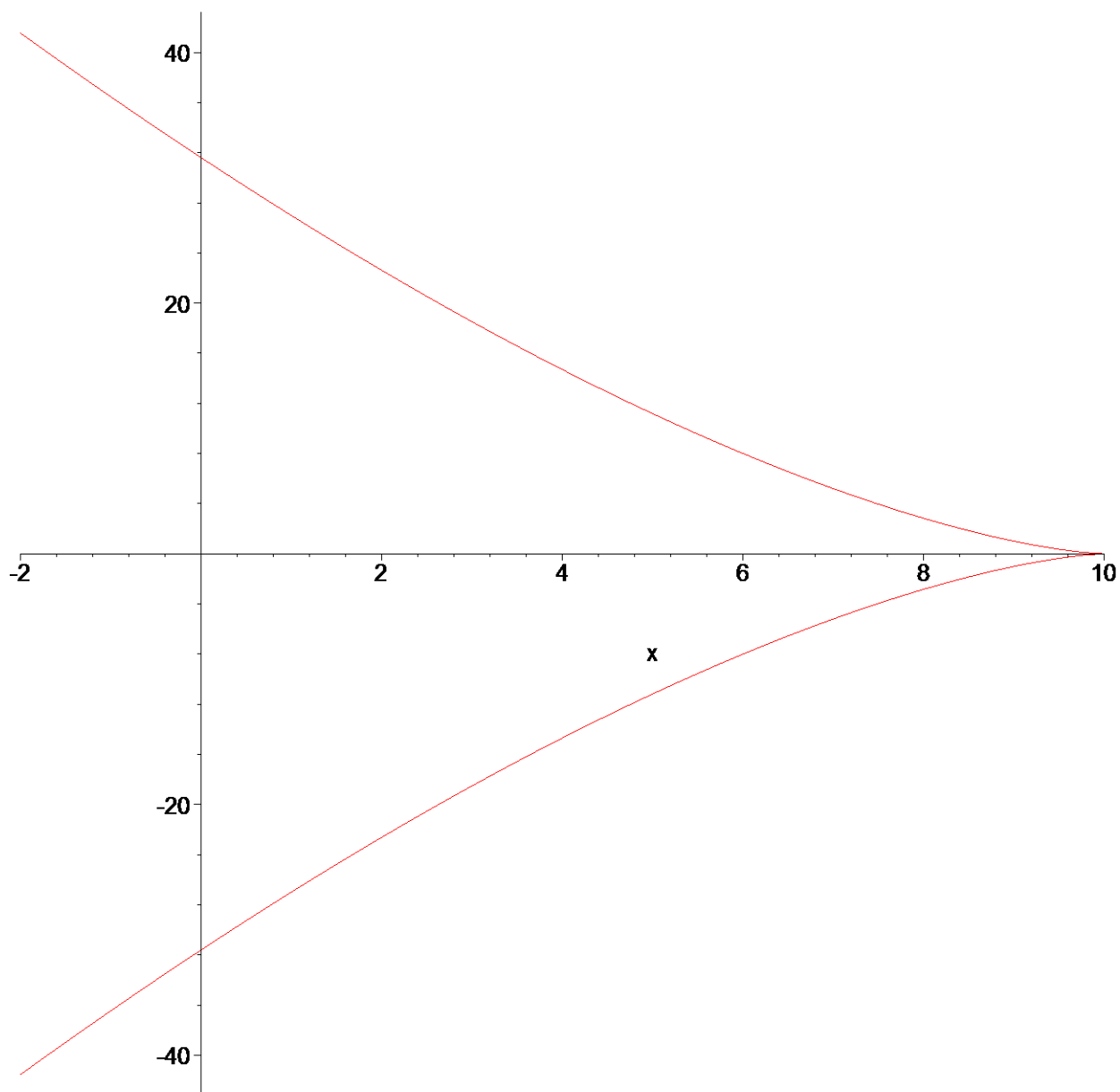
$$g := -\sqrt{(a-x)^3}$$

Vykreslíme graf funkcí  $h, g$  pro pevně zvolené  $a=10$ .

```
> p1:=plot(sqrt((10-x)^3),x=-2..10):
```

```
> p2:=plot(-sqrt((10-x)^3),x=-2..10):
```

```
> display({p1,p2});
```



Určíme průsečík s osou  $x$ , tj.  $y=0$ . Příkaz **intercept** spočítá průsečík dvou křivek.

> **intercept (h=0, y=0) ;**

$$\{y = 0, a = x, x = x\}$$

Počítáme tedy integrál

> **i2 := Int (h-g, x=0..a) ;**

$$i2 := \int_0^a 2\sqrt{(a-x)^3} dx$$

Výsledný obsah

> **value (i2) ;**

$$\frac{4}{5} a \sqrt{a^3}$$

>

### Příklad 3

Určete obsah obrazce ohraničeného obloukem cykloidy a osou  $x$ .

> **restart:with (plots) :with (student) :**



Warning, the name changecoords has been redefined

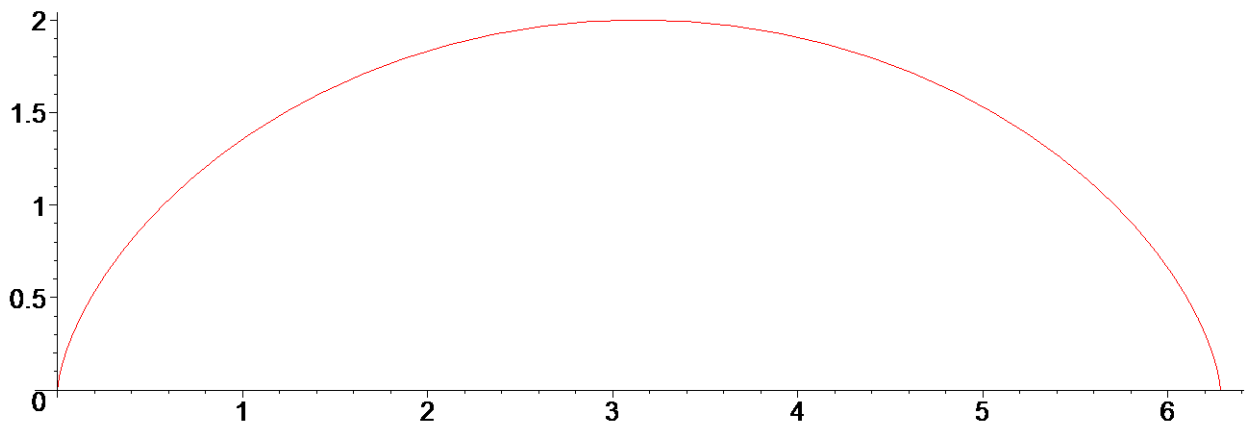
Rovnice cykloidy

```
> rce := [r*(t-sin(t)), r*(1-cos(t))];
#assume(0<=t, t<=2*Pi);
```

```
rce := [r(t-sin(t)), r(1-cos(t))]
```

Nakreslení grafu. Rovnice jsou děleny v argumentu  $r$ , abychom dostali konkrétní vyjádření.

```
> plot([rce[1]/r, rce[2]/r, t=0..2*Pi], scaling=constrained);
```



Užijeme vzorce pro výpočet obsahu obrazce ohraničeným funkcemi danými parametricky

```
> P(M) = Int(psi(t)*D(phi(t)), t=alpha..beta);
```

$$P(M) = \int_{\alpha}^{\beta} \psi(t) D(\phi(t)) dt$$

Dosazením dostáváme

```
> i3 := Int(rce[2]*diff(rce[1], t), t=0..2*Pi);
```

$$i3 := \int_0^{2\pi} r^2 (1 - \cos(t))^2 dt$$

Spočteme obsah

```
> value(i3);
```

$$3 \pi r^2$$

```
>
```

```
>
```

#### Příklad 4

Určete obsah obrazce 1.kvadrantu ohraničeného Archimedovou spirálou a přímkami

```
> restart:with(plots):with(student):
```

Warning, the name changecoords has been redefined

Omezení na plochu

```
> p:=rho=a*phi; # archimedova spirála
```

```
phi=Pi/3; # přímka
```

```
phi=Pi/2; # přímka
```

$$p := \rho = a \phi$$

$$\phi = \frac{\pi}{3}$$

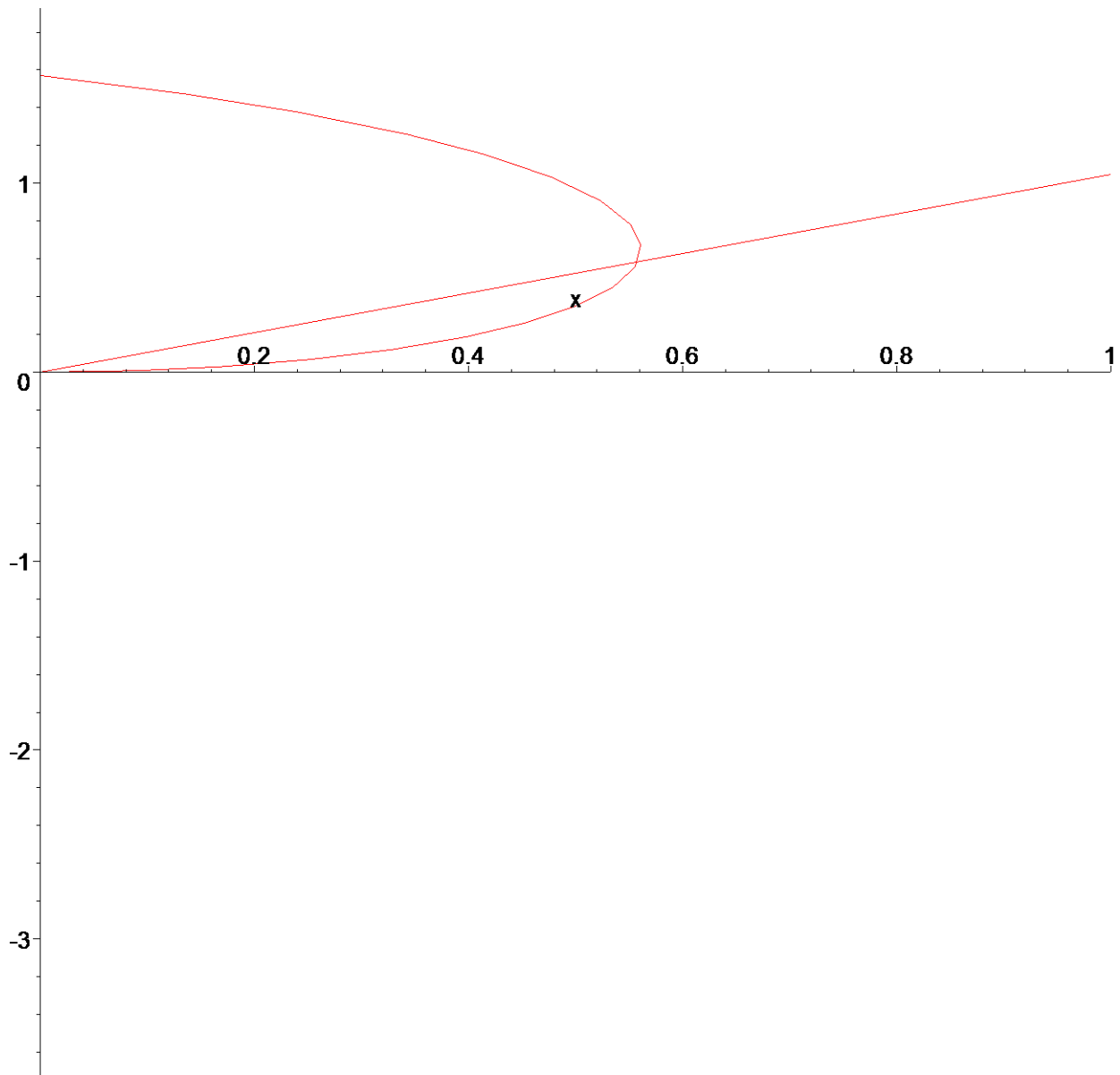
$$\phi = \frac{\pi}{2}$$

Vykreslení plochy

```
> p1:=polarplot(phi,phi=0..4/3*Pi):
```

```
> p12:=plot(Pi/3*x,x=0..1):
```

```
> display({p11,p12});
```



Dosadíme-li do odpovídajícího intergrálu dostaneme

> `i4:=1/2*Int(a^2*phi^2,phi=Pi/3..Pi/2);`

$$i4 := \frac{1}{2} \int_{\frac{\pi}{3}}^{\frac{\pi}{2}} a^2 \phi^2 d\phi$$

Výsledná plocha je

> `value(%);`

$$\frac{19 a^2 \pi^3}{1296}$$

>  
>

## Délka křivky

Při výpočtu délky  $l$  křivky  $C$  vycházíme ze vztahů

- $l = \int_a^b \sqrt{1 + [f'(x)]^2} dx$  v případě, kdy rovinná křivka  $C$  je zadána rovnicí  $y = f(x), a \leq x \leq b$ , kde  $f$  je funkce spojitá zároveň se svojí první derivací v uzavřeném intervalu  $\langle a, b \rangle$ .

- $l = \int_\alpha^\beta \sqrt{\dot{\varphi}^2(t) + \dot{\psi}^2(t) + \dot{\chi}^2(t)} dt$  v případě, kdy prostorová křivka  $C$  je zadána parametrickými rovnicemi  $x = \varphi(t), y = \psi(t), z = \chi(t), t \in \langle \alpha, \beta \rangle$ , kde vektorová funkce  $\vec{d}(t) = [\varphi(t), \psi(t), \chi(t)]$  je spojitá zároveň se svojí první derivací  $\dot{\vec{d}}$  v  $\langle \alpha, \beta \rangle$ , přičemž  $\dot{\vec{d}} \neq [0, 0, 0], \forall t \in \langle \alpha, \beta \rangle$ .

- $l = \int_\alpha^\beta \sqrt{f^2(\theta) + [f'(\theta)]^2} d\theta$  v případě, kdy rovinná křivka  $C$  je zadána rovnicí  $\gamma = f(\theta), \alpha \leq \theta \leq \beta$  v polárních souřadnicích, kde  $f$  je funkce spojitá s první derivací v intervalu  $\langle \alpha, \beta \rangle$ .

>

### Příklad 1

Vypočítejte délku grafu funkce  $\ln$  mezi danými body

```
> restart:with(plots):with(student):
```

```
Warning, the name changecoords has been redefined
```

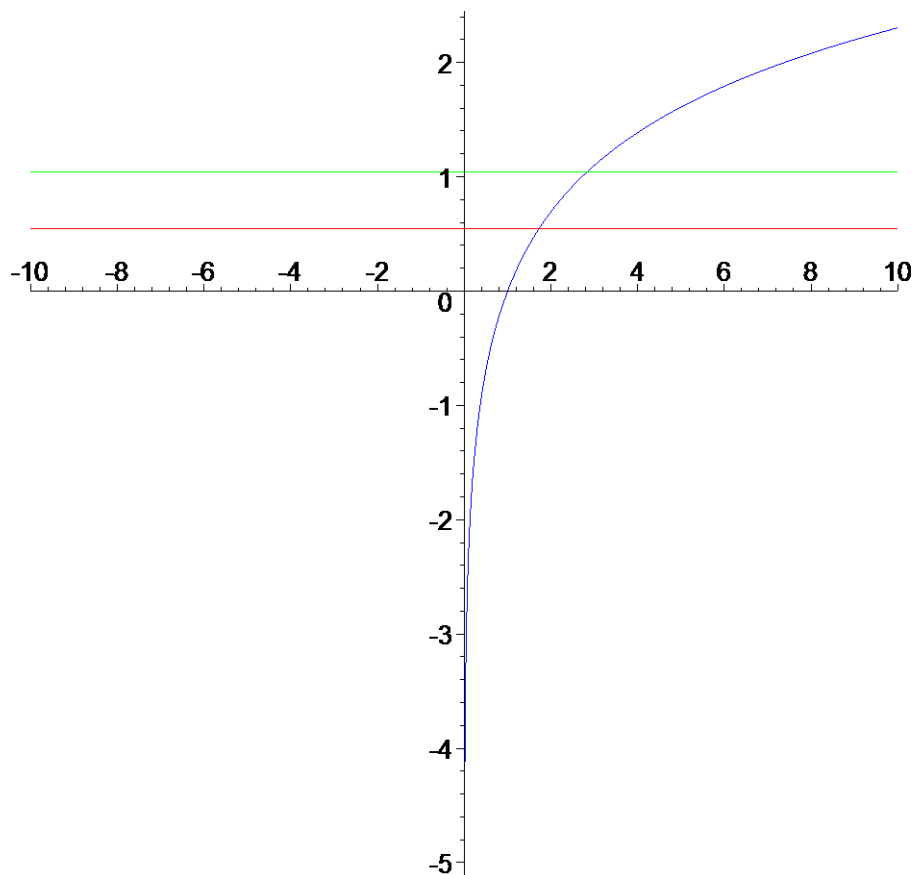
Zadané body

```
> body := [sqrt(3), ln(sqrt(3))], [sqrt(8), ln(sqrt(8))];
```

$$body := \left[ \sqrt{3}, \frac{1}{2} \ln(3) \right], [2\sqrt{2}, \ln(2\sqrt{2})]$$

Počítáme délku křivky mezi dvěma vodorovnými přímkami

```
> p1:=plot(ln(x), x=0..10, color=[blue], legend=`ln(x)`):
p2:=plot([1/2*ln(3), ln(2*2^(1/2))], legend=[`1/2*ln(3)`, `ln(2*2^(1/2))`]):
display({p1,p2});
```



|                                                                 |                        |
|-----------------------------------------------------------------|------------------------|
| <hr style="border: 1px solid red; width: 100px; margin: 0;"/>   | $\frac{1}{2} \ln(3)$   |
| <hr style="border: 1px solid green; width: 100px; margin: 0;"/> | $\ln(2 \cdot 2^{1/2})$ |
| <hr style="border: 1px solid blue; width: 100px; margin: 0;"/>  | $\ln(x)$               |

Dosadíme do prvního vzorce

> `il := Int(sqrt(1 + (D(ln)(x))^2), x = body[1][1] .. body[2][1]);`

$$il := \int_{\sqrt{3}}^{2\sqrt{2}} \sqrt{1 + \frac{1}{x^2}} dx$$

Upravíme na

> `ilp := Int(sqrt(1+x^2)/x, x = body[1][1] .. body[2][1]);`

$$ilp := \int_{\sqrt{3}}^{2\sqrt{2}} \frac{\sqrt{1+x^2}}{x} dx$$

Zavedeme substituci

> `changevar(1+x^2=u, ilp, u);`

$$\int_4^9 \frac{\sqrt{u}}{2(-1+u)} du$$

Délka křivky je rovna

> **d1:=value(%);**

$$d1 := 1 - \frac{1}{2} \ln(2) + \frac{1}{2} \ln(3)$$

Přímý výpočet integrálu

> **d2:=value(ilp);**

$$d2 := 1 - \operatorname{arctanh}\left(\frac{1}{3}\right) + \operatorname{arctanh}\left(\frac{1}{2}\right)$$

Zjednodušíme a převedeme na členy s  $\ln$

> **simplify(convert(d2,ln));**

$$1 - \frac{1}{2} \ln(2) + \frac{1}{2} \ln(3)$$

>

>

#### **Příklad 2**

Vypočítejte délku asteroidy

> **restart:with(plots):with(student):**

Warning, the name changecoords has been redefined

Parametrické rovnice jsou

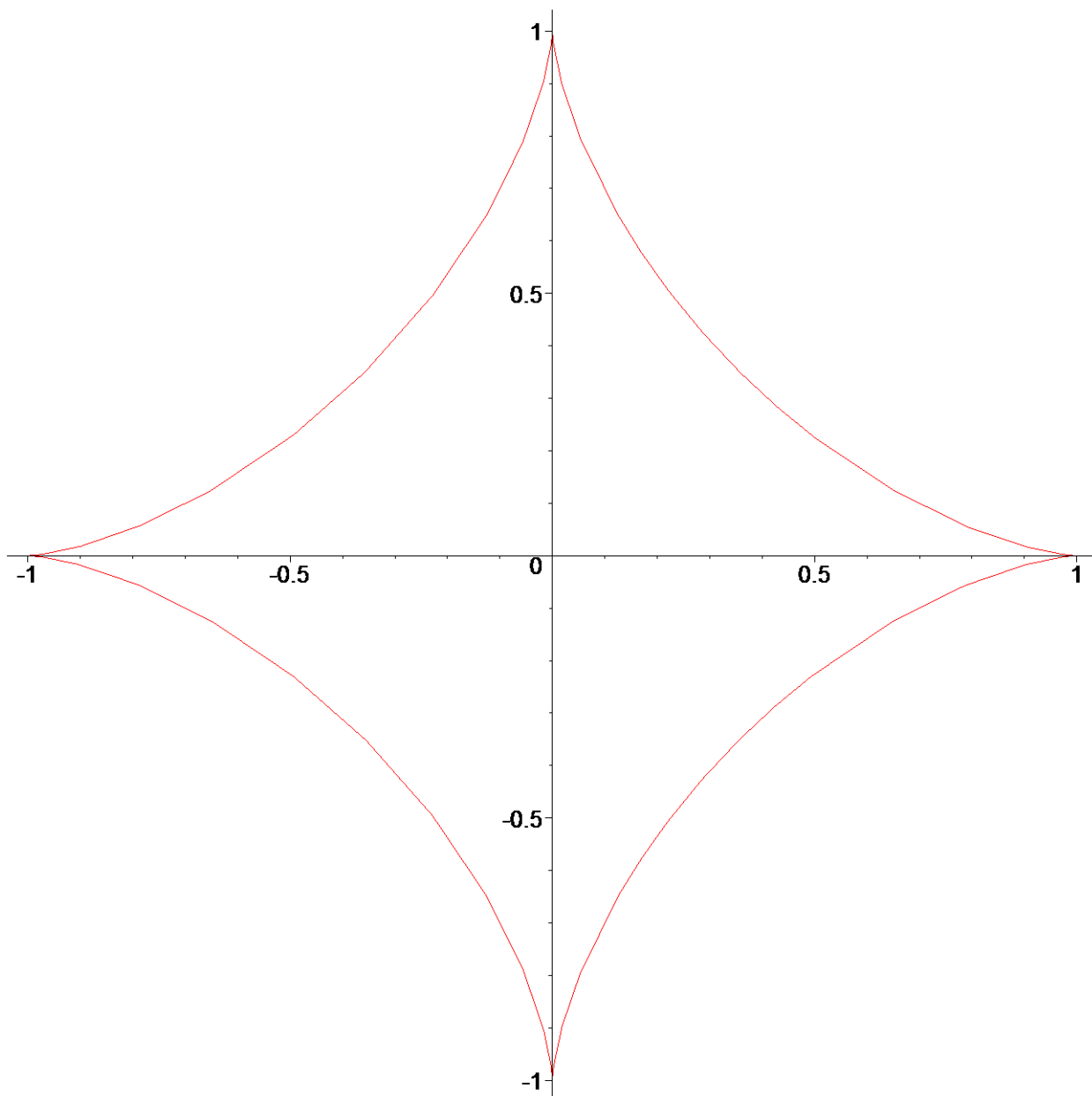
> **f:=[a\*cos(x)^3,a\*sin(x)^3];**

$$f := [a \cos(x)^3, a \sin(x)^3]$$

a proto použijeme druhý vzorec

Graf asteroidy pro  $a=1$

> **plot([cos(x)^3,sin(x)^3,x=-Pi..Pi]);**



Konstanta  $a > 0$

> **assume(a, positive);**

Křivka je symetrická, proto budeme počítat pouze integrál

> **i2 := 4 \* Int(sqrt(Diff(f[1], x)^2 + Diff(f[2], x)^2), x = 0 .. Pi/2);**

$$i2 := 4 \int_0^{\frac{\pi}{2}} \sqrt{\left(\frac{\partial}{\partial x}(a \cos(x)^3)\right)^2 + \left(\frac{\partial}{\partial x}(a \sin(x)^3)\right)^2} dx$$

Výsledek je

> **simplify(value(%));**

$6a$

>

>

### Příklad 3

Délka oblouku cykloidy na  $\langle 0, 2\pi \rangle$

```
> restart:with(plots):with(student):
Warning, the name changecoords has been redefined
```

Parametrizace

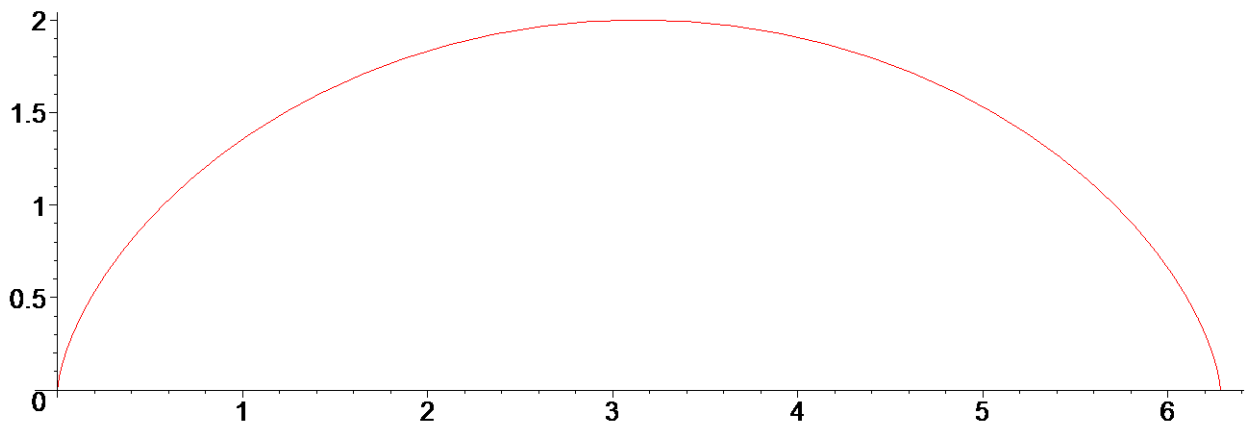
```
> f:=[a*(x-sin(x)),a*(1-cos(x))];
f:= [a(x - sin(x)), a(1 - cos(x))]
```

Nechť  $a > 0$

```
> assume(a>0);
```

Vykreslíme graf funkce

```
> plot([(t-sin(t)), (1-cos(t)), t=0..2*Pi], scaling=constrained);
```



Počítáme integrál

```
> i3:=Int(sqrt(Diff(f[1],x)^2+Diff(f[2],x)^2),x=0..2*Pi);
```

$$i3 := \int_0^{2\pi} \sqrt{\left(\frac{\partial}{\partial x}(a(x - \sin(x)))\right)^2 + \left(\frac{\partial}{\partial x}(a(1 - \cos(x)))\right)^2} dx$$



Délka křivky je

```
> value(%);
```

8 a~

```
>
```

```
>
```

#### Příklad 4

Vypočítejte délku kardioidy zadané v polárních souřadnicích rovnicí

```
> restart:with(student):with(plots):
```

Warning, the name changecoords has been redefined

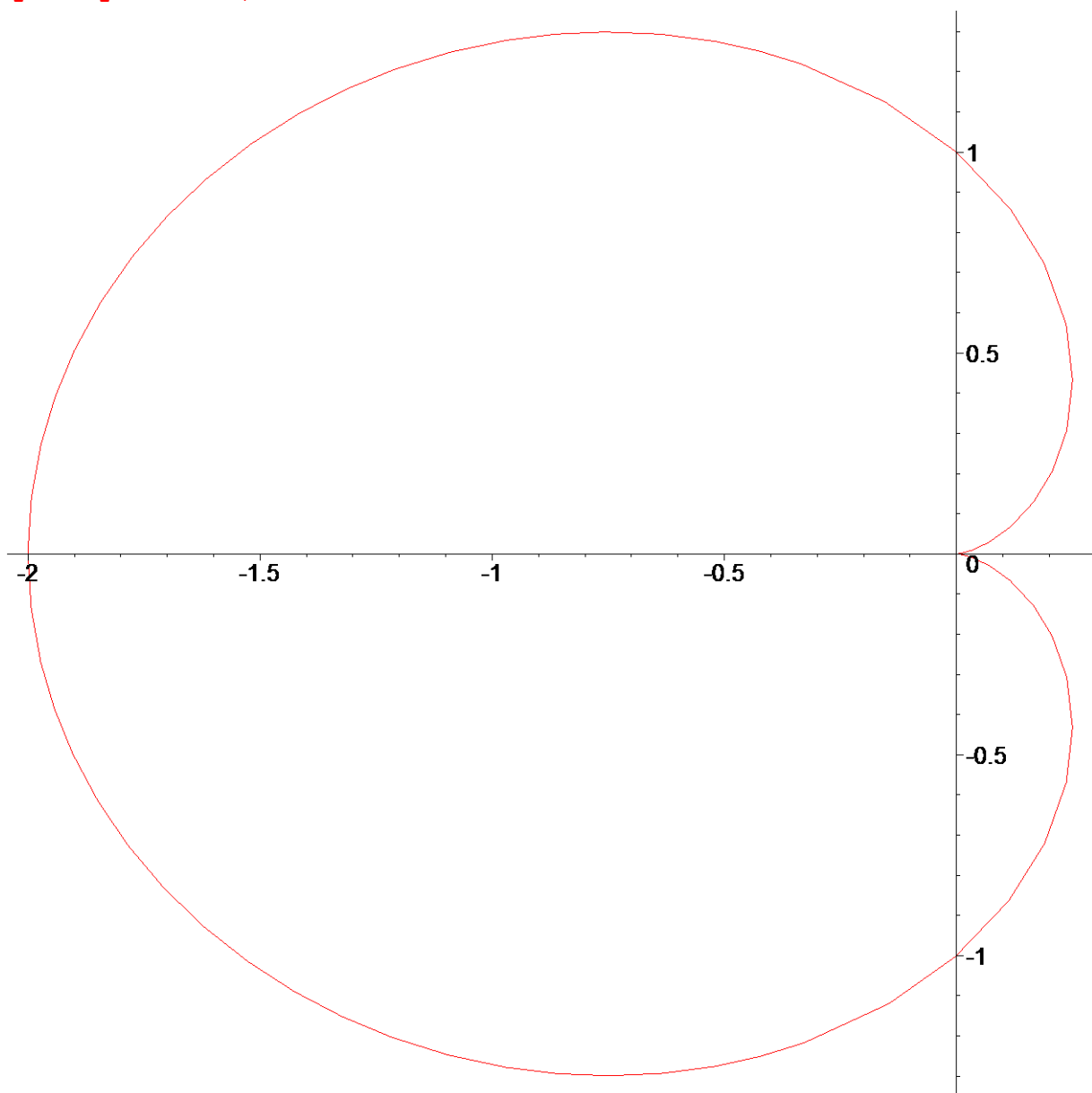
```
> rce:=1-cos(theta);
```

$rce := 1 - \cos(\theta)$

pro  $0 \leq \theta \leq 2\pi$ .

Nakreslíme obrázek

```
> polarplot(rce,theta=0..2*Pi);
```



Použijeme třetí vzorec. Po dosazení dostáváme

```
> Int(sqrt(rce^2+(Diff(rce,theta)^2)),theta=0..2*Pi);
```

$$\int_0^{2\pi} \sqrt{(1 - \cos(\theta))^2 + \left(\frac{d}{d\theta}(1 - \cos(\theta))\right)^2} d\theta$$

Spočteme délku této křivky

> **value(%)** ;

8

>

#### Příklad 5

Vypočítejte délku Archimedovy spirály

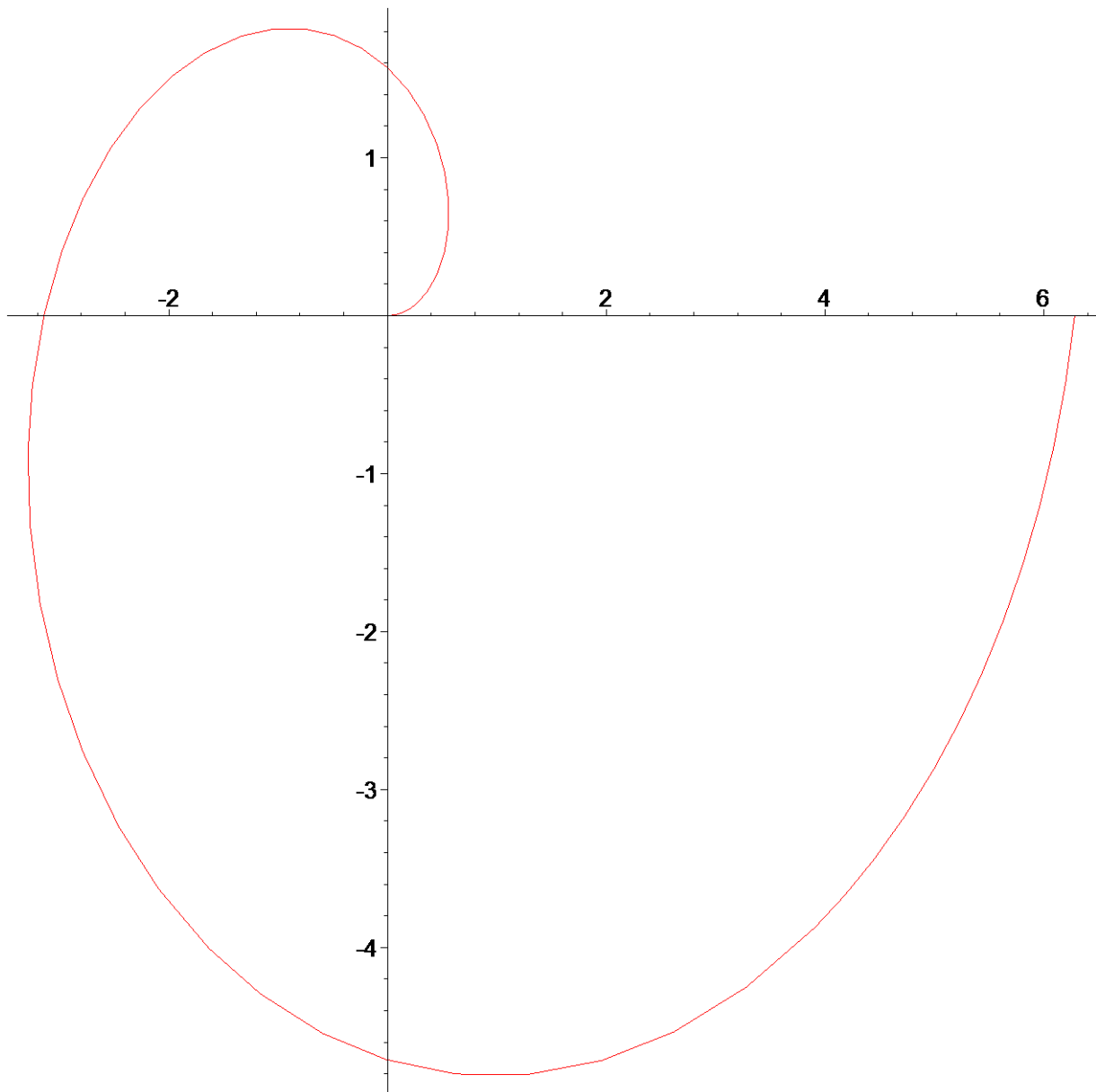
> **rce:=a\*theta** ;

*rce := a θ*

pro  $a > 0$  od počátku do konce prvního závitu.

Nakresleme obrázek pro  $a=1$

> **polarplot(theta, theta=0..2\*Pi)** ;



Použijeme třetí vzorec. Po dosazení dostáváme

> Int(sqrt(rce^2+(Diff(rce,theta)^2)),theta=0..2\*Pi);

$$\int_0^{2\pi} \sqrt{a^2 \theta^2 + \left(\frac{\partial}{\partial \theta}(a \theta)\right)^2} d\theta$$

Délka prvního závitu spirály je

> assume(a>0):

simplify(value(%));

$$\frac{1}{2} a \left( 2\pi \sqrt{4\pi^2 + 1} + \ln(2\pi + \sqrt{4\pi^2 + 1}) \right)$$

>

>

## Objem rotačního tělesa

Při výpočtu objemu  $|V|$  tělesa  $V$  vycházíme ze vztahů

- $|V| = \pi \int_a^b f^2(x) dx$  v případě, kdy těleso  $V$  vzniká rotací rovinné oblasti

$A = \{(x, y) \in \mathbb{R}_2 \mid a \leq x \leq b, 0 \leq y \leq f(x)\}$  kolem osy  $O_x$ , kde  $f$  je nezáporná spojitá funkce v intervalu  $\langle a, b \rangle$ .

- $|V| = 2\pi \int_\alpha^\beta y(f(y) - g(y)) dy$  v případě, kdy těleso  $V$  vzniká rotací rovinné oblasti

$B = \{(x, y) \in \mathbb{R}_2 \mid g(y) \leq x \leq f(y), \alpha \leq y \leq \beta\}$  kolem osy  $O_x$ , kde  $g(y), f(y)$  jsou spojitě funkce definované v intervalu  $\langle \alpha, \beta \rangle$ ,  $\alpha \geq 0$  takové, že  $g(y) \leq f(y), \forall y \in \langle \alpha, \beta \rangle$ .

>

### Příklad 1

Hledáme objem koule o poloměru  $r$ . Kouli dostaneme rotací poloviny kruhu

$$P = \{(x, y) \in \mathbb{R}_2 \mid -r \leq x \leq r, 0 \leq y \leq \sqrt{r^2 - x^2}\}$$

kolem osy  $O_x$ .

> **restart:with(student):with(plots):**

Warning, the name changecoords has been redefined

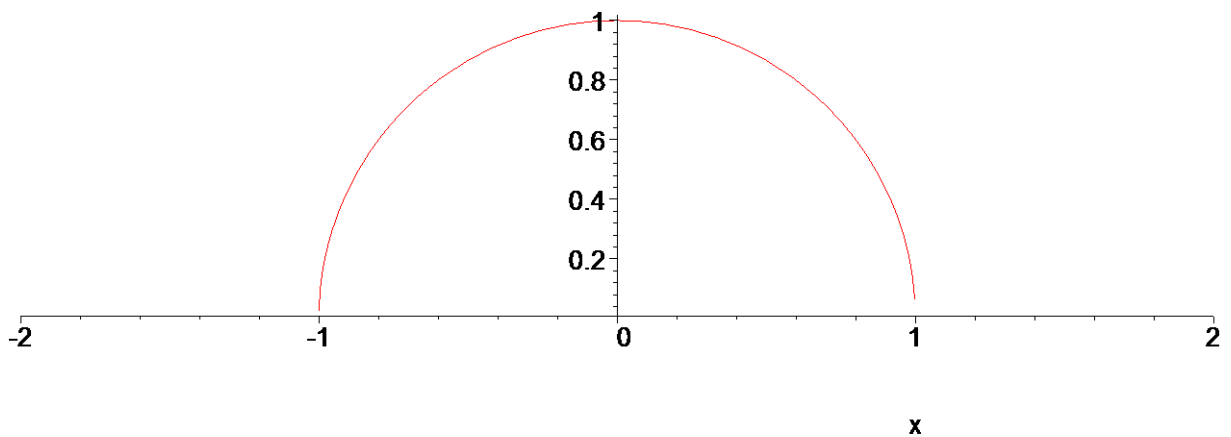
Rovnice poloroviny kruhu

> **f:=sqrt(r^2-x^2);**

$$f := \sqrt{r^2 - x^2}$$

Nakresleme obrázek pro  $r=1$

> **plot(sqrt(1-x^2), x=-2..2, scaling=constrained);**



Dosadíme do prvního vztahu

```
> Int(Pi*f^2,x=-r..r);
```

$$\int_{-r}^r \pi (r^2 - x^2) dx$$

Výsledný objem je roven

```
> value(%);
```

$$\frac{4 \pi r^3}{3}$$

```
>
```

```
>
```

### Příklad 2

Vypočítejte objem tělesa  $V$  vzniklého rotací množiny  $A$  ležící mezi křivkami

```
> restart:with(plots):
```

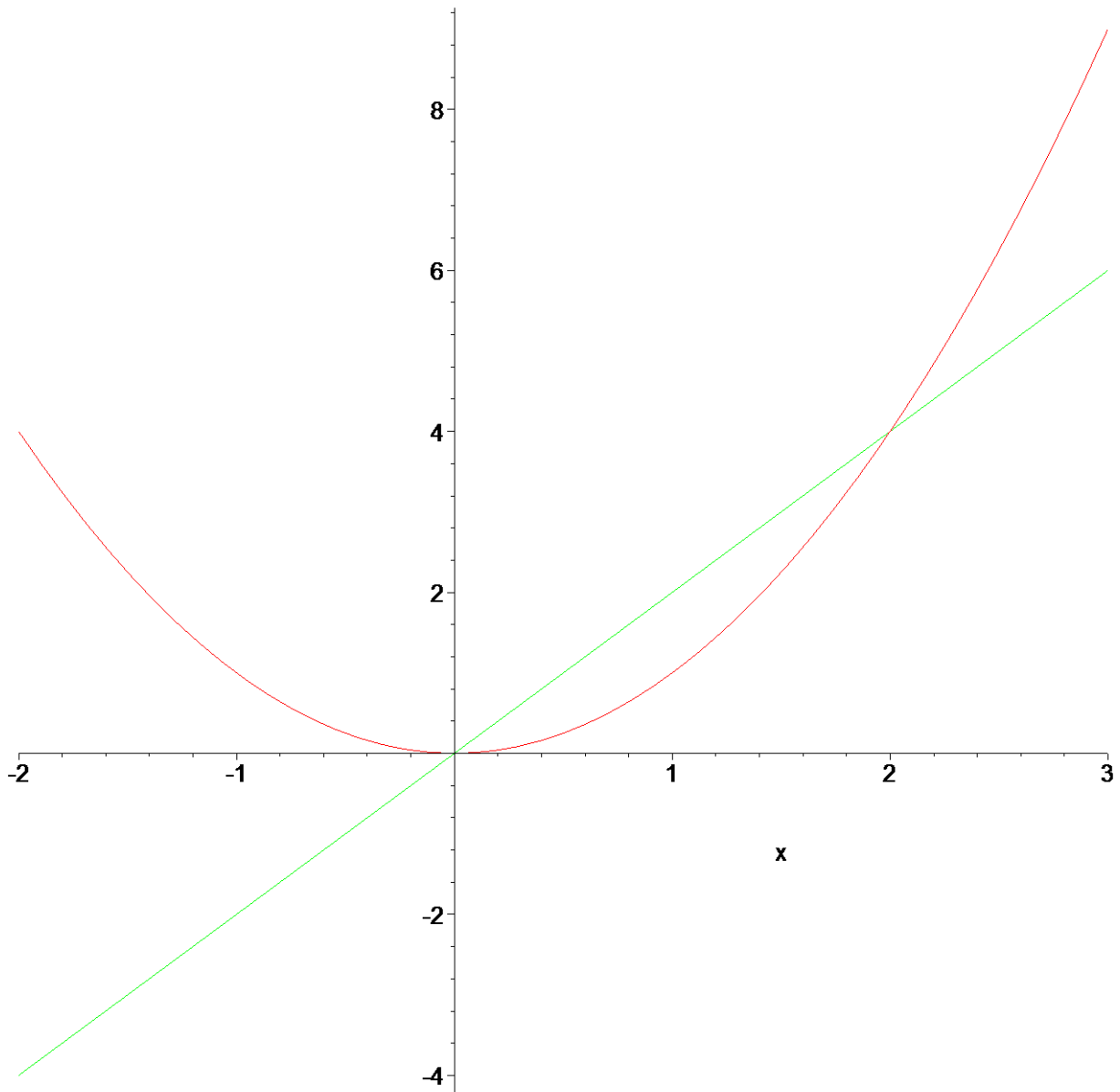
Warning, the name changecoords has been redefined

```
> rce:=y=x^2,y=2*x;
```

$$rce := y = x^2, y = 2x$$

Nakreslíme obrázek

```
> plot([x^2, 2*x], x=-2..3);
```



Odečteme objem vzniklý rotací křivky  $y=x^2$  od objemu vyniklého rotací křivky  $y=2x$ .

Dosazením do prvního vztahu dostáváme

```
> Int(Pi*(2*x)^2, x=0..2) - Int(Pi*(x^2)^2, x=0..2);
```

$$\int_0^2 4\pi x^2 dx - \int_0^2 \pi x^4 dx$$

Výsledný objem je roven

```
> value(%);
```

$$\frac{64\pi}{15}$$

```
>
>
```

*Příklad 3*

Hledáme objem koule o poloměru  $r$ . Kouli dostaneme rotací rovinné množiny

$$B = \left\{ (x, y) \in \mathbb{R}_2 \mid -\sqrt{r^2 - y^2} \leq x \leq \sqrt{r^2 - y^2}, 0 \leq y \leq r \right\} \text{ kolem osy } O_x.$$

```
> restart:with(student):with(plots):
```

Warning, the name changecoords has been redefined

```
> rce:=sqrt(r^2-y^2);
```

$$rce := \sqrt{r^2 - y^2}$$

Dosazením do druhého vztahu dostaneme

```
> V:=2*Pi*Int(y*(rce-(-rce)),y=0..r);
```

$$V := 2 \pi \int_0^r 2 y \sqrt{r^2 - y^2} dy$$

Spočteme nyní výsledný objem, který je pochopitelně roven objemu v příkladu 1

```
> assume(r>0);
```

```
value(%);
```

$$\frac{4 \pi r^3}{3}$$

```
>
```

#### Příklad 4

Vypočítejte objem tělesa  $V$  vzniklého rotací kolem osy  $O_y$  množiny ohraničené křivkami

$y = \sqrt{1+x^2}, x = 0, y = 0, x = 1$  (viz. obrázek 1). Zavedeme novou soustavu souřadnic  $u, v$  takovou, že  $u = -y, v = x$ . Oblast  $B$  je pak z hlediska výpočtu stejná jako na obrázku 2 a rotační těleso vzniká její rotací kolem osy  $O_u$ .

```
> restart:with(student):with(plots):
```

Warning, the name changecoords has been redefined

Nakreslíme obrázky

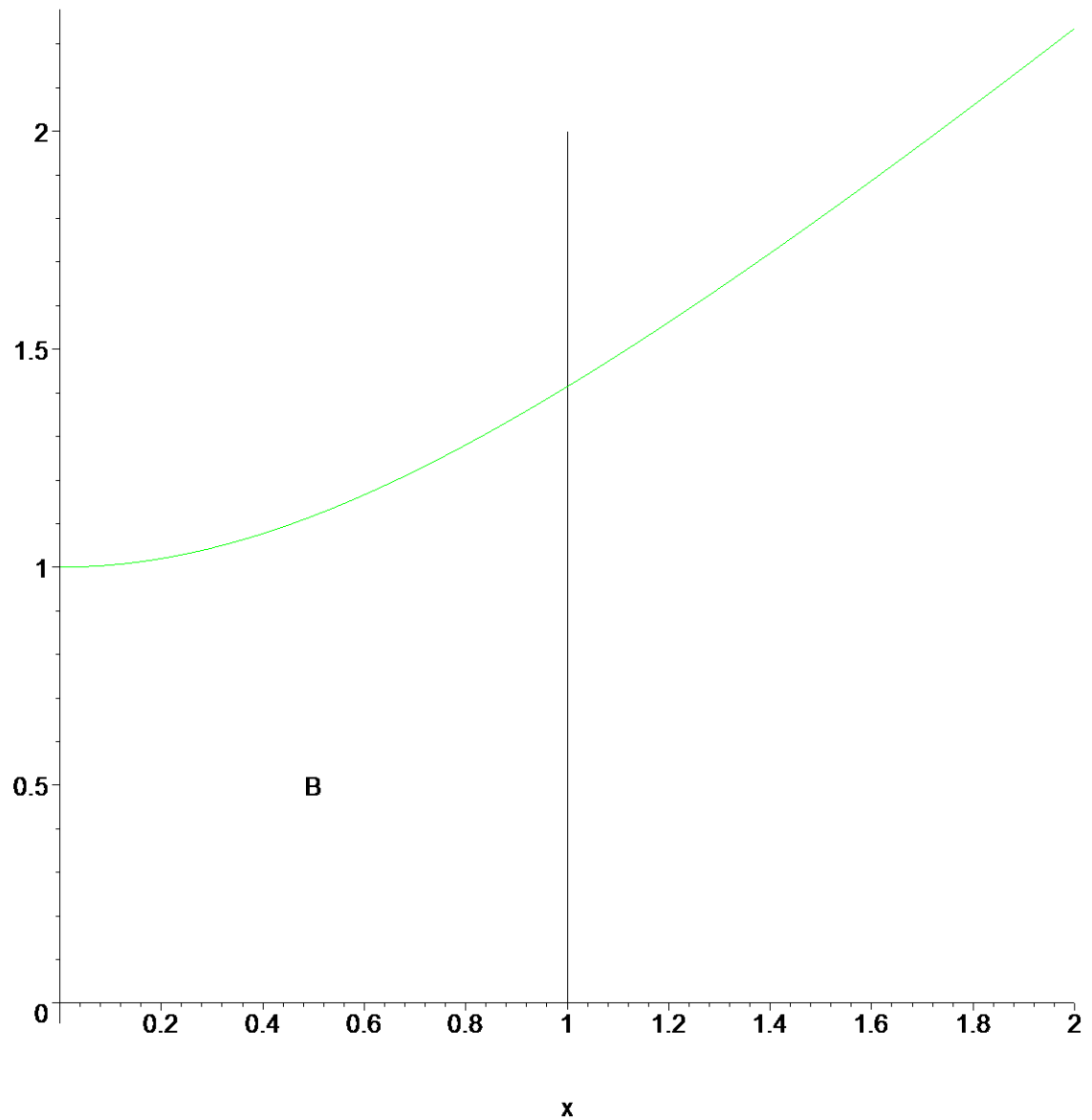
```
> p1:=plot([0,sqrt(1+x^2)],x=0..2):
```

```
p2:=plottools[line]([1,0],[1,2]):
```

```
p3:=textplot([0.5,0.5,`B`]):
```

```
display({p1,p2,p3},title=`Obrázek 1`);
```

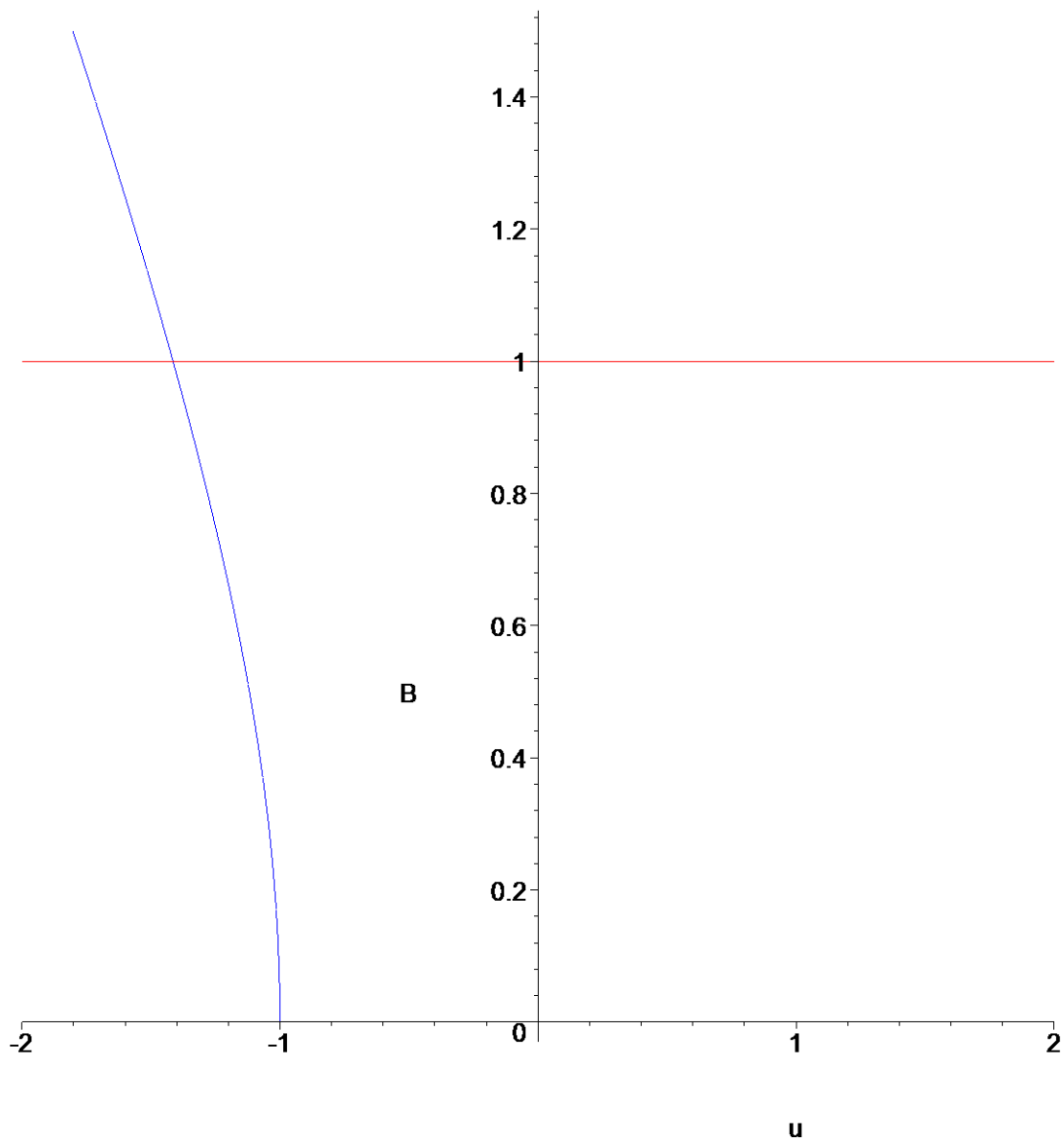
Obrázek 1



```
> p1:=plot(1,u=-2..2):
p2:=implicitplot(u+sqrt(1+v^2),u=-2..2,v=0..1.5,color=blue):
p3:=textplot([-0.5,0.5,`B`]):
display({p1,p2,p3},title=`Obrázek 2`);
```



Obrázek 2



Dosazením do druhého vztahu dostáváme

```
> 2*Pi*Int(v*(0-(-sqrt(1+v^2))),v=0..1);
```

$$2 \pi \int_0^1 v \sqrt{1+v^2} dv$$

Výsledný objem je roven

```
> simplify(value(%));
```

$$\frac{2 \pi (2\sqrt{2} - 1)}{3}$$

>

>

## Obsah rotační plochy

Při výpočtu obsahu  $|S|$  plochy  $S$  vycházíme ze vztahů

- $|S| = 2\pi \int_a^b f(x) \sqrt{1 + [f'(x)]^2} dx$  v případě, kdy plocha  $S$  vzniká rotací grafu  $y = f(x)$ ,  $a \leq x \leq b$

funkce  $f$  kolem osy  $O_x$ , kde funkce  $f$  je nezáporná a spojitá zároveň se svou derivací  $f'(x)$  v uzavřeném intervalu  $\langle a, b \rangle$ .

- $|S| = 2\pi \int_\alpha^\beta y \sqrt{1 + [g'(y)]^2} dy$  v případě, kdy plocha  $S$  vzniká rotací grafu  $x = g(y)$ ,  $\alpha \leq y \leq \beta$

funkce  $g$  kolem osy  $O_x$ , kde funkce  $g$  je spojitá zároveň se svou derivací  $g'(y)$  v uzavřeném intervalu  $\langle \alpha, \beta \rangle$ ,  $\alpha \geq 0$ .

>

### Příklad 1

Odvodíme vzorec pro výpočet obsahu pláště rotačního kužele o poloměru  $r$  a výšce  $h$ . Ukážeme použití obou výše uvedených vztahů.

Nejdříve nakreslíme obrázek

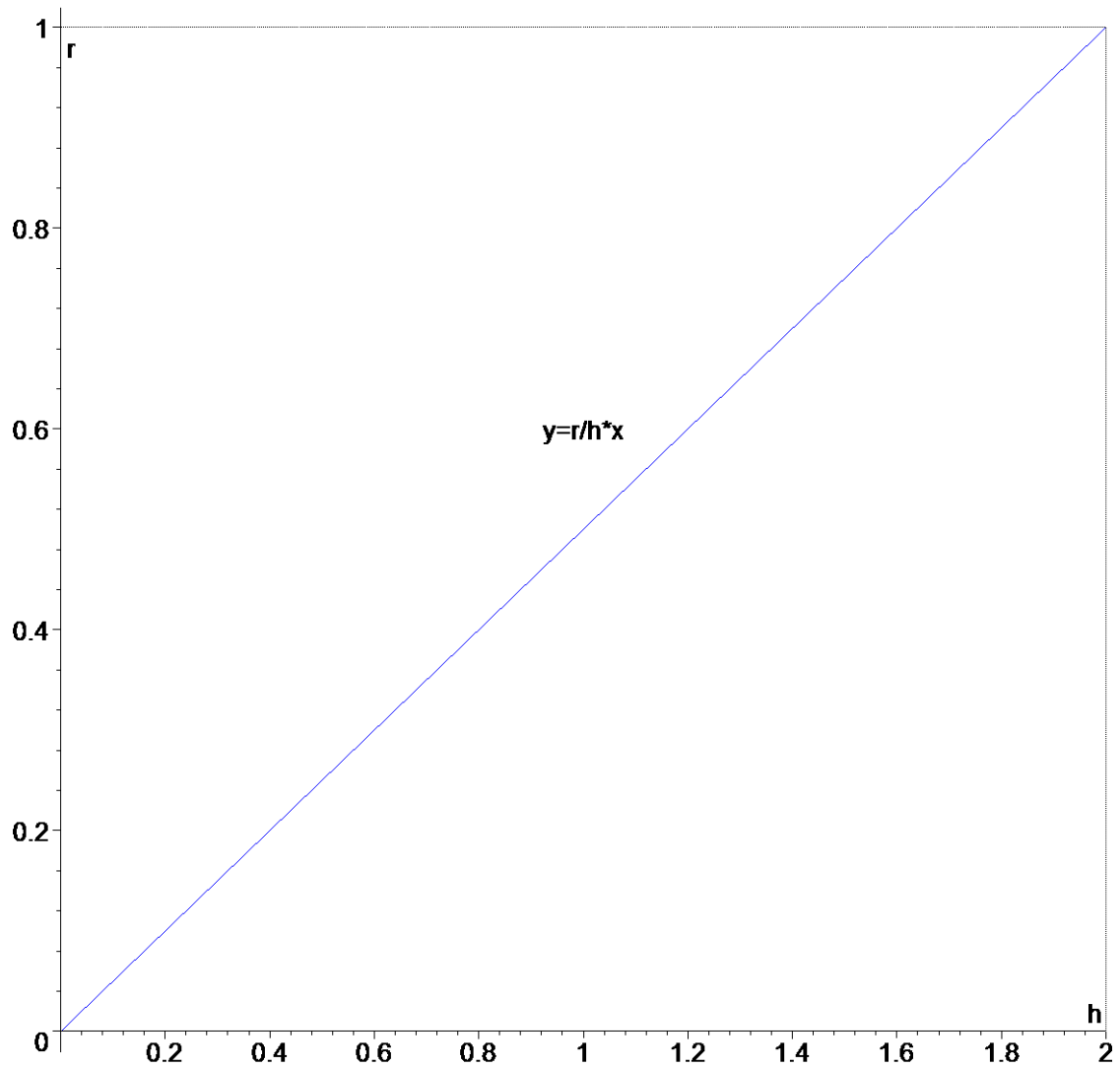
```
> restart:with(plots):with(plottools):
```

```
Warning, the name changecoords has been redefined
```

```
Warning, the name arrow has been redefined
```

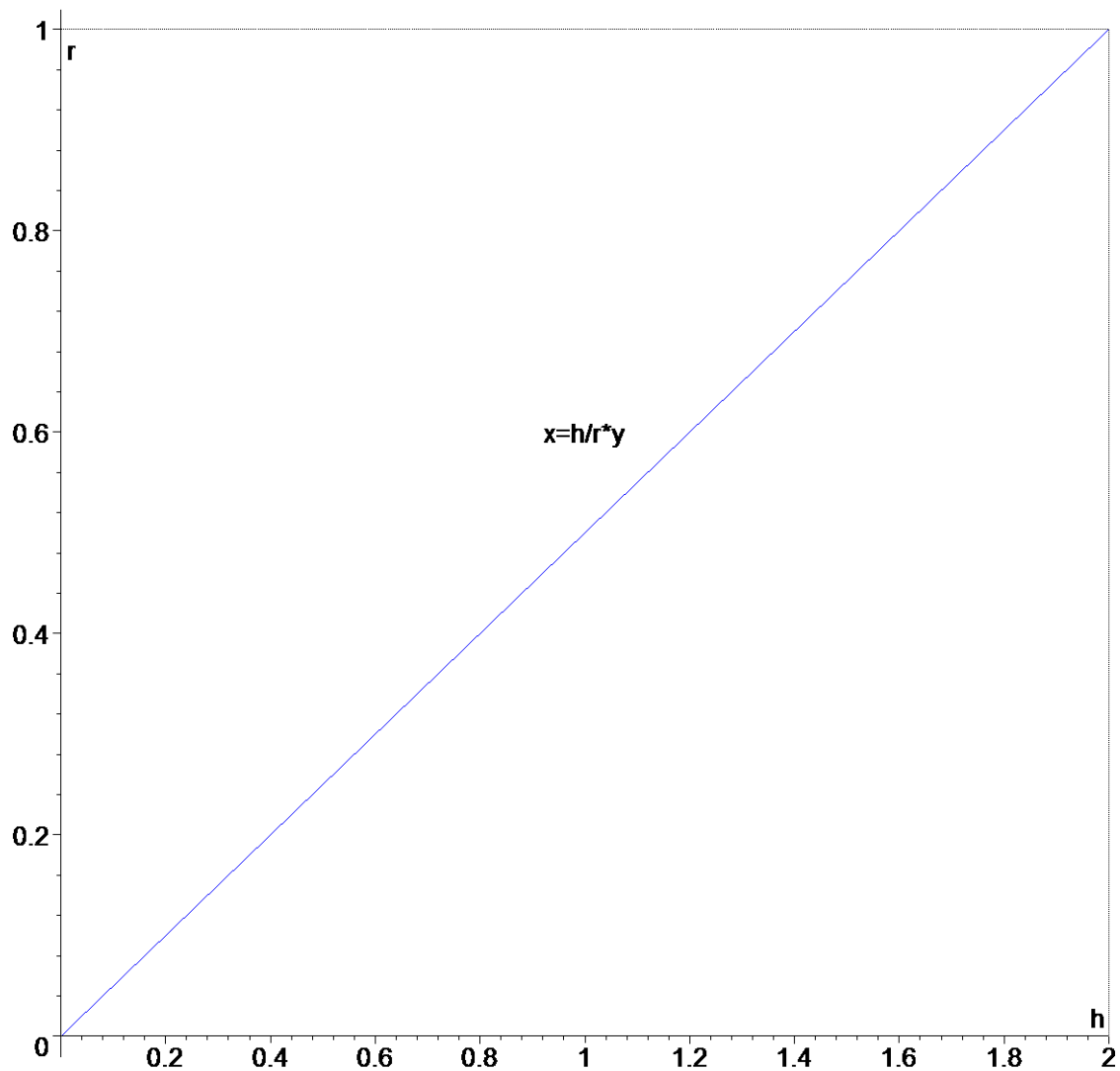
```
> p1:=plot([0,1,1/2*x],x=0..2,color=[black,black,blue],linestyle=[1,2,1]):
p2:=line([2,0],[2,1],color=black,linestyle=2):
p3:=textplot({[1,0.6,`y=r/h*x`],[0.02,0.98,`r`],[1.98,0.019,`h`]
}):
display({p1,p2,p3},title=`Obrázek 1`);
```

Obrázek 1



```
> p1:=plot([0,1,1/2*x],x=0..2,color=[black,black,blue],linestyle=[
1,2,1]):
p2:=line([2,0],[2,1],color=black,linestyle=2):
p3:=textplot({[1,0.6,`x=h/r*y`],[0.02,0.98,`r`],[1.98,0.019,`h`]
}):
display({p1,p2,p3},title=`Obrázek 2`);
```

Obrázek 2



Nejprve uijeme první vztah, pro něj z prvního obrázku platí

> **f:=r/h\*x;**

$$f := \frac{r x}{h}$$

Dosazením dostáváme

> **2\*Pi\*Int(f\*sqrt(1+(Diff(f,x))^2),x=0..h);**

$$2 \pi \int_0^h \frac{r x \sqrt{1 + \left( \frac{\partial}{\partial x} \left( \frac{r x}{h} \right) \right)^2}}{h} dx$$

Výsledkem je

> **assume(h>0);**

**simplify(value(%));**

$$\pi r \sqrt{h^2 + r^2}$$

Nyní uijeme druhý ze vztahů, pro něj z z druhého obrázku platí

> **f:=h/r\*y;**

$$f := \frac{h \cdot y}{r}$$

Dosazením dostáváme

> **2\*Pi\*Int(y\*sqrt(1+(Diff(f,y))^2),y=0..r);**

$$2 \pi \int_0^r y \sqrt{1 + \left( \frac{\partial}{\partial y} \left( \frac{h \cdot y}{r} \right) \right)^2} dy$$

Výsledkem je

> **assume(h>0,r>0);**

**simplify(value(%));**

$$\pi r \sqrt{r^2 + h^2}$$

>

>

### Příklad 2

Spočítejte obsah kulového pásu o poloměru  $r$  a výšce  $h$ . Vznikne rotací subgrafu funkce

$$\sqrt{r^2 - x^2}, -r \leq b \leq r, h = b - a$$

> **restart:with(plots):with(student):**

Warning, the name changecoords has been redefined

> **f:=sqrt(r^2-x^2);**

$$f := \sqrt{r^2 - x^2}$$

Po dosazení do prvního vztahu počítáme integrál

> **i1:=2\*Pi\*Int(f\*sqrt(1+diff(f,x)^2),x=a..b);**

$$i1 := 2 \pi \int_a^b \sqrt{r^2 - x^2} \sqrt{1 + \frac{x^2}{r^2 - x^2}} dx$$

> **simplify(i1);**

$$2 \pi \int_a^b \sqrt{r^2 - x^2} \sqrt{\frac{r^2}{r^2 - x^2}} dx$$

Vidíme, že integrál je roven

> **v1:=2\*Pi\*r\*(b-a);**

$$v1 := 2 \pi r (b - a)$$

System Maple plochu vyjádří následovně

> **v2:=value(i1);**

$$v2 := 2 \pi \left( \sqrt{\frac{r^2}{r^2 - b^2}} \sqrt{r^2 - b^2} b - \sqrt{\frac{r^2}{r^2 - a^2}} \sqrt{r^2 - a^2} a \right)$$

>

>

>

## 5.5 Dvojný a trojný integrál

V tomto odstavci budou postupně vysvětleny příkazy pro zápis a výpočet dvojných a trojných integrálů. Jde o příkazy *Doubleint*, *Tripleint*, které jsou součástí rozšiřující knihovny *student*.

- Dvojný integrál ( matematický zápis )  
*Doubleint( f(x,y), x, y, oblast)*  
*Doubleint( f(x,y), x=a..b, y=c..d)*
- Trojný integrál ( matematický zápis )  
*Tripleint( f(x,y,z), x,y,z, oblast)*  
*Tripleint( f(x,y,z), x=a..b, y=c..d, z=g..h)*
- Dvojnásobný integrál  
*Int( Int(f(x,y),x=a..b), y=c..d)* - matematický zápis  
*int( int(f(x,y),x=a..b), y=c..d)* - přímý výpočet
- Trojnásobný integrál  
*Int( Int( Int( f(x,y,z),x=a..b), y=c..d),z=g..h)* - matematický zápis  
*int( int( int( f(x,y,z),x=a..b), y=c..d),z=g..h)* - přímý výpočet

Ve všech předchozích příkazech lze místo funkce *f* užít výraz pro výpočet hodnoty této funkce. Pro výpočet hodnoty integrálu užijeme příkaz *value*.

>

## Dvojný integrál a jeho aplikace

Nyní uveďme několik příkladů.

### Příklad 1

Vypočítejte dvojný integrál přes obdélník zadaný nerovnostmi  $0 \leq x \leq 2$ ,  $0 \leq y \leq 1$

```
> restart:with(student):
> Doubleint(x^2+y^2,x=0..2,y=0..1);
```

$$\int_0^1 \int_0^2 x^2 + y^2 dx dy$$

```
> value(%);
```

$$\frac{10}{3}$$

```
>
```

```
>
```

### Příklad 2

Vypočítejte dvojný integrál přes oblast  $M$

```
> restart:with(student):with(plots):
Warning, the name changecoords has been redefined
```

Zadaný integrál

```
> i2:=Doubleint(x^2+y^2,x,y,M);
```

$$i2 := \int \int_M x^2 + y^2 dx dy$$

$M$  je množina mezi dvěma křivkami

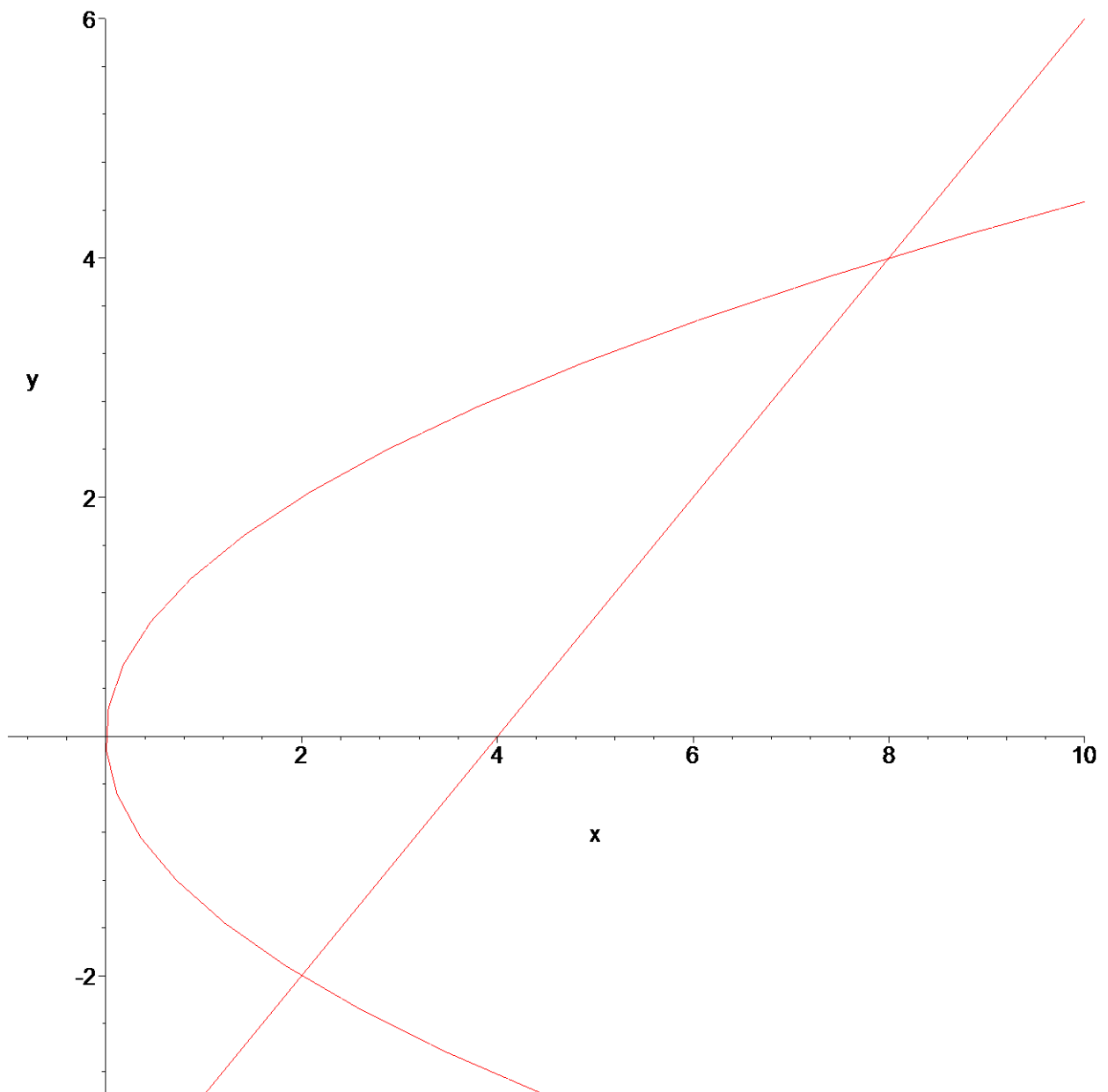
```
> M:={y^2=2*x,y=x-4};
```

$$M := \{y^2 = 2x, y = x - 4\}$$

Nakreslíme oblast  $M$

```
> p1:=implicitplot(M[1],x=-1..10,y=-3..6):
> p2:=plot(rhs(M[2]),x=-1..10,y=-3..6): # pomocí rhs získáme
pravou stranu rovnice
> display({p1,p2});
```





Nalezneme průsečíky křivek pomocí příkazu *intersect*

```
> intersect(M[1],M[2],{x,y});
```

```
{y = -2, x = 2}, {y = 4, x = 8}
```

Výhodnější je integrovat nejprve dle  $x$  a pak dle  $y$ .

Pro získání mezí pro  $x$  použijeme příkaz *isolate*, který vyjádří danou proměnnou z rovnice.

```
> omez := {y = -2 .. 4, x = rhs(isolate(M[1], x)) .. rhs(isolate(M[2], x))};
```

```
omez := {y = -2 .. 4, x = $\frac{y^2}{2}$.. y + 4}
```

Počítáme tedy integrál

```
> i2u := Doubleint(x^2+y^2, omez[2], omez[1]);
```

$$i2u := \int_{-2}^4 \int_{\frac{y^2}{2}}^{y+4} x^2 + y^2 \, dx \, dy$$

Všimněte si, že pokud zadáme meze v opačném pořadí, změní se i pořadí integrace.

Vždy je nutné mít konstantní meze při poslední integraci! Ukažme to následujícími dvěma příkazy.

```
> _i2u:=Doubleint(x^2+y^2,omez[1],omez[2]);
```

$$_i2u := \int_{\frac{y^2}{2}}^{y+4} \int_{-2}^4 x^2 + y^2 dy dx$$

Výpočet integrálu

```
> value(_i2u);
```

$$2(y+4)^3 - \frac{y^6}{4} + 24y + 96 - 12y^2$$

Spočítejme výslednou hodnotu, ze správně zadaného dvojného integrálu

```
> value(i2u);
```

$$\frac{10224}{35}$$

```
>
```

```
>
```

*Příklad 3*

```
> restart:with(student):with(plots):
```

Warning, the name changecoords has been redefined

Vypočítejte dvojný integrál přes trojúhelník daný 3 body

```
> i3:=Doubleint(x*y,x,y,M);
```

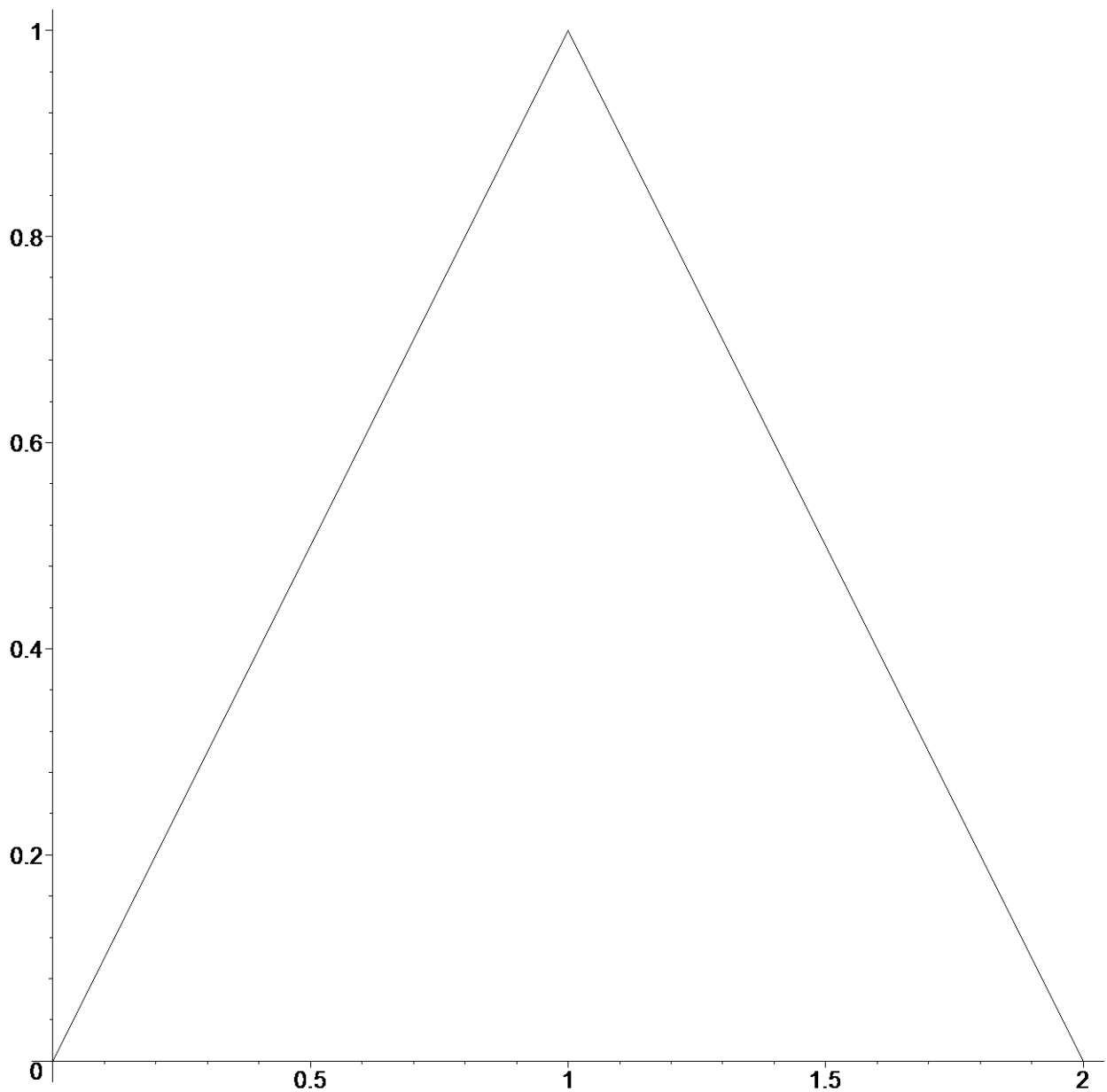
$$i3 := \iint_M x y dx dy$$

```
> M:={ [0,0], [1,1], [2,0] };
```

$$M := \{ [1, 1], [0, 0], [2, 0] \}$$

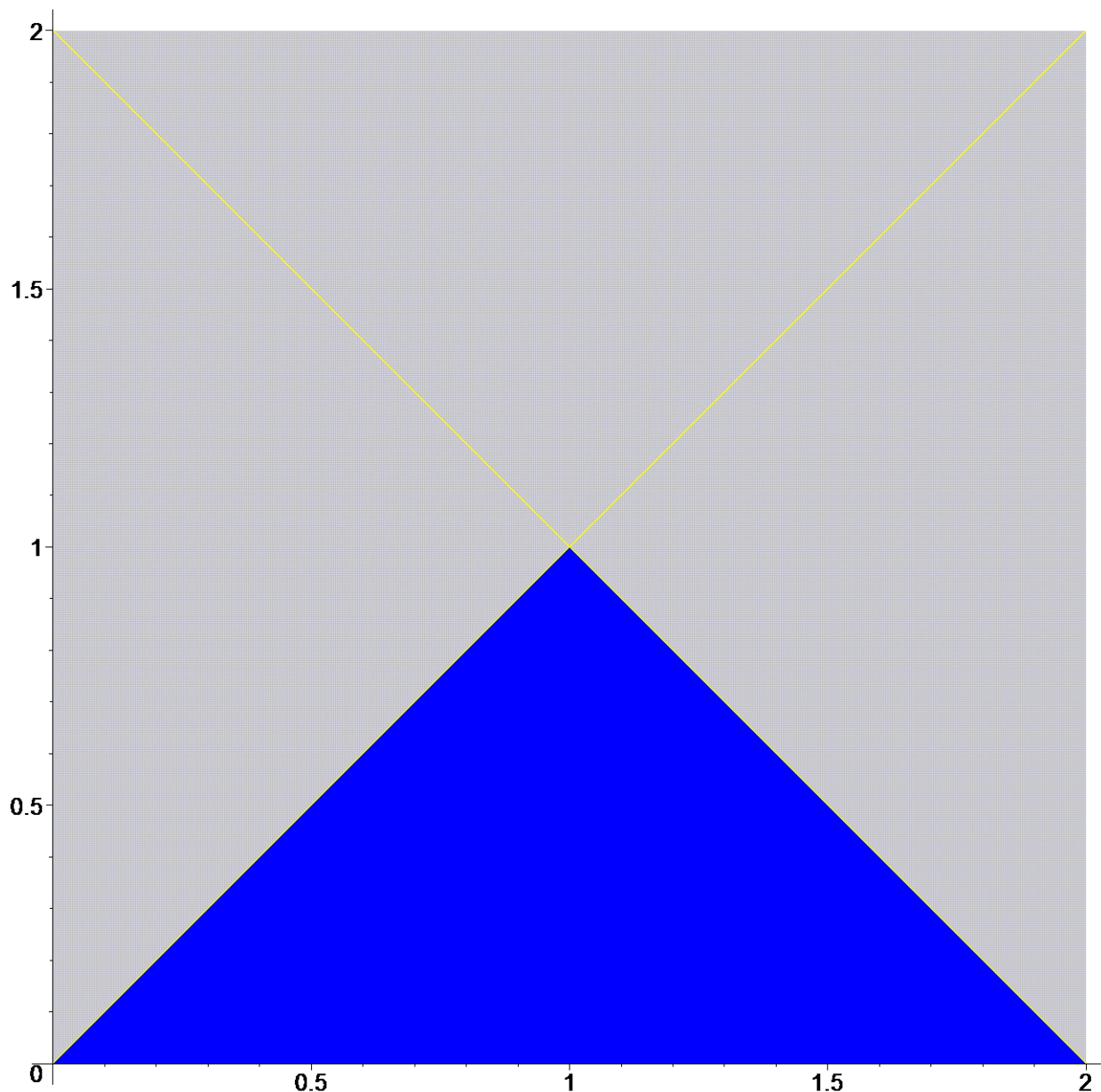
Nakreslíme oblast  $M$

```
> polygonplot(M);
```



Pro lepší zobrazení integrační oblasti použijeme příkaz *inequal*, kde prvním parametrem je množina rovnic přímek, které procházejí danými body.

```
> inequal ({x>=y,x<=2-y},x=-0..2,y=0..2,optionsfeasible=(color
=blue) ,
optionsexcluded=(color=gray) ,optionsclosed=(color=yellow) ,thickn
ess=2) ;
```



Určíme meze integrace

```
> omez := x=y .. 2-y, y=0 .. 1;
```

```
omez := x = y .. 2 - y, y = 0 .. 1
```

Počítáme tedy integrál

```
> i3u := Doubleint (x*y, omez) ;
```

$$i3u := \int_0^1 \int_y^{2-y} x y \, dx \, dy$$

Spočítáme výslednou hodnotu

```
> value (i3u) ;
```

$$\frac{1}{3}$$

```
>
```

#### Příklad 4

Tento příklad ukazuje důležitost volby pořadí integrace

```
> restart:with(student):with(plots):
```

Warning, the name changecoords has been redefined

Vypočítejme dvojný integrál přes zadanou oblast  $M$

```
> Doubleint(sin(y^2), x, y, M);
```

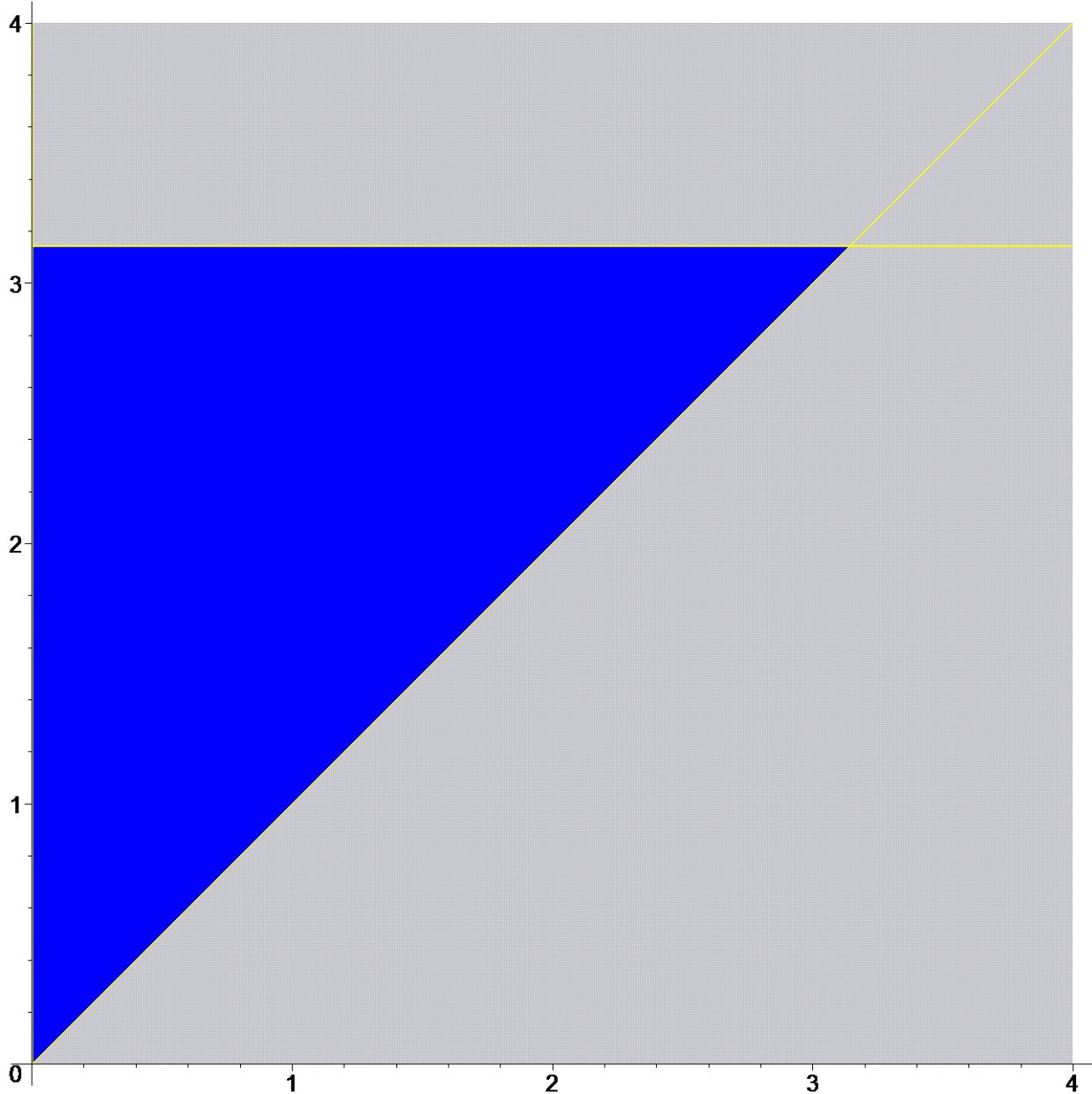
$$\iint_M \sin(y^2) dx dy$$

```
> M:=x=0, y=x, y=Pi;
```

$$M := x = 0, y = x, y = \pi$$

Nakreslíme oblast integrace  $M$ .

```
> inequal({x>=0, y>=x, y<=Pi}, x=-0..4, y=0..4, optionsfeasible=(color=blue),
optionsexcluded=(color=gray), optionsclosed=(color=yellow), thickness=2);
```



Počítáme tedy integrál

```
> i4:=Doubleint(sin(y^2), y=rhs(M[2])..Pi, x=0..Pi);
```

$$i4 := \int_0^\pi \int_x^\pi \sin(y^2) dy dx$$

Tento integrál neumíme jednoduše vypočítat, protože  $\int_0^\pi \sin y^2 dy$  neumíme vyjádřit pomocí

elementárních funkcí

Výsledkem je

> **value(%);**

$$\frac{1}{2} \text{FresnelS}(\sqrt{2} \sqrt{\pi}) \sqrt{2} \pi^{(3/2)} - \frac{1}{12} \pi^4 \text{hypergeom}\left(\left[\frac{3}{4}, 1\right], \left[\frac{3}{2}, \frac{7}{4}, 2\right], -\frac{\pi^4}{4}\right)$$

Nyní zaměníme pořadí integrace. Nejprve budeme integrovat dle  $x$ .

> **i4u:=Doubleint(sin(y^2),x=0..lhs(M[2]),y=0..Pi);**

$$i4u := \int_0^\pi \int_0^y \sin(y^2) dx dy$$

Tento integrál snadno spočítáme

> **value(%);**

$$-\frac{1}{2} \cos(\pi^2) + \frac{1}{2}$$

Jak je z tohoto příkladu vidět, volba pořadí integrace je velmi důležitá.

>

**Substituce** ve dvojném integrálu se provádí pomocí příkazu **changevar( {relace} , Doubleint( f(x,y) , x, y, oblast), [nové proměnné])** *relace* uvádí vztah mezi původními a novými proměnnými. V případě jednoduchého integrálu byly transformovány i meze integrace. *V případě vícenásobných integrálů nejsou meze integrace transformovány!*

**Příklad 5**

Počítejme dvojný integrál přes čtvrtkruh o poloměru  $r$  v I. kvadrantu.

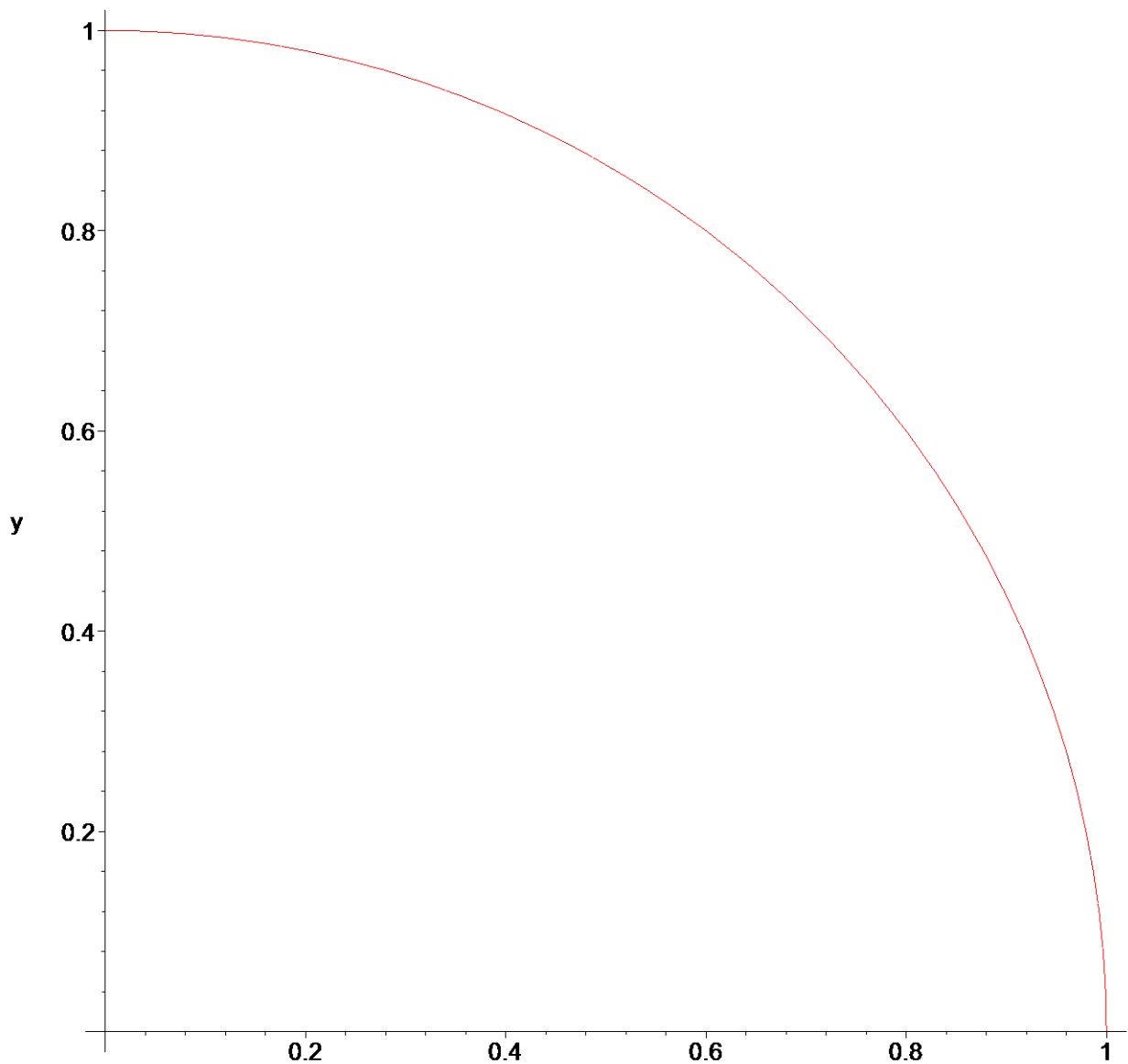
```
> restart:with(student):with(plots):
Warning, the name changecoords has been redefined
```

```
> i5:=Doubleint(sqrt(x^2+y^2),x,y,M);
```

$$i5 := \iint_M \sqrt{x^2 + y^2} \, dx \, dy$$

Vykreslíme oblast pro  $r = 1$

```
> implicitplot(x^2+y^2=1,x=0..1,y=0..1); # vykreslí funkci danou
implicitně
```



Použijeme substituci do polárních souřadnic. Oblast  $M$  v kartézských souřadnicích přejde po substituci v oblast  $N$  v polárních souřadnicích.

> `i5u:=changevar({x=u*cos(v),y=u*sin(v)},subs(M=N,i5),[u,v]);`

$$i5u := \int \int_N |u|^2 du dv$$

Jak již bylo řečeno, příkaz *changevar* v tomto případě neprovede transformaci mezí. Transformaci provedeme sami. Jde o čtvrtkruh, tzn. že pro nové proměnné platí  $0 \leq u \leq r$ ,  $0 \leq v \leq \pi/2$ .

Tedy počítáme integrál

> `Doubleint(integrand(i5u),u=0..r,v=0..Pi/2);`

$$\int_0^{\pi/2} \int_0^r |u|^2 du dv$$

Výsledkem výpočtu je

> `value(%);`



$$\frac{r^3 \pi}{6}$$

>  
>

**Příklad 6**

> **restart:with(student):with(plots):**  
Warning, the name `changecoords` has been redefined

Počítejme integrál přes oblast  $A$ .

> **i6:=Doubleint(1,x,y,A);**

$$i6 := \iint_A 1 \, dx \, dy$$

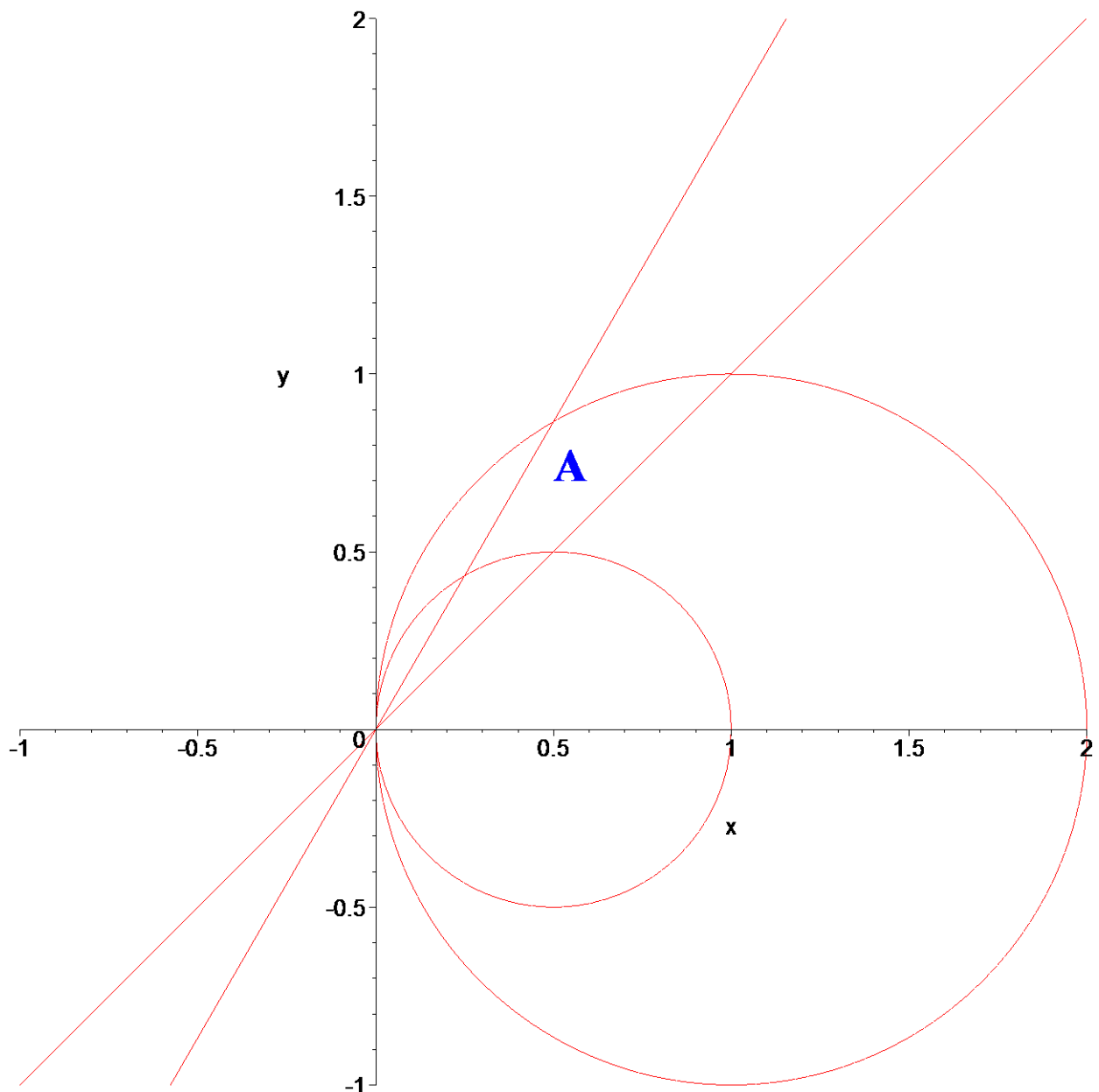
Oblast  $A$  je omezena křivkami zadanými rovnicemi

> **omez:=[x^2+y^2=x,x^2+y^2=2\*x,y=x,y=sqrt(3)\*x];**

$$omez := [x^2 + y^2 = x, x^2 + y^2 = 2x, y = x, y = \sqrt{3}x]$$

Pro názornost oblast nakreslíme

> **p1:=implicitplot(omez[1],x=-1..2,y=-1..2,numpoints=5000):**  
> **p2:=implicitplot(omez[2],x=-1..2,y=-1..2,numpoints=5000):**  
> **p3:=plot(rhs(omez[3]),x=-1..2,y=-1..2):**  
> **p4:=plot(rhs(omez[4]),x=-1..2,y=-1..2):**  
    **p5:=textplot([0.5,0.70,"A"],align={ABOVE,RIGHT},color=blue,**  
    **font=[TIMES,BOLD,20]):**  
> **display({p1,p2,p3,p4,p5});**



Použijeme substituci do polárních souřadnic. Oblast  $A$  v kartézských souřadnicích po substituci přejde v oblast  $B$  v polárních souřadnicích. Proto bylo použito příkazu `subs`.

> `s:=x=u*cos(v),y=u*sin(v);`

$$s := x = u \cos(v), y = u \sin(v)$$

> `i6u:=changevar({s},subs(A=B,i6),[u,v]);`

$$i6u := \int \int_B |u| du dv$$

Je třeba transformovat meze. Dosadíme rovnice transformace do rovnic křivek

> `o1:=simplify(subs(s,omez[1]));`

$$o1 := u^2 = u \cos(v)$$

> `o2:=simplify(subs(s,omez[2]));`

$$o2 := u^2 = 2 u \cos(v)$$

Dále z rovnic přímek získáme informace o úhlu pomocí směrnice

> `o3:=solve(tan(x)=1,x);`

$$o3 := \frac{\pi}{4}$$

> `o4:=solve(tan(x)=sqrt(3),x);`

$$o4 := \frac{\pi}{3}$$

Počítáme tedy integrál (protože jsme dále neupravovali meze, sami vykrátíme  $u$  v  $o1$  a  $o2$ )

> `i6p:=Doubleint(u,u=rhs(o1)/u..rhs(o2)/u,v=o3..o4);`

$$i6p := \int_{\frac{\pi}{4}}^{\frac{\pi}{3}} \int_{\cos(v)}^{2\cos(v)} u \, du \, dv$$

Výsledek spočítáme pomocí

> `value(i6p);`

$$\frac{3\sqrt{3}}{16} + \frac{\pi}{16} - \frac{3}{8}$$

>

*Příklad 7*

> `restart:with(student):with(plots):`

Warning, the name `changecoords` has been redefined

Počítejte obsah elipsy

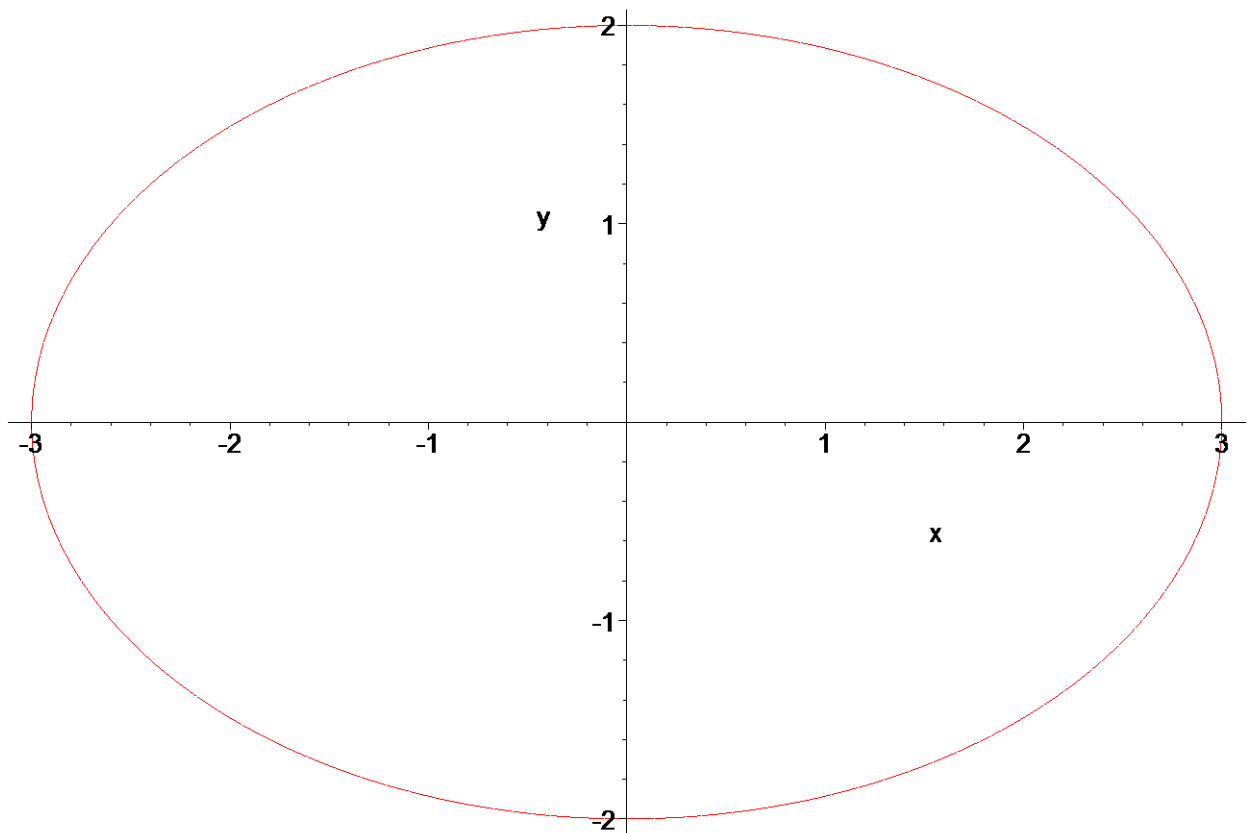
> `e1:=(x/a)^2+(y/b)^2<=1;`

`assume(a>0,b>0):`

$$e1 := \frac{x^2}{a^2} + \frac{y^2}{b^2} \leq 1$$

Nakreslíme obrázek pro pevně zvolené parametry  $a=3$ ,  $b=2$

> `implicitplot((x/3)^2+(y/2)^2=1,x=-3..3,y=-3..3,scaling=constrained,numpoints=5000);`



Zavedeme zobecněné polární souřadnice

>  $s := x = a \cdot u \cdot \cos(v), y = b \cdot u \cdot \sin(v);$

$$s := x = a \cdot u \cdot \cos(v), y = b \cdot u \cdot \sin(v)$$

Určíme meze pro novou proměnnou  $u$  dosazením do rovnice elipsy

>  $\text{simplify}(\text{subs}(s, el));$

$$u^2 \leq 1$$

Integrační meze pro  $v$ ,  $0 \leq v \leq 2 \cdot \text{Pi}$ . Po provedení substituce a dosazením mezei dostáváme

>  $\text{changevar}(\{s\}, \text{Doubleint}(1, x, y, M), [u, v]);$

$$\int \int_M a \cdot b \cdot |u| \, du \, dv$$

>  $i7 := \text{Doubleint}(\text{integrand}(\%), u=0..1, v=0..2 \cdot \text{Pi});$

$$i7 := \int_0^{2\pi} \int_0^1 a \cdot b \cdot |u| \, du \, dv$$

Obsah elipsy je roven

```
> value(i7);
```

$$a \sim b \sim \pi$$

```
>
```

#### Příklad 8

Vypočtěte integrál

```
> restart:with(student):with(plots):
```

```
Warning, the name changecoords has been redefined
```

```
> i8:=Doubleint(x,x,y,M);
```

$$i8 := \iint_M x \, dx \, dy$$

kde  $M$  je kruh

```
> kruh:=x^2+y^2<=2*r*x;
```

```
assume(r>0);
```

$$kruh := x^2 + y^2 \leq 2 r x$$

Tento dvojný integrál spočteme v polárních souřadnicích.

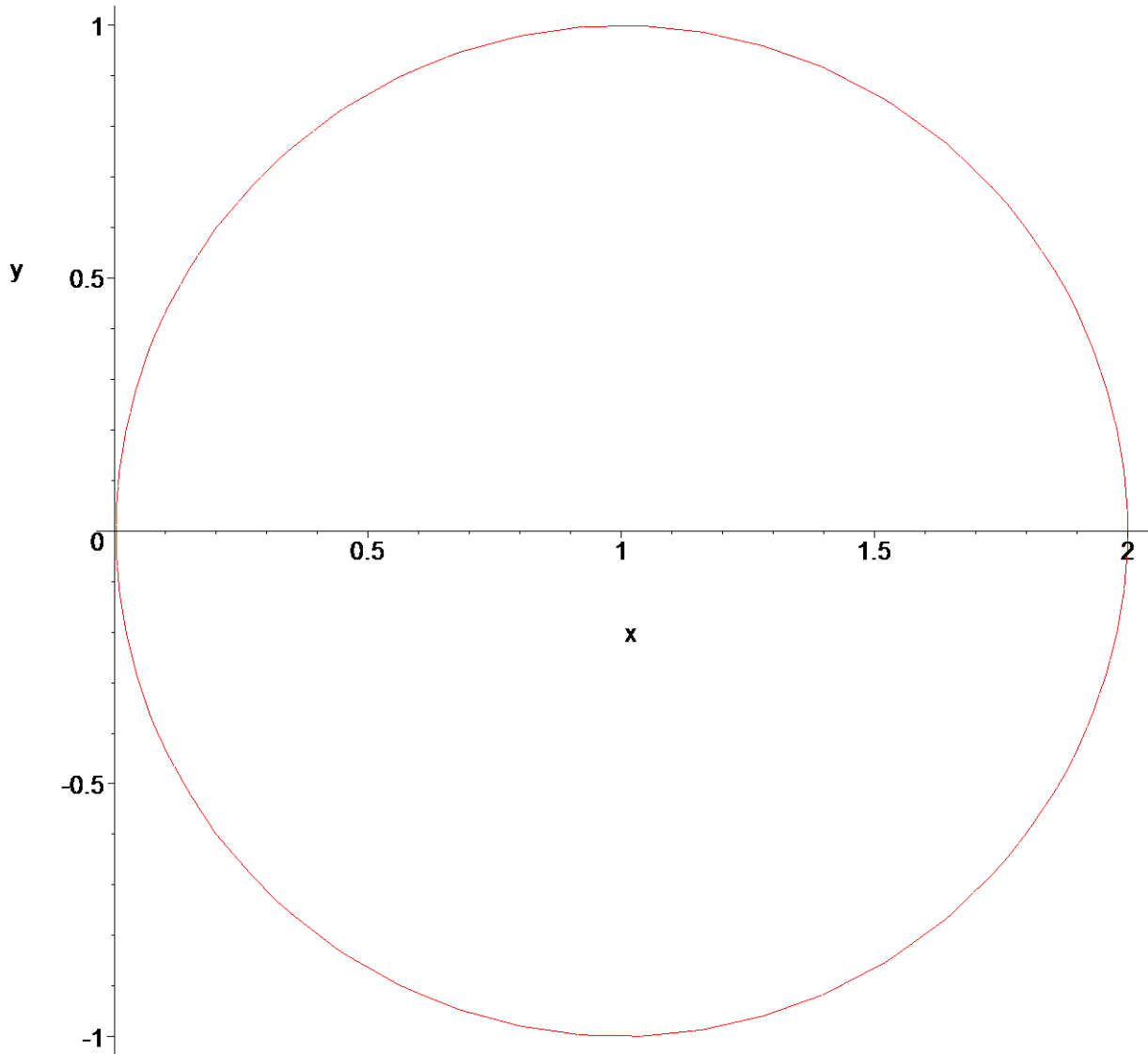
Kruh  $M$  má střed na ose  $x$  a dotýká se  $y$  v počátku, což plyne z dané nerovnice doplněním na součet úplných čtverců.

```
> completesquare(x^2+y^2-2*r*x<=0,x); # doplnění na součet
úplných čtverců
```

$$(x - r)^2 - r^2 + y^2 \leq 0$$

Nakreslíme oblast  $M$  pro pevně zvolené  $r=l$

```
> implicitplot(x^2+y^2=2*x,x=-1..2,y=-1..1,numpoints=600);
```



Zavedeme polární souřadnice. Oblasti  $M$  zadané v kartézských souřadnicích odpovídá v polárních souřadnicích oblast  $N$ .

> **s:=x=rho\*cos(phi), y=rho\*sin(phi);**

$$s := x = \rho \cos(\phi), y = \rho \sin(\phi)$$

> **i8u:=changevar({s}, subs(M=N, i8), [rho, phi]);**

$$i8u := \int \int_N \rho \cos(\phi) |\rho| d\rho d\phi$$

Nerovnice kruhu se transformuje v nerovnici

> **subs(s, kruh);**

$$\rho^2 \cos(\phi)^2 + \rho^2 \sin(\phi)^2 \leq 2 r \rho \cos(\phi)$$

> **simplify(%);**

$$\rho^2 \leq 2 r \rho \cos(\phi)$$

Po vykrácení dostáváme

> **rho<=2\*r\*cos(phi);**

$$\rho \leq 2 r \sim \cos(\phi)$$

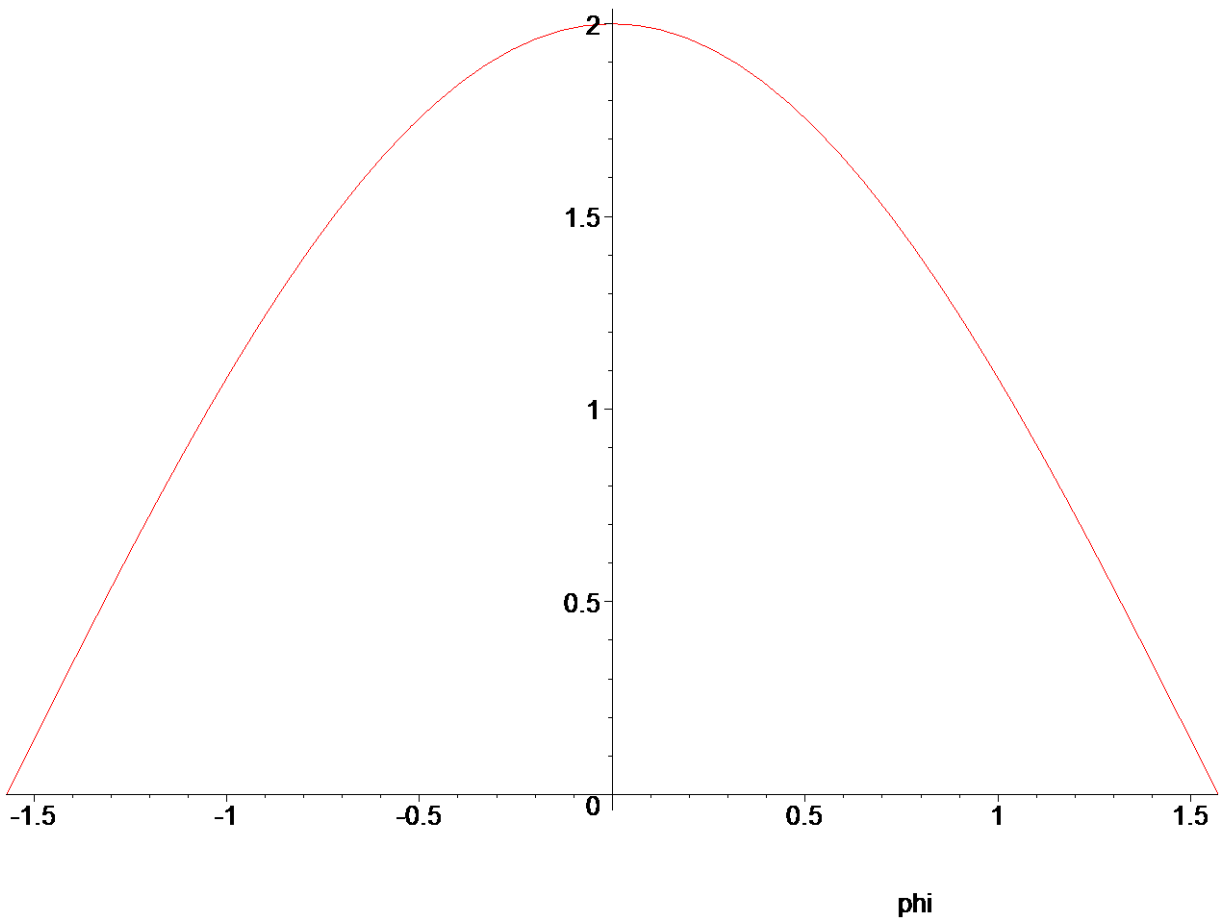
Určíme meze pro integraci

> `meze:=rho=0..rhs(%),phi=-Pi/2..Pi/2;`

$$meze := \rho = 0 \dots 2 r \sim \cos(\phi), \phi = -\frac{\pi}{2} \dots \frac{\pi}{2}$$

Nakreslíme obrázek pro  $r=1$

> `plot(2*1*cos(phi),phi=-Pi/2..Pi/2,scaling=constrained);`



Počítáme tedy integrál

> `assume(rho>0);`

> `i8u:=Doubleint(integrand(i8u),meze);`

$$i8u := \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \int_0^{2 r \sim \cos(\phi)} \rho^2 \cos(\phi) d\rho d\phi$$

Postupnou integrací dostáváme

> `value(%);`

$$\pi r^3$$

>

>

*Příklad 9*

```
> restart:with(student):with(plots):
```

```
Warning, the name changecoords has been redefined
```

Stanovme obsah kardioidy o rovnici  $\rho = a(1 + \cos \varphi)$ ,  $\varphi \in \langle 0, 2\pi \rangle$ . Kardioida je křivka patřící mezi epicykloidy, kdy pevná i kotálející se kružnice mají stejný poloměr.

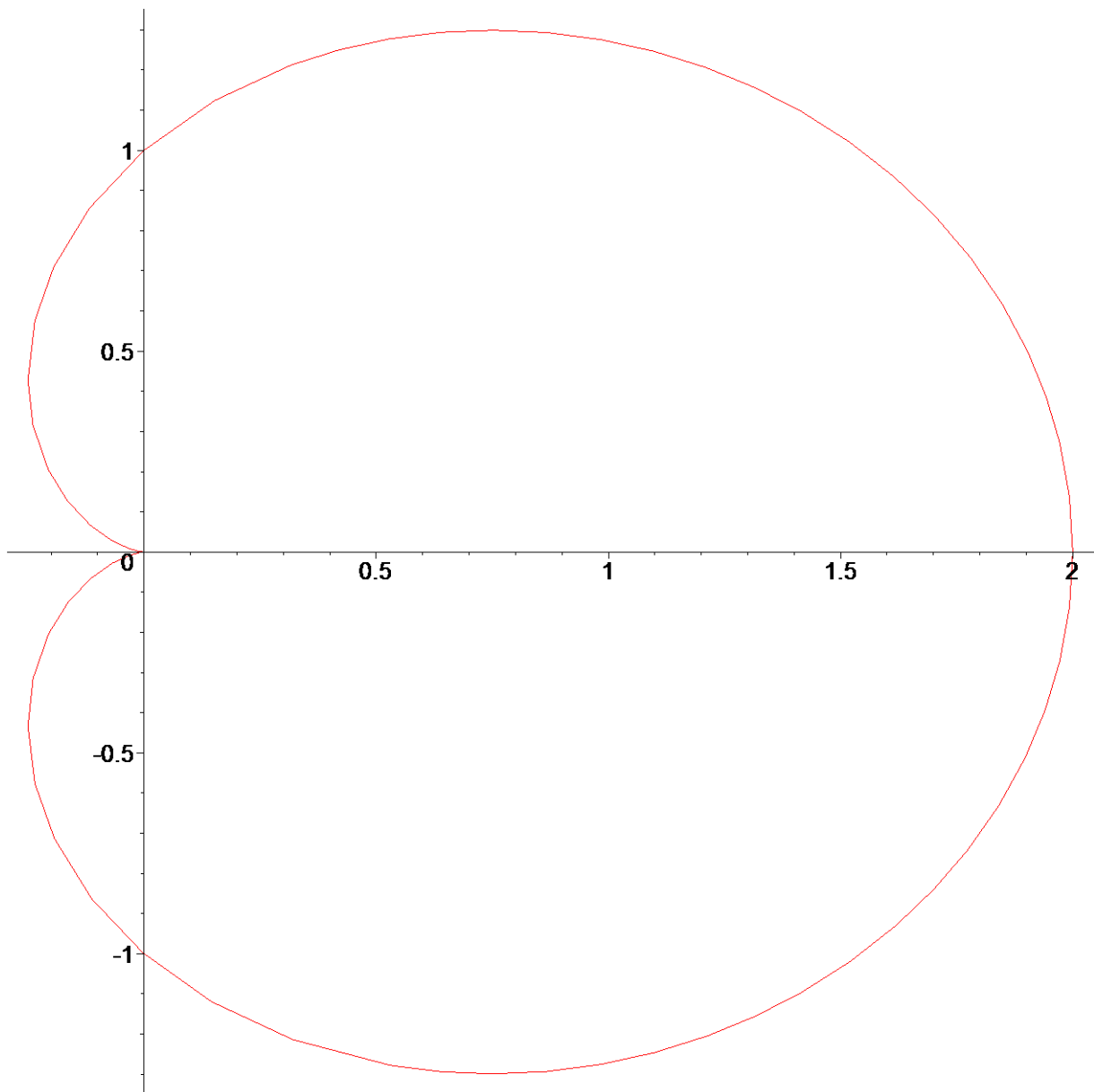
Počítáme integrál

```
> i9:=Doubleint(1,x,y,M);
```

$$i9 := \iint_M 1 \, dx \, dy$$

Nakreslíme obrázek pro  $a=1$

```
> polarplot(1+cos(phi),phi=0..2*Pi);
```



Zavedeme substituci do polárních souřadnic



> **s:=x=rho\*cos(phi),y=rho\*sin(phi);**

$$s := x = \rho \cos(\phi), y = \rho \sin(\phi)$$

> **i9u:=changevar({s},subs(M=N,i9),[rho,phi]);**

**# oblasti M zadané v kartézských souřadnicích odpovídá  
# v polárních souřadnicích oblast N**

$$i9u := \iint_N |\rho| d\rho d\phi$$

Integrační meze jsou

> **meze:=rho=0..a\*(1+cos(phi)),phi=0..2\*Pi;**

$$meze := \rho = 0 .. a(1 + \cos(\phi)), \phi = 0 .. 2\pi$$

Počítáme integrál

> **i9u:=Doubleint(integrand(i9u),meze);**

$$i9u := \int_0^{2\pi} \int_0^{a(1+\cos(\phi))} |\rho| d\rho d\phi$$

Postupnou integrací dostaneme

> **assume(a>0);**

> **value(%);**

$$\frac{3\pi a^2}{2}$$

>

### Objem tělesa

Nechť  $g(x,y)$ ,  $h(x,y)$  jsou spojité funkce na oblasti  $M$  a platí  $g(x,y) < h(x,y)$  uvnitř obrazce  $M$ .

Označme  $T = \{(x, y, z) \in \mathbb{R}_3 \mid (x, y) \in M \wedge g(x, y) \leq z \leq h(x, y)\}$ . Pak  $T$  je těleso a jeho objem

definujeme  $V(T) = \iint_M (h(x, y) - g(x, y)) dx dy$ .

#### Příklad 10

```
> restart:with(plots):with(student):
```

```
Warning, the name changecoords has been redefined
```

Určete objem koule  $T$ , určené nerovností  $x^2 + y^2 + z^2 \leq r^2$ . Z rovnice kulové plochy

$x^2 + y^2 + z^2 = r^2$  vypočítáme  $z_{1,2} = \pm\sqrt{r^2 - x^2 - y^2}$  a volíme

```
> g:=-sqrt(r^2-x^2-y^2):
```

```
h:=sqrt(r^2-x^2-y^2):
```

Obě funkce jsou definovány na kruhu  $M: x^2 + y^2 \leq r^2$ , který získáme také kolmým průmětem do roviny  $xy$ . Dosazením do vzorce pro výpočet objemu dostáváme

```
> i10:=Doubleint(h-g,x,y,M);
```

$$i10 := \iint_M 2\sqrt{r^2 - x^2 - y^2} dx dy$$

Po transformaci do polárních souřadnic dostáváme

```
> s:=x=rho*cos(phi),y=rho*sin(phi);
```

$$s := x = \rho \cos(\phi), y = \rho \sin(\phi)$$

```
> i10u:=changevar({s},subs(M=N,i10),[rho,phi]);
```

$$i10u := \iint_N 2\sqrt{r^2 - \rho^2} |\rho| d\rho d\phi$$

kde  $N$  je oblast v polárních souřadnicích ( odpovídá oblasti  $M$  zadané v kartézských souřadnicích ).

Nové meze po transformaci

```
> meze:=rho=0..r,phi=0..2*Pi;
```

$$meze := \rho = 0 .. r, \phi = 0 .. 2\pi$$

Počítáme tedy integrál

```
> Doubleint(integrand(i10u),meze);
```

$$\int_0^{2\pi} \int_0^r 2\sqrt{r^2 - \rho^2} |\rho| d\rho d\phi$$

Postupnou integrací dostaneme

```
> assume(rho>0):
```

```
> value(%);
```

$$\frac{4(r^2)^{(3/2)}\pi}{3}$$

```
> assume(r>0):
```

```
> simplify(%);
```

$$\frac{4 r^3 \pi}{3}$$

>

*Příklad 11*

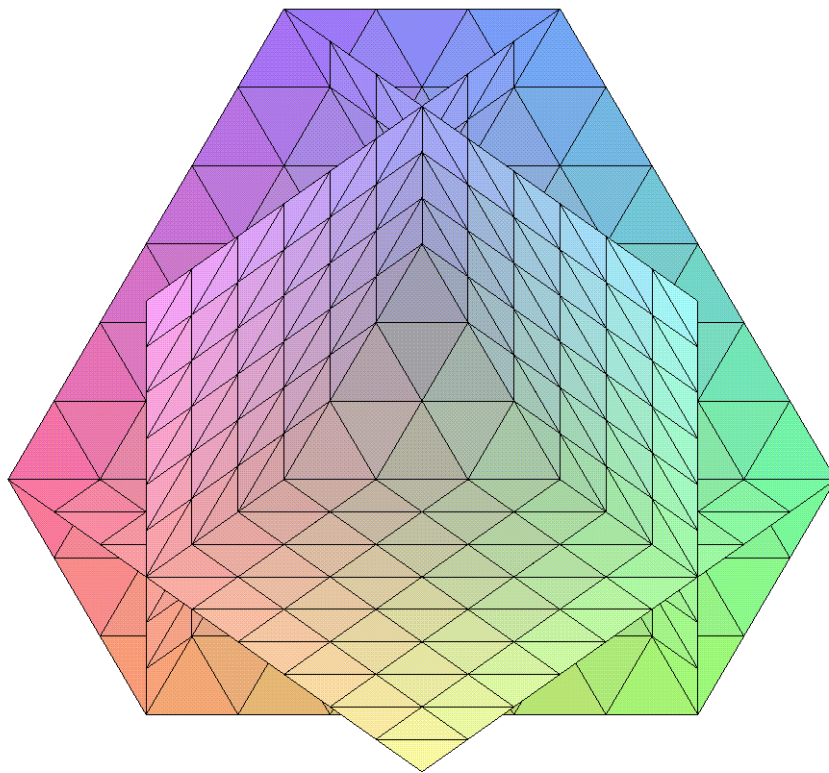
> **restart:with(plots):with(student):**

Warning, the name changecoords has been redefined

Určete objem tělesa ohraničeného rovinami  $x + y + z = 1, x = 0, y = 0, z = 0$ .

Nejprve nakresleme obrázek

> **implicitplot3d({x+y+z-1,x=0,y=0,z=0},x=-1..2,y=-1..2,z=-1..2);**



Průmětem tělesa do roviny  $xy$  je trojúhelník  $M$  ohraničený přímkami  $x = 0, y = 0, x + y = 1$ . Těleso je zdola ohraničeno rovinou  $xy$ , proto

> **g:=0;**

$g := 0$

horní plochu tvoří rovina

> **h:=1-x-y;**

$h := 1 - x - y$

Dosazením do vzorce pro výpočet objemu dostaneme

```
> i11:=Doubleint(h-g,y=0..1-x,x=0..1);
```

$$i11 := \int_0^1 \int_0^{1-x} 1-x-y \, dy \, dx$$

Postupnou integrací dostáváme

```
> i11=value(%);
```

$$\int_0^1 \int_0^{1-x} 1-x-y \, dy \, dx = \frac{1}{6}$$

>

>

*Příklad 12*

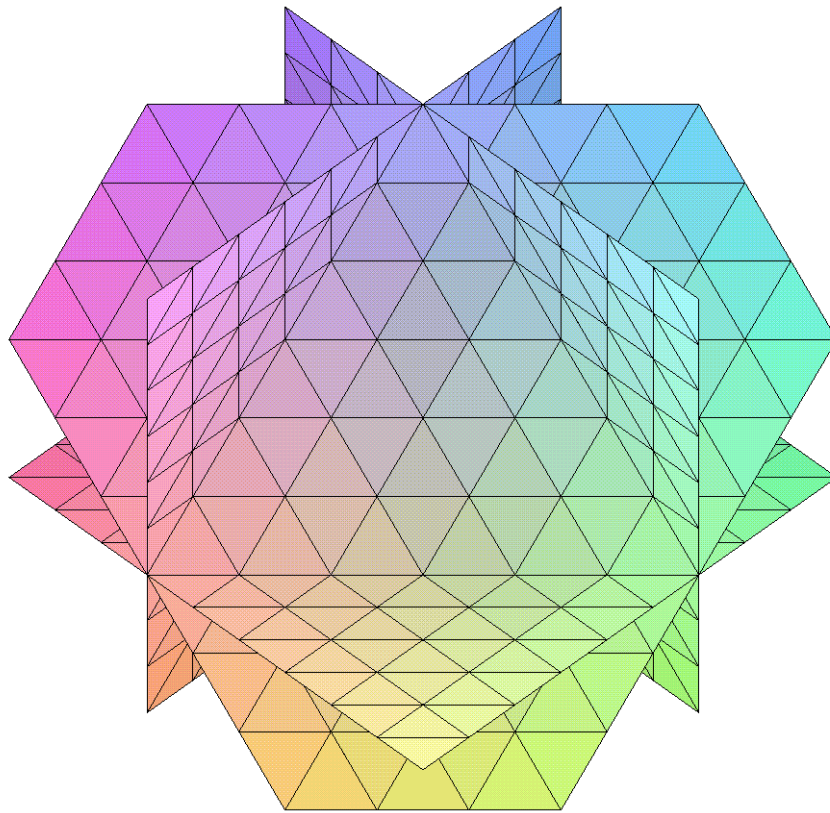
```
> restart:with(plots):with(student):
```

Warning, the name changecoords has been redefined

Vypočtěme objem tělesa, jehož hranici tvoří souřadnicové roviny a rovina  $\frac{x}{a} + \frac{y}{b} + \frac{z}{c} = 1, a, b, c > 0$ .

Nakresleme obrázek pro  $a = b = c = 2$

```
> implicitplot3d({x/2+y/2+z/2-1,x=0,y=0,z=0},x=-1..2,y=-1..2,z=-1..2);
```



Průmětem tělesa je v rovině  $xy$  trojúhelník  $M = \left\{ (x, y) \in \mathbb{R}_2 \mid 0 \leq x \leq a, 0 \leq y \leq b \left( 1 - \frac{x}{a} \right) \right\}$ . Dolní

plochou je

> **g:=0;**

$$g := 0$$

Horní plochou je

> **h:=c\*(1-x/a-y/b);**

$$h := c \left( 1 - \frac{x}{a} - \frac{y}{b} \right)$$

Dosazením do vzorce pro výpočet objemu dostáváme

> **i12:=Doubleint (h-g,y=0..b\*(1-x/a),x=0..a);**

$$i12 := \int_0^a \int_0^{b \left( 1 - \frac{x}{a} \right)} c \left( 1 - \frac{x}{a} - \frac{y}{b} \right) dy dx$$

Integrací dostáváme

> `i12=value (%) ;`

$$\int_0^a \int_0^{b\left(1-\frac{x}{a}\right)} c\left(1-\frac{x}{a}-\frac{y}{b}\right) dy dx = \frac{c b a}{6}$$

>

>

*Příklad 13*

> `restart:with(plots):with(student) :`

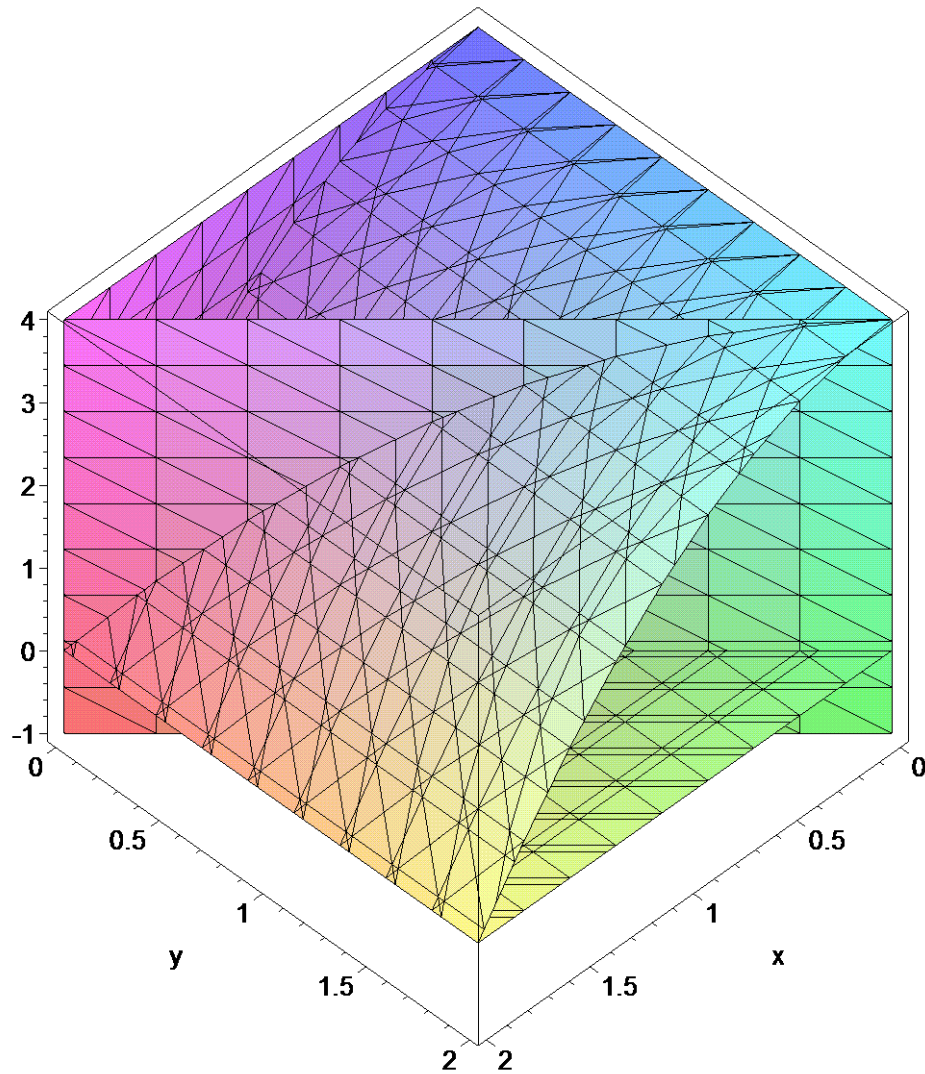
Warning, the name `changecoords` has been redefined

Stanovme objem tělesa ohraničeného souřadnicovými rovinami, rovinou  $\frac{x}{a} + \frac{y}{b} = 1$  a parabolickým

válcem  $z = c^2 - x^2, x \geq 0$ . Dále platí  $a, b, c > 0, a \leq c$ .

Nakresleme obrázek pro  $a = b = c = 2$

> `implicitplot3d({x/2+y/2-1, x=0, y=0, z=0, z+x^2-4}, x=0..2, y=0..2, z=-1..4, axes=boxed) ;`



Průmětem tělesa do roviny  $xy$  je trojúhelník  $M = \left\{ (x, y) \in \mathbb{R}_2 \mid 0 \leq x \leq a, 0 \leq y \leq b \left(1 - \frac{x}{a}\right) \right\}$ . Dolní plochou je rovina  $xy$  a horní plochou je parabolická plocha válce.

> **g:=0;**

$$g := 0$$

> **h:=c^2-x^2;**

$$h := c^2 - x^2$$

Dosazením do vzorce pro výpočet objemu dostáváme

> **Doubleint (h-g, y=0..b\*(1-x/a), x=0..a);**

$$\int_0^a \int_0^{b \left(1 - \frac{x}{a}\right)} c^2 - x^2 \, dy \, dx$$

Integrací dostáváme

> **value(%);**

$$-\frac{1}{12} b a^3 + \frac{1}{2} c^2 b a$$

>



**První momenty** obrazce  $M$  vzhledem k ose

-  $x$ :  $U_x(M) = \iint_M y dx dy$

-  $y$ :  $U_y(M) = \iint_M x dx dy$

**Těžiště obrazce**  $T = [x_T, y_T]$

$$x_T = \frac{U_y(M)}{P(M)}, y_T = \frac{U_x(M)}{P(M)}$$

**Kvadratické momenty** obrazce  $M$  vzhledem k osám  $x, y, z$  definujeme

$$I_x(M) = \iint_M y^2 dx dy, \quad I_y(M) = \iint_M x^2 dx dy, \quad I_z(M) = \iint_M (x^2 + y^2) dx dy$$

Vzdálenost od osy  $z$  v naše případě znamená vzdálenost od počátku.

>

**Příklad 14**

> **restart:with(plots):with(student):**

Warning, the name changecoords has been redefined

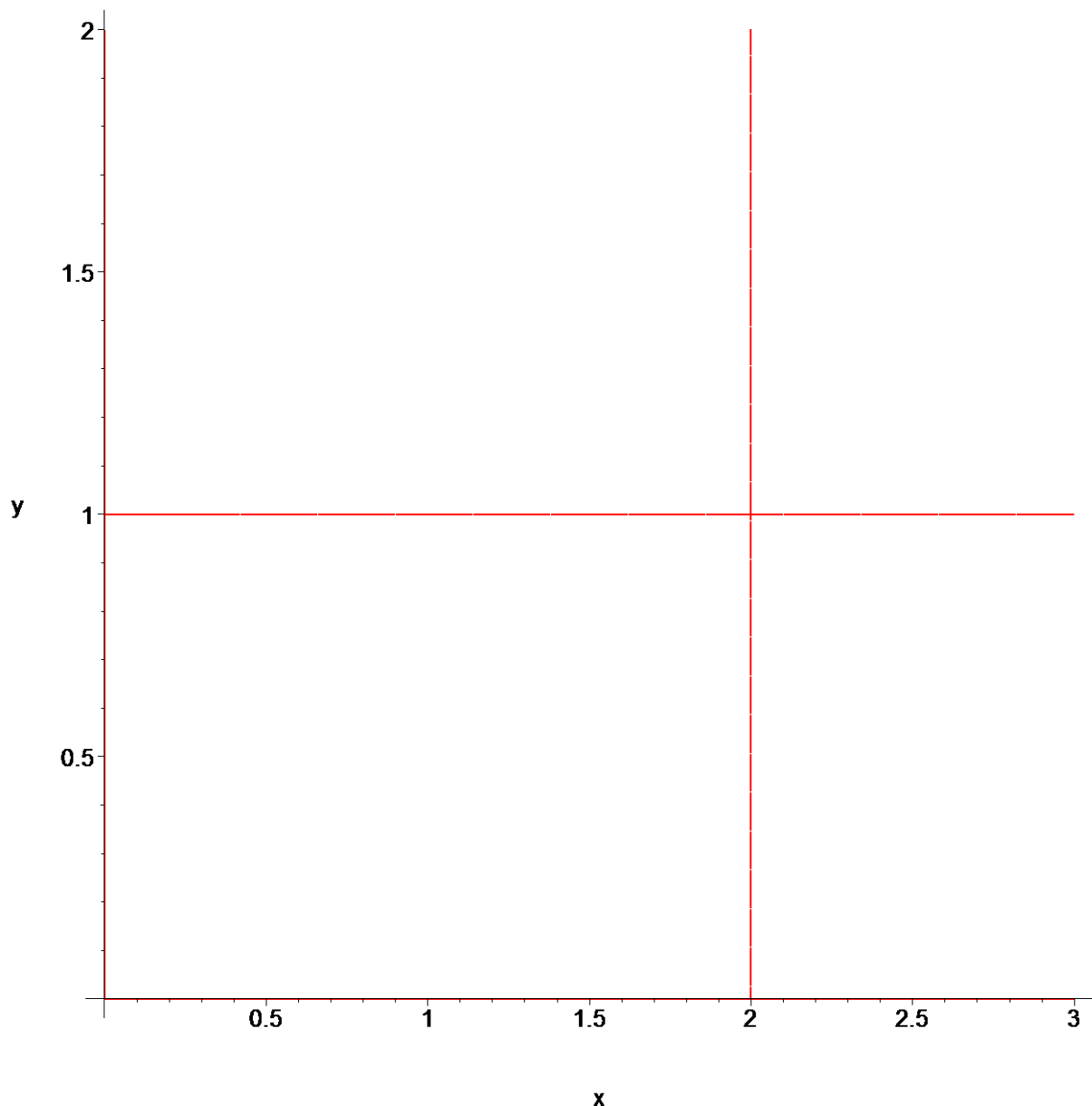
Určete první moment obdélníka o stranách  $a, b$  vzhledem k přímce, na které leží strana  $a$ . Umístíme ho do souřadnicového systému, jeho hranici tedy tvoří přímky o rovnicích.

> **rce:=x=0, x=a, y=0, y=b;**

*rce := x = 0, x = a, y = 0, y = b*

Nakreslíme obrázek pro  $a=2, b=1$

> **implicitplot({x,x-2,y,y-1},x=0..3,y=0..2,thickness=2);**



Protože strana  $a$  leží na ose  $x$ , počítáme

> `U[x] (M) :=Doubleint (y,x,y,M) ;`

$$U_x(M) := \iint_M y \, dx \, dy$$

Dosadíme hranice oblasti  $M$  do integrálu

> `Doubleint (y,x=0..a,y=0..b) ;`

$$\int_0^b \int_0^a y \, dx \, dy$$

Spočítáme integrál

> `value (%) ;`

$$\frac{a b^2}{2}$$

Tento jednoduchý příklad slouží k odvození rozměru prvního momentu. V analytické geometrii rozměry neuvažujeme a můžeme použít i rovnice, které nejsou rozměrově správné. Vypočtené

integrály však znamenají pouze hodnoty prvních momentů, které mají vždy rozměr  $m^3$ , odpovídá-li jednotce 1 m.

>  
>

**Příklad 15**

> **restart:with(plots):with(student):**

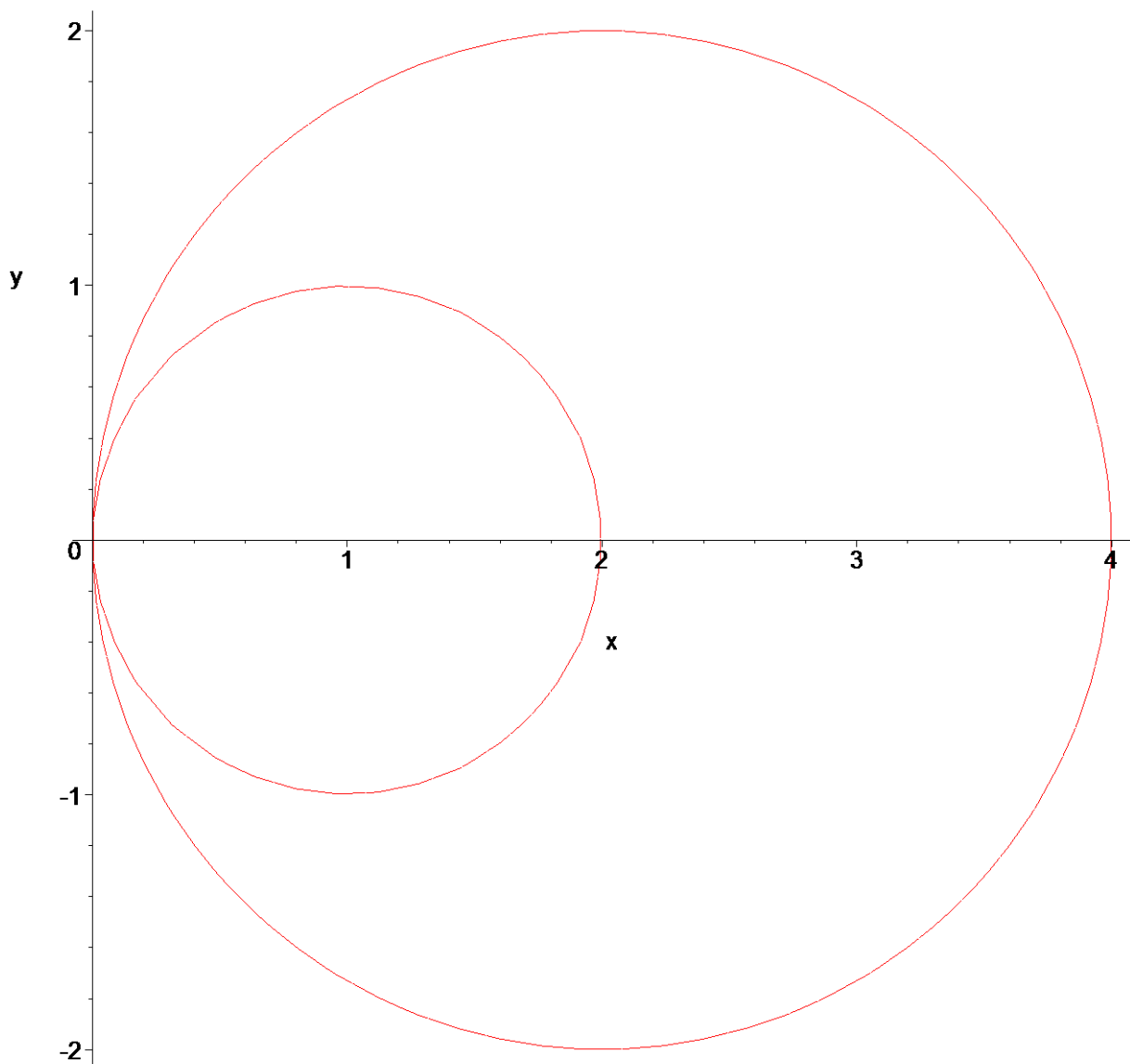
Warning, the name changecoords has been redefined

Určeme souřadnice těžiště homogenního obrazce  $M$ , který je dán podmínkami

$$2r_1x \leq x^2 + y^2 \leq 2r_2x, 0 < r_1 < r_2 .$$

Nejdříve si obrazec nakresleme pro  $r_1=1, r_2=2$ .

> **implicitplot({2\*1\*x-x^2-y^2,x^2+y^2-2\*2\*x},x=0..4,y=-2..2);**



Z obrázku je patrné, že  $y_T = 0$ . Pro určení druhé souřadnice těžiště musíme spočítat obsah

> **P:=Doubleint(1,x,y,M);**

$$P := \iint_M 1 \, dx \, dy$$

a první moment

> **My:=Doubleint(x,x,y,M);**

$$My := \iint_M x \, dx \, dy$$

Zavedeme polární souřadnice

> **s:=x=rho\*cos(phi), y=rho\*sin(phi);**

$$s := x = \rho \cos(\phi), y = \rho \sin(\phi)$$

> **P:=changevar({s},subs(M=N,P),[rho,phi]);**

$$P := \iint_N |\rho| \, d\rho \, d\phi$$

> **My:=changevar({s},subs(M=N,My),[rho,phi]);**

$$My := \iint_N \rho \cos(\phi) |\rho| \, d\rho \, d\phi$$

Obrazec  $M$  se zobrazí na obrazec  $N$ , pro který platí

> **meze:=rho=2\*r1\*cos(phi)..2\*r2\*cos(phi), phi=-Pi/2..Pi/2;**

$$meze := \rho = 2 r_1 \cos(\phi) \dots 2 r_2 \cos(\phi), \phi = -\frac{\pi}{2} \dots \frac{\pi}{2}$$

Dosazením dostaneme

> **P:=Doubleint(integrand(P),meze);**

$$P := \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \int_{2 r_1 \cos(\phi)}^{2 r_2 \cos(\phi)} |\rho| \, d\rho \, d\phi$$

> **My:=Doubleint(integrand(My),meze);**

$$My := \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \int_{2 r_1 \cos(\phi)}^{2 r_2 \cos(\phi)} \rho \cos(\phi) |\rho| \, d\rho \, d\phi$$

Spočteme hodnoty integrálů

> **assume(rho>0);**

> **P:=value(P);**

$$P := r^2 \pi - r l^2 \pi$$

> **My:=value(My);**

$$My := r^2 \pi - r l^3 \pi$$

Dosazením do vzorce pro výpočet těžiště dostaneme

> **x[T]:=simplify(My/P);**

$$x_T := \frac{r l^2 + r l r_2 + r_2^2}{r l + r_2}$$

>

>

**Příklad 16**

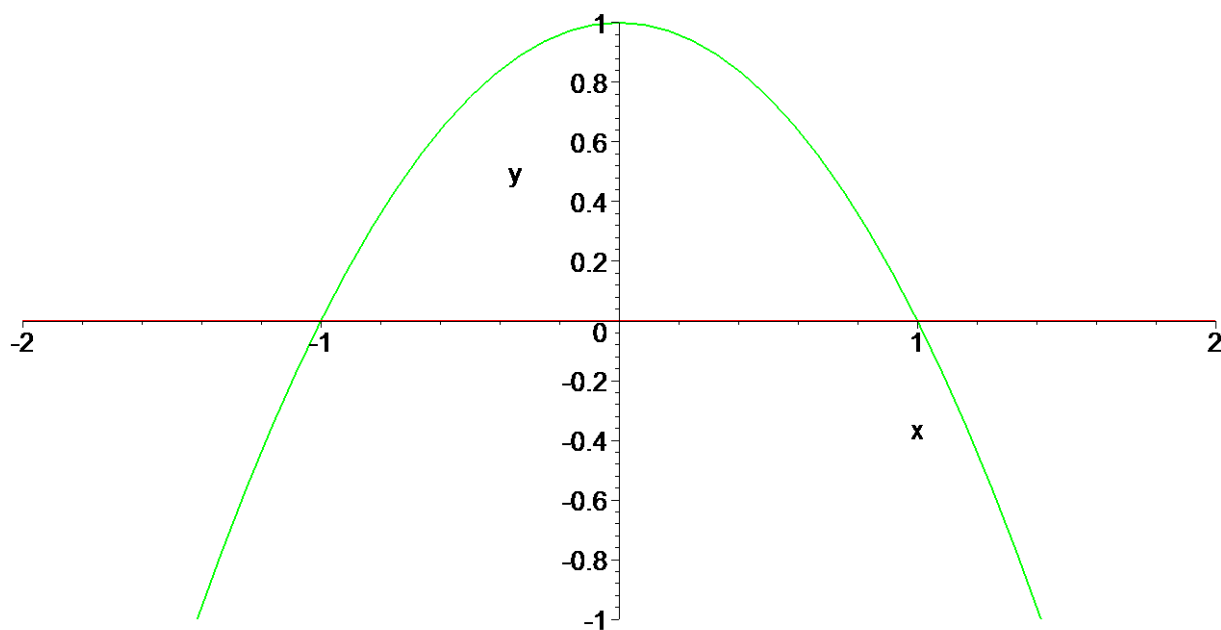
> **restart:with(plots):with(student):**

Warning, the name changecoords has been redefined

Určete těžiště obrazce  $M$  ohraničeného křivkami  $y = 1 - x^2, y = 0$ .

Nakresleme obrázek

> `plot([0,1-x^2],x=-2..2,y=-1..1,scaling=constrained,thickness=2);`



Spočteme obsah obrazce. Počítáme tedy integrál

> `P:=Int(1-x^2,x=-1..1);`

$$P := \int_{-1}^1 1 - x^2 dx$$

> `P:=value(%);`

$$P := \frac{4}{3}$$

Dále spočteme první momenty. Jde o integrály

> `Ux:=Doubleint(y,x,y,M);`

$$U_x := \iint_M y \, dx \, dy$$

> `Uy:=Doubleint(x,x,y,M);`

$$U_y := \iint_M x \, dx \, dy$$

Dosadíme meze integrace

> `Ux:=Doubleint(y,y=0..1-x^2,x=-1..1);`

$$U_x := \int_{-1}^1 \int_0^{1-x^2} y \, dy \, dx$$

> `Ux:=value(%);`

$$U_x := \frac{8}{15}$$

> `Uy:=Doubleint(x,x=-sqrt(1-y)..sqrt(1-y),y=0..1);`

$$U_y := \int_0^1 \int_{-\sqrt{1-y}}^{\sqrt{1-y}} x \, dx \, dy$$

> `Uy:=value(%);`

$$U_y := 0$$

Dosazením do vzorců pro výpočet těžiště dostáváme

> `x[T]:=Uy/P;`

$$x_T := 0$$

> `y[T]:=Ux/P;`

$$y_T := \frac{2}{5}$$

>

>

#### Příklad 17

> `restart:with(plots):with(student):`

Warning, the name changecoords has been redefined

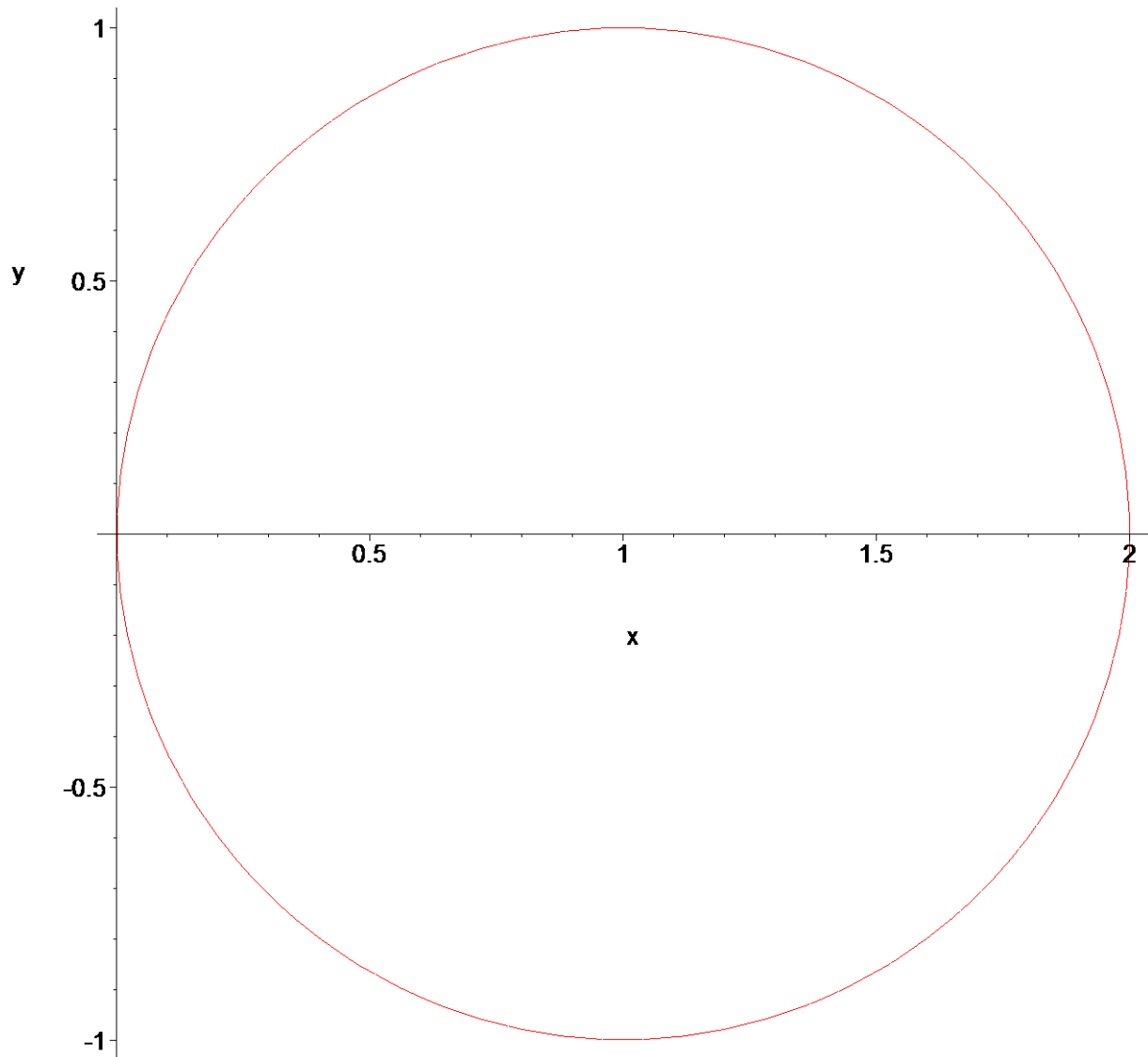
Určete kvadratický moment kruhu vzhledem k přímce procházející bodem na obvodu kolmo k rovině kruhu. Za střed  $S$  kruhu o poloměru  $r$  a zvolme bod  $(r,0)$ . Hranicí kruhu  $M$  je kružnice  $(x-r)^2 + y^2 = r^2$ . Hledaný kvadratický moment je  $I_z(M) = \iint_M (x^2 + y^2) \, dx \, dy$ .

Nakreslíme obrázek pro  $r=1$

> `rce:=(x-r)^2+y^2=r^2;`

$$rce := (x-r)^2 + y^2 = r^2$$

> `implicitplot((x-1)^2+y^2-1,x=0..2,y=-1..1);`



Výpočet provedeme transformací do polárních souřadnic

```
> s:=x=rho*cos(phi),y=rho*sin(phi);
```

$$s := x = \rho \cos(\phi), y = \rho \sin(\phi)$$

Transformací kružnice dostaneme

```
> subs({s},rce);
```

$$(\rho \cos(\phi) - r)^2 + \rho^2 \sin^2(\phi) = r^2$$

```
> simplify(%);
```

$$-2 \rho \cos(\phi) r + r^2 + \rho^2 = r^2$$

Vyřešíme rovnici vzhledem k  $\rho$

```
> solve(%,rho);
```

$$0, 2 \cos(\phi) r$$

Protože  $\rho \geq 0$ , plyne z nerovnosti  $2r \cos \varphi \geq 0$  omezení  $-\frac{\pi}{2} \leq \varphi \leq \frac{\pi}{2}$ .

```
> meze:=rho=0..2*r*cos(phi),phi=-Pi/2..Pi/2;
```

$$\text{meze} := \rho = 0 \dots 2 \cos(\phi) r, \phi = -\frac{\pi}{2} \dots \frac{\pi}{2}$$

Dosazením do integrálu

> `Ix:=Doubleint((x^2+y^2),x,y,M);`

$$Ix := \iint_M x^2 + y^2 dx dy$$

dostaneme integrál

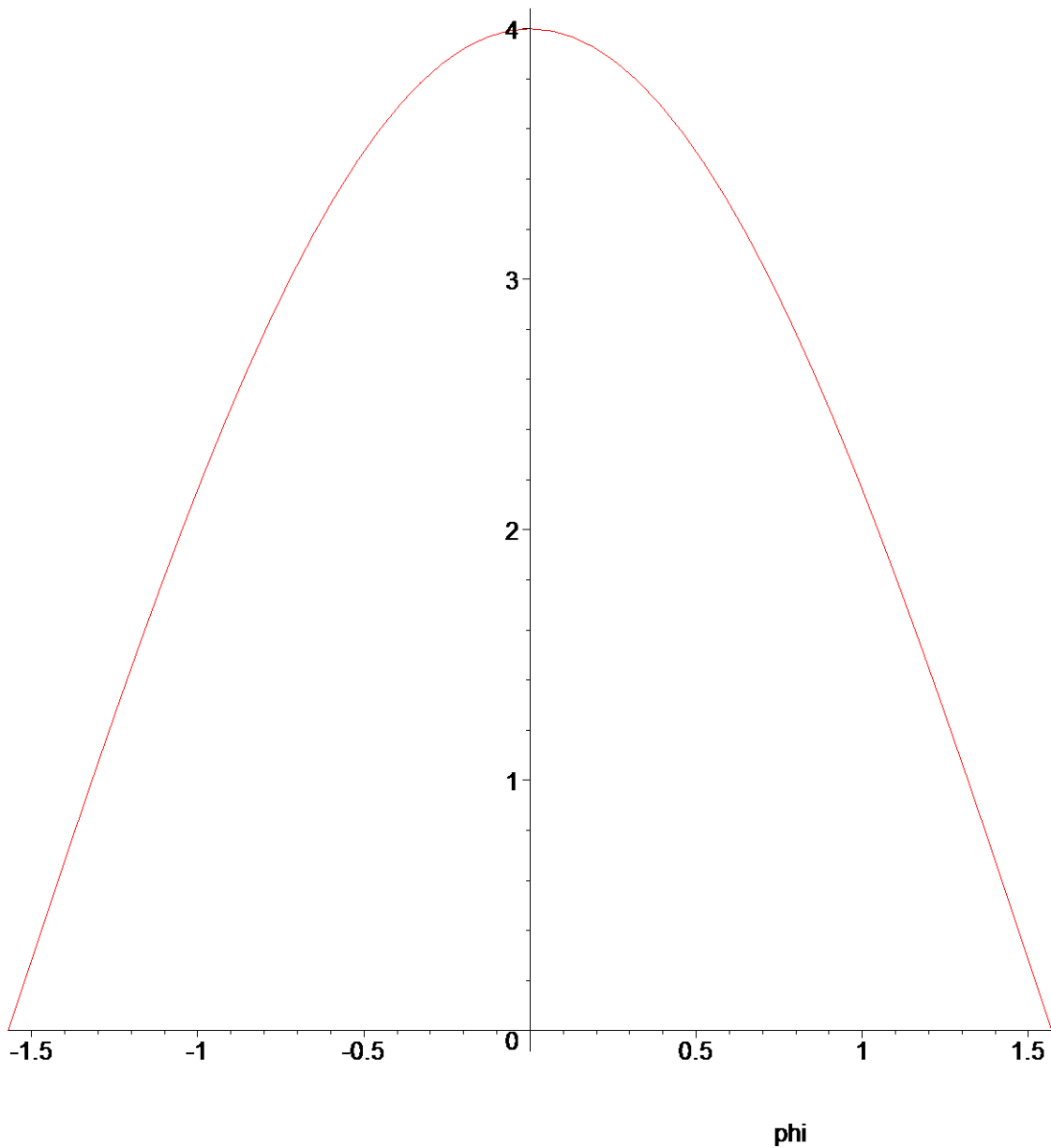
> `Ixu:=changevar({s},subs(M=N,Ix),[rho,phi]);`

$$Ixu := \iint_N \rho^2 |\rho| d\rho d\phi$$

Obrazec  $N$  v polárních souřadnicích odpovídá obrazci  $M$  v kartézských.

Nakreslíme obrázek pro  $r=2$

> `plot(2*2*cos(phi),phi=-Pi/2..Pi/2);`



Dosazením mezi dostaneme



> `Ixu:=Doubleint(integrand(Ixu),meze);`

$$I_{xu} := \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \int_0^{2 \cos(\phi) r} \rho^2 |\rho| d\rho d\phi$$

Integrací spočteme hodnotu kvadratického momentu

> `assume(rho>0,r>0);`  
`map(value,%);`

$$\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} 4 \cos(\phi)^4 r^4 d\phi$$

> `value(%);`

$$\frac{3 \pi r^4}{2}$$

Jednotkou kvadratického momentu je  $m^4$ .

>  
>

### Trojný integrál a jeho aplikace

Nyní uveďme několik příkladů

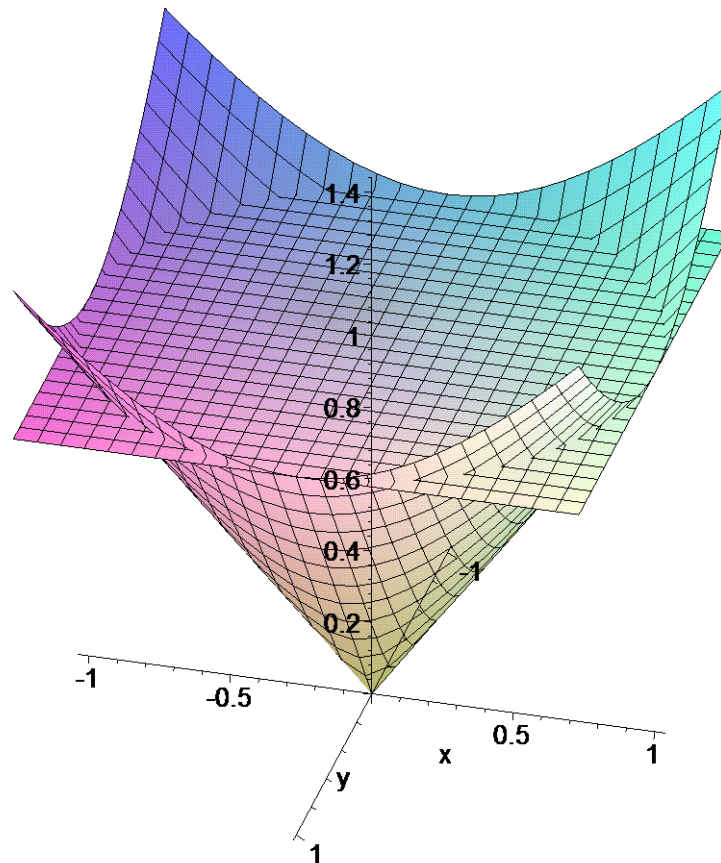
#### Příklad 1

```
> restart:with(student):with(plots):
Warning, the name changecoords has been redefined
```

Vypočítejte  $\iiint_T (x^2 + y^2) z dx dy dz$ , kde těleso  $T$  je ohraničeno plochami  $z = \sqrt{x^2 + y^2}$ ,  $z = 1$ .

Plochou je část kuželové plochy s vrcholem v počátku. Nakresleme obrázek

```
> plot3d({sqrt(x^2+y^2),1},x=-1..1,y=-1..1,axes=normal,orientation
=[15,60]);
```



Z obrázku je patrné, že těleso  $T$  je rotační kužel. Dolní plochou je daná kuželová plocha, proto

```
> g:=sqrt(x^2+y^2);
```

$$g := \sqrt{x^2 + y^2}$$

a horní plochou je rovina

```
> h:=1;
```

$$h := 1$$

Obrazec  $M$  získáme průmětem tělesa  $T$  do roviny  $xy$ . Jde o kruh, jehož hranici získáme řešením soustavy rovnic  $z = \sqrt{x^2 + y^2}, z = 1 \Rightarrow x^2 + y^2 = 1$ .

Počítáme tedy integrál

> **Tripleint**  $((x^2+y^2)*z, x, y, z, T)$  ;

$$\iiint_T (x^2 + y^2) z \, dx \, dy \, dz$$

který je dle Fubiniovy věty roven

> **Doubleint**  $(\text{Int}((x^2+y^2)*z, z=g..h), x, y, M)$  ;

$$\iint_M \int_{\sqrt{x^2+y^2}}^1 (x^2 + y^2) z \, dz \, dx \, dy$$

Integrací podle  $z$  dostaneme

> **iz:=value**  $(\text{Int}((x^2+y^2)*z, z = (x^2+y^2)^{(1/2)} .. 1))$  ;

$$iz := \frac{(x^2 + y^2)(1 - x^2 - y^2)}{2}$$

Počítáme tedy integrál

> **i:=Doubleint**  $(iz, x, y, M)$  ;

$$i := \iint_M \frac{(x^2 + y^2)(1 - x^2 - y^2)}{2} \, dx \, dy$$

Zavedeme polární souřadnice a obrazec  $M$  zadaný v kartézských souřadnicích přejde na obrazec  $N$  v polárních souřadnicích.

> **s:=x=rho\*cos(phi), y=rho\*sin(phi)** ;

$$s := x = \rho \cos(\phi), y = \rho \sin(\phi)$$

> **i:=changevar**  $(\{s\}, \text{subs}(M=N, i), [\rho, \phi])$  ;

$$i := \iint_N \left( \frac{1}{2} \rho^2 |\rho| - \frac{1}{2} \rho^4 |\rho| \right) d\rho \, d\phi$$

Omezení na obrazec  $N$  jsou

> **meze:=rho=0..1, phi=0..2\*Pi** ;

$$meze := \rho = 0 .. 1, \phi = 0 .. 2 \pi$$

Po dosazení dostáváme

> **Doubleint**  $(\text{integrand}(i), meze)$  ;

$$\int_0^{2\pi} \int_0^1 \left( \frac{1}{2} \rho^2 |\rho| - \frac{1}{2} \rho^4 |\rho| \right) d\rho \, d\phi$$

Integrací obdržíme

> **value**  $(\%)$  ;

$$\frac{\pi}{12}$$

S tímto příkladem si přímo poradí i systém Maple. Stačí si uvědomit, že těleso  $T$  je množinou bodů  $(x, y, z) \in \mathbb{R}_3$  definovanou nerovnostmi  $-1 \leq x \leq 1, -\sqrt{1-x^2} \leq y \leq \sqrt{1-x^2}, \sqrt{x^2+y^2} \leq z \leq 1$ .

Přímým výpočtem tedy dostáváme

```
> Tripleint(
 (x^2+y^2)*z, z=sqrt(x^2+y^2)..1, y=-sqrt(1-x^2)..sqrt(1-x^2), x=-1..1);
```

$$\int_{-1}^1 \int_{-\sqrt{1-x^2}}^{\sqrt{1-x^2}} \int_{\sqrt{x^2+y^2}}^1 (x^2+y^2) z \, dz \, dy \, dx$$

```
> value(%);
```

$$\frac{\pi}{12}$$

```
>
>
```

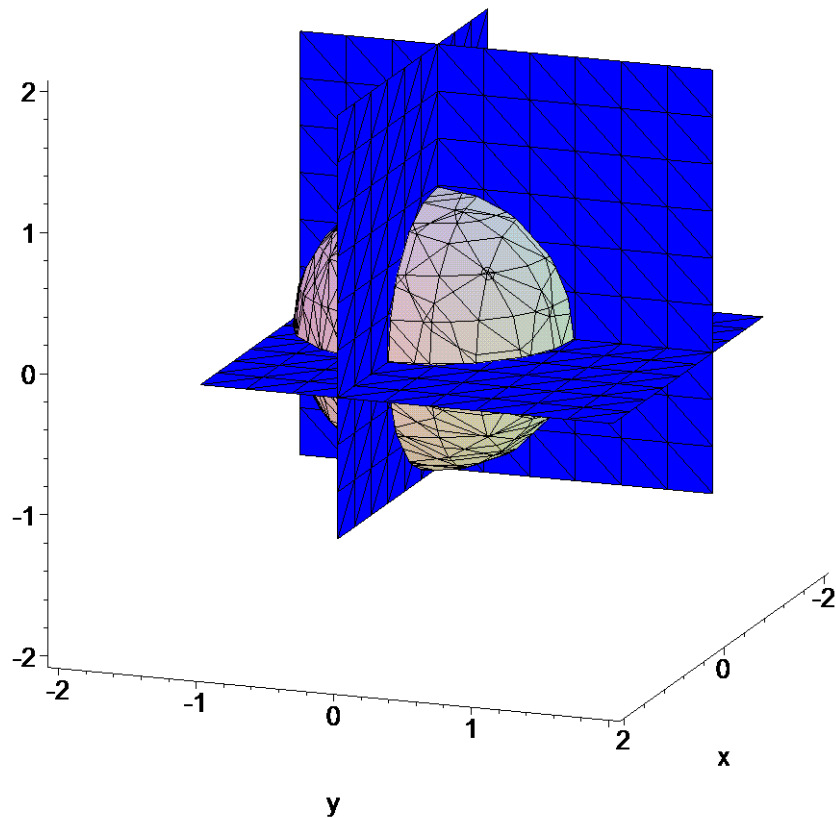
### Příklad 2

```
> restart:with(student):with(plots):
Warning, the name changecoords has been redefined
```

Vypočítejte  $\iiint_T (x+y) z \, dx \, dy \, dz$ , kde  $T$  je osmina koule  $x^2 + y^2 + z^2 \leq 1$  v prvním oktantu.

Nakresleme obrázek

```
> p1:=implicitplot3d(x^2+y^2+z^2-1, x=-2..2, y=-2..2, z=-2..2, numpoints=2000):
> p2:=implicitplot3d({z=0, y=0, x=0}, x=-1..2, y=-1..2, z=-1..2, color=blue):
> display({p1, p2}, orientation=[20, 75], axes=framed);
```



Průmětem tělesa  $T$  do roviny  $xy$  dostaneme čtvrtkruh  $M : x^2 + y^2 \leq 1, x \geq 0, y \geq 0$ . Dolní plocha je rovina  $xy$

>  $g:=0$ ;

$$g := 0$$

horní plochou je část kulové plochy  $x^2 + y^2 + z^2 = 1 \Rightarrow z = \pm\sqrt{1-x^2-y^2}$ , proto

>  $h:=\text{sqrt}(1-x^2-y^2)$ ;

$$h := \sqrt{1-x^2-y^2}$$

Počítáme tedy

>  $\text{Tripleint}((x+y)*z, x, y, z, T) = \text{Doubleint}(\text{Int}((x+y)*z, z=g..h), x, y, M)$ ;

$$\iiint_T (x+y)z \, dx \, dy \, dz = \iint_M \int_0^{\sqrt{1-x^2-y^2}} (x+y)z \, dz \, dx \, dy$$

Integrací dle  $z$  dostaneme

>  $iz:=\text{int}((x+y)*z, z=g..h)$ ;

$$iz := \frac{(x+y)(1-x^2-y^2)}{2}$$

Dostáváme tedy

> `Doubleint(iz,y=0..sqrt(1-x^2),x=0..1);`

$$\int_0^1 \int_0^{\sqrt{1-x^2}} \frac{(x+y)(1-x^2-y^2)}{2} dy dx$$

Výsledkem je

> `value(%);`

$$\frac{2}{15}$$

Přímým výpočtem dostáváme

> `meze:=z=0..sqrt(1-x^2-y^2),y=0..sqrt(1-x^2),x=0..1;`

$$meze := z = 0 .. \sqrt{1-x^2-y^2}, y = 0 .. \sqrt{1-x^2}, x = 0 .. 1$$

> `Tripleint((x+y)*z,meze);`

$$\int_0^1 \int_0^{\sqrt{1-x^2}} \int_0^{\sqrt{1-x^2-y^2}} (x+y)z dz dy dx$$

> `value(%);`

$$\frac{2}{15}$$

>

>

**Transformace do cylindrických (válcových) souřadnic.**

Transformace je určena rovnicemi

$$x = \rho \cos \varphi$$

$$y = \rho \sin \varphi$$

$$z = z$$

Jakobián této transformace je roven  $J(\rho, \varphi, z) = \rho$ .

>

**Příklad 3**

> **restart:with(student):with(plots):**

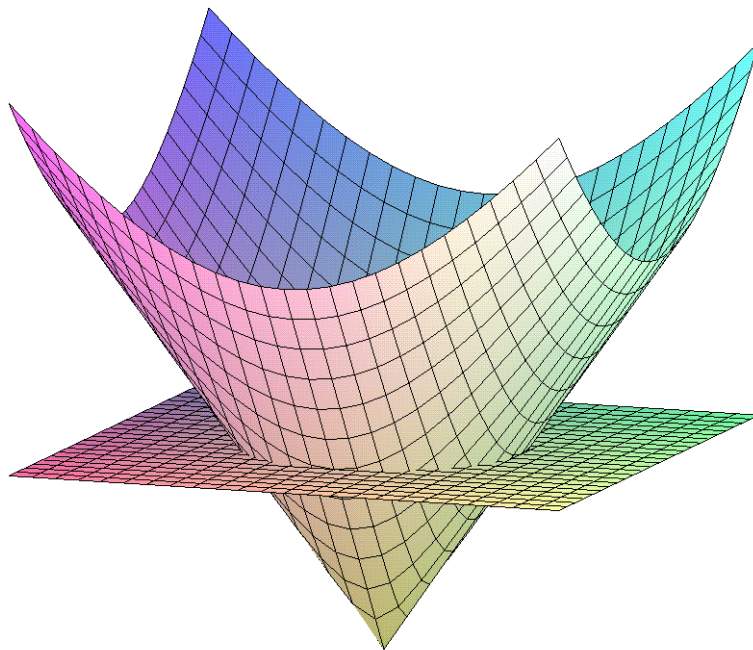
Warning, the name changecoords has been redefined

Transformací do cylindrických souřadnic vypočítejte  $\iiint_T (x^2 + y^2) z dx dy dz$ , kde těleso  $T$  je

ohraničeno plochami  $z = 1, z = \sqrt{x^2 + y^2}$ . Jeho průmět do roviny  $xy$  je kruh  $x^2 + y^2 \leq 1$ .

Nakresleme obrázek

> **plot3d({1, sqrt(x^2+y^2)}, x=-2..2, y=-2..2, orientation=[20, 80]);**

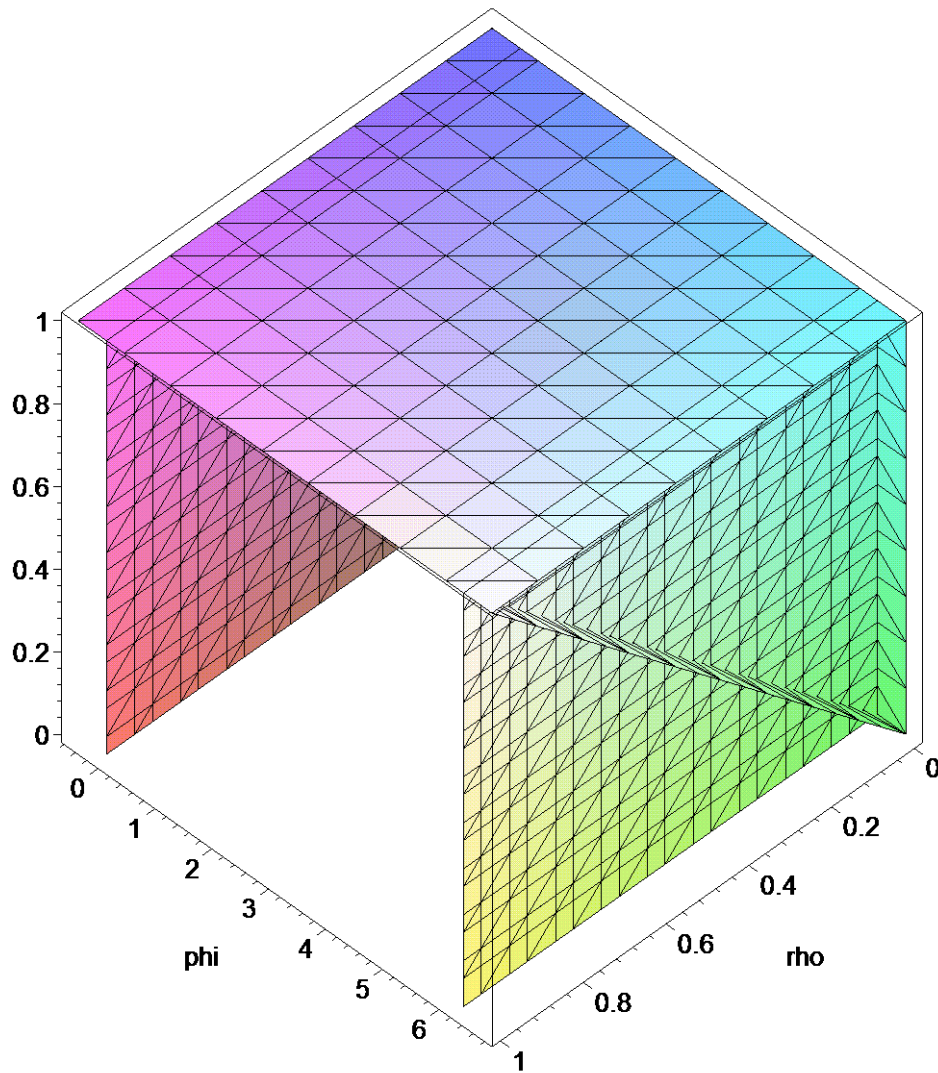


Těleso  $T$  je množina bodů  $(x, y, z) \in \mathbb{R}_3$  takových, že

$-1 \leq x \leq 1, -\sqrt{1-x^2} \leq y \leq \sqrt{1-x^2}, \sqrt{x^2+y^2} \leq z \leq 1$ . Při transformaci přejde na množinu  $U$  bodů  $(\rho, \phi, z)$  takových, že  $0 \leq \rho \leq 1, 0 \leq \phi \leq 2\pi, \rho \leq z \leq 1$ .

Nakresleme obrázek

```
> implicitplot3d({z-1, z-rho, phi, phi-2*Pi, rho}, rho=0..1,
 phi=-0.5..2*Pi+0.5, z=0..1, axes=boxed);
```



Počítáme integrál

```
> i:=Tripleint((x^2+y^2)*z, x, y, z, T);
```

$$i := \iiint_T (x^2 + y^2) z \, dx \, dy \, dz$$

Transformací dostaneme integrál

```
> s:=x=rho*cos(phi), y=rho*sin(phi), z=z;
```

$$s := x = \rho \cos(\phi), y = \rho \sin(\phi), z = z$$

```
> changevar({s}, subs(T=U, i), [rho, phi, z]);
```



$$\iiint_U \rho^2 z |\rho| d\rho d\phi dz$$

který je roven

> `Doubleint (Int (integrand (%), z=rho..1), rho, phi, N);`

$$\iiint_N \rho^2 z |\rho| dz d\rho d\phi$$

Integrací podle z obdržíme

> `value (Int (rho^2*z*abs (rho), z = rho .. 1));`

$$\frac{1}{2} \rho^2 |\rho| (1 - \rho^2)$$

Počítáme tedy integrál

> `Doubleint (% , rho=0..1, phi=0..2*Pi);`

$$\int_0^{2\pi} \int_0^1 \frac{1}{2} \rho^2 |\rho| (1 - \rho^2) d\rho d\phi$$

Integrací dostaneme

> `value (%);`

$$\frac{\pi}{12}$$

I v tomto případě lze integrál počítat přímo

> `Tripleint ((x^2+y^2)*z, z=sqrt (x^2+y^2) .. 1, y=-sqrt (1-x^2) .. sqrt (1-x^2), x=-1..1);`

$$\int_{-1}^1 \int_{-\sqrt{1-x^2}}^{\sqrt{1-x^2}} \int_{\sqrt{x^2+y^2}}^1 (x^2 + y^2) z dz dy dx$$

> `value (%);`

$$\frac{\pi}{12}$$

>

>

**Objem tělesa  $T$**  se vypočítá pomocí vzorce

$$V(T) = \iiint_T dx dy dz$$

>

**Příklad 4**

> **restart:with(student):with(plots):**

Warning, the name changecoords has been redefined

Určete objem rotačního tělesa  $T: x^2 + y^2 \leq f^2(z), a \leq z \leq b$ , kde  $f(z)$  je spojitá na  $\langle a, b \rangle$  a platí  $f(z) > 0$  na  $(a, b)$ .

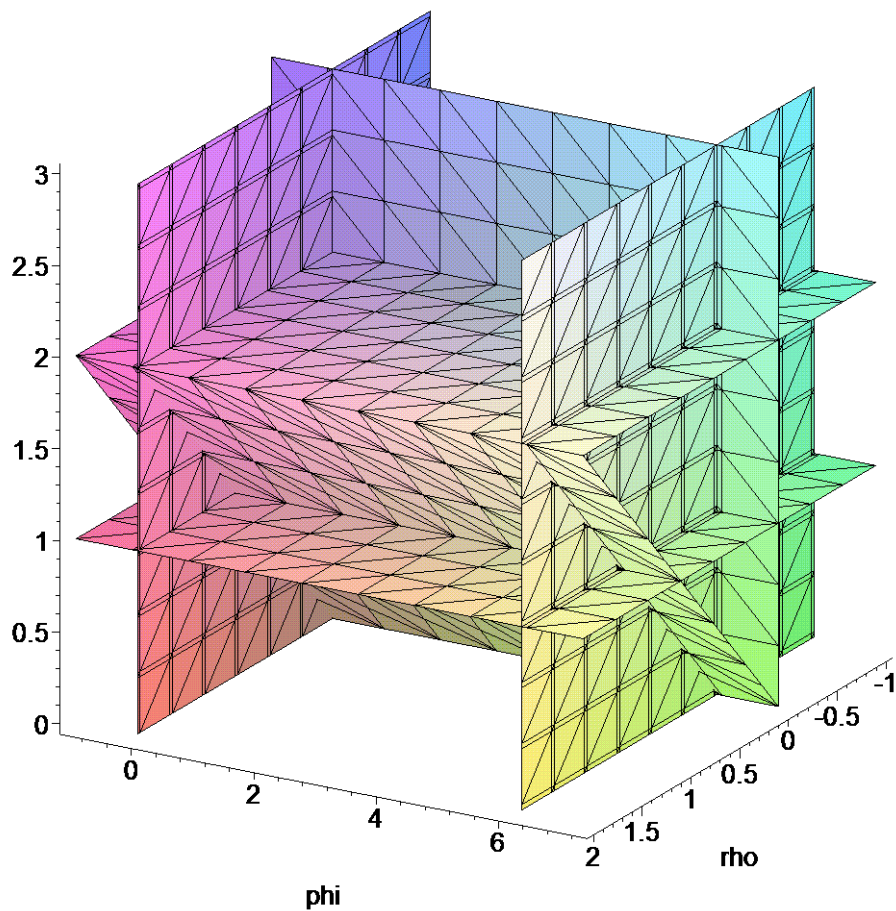
Transformací do cylindrických souřadnic dostaneme těleso  $U$  ohraničené plochami

$\rho = f(z), \rho = 0, z = a, z = b, \varphi = 0, \varphi = 2\pi$ . Pravoúhlým průmětem je obrazec

$M: 0 \leq \rho \leq f(z), a \leq z \leq b$ .

Nakresleme obrázek pro  $f(z) = z, a = 1, b = 2$

> **implicitplot3d({rho=z, rho=0, z=1, z=2, phi=0, phi=2\*Pi},  
rho=-1..2, phi=-1..2\*Pi+1, z=0..3, axes=framed, orientation=[30, 70])  
;**



Počítáme integrál

```
> i:=Tripleint(1,x,y,z,T);
```

$$i := \iiint_T 1 \, dx \, dy \, dz$$

Transformací

```
> s:=x=rho*cos(phi),y=rho*sin(phi),z=z;
```

$$s := x = \rho \cos(\phi), y = \rho \sin(\phi), z = z$$

dostaneme

```
> iu:=changevar({s},subs(T=U,i),[rho,phi,z]);
```

$$iu := \iiint_U |\rho| \, d\rho \, d\phi \, dz$$

Dosazením mezí

```
> meze:=rho=0..f(z),phi=0..2*Pi,z=a..b;
```

$$meze := \rho = 0 \dots f(z), \phi = 0 \dots 2\pi, z = a \dots b$$

dostaneme

> **Tripleint (integrand (iu) ,meze) ;**

$$\int_a^b \int_0^{2\pi} \int_0^{f(z)} |\rho| d\rho d\phi dz$$

Integrací dostaneme

> **assume (f (z) >0) :**

> **value (%) ;**

$$\int_a^b f(z)^2 \pi dz$$

>

>

### Transformace do sférických (kulových) souřadnic

Transformace je určena rovnicemi

$$x = \rho \sin \nu \cos \phi$$

$$y = \rho \sin \nu \sin \phi$$

$$z = \rho \cos \nu$$

Jakobián této transformace je roven  $\rho^2 \cos \nu$ .

>

#### Příklad 5

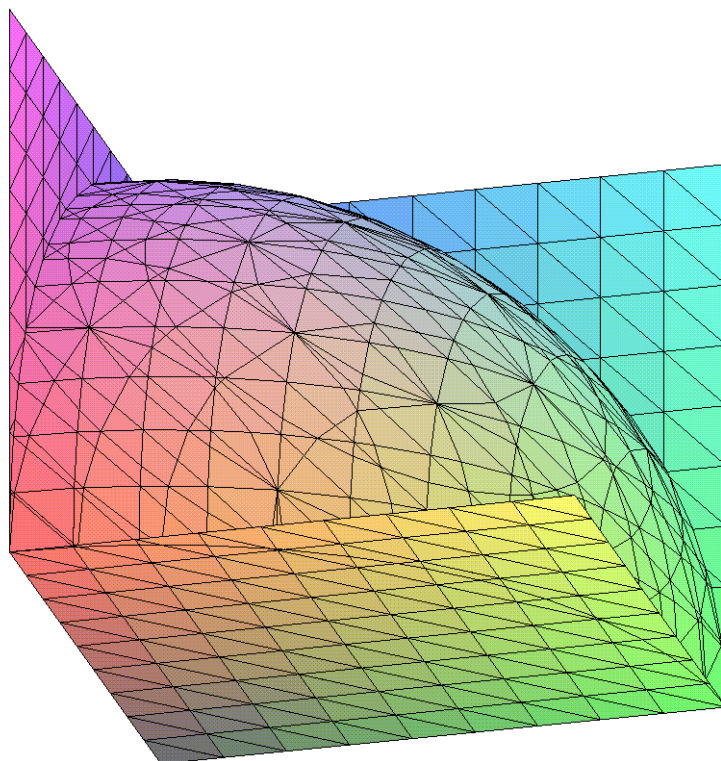
```
> restart:with(plots):with(student):
```

```
Warning, the name changecoords has been redefined
```

Vypočítejte  $\iiint_T z dx dy dz$ , kde  $T$  je určeno  $x^2 + y^2 + z^2 \leq r^2, x \geq 0, y \geq 0, z \geq 0$ .

Nakresleme obrázek pro  $r=1$

```
> implicitplot3d({x^2+y^2+z^2-1,x=0,y=0,z=0},x=0..1,y=0..1,z=0..1,orientation=[15,112]);
```



Těleso  $T$  je osmina koule ležící v prvním oktantu, tj. množina bodů  $(x, y, z) \in \mathbb{R}_3$  takových, že  $0 \leq x \leq r, 0 \leq y \leq \sqrt{r^2 - x^2}, 0 \leq z \leq \sqrt{r^2 - x^2 - y^2}$ . Při transformaci přechází do množiny bodů  $(\rho, \nu, \phi)$  takových, že  $0 \leq \rho \leq r, 0 \leq \nu \leq \frac{\pi}{2}, 0 \leq \phi \leq \frac{\pi}{2}$ .

> **meze:=rho=0..r,upsilon=0..Pi/2,phi=0..Pi/2;**

$$meze := \rho = 0 .. r, \nu = 0 .. \frac{\pi}{2}, \phi = 0 .. \frac{\pi}{2}$$

Integrál

> **i5:=Tripleint(z,x,y,z,T);**

$$i5 := \int \int \int_T z \, dx \, dy \, dz$$

je tedy po transformaci

> **s:=x=rho\*sin(upsilon)\*cos(phi),y=rho\*sin(upsilon)\*sin(phi),z=rho\*cos(upsilon);**

$$s := x = \rho \sin(\nu) \cos(\phi), y = \rho \sin(\nu) \sin(\phi), z = \rho \cos(\nu)$$

roven

> **changevar({s},subs(T=U,i5),[rho,upsilon,phi]);**

$$\int \int \int_U \rho \cos(\nu) |\rho|^2 |\sin(\nu)| \, d\rho \, d\nu \, d\phi$$

Těleso  $T$  bylo transformací převedeno na těleso  $U$  s výše určenými omezeními. Do integrálu dosadíme meze

> **Tripleint(integrand(%),meze);**

$$\int_0^{\frac{\pi}{2}} \int_0^{\frac{\pi}{2}} \int_0^r \rho \cos(\nu) |\rho|^2 |\sin(\nu)| \, d\rho \, d\nu \, d\phi$$

Výsledkem je

> **value(%);**

$$\frac{r^4 \pi}{16}$$

>  
>

**Příklad 6**

> **restart:with(plots):with(student):**

Warning, the name changecoords has been redefined

Vypočítejte  $\iiint_T \frac{1}{\sqrt{x^2 + y^2 + (z-a)^2}} \, dx \, dy \, dz$ , kde  $T$  je koule  $x^2 + y^2 + z^2 \leq r^2, a > r > 0$ .

Integrand je spojitý všude s výjimkou bodu  $(0,0,a)$ , který však leží mimo kouli. Zavedením sférických souřadnic přejde koule  $T$  na kvádr  $U$  s následujícími omezeními

> **meze:=rho=0..r,upsilon=0..Pi,phi=0..2\*Pi;**

$$meze := \rho = 0 .. r, \nu = 0 .. \pi, \phi = 0 .. 2 \pi$$

Počítáme tedy integrál

```
> i:=Tripleint(1/sqrt(x^2+y^2+(z-a)^2),x,y,z,T);
```

$$i := \iiint_T \frac{1}{\sqrt{x^2 + y^2 + z^2 - 2za + a^2}} dx dy dz$$

který po transformaci

```
> s:=x=rho*sin(upsilon)*cos(phi),y=rho*sin(upsilon)*sin(phi),z=rho*cos(upsilon);
```

$$s := x = \rho \sin(\upsilon) \cos(\phi), y = \rho \sin(\upsilon) \sin(\phi), z = \rho \cos(\upsilon)$$

přejde na

```
> changevar({s},subs(T=U,i),[rho,upsilon,phi]);
```

$$\iiint_U \frac{|\rho|^2 |\sin(\upsilon)|}{\sqrt{a^2 - 2\rho \cos(\upsilon) a + \rho^2}} d\rho d\upsilon d\phi$$

Dosadíme výše určené meze

```
> i:=Tripleint(integrand(%),meze);
```

$$i := \int_0^{2\pi} \int_0^{\pi} \int_0^r \frac{|\rho|^2 |\sin(\upsilon)|}{\sqrt{a^2 - 2\rho \cos(\upsilon) a + \rho^2}} d\rho d\upsilon d\phi$$

Vypočítáme tento integrál

```
> assume(a>0,rho>0,r>0,a>r);
```

```
> # value(i);
```

```
zakomentováno kvůli možnému nedostatku paměti počítače při výpočtu, tzn. ukončení
```

```
kernelu
```

System Maple neumí tento integrál spočítat, my však ano.

Integrujme nejdříve podle  $\upsilon$ , což je možné díky konstantním mezím. Dostáváme tedy integrál

```
> iu:=Tripleint(integrand(%),meze[2],meze[1],meze[3]);
```

$$iu := \int_0^{2\pi} \int_0^r \int_0^{\pi} \frac{\rho^2 |\sin(\upsilon)|}{\sqrt{a^2 - 2\rho \cos(\upsilon) a + \rho^2}} d\upsilon d\rho d\phi$$

Do vnitřního integrálu zavedeme substituci

```
> s1:=a^2-2*rho*cos(upsilon)*a+rho^2;
```

$$s1 := a^2 - 2\rho \cos(\upsilon) a + \rho^2$$

```
> iu1:=changevar({s1=t^2},iu,[t,rho,phi]);
```

Warning, Computation of new ranges not implemented

$$iul := \iiint \frac{2 \rho^2 \left| \sin \left( \arccos \left( \frac{a^2 + \rho^2 - t^2}{2 \rho a} \right) \right) \right|}{\sqrt{|-2 \rho^2 a^2 + a^4 - 2 a^2 t^2 + \rho^4 - 2 \rho^2 t^2 + t^4|}} dt d\rho d\phi$$

Zjednodušíme integrand

> **integr:=simplify(integrand(iul));**

$$integr := \frac{\rho}{a}$$

Dosadíme meze a integrál spočteme

> **meze1:=t=abs(rho-a)..abs(rho+a),meze[1],meze[3];**

$$meze1 := t = |\rho - a| .. \rho + a, \rho = 0 .. r, \phi = 0 .. 2\pi$$

> **Tripleint(integr,meze1);**

$$\int_0^{2\pi} \int_0^r \int_{|\rho-a|}^{\rho+a} \frac{\rho}{a} dt d\rho d\phi$$

> **value(%);**

$$\frac{4 r^3 \pi}{3 a}$$

>

>



**První momenty tělesa  $T$**  vzhledem k souřadnicovým rovinám  $xy$ ,  $yz$ ,  $xz$  definujeme

$$U_{xy}(T) = \iiint_T z dx dy dz$$

$$U_{yz}(T) = \iiint_T x dx dy dz$$

$$U_{xz}(T) = \iiint_T y dx dy dz$$

**Souřadnice těžiště**

$$x_T = \frac{U_{yz}(T)}{V(T)}, y_T = \frac{U_{xz}(T)}{V(T)}, z_T = \frac{U_{xy}(T)}{V(T)}$$

>

**Příklad 7**

> **restart:with(plots):with(student):**

Warning, the name changecoords has been redefined

Určete první moment kvádrů o hranách  $a$ ,  $b$ ,  $c$  vzhledem k rovině, ve které leží jeho strany  $a$ ,  $b$ . Jeho hrany vytvářejí roviny  $x=0, x=a, y=0, y=b, z=0, z=c$ . Protože hrany  $a$ ,  $b$  leží v rovině  $xy$ , počítáme

> **Uxy:=Tripleint(z,x,y,z,T);**

$$U_{xy} := \iiint_T z dx dy dz$$

Integrál je roven

> **Doubleint(Int(z,z=0..c),x,y,M);**

$$\iint_M \int_0^c z dz dx dy$$

kde  $M$  je obrazec, který získáme kolmým průmětem kvádrů do roviny  $xy$ .

Integrací dle  $z$  dostaneme

> **int(z,z=0..c);**

$$\frac{c^2}{2}$$

Počítáme integrál

> **Doubleint(% ,x=0..a,y=0..b);**

$$\int_0^b \int_0^a \frac{c^2}{2} dx dy$$

> **value(%);**

$$\frac{c^2 a b}{2}$$

Tento jednoduchý příklad slouží k určení jednotky prvních momentů. Odpovídá-li matematická jednotka jednotce  $1 m$ , je rozměr prvních momentů

$m^4$ .

>

**Příklad 8**

```
> restart:with(plots):with(student):
Warning, the name changecoords has been redefined
```

Určete souřadnice těžiště osminy koule  $T : x^2 + y^2 + z^2 \leq 1, x \geq 0, y \geq 0, z \geq 0$ .

Nejdříve spočítáme objem

```
> V:=Tripleint(1,x,y,z,T);
```

$$V := \int \int \int_T 1 \, dx \, dy \, dz$$

Zavedeme sférické souřadnice, těleso  $T$  přejde transformací v těleso  $U$ ,

```
> s:=x=rho*sin(upsilon)*cos(phi),y=rho*sin(upsilon)*sin(phi),z=rho*cos(upsilon);
```

$$s := x = \rho \sin(\upsilon) \cos(\phi), y = \rho \sin(\upsilon) \sin(\phi), z = \rho \cos(\upsilon)$$

```
> iu:=changevar({s},subs(T=U,V),[rho,upsilon,phi]);
```

$$iu := \int \int \int_U |\rho|^2 |\sin(\upsilon)| \, d\rho \, d\upsilon \, d\phi$$

Dosadíme meze integrace

```
> meze:=rho=0..1,upsilon=0..Pi/2,phi=0..Pi/2;
```

$$meze := \rho = 0 .. 1, \upsilon = 0 .. \frac{\pi}{2}, \phi = 0 .. \frac{\pi}{2}$$

```
> Tripleint(integrand(iu),meze);
```

$$\int_0^{\frac{\pi}{2}} \int_0^{\frac{\pi}{2}} \int_0^1 |\rho|^2 |\sin(\upsilon)| \, d\rho \, d\upsilon \, d\phi$$

```
> V:=value(%);
```

$$V := \frac{\pi}{6}$$

Dále počítejme

```
> Uzy:=Tripleint(x,x,y,z,T);
```

$$Uzy := \int \int \int_T x \, dx \, dy \, dz$$

Opět zavedeme sférické souřadnice

```
> changevar({s},subs(T=U,%),[rho,upsilon,phi]);
```

$$\int \int \int_U \rho \sin(\upsilon) \cos(\phi) |\rho|^2 |\sin(\upsilon)| \, d\rho \, d\upsilon \, d\phi$$

Dosazením mezí dostáváme

```
> Tripleint(integrand(%),meze);
```

$$\int_0^{\frac{\pi}{2}} \int_0^{\frac{\pi}{2}} \int_0^1 \rho \sin(\upsilon) \cos(\phi) |\rho|^2 |\sin(\upsilon)| \, d\rho \, d\upsilon \, d\phi$$

```
> Uzy:=value(%);
```

$$U_{zy} := \frac{\pi}{16}$$

Použitím vzorce pro výpočet těžiště dostaneme

> **x[T] := Uzy/V;**

$$x_T := \frac{3}{8}$$

Z důvodů symetrie nemusíme další momenty počítat a hledané těžiště je

> **T := [3/8, 3/8, 3/8];**

$$T := \begin{bmatrix} \frac{3}{8} & \frac{3}{8} & \frac{3}{8} \end{bmatrix}$$

>

>

*Příklad 9*

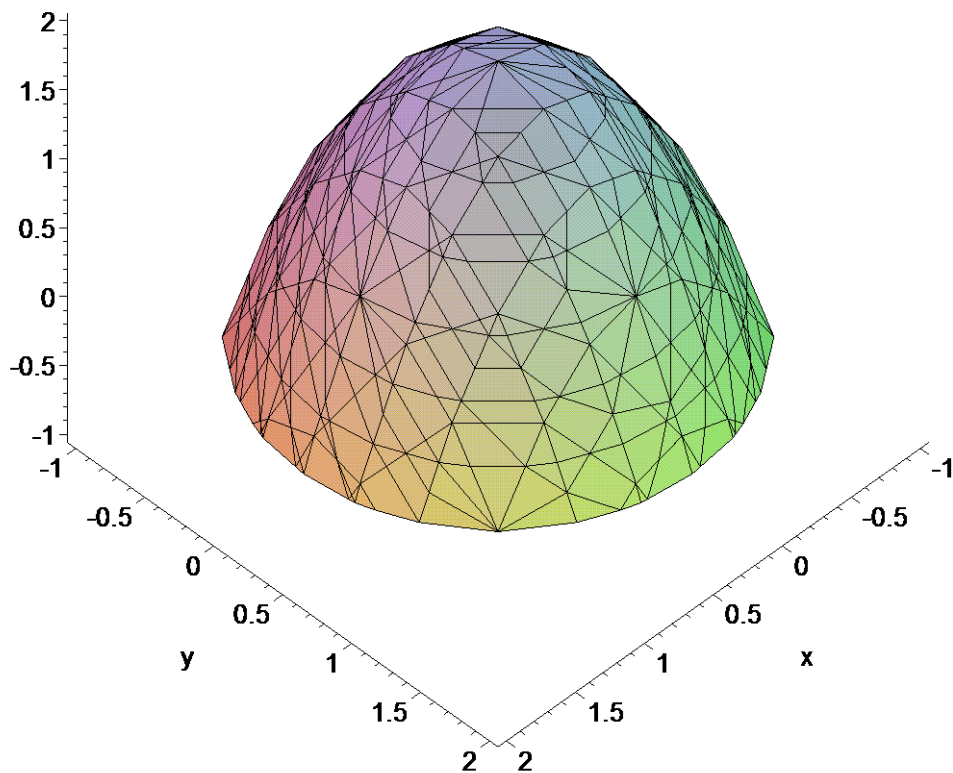
> **restart:with(plots):with(student):**

Warning, the name changecoords has been redefined

Nalezněte souřadnice těžiště přímé úseče rotačního paraboloidu  $0 \leq z \leq a^2 - x^2 - y^2$ .

Nakresleme obrázek pro  $a=1$

> **implicitplot3d(x^2+y^2+z-1,x=-1..2,y=-1..2,z=-1..2,axes=framed);**



Z názoru je zřejmě  $x_T = y_T = 0$

Počítáme tedy  $z_T = \frac{U_{xy}(T)}{V(T)}$ , tzn. následující integrály

> **V:=Tripleint(1,x,y,z,T);**

$$V := \iiint_T 1 \, dx \, dy \, dz$$

> **Uxy:=Tripleint(z,x,y,z,T);**

$$U_{xy} := \iiint_T z \, dx \, dy \, dz$$

Zavedeme válcové souřadnice

> **s:=x=rho\*cos(phi), y=rho\*sin(phi), z=z;**

$$s := x = \rho \cos(\phi), y = \rho \sin(\phi), z = z$$

a tedy těleso  $T$  přejde v  $U$  s omezeními

> **meze:=z=0..a^2-rho^2, rho=0..a, phi=0..2\*Pi;**

$$\text{meze} := z = 0 \dots a^2 - \rho^2, \rho = 0 \dots a, \phi = 0 \dots 2\pi$$

Počítáme tedy integrály

> **V:=changevar({s},subs(T=U,V),[rho,phi,z]);**

$$V := \int \int \int_U |\rho| d\rho d\phi dz$$

> **Uxy:=changevar({s},subs(T=U,Uxy),[rho,phi,z]);**

$$Uxy := \int \int \int_U z |\rho| d\rho d\phi dz$$

Dosazením dostaneme

> **V:=Tripleint(integrand(V),meze);**

$$V := \int_0^{2\pi} \int_0^a \int_0^{a^2-\rho^2} |\rho| dz d\rho d\phi$$

> **Uxy:=Tripleint(integrand(Uxy),meze);**

$$Uxy := \int_0^{2\pi} \int_0^a \int_0^{a^2-\rho^2} z |\rho| dz d\rho d\phi$$

Určíme hodnoty integrálů

> **assume(rho>0,a>0);**

> **V:=value(V);**

$$V := \frac{a^4 \pi}{2}$$

> **Uxy:=value(Uxy);**

$$Uxy := \frac{a^6 \pi}{6}$$

Souřadnice těžiště je

> **z[T]:=Uxy/V;**

$$z_T := \frac{a^2}{3}$$

>

**Objemy těles**

>

**Příklad 10**

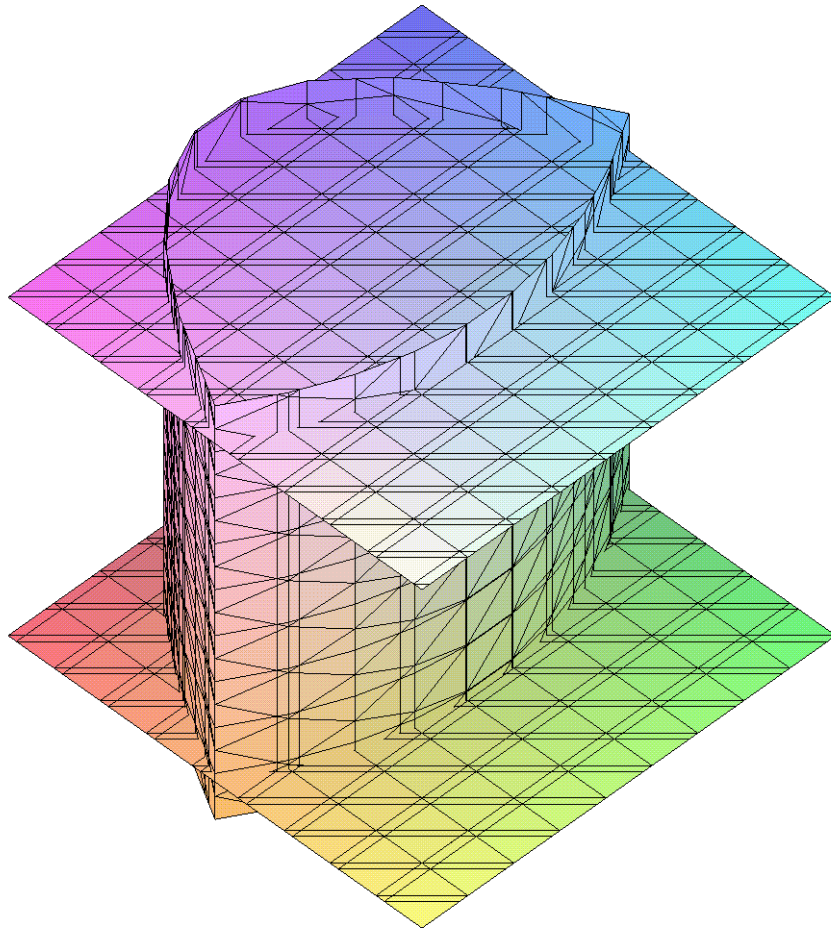
> **restart:with(plots):with(student):**

Warning, the name changecoords has been redefined

Určete objem tělesa  $T$ , jehož hranici tvoří části daných ploch: rovin  $z=0, z=9$  a parabolických válcových ploch  $x^2 - y = 0, x^2 + 3y = 4$

Nakreslíme obrázek

> **implicitplot3d({z=0,z=9,x^2-y,x^2+3\*y-4},x=-1..1,y=0..2,z=-1..10);**



Omezení jsou

```
> meze := y = x^2 .. 1/3 * (4 - x^2), x = -1 .. 1, z = 0 .. 9;
```

$$meze := y = x^2 .. \frac{4}{3} - \frac{x^2}{3}, x = -1 .. 1, z = 0 .. 9$$

Počítáme integrál

```
> Tripleint(1, meze);
```

$$\int_0^9 \int_{-1}^1 \int_{x^2}^{4/3 - \frac{x^2}{3}} 1 \, dy \, dx \, dz$$

```
> value(%);
```

16

```
>
```

```
>
```

*Příklad 11*

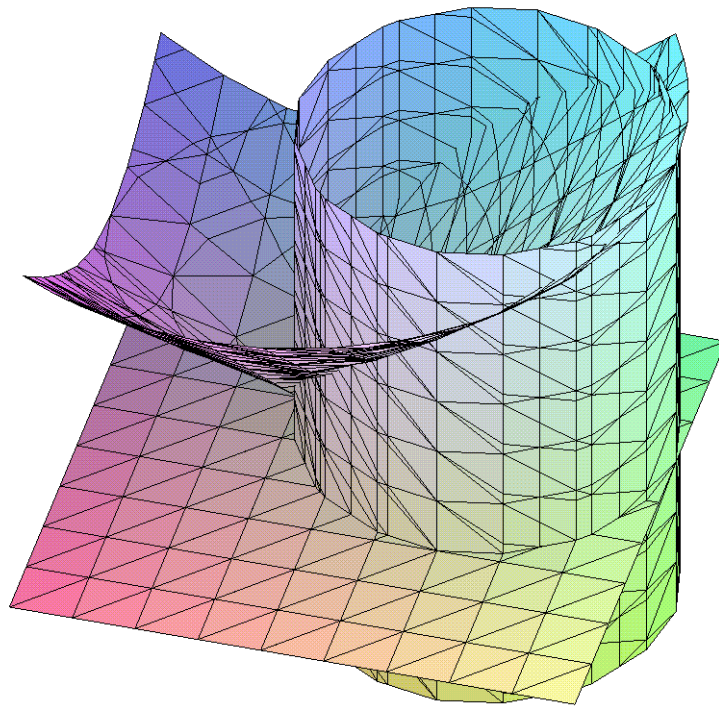
```
> restart:with(plots):with(student):
```

Warning, the name changecoords has been redefined

Stanovme objem tělesa určeného podmínkami  $0 \leq z \leq k\sqrt{x^2 + y^2}$ ,  $x^2 + y^2 \leq 2ry$ ,  $k, r > 0$ .

Nakresleme obrázek pro  $k=r=1$

```
> implicitplot3d({z-sqrt(x^2+y^2), x^2+y^2-2*y, z}, x=-1..2, y=-1..2, z
=-1..2,
orientation=[15,50]);
```



Těleso  $T$  má za dolní plochu půdorysnu, za horní plochu část kuželové plochy  $z = k\sqrt{x^2 + y^2}$ .

Zavedeme cylindrické souřadnice a těleso  $T$  přejde v těleso  $U$  s těmito omezeními

```
> meze:=z=0..k*rho, rho=0..2*r*sin(phi), phi=0..Pi;
meze := z = 0 .. k rho, rho = 0 .. 2 r sin(phi), phi = 0 .. pi
```

Počítáme integrál

```
> V:=Tripleint(1,x,y,z,T);
```

$$V := \iiint_T 1 \, dx \, dy \, dz$$

Transformací

> **s:=x=rho\*cos(phi), y=rho\*sin(phi), z=z;**

$$s := x = \rho \cos(\phi), y = \rho \sin(\phi), z = z$$

dostaneme

> **changevar({s}, subs(T=U,V), [rho, phi, z]);**

$$\iiint_U |\rho| d\rho d\phi dz$$

Dosadíme meze a integrál spočítáme

> **V:=Tripleint(integrand(%), meze);**

$$V := \int_0^\pi \int_0^{2r \sin(\phi)} \int_0^{k\rho} |\rho| dz d\rho d\phi$$

> **assume(r>0, rho>0, k>0);**

> **value(%);**

$$\frac{32 k r^3}{9}$$

>  
>  
>  
>  
>



## 6. Křivkový integrál

System Maple nabízí určitou podporu pro výpočty křivkových integrálů. Nejprve připomeneme, že křivkové integrály jsou dvojího druhu. Přesná definice je uvedena např. v [1].

### 6.1 Křivkový integrál I. druhu

Podpora pro výpočet tohoto integrálu je v rozšiřující knihovně *student*, která obsahuje příkaz *Lineint*. V tomto případě jde jen o matematický zápis křivkového integrálu. Další možností je užít příkaz *PathInt* z rozšiřující knihovny *VectorCalculus*. Předpokládejme, že hladká křivka je zadána parametricky  $x=x(t)$ ,  $y=y(t)$ ,  $z=z(t)$ ,  $a \leq t \leq b$ .

Syntaxe je

*Lineint*( *integrand*( $x,y,z$ ),  $x=x(t),y=y(t),z=z(t),t=a..b$  ) ... *student package*  
*PathInt*( *integrand*( $x,y,z$ ), [ $x,y,z$ ] = *křivka* ) ... *VectorCalculus package*

Ještě připomeňme, že systém Maple rozlišuje velká a malá písmena, protože pro výpočet křivkového integrálu II. druhu se používá podobný příkaz (*LineInt*).

> **restart:**

**with(student):**

# První příkaz vynuluje všechny dosavadní výsledky, druhý nahraje rozšiřující knihovnu do paměti

> **Int(F(x,y,z), s=gamma..` `)=Lineint(F(x,y,z), x,y,z, t=a..b);**

$$\int_{\gamma} F(x,y,z) ds = \int_a^b F(x(t), y(t), z(t)) \sqrt{\left(\frac{d}{dt}y(t)\right)^2 + \left(\frac{d}{dt}z(t)\right)^2 + \left(\frac{d}{dt}x(t)\right)^2} dt$$

Druhá možnost výpočtu, pomocí *PathInt*, bude uvedena až přímo u jednotlivých příkladů, protože obecný zápis je spíše matoucí.

Nyní uveďme několik příkladů na výpočet tohoto druhu integrálu.

>

#### Příklad 1

Spočítejte následující křivkový integrál po částech hladké křivce, která je zadána třemi vrcholy trojúhelníka.

> **Int(x+y, s=gamma..` `);**

$$\int_{\gamma} x + y ds$$

> **body:=[[1,-1],[2,-1],[1,0]];**

*body* := [[1, -1], [2, -1], [1, 0]]

> **with(plots):**

# Pro vykreslení křivky a její popis je nutné nahrát tuto knihovnu do paměti

> **tr:=polygonplot(body):**

# Příkaz pro vykreslení oblasti zadané bodově

> **t1:=textplot([1.3,-1.05,gamma\_1],align={ABOVE,RIGHT}):**

**t2:=textplot([1.5,-0.5,gamma\_2],align={ABOVE,RIGHT}):**

**t3:=textplot([0.8,-0.5,gamma\_3],align={ABOVE,RIGHT}):**

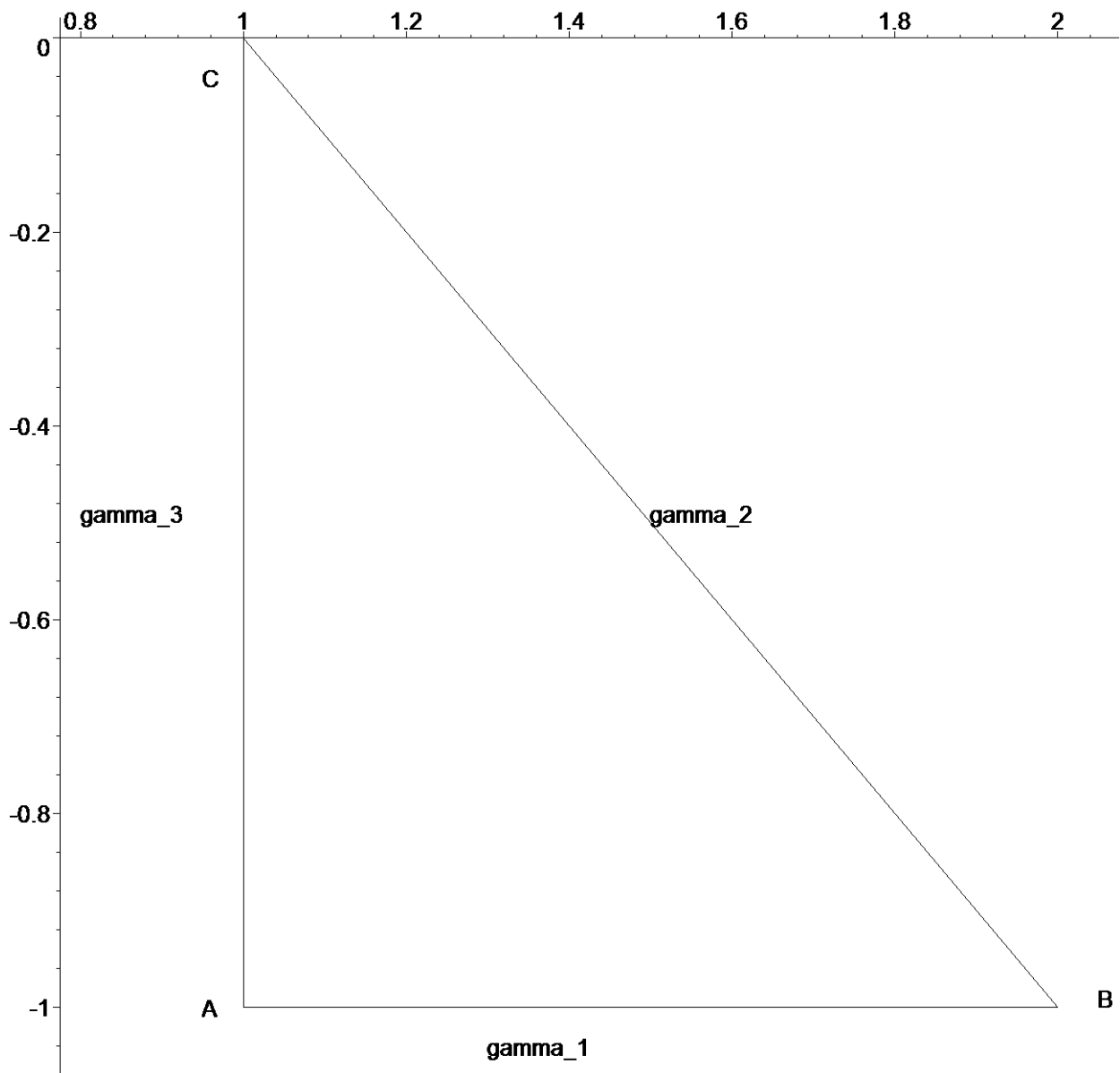
**\_t1:=textplot([0.95,-1.01,A],align={ABOVE,RIGHT}):**

**\_t2:=textplot([2.05,-1,B],align={ABOVE,RIGHT}):**

**\_t3:=textplot([0.95,-0.05,C],align={ABOVE,RIGHT}):**

# Příkaz *textplot* se užívá k zápisu textu do obrázku

```
> display({tr,t1,t2,t3,_t1,_t2,_t3});
```



# Zobrazení grafu a textu do jednoho obrázku

```
>
```

Z obrázku je vidět, že jde opravdu o křivku po částech hladkou, proto integrál rozdělíme na tři dílčí integrály po jednotlivých hladkých křivkách. Na výpočtu integrálu přes křivky gamma\_1, tj. úsečka AB, si ukážeme jednotlivé kroky, které pak budeme považovat za jasné.

Nejprve je třeba určit parametrizaci křivky gamma\_1. Jde o úsečku AB a jednou z možných parametrizací je

```
> par1:=x=1+t,y=-1,t=0..1;
```

$$par1 := x = 1 + t, y = -1, t = 0 .. 1$$

# Identifikátoru *par1* jsem přiřadili posloupnost parametrických rovnic, tzn. při každém dalším použití *par1* bude na jeho místo vložena tato posloupnost.

Dosadíme do křivkového integrálu, tj. počítáme

```
> i1:=Lineint(x+y,par1);
```

$$i1 := \int_0^1 t \sqrt{\left(\frac{d}{dt}(-1)\right)^2 + \left(\frac{d}{dt}(1+t)\right)^2} dt$$

Určíme hodnotu tohoto integrálu pomocí příkazu

> **value(i1);**

$$\frac{1}{2}$$

Určeme nyní parametrizace zbylých dvou křivek

> **par2:=x=2-t,y=-1+t,t=0..1;**

$$par2 := x = 2 - t, y = -1 + t, t = 0 .. 1$$

> **par3:=x=1,y=-1+t,t=0..1;**

$$par3 := x = 1, y = -1 + t, t = 0 .. 1$$

Dosadíme do integrálu

> **i2:=Lineint(x+y,par2[1],par2[2],par2[3]);**

> **i3:=Lineint(x+y,par3[1],par3[2],par3[3]);**

Určíme hodnoty a sečteme jednotlivé integrály

> **value(i1+i2+i3);**

$$1 + \sqrt{2}$$

Nyní se podívejme na výpočet s využitím funkcí z rozšiřující knihovny *VectorCalculus*.

Vysvětlíme pouze základní pojmy a postupy, více naleznete v příloze *Knihovna VectorCalculus*.

Nejprve určíme křivku. Máme zadán trojúhelník pomocí tří bodů. Užijeme konstrukce *LineSegments*, která ze zadaných bodů vytvoří po částech lineární křivku.

> **with(VectorCalculus):**

Warning, the assigned names <, > and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: \*, +, .., Vector, diff, int, limit, series

> **l:=LineSegments(<1,-1>, <2,-1>, <1,0>, <1,-1>);**

$$l := \text{LineSegments}\left(\begin{matrix} e_x - e_y \\ x \quad y \end{matrix}, \begin{matrix} 2e_x - e_y \\ x \quad y \end{matrix}, \begin{matrix} e_x \\ x \end{matrix}, \begin{matrix} e_x - e_y \\ x \quad y \end{matrix}\right)$$

# Je potřeba zadat čtyři body, protože každá úsečka je dána dvěma body a my potřebujeme získat trojúhelník.

Dále pro výpočet užijeme příkaz *PathInt*, která počítá integrál I. druhu.

> **PathInt(x+y, [x,y] = l);**

$$1 + \sqrt{2}$$

Jak je vidět, tento postup je velmi efektivní, ale je potřeba dávat pozor na zadání křivky.

>

## Příklad 2

Spočítejte křivkový integrál přes úsek paraboly  $y^2=2px$  vymezený průsečíky s parabolou  $x^2=2py$

> **restart:with(student):with(plots):with(VectorCalculus):**

Warning, the name changecoords has been redefined

Warning, the assigned names <, > and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: \*, +, .., Vector, diff, int, limit, series

```
> Int(y, s=gamma..`);
```

$$\int_{\gamma} y ds$$

Nakreslení obrázku křivek - nutné zvolit pevné  $\rho=1$ .

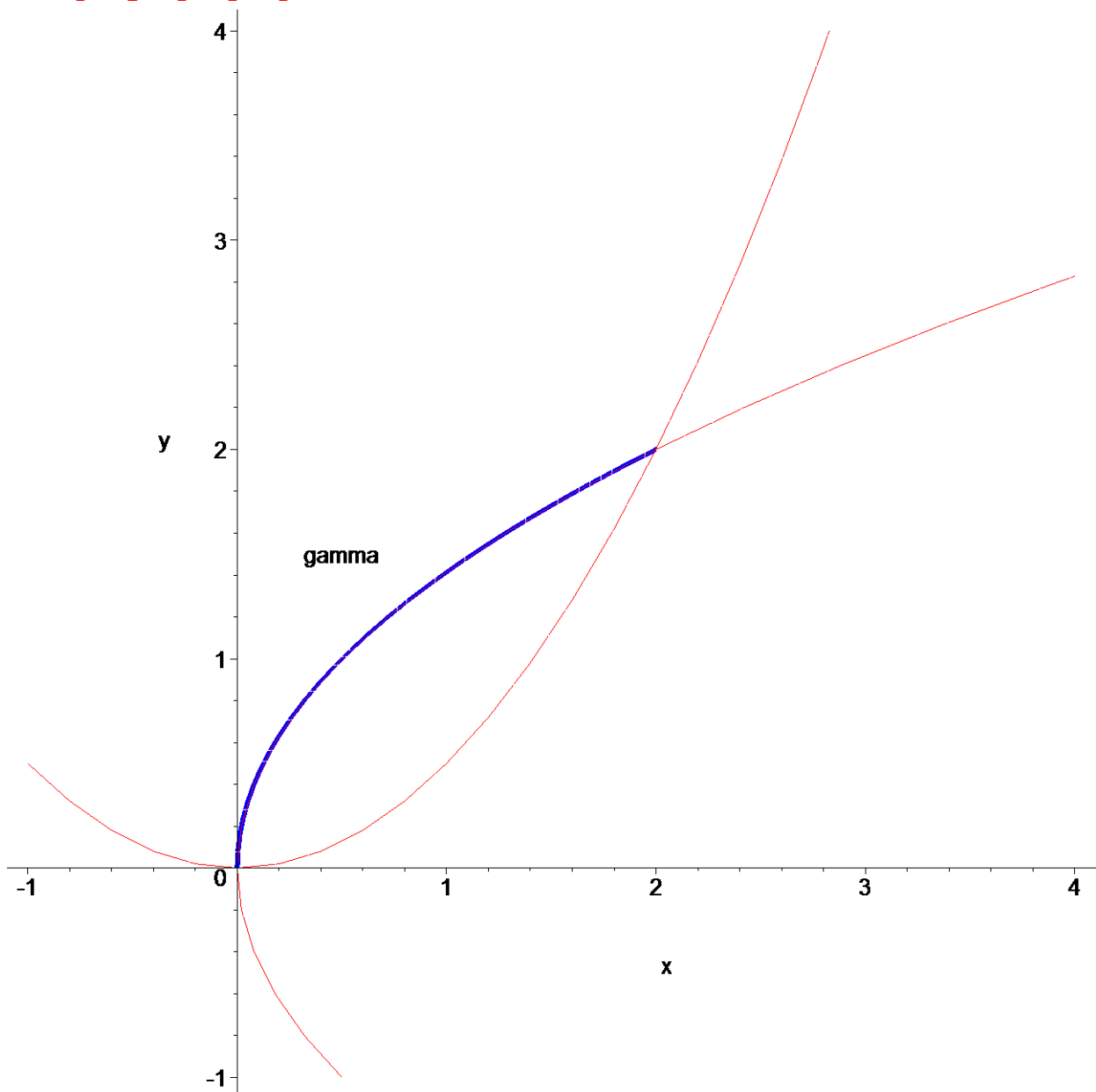
```
> p1:=implicitplot(y^2=2*1*x,x=-1..4,y=-1..4):
p2:=implicitplot(x^2=2*1*y,x=-1..4,y=-1..4):
p:=implicitplot(y^2=2*1*x,x=0..2,y=0..2,color=blue,
thickness=6):
```

# Příkaz *implicitplot* vykresluje křivky dané implicitně.

```
> t1:=textplot([0.5,1.5,gamma]):
```

# Zapsání textu do obrázku

```
> display({p1,p2,p,t1});
```



Zavedeme parametrizaci

```
> par:=x=t^2/(2*rho),y=t,t=0..2*rho;
```

$$par := x = \frac{t^2}{2\rho}, y = t, t = 0 .. 2\rho$$

Počítáme tedy integrál

> **i := Lineint(y, par) ;**

$$i := \int_0^{2\rho} t \sqrt{\left(\frac{d}{dt} t\right)^2 + \left(\frac{\partial}{\partial t} \left(\frac{t^2}{2\rho}\right)\right)^2} dt$$

Výsledná hodnota je

> **value(i) ;**

$$\frac{5}{3}\sqrt{5}\rho^2 - \frac{1}{3}\rho^2$$

Výpočet pomocí knihovny *VectorCalculus*

> **PathInt(y, [x, y] = Path(<t^2/(2\*rho), t>, t=0..2\*rho) ) ;**

$$\frac{5}{3}\sqrt{5}\rho^2 - \frac{1}{3}\rho^2$$

# Využili jsme odvozené parametrizace a příkazu *Path*, která vytvoří požadovanou křivku. Tento příkaz vyžaduje nejméně dva parametry, prvním jsou parametrické rovnice křivky zadané jako vektor a druhým je rozsah daného parametru.

>  
>  
>  
>

### Příklad 3

Spočítejte křivkový integrál podél křivky  $x^2+y^2=r*y$ , jde o kružnici.

> **restart:with(student):with(plots):with(VectorCalculus) :**

Warning, the name changecoords has been redefined

Warning, the assigned names <, > and <|> now have a global binding

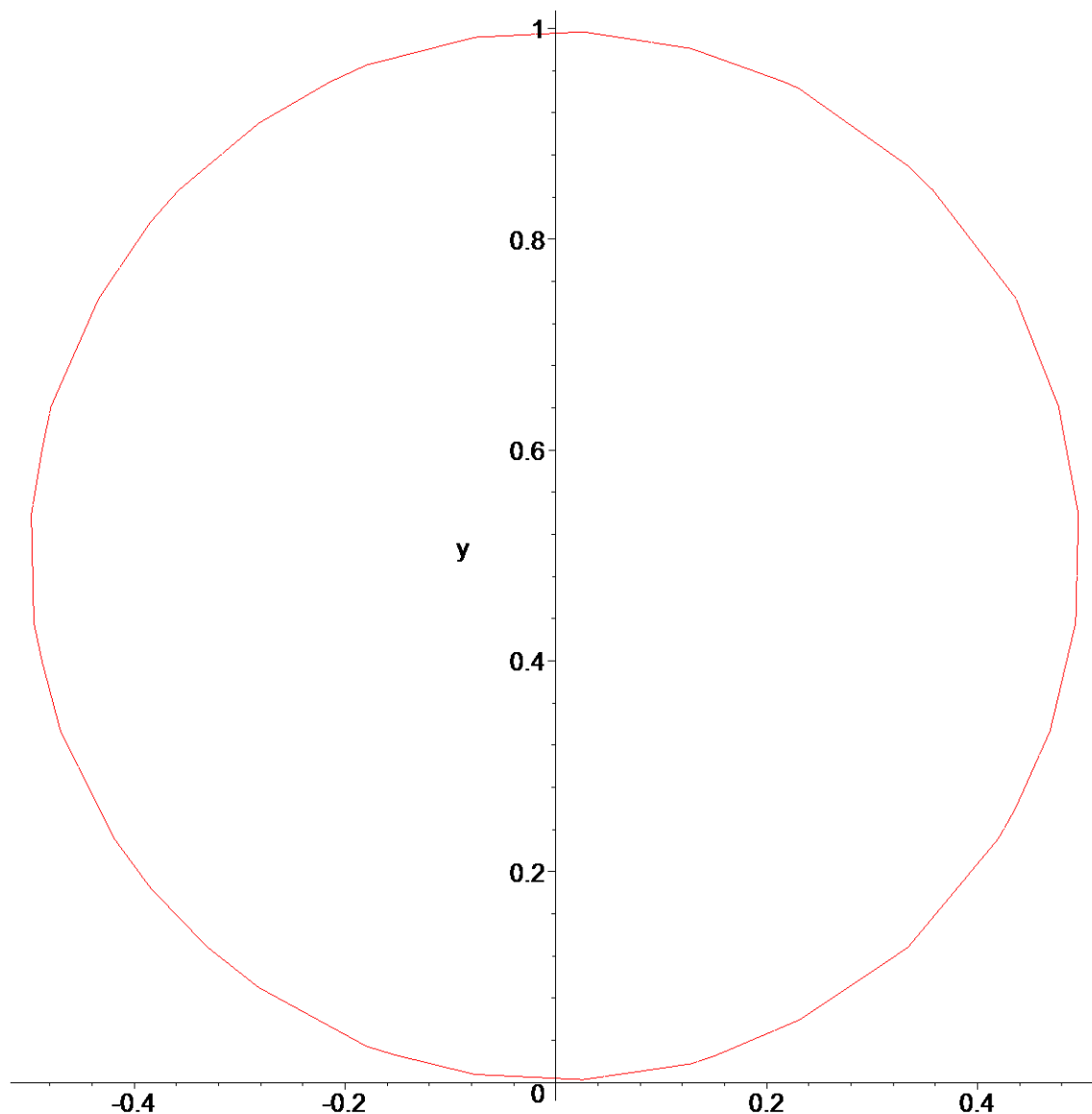
Warning, these protected names have been redefined and unprotected: \*, +, .., Vector, diff, int, limit, series

> **Int(2\*r+y-x, s=gamma..` `) ;**

$$\int_{\gamma} 2r + y - x ds$$

Nakreslení křivky - nutné zvolit pevné  $r=l$ .

> **implicitplot(x^2+y^2=1\*y, x=-1..2\*Pi, y=-1..2\*Pi, numpoints=5000, scaling=constrained) ;**



# Parametr *scaling* určuje měřítka na souřadnicových osách, CONSTRAINED (stejná), UNCONSTRAINED (různá).  
 Parametr *numpoints* zadán pro "hladké" vykreslení kružnice.

Křivku parametrizujeme známými rovnicemi

> **par:=x=r/2\*cos(t), y=r/2+r/2\*sin(t), t=0..2\*Pi;**

$$par := x = \frac{1}{2} r \cos(t), y = \frac{r}{2} + \frac{1}{2} r \sin(t), t = 0 .. 2 \pi$$

Počítáme tedy integrál

> **Lineint(2\*r+y-x, par);**

$$\int_0^{2\pi} \left( \frac{5r}{2} + \frac{1}{2} r \sin(t) - \frac{1}{2} r \cos(t) \right) \sqrt{\left( \frac{\partial}{\partial t} \left( \frac{1}{2} r \cos(t) \right) \right)^2 + \left( \frac{\partial}{\partial t} \left( \frac{r}{2} + \frac{1}{2} r \sin(t) \right) \right)^2} dt$$

Výsledkem je

> **v:=value(%);**

$$v := \int_0^{2\pi} \left( \frac{5r}{2} + \frac{1}{2}r \sin(t) - \frac{1}{2}r \cos(t) \right) \sqrt{\frac{1}{4}r^2 \sin(t)^2 + \frac{1}{4}r^2 \cos(t)^2} dt$$

System Maple neumí tento integrál spočítat přímo, ale po zavedení známé rovnice a požadavku na  $r > 0$  vše dopočítá

> **assume (r>0) ;**

# Přidá podmínku pro hodnoty  $r$ .

> **simplify (v, {sin (t) ^2+cos (t) ^2=1}) ;**

$$\frac{5 r^2 \pi}{2}$$

# Pomocí druhého parametru je možné zadat rovnice, kterými Maple zjednoduší výraz zadaný prvním parametrem. Připomeňme ještě možnost odstranění podmínky na  $r$ .

> **r:='r' ;**

$$r := r$$

# Následující příkaz ukazuje, že podmínka na  $r$  byla odstraněna.

> **simplify (v, {sin (t) ^2+cos (t) ^2=1}) ;**

$$\frac{5 r \sqrt{r^2} \pi}{2} + \frac{1}{2} r \sqrt{r^2} \sin(t) \pi - \frac{1}{2} r \sqrt{r^2} \cos(t) \pi$$

Výpočet pomocí knihovny *VectorCalculus*

> **PathInt ( 2\*r+y-x, [x,y] = Path (<r/2\*cos (t), r/2+r/2\*sin (t)>, t=0..2\*Pi) ) ;**

$$\int_0^{2\pi} \frac{1}{2} \left( \frac{5r}{2} + \frac{1}{2}r \sin(t) - \frac{1}{2}r \cos(t) \right) \sqrt{r^2 \sin(t)^2 + r^2 \cos(t)^2} dt$$

# Využili jsme odvozené parametrizace a příkazu *Path*, která vytvoří požadovanou křivku

> **PathInt ( 2\*r+y-x, [x,y] = Circle (<0,r/2>,r/2) ) ;**

$$\int_0^{2\pi} \frac{1}{2} \left( \frac{5r}{2} + \frac{1}{2}r \sin(t) - \frac{1}{2}r \cos(t) \right) \sqrt{r^2 \sin(t)^2 + r^2 \cos(t)^2} dt$$

# Jiné určení křivky a to pomocí konstrukce *Circle(střed, poloměr)*

>

#### Příklad 4

Spočtete křivkový integrál po oblouku evolventy kružnice

> **restart:with(student):with(plots):with(VectorCalculus) :**

Warning, the name changecoords has been redefined

Warning, the assigned names <, > and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: \*, +, .., Vector, diff, int, limit, series

> **i:=Int (sqrt (x^2+y^2) , s=gamma..` `) ;**

$$i := \int_{\gamma} \sqrt{x^2 + y^2} ds$$

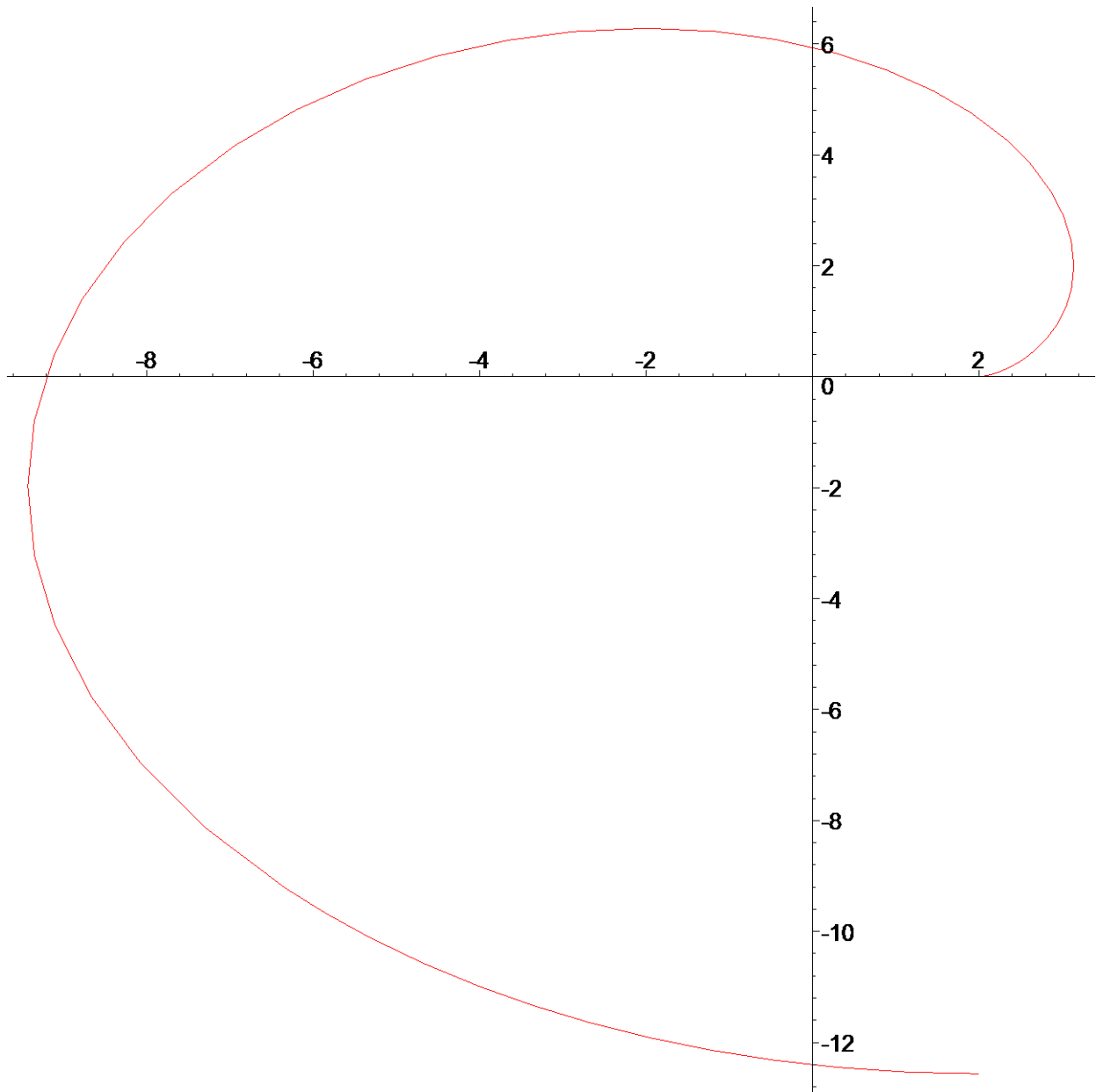
Rovnice evolventy kružnice

```
> par:=x=r*(cos(t)+t*sin(t)),y=r*(sin(t)-t*cos(t)),
t=0..2*Pi;
```

*par := x = r (cos(t) + t sin(t)), y = r (sin(t) - t cos(t)), t = 0 .. 2 π*

Vykresleme graf pro pevně zvolené  $r=2$ .

```
> plot([2*(cos(t)+t*sin(t)),2*(sin(t)-t*cos(t)),
t=0..2*Pi]);
```



# Pro vykreslení křivky dané parametricky je nutné parametrické rovnice zapsat jako seznam, tj. do [...].  
 Dosadíme do křivkového integrálu

```
> Lineint(sqrt(x^2+y^2),par);
```



$$\int_0^{2\pi} \sqrt{r^2 (\cos(t) + t \sin(t))^2 + r^2 (\sin(t) - t \cos(t))^2} dt$$

$$\sqrt{\left(\frac{\partial}{\partial t}(r(\cos(t) + t \sin(t)))\right)^2 + \left(\frac{\partial}{\partial t}(r(\sin(t) - t \cos(t)))\right)^2} dt$$

Zjednodušíme

> **simplify(%);**

$$r^2 \int_0^{2\pi} \sqrt{1+t^2} t dt$$

Výslednou hodnotu dostaneme výpočtem předchozího integrálu

> **v:=value(%);**

$$v := r^2 \left( \frac{(1+4\pi^2)^{(3/2)}}{3} - \frac{1}{3} \right)$$

Výpočet pomocí *PathInt*.

> **v1:=PathInt(sqrt(x^2+y^2), [x,y] = Path(<r\*(cos(t)+t\*sin(t)), r\*(sin(t)-t\*cos(t))>, t=0..2\*Pi));**

$$v1 := \frac{\sqrt{r^2 (r^2 + 4 r^2 \pi^2)^{(3/2)} - r^4}}{3 r^2}$$

> **is(simplify(v1)=simplify(v));**

*true*

# Kontrola, že jde o stejný výraz

>

### Příklad 5

Spočítejte křivkový integrál po druhém závitě pravotočivé šroubovice

> **restart:with(student):with(plots):with(VectorCalculus):**

Warning, the name changecoords has been redefined

Warning, the assigned names <, > and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: \*, +, .., Vector, diff, int, limit, series

> **Int(x\*z, s=gamma..`);**

$$\int_{\gamma} x z ds$$

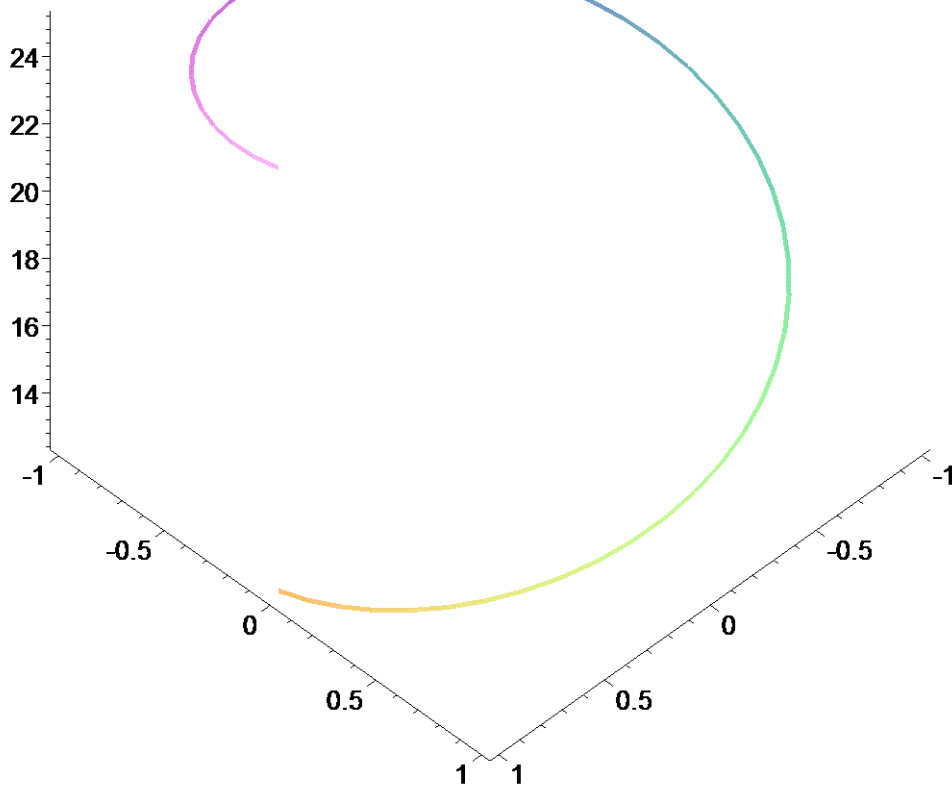
Parametrizace šroubovice

> **par:=x=r\*cos(t), y=r\*sin(t), z=t\*v, t=2\*Pi..4\*Pi;**

$$par := x = r \cos(t), y = r \sin(t), z = t v, t = 2 \pi .. 4 \pi$$

Vykreslení křivky pro  $r=1$  a  $v=2$ .

> **spacecurve([cos(t), sin(t), t\*2], t=2\*Pi..4\*Pi, axes=framed, thickness=5);**



# Příkaz pro vykreslení křivky v prostoru, parametr *thickness* určuje šířku pera pro kreslení.  
 Dosadíme parametrické vyjádření křivky do křivkového integrálu

> `Lineint(x*z, par);`

$$\int_{2\pi}^{4\pi} r \cos(t) t v \sqrt{\left(\frac{\partial}{\partial t}(t v)\right)^2 + \left(\frac{\partial}{\partial t}(r \sin(t))\right)^2 + \left(\frac{\partial}{\partial t}(r \cos(t))\right)^2} dt$$

Hodnotu integrálu určíme pomocí příkazu

> `value(%);`

0

Výpočet pomocí *PathInt*.

> `PathInt(x*z, [x,y,z] = Path(<r*cos(t), r*sin(t), t*v>, t=2*Pi..4*Pi));`

0

# Jde o výpočet ve 3D, stačí pouze změnit počet proměnných, tj. [x,y,z]

>

### Příklad 6

Spočteme následující integrál po křivce, která je průnikem koule a roviny  $x=y$ .

```
> restart:with(student):with(plots):with(VectorCalculus):
```

```
Warning, the name changecoords has been redefined
```

```
Warning, the assigned names <, > and <|> now have a global binding
```

```
Warning, these protected names have been redefined and unprotected: *, +, .., Vector, diff, int, limit, series
```

```
> Int(x+y, s=gamma..` `);
```

$$\int_{\gamma} x + y \, ds$$

Rovnice křivky

```
> x^2+y^2+z^2=r^2;x=y;r>0;
```

$$x^2 + y^2 + z^2 = r^2$$

$$x = y$$

$$0 < r$$

Nejprve nakresleme obrázek, pro který je nutné zvolit pevné  $r$ , v našem případě pro jednoduchost  $r=1$

```
> p:=implicitplot3d(x^2+y^2+z^2=1^2,x=0..2,y=0..2,z=0..2,
 numpoints=5000,color=green,style=patchnogrid):
```

# Příkaz pro vykreslení funkce ve 3D, která je dána implicitně. Parametr *numpoints* je nastaven na vyšší hodnotu, aby byla koule zobrazena jako koule. Parametrem *patchnogrid* nebudeme zobrazovat triangulaci na dané ploše.

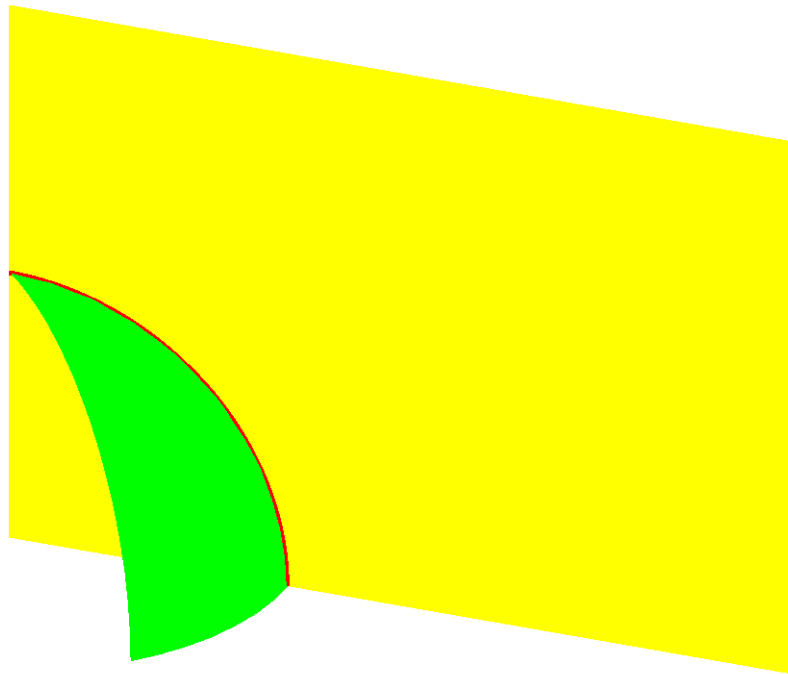
```
> p1:=implicitplot3d(x-y,x=0..2,y=0..2,z=0..2,
 numpoints=5000,color=yellow,style=patchnogrid):
```

# Pro lepší zobrazení průniku obou ploch zobrazíme křivku, a to pomocí jejího parametrického vyjádření, které odvodíme později.

```
> p2:=spacecurve([sqrt(2)/2*sin(t),sqrt(2)/2*sin(t),cos(t)],
 t=0..Pi/2,color=red,thickness=8,numpoints=5000,
 style=patchnogrid):
```

# Následujícím příkazem vykreslíme předchozí grafy do jednoho obrázku a pomocí parametru *orientation* určíme výchozí pohled.

```
> display({p,p1,p2},orientation=[-24,63]);
```



Zavedeme parametrizaci této křivky ( zvolíme nejprve  $z$  a pak dosadíme do rovnice koule).

```
> par:=x=sqrt(2)/2*r*sin(t),y=sqrt(2)/2*r*sin(t),
z=r*cos(t),t=0..Pi/2;
```

$$par := x = \frac{1}{2}\sqrt{2} r \sin(t), y = \frac{1}{2}\sqrt{2} r \sin(t), z = r \cos(t), t = 0 \dots \frac{\pi}{2}$$

Počítáme tedy integrál

```
> i:=Lineint(x+y,par);
```

$$i := \int_0^{\frac{\pi}{2}} \sqrt{2} r \sin(t) \sqrt{2 \left( \frac{\partial}{\partial t} \left( \frac{1}{2} \sqrt{2} r \sin(t) \right) \right)^2 + \left( \frac{\partial}{\partial t} (r \cos(t)) \right)^2} dt$$

```
> assume(r>0);
```

# Podmínka na  $r$

Výsledek je po

> `value(i);`

$$\sqrt{2} r^2$$

Výpočet pomocí *PathInt*.

> `PathInt(x+y, [x,y,z] = Path(<sqrt(2)/2*r*sin(t),  
sqrt(2)/2*r*sin(t), r*cos(t)>, t=0..Pi/2));`

$$\sqrt{2} r^2$$

>

>

Na dalších příkladech ukážeme využití křivkového integrálu I. druhu v praxi a to při výpočtu délky křivky, těžiště křivky, hmotnosti křivky popř. plošného obsahu.

### Příklad 7

> `restart:with(student):with(plots):with(VectorCalculus):`

Warning, the name `changecoords` has been redefined

Warning, the assigned names `<, >` and `<|>` now have a global binding

Warning, these protected names have been redefined and unprotected: `*`, `+`, `..`, `Vector`, `diff`, `int`, `limit`, `series`

Vypočtěte délku oblouku cykloidy. Při určování **délky křivky** jde o výpočet integrálu

> `Int(1, s=gamma..`);`

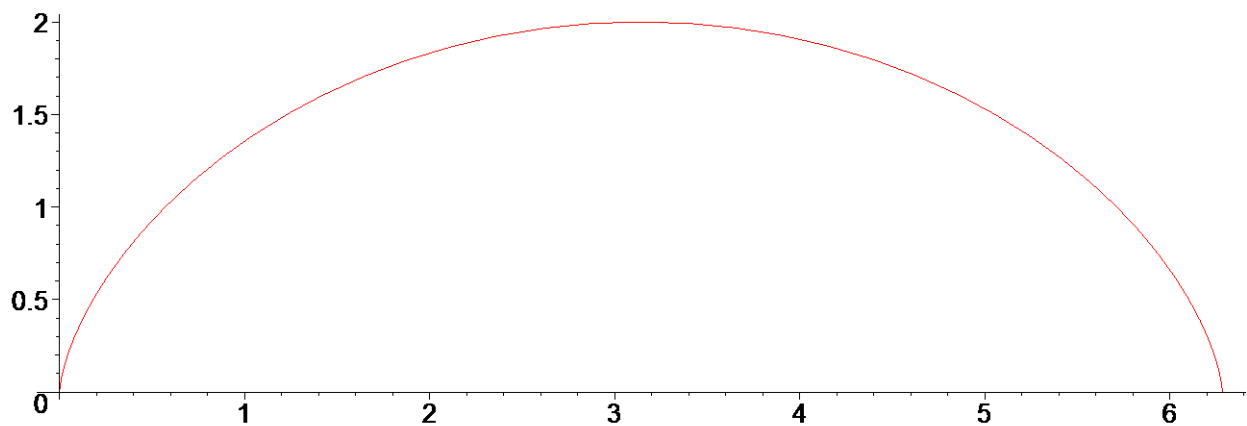
$$\int_{\gamma} 1 ds$$

Parametrické rovnice cykloidy jsou

> `par:=x=r*(t-sin(t)), y=r*(1-cos(t)), t=0..2*Pi;`

$$par := x = r(t - \sin(t)), y = r(-\cos(t) + 1), t = 0 .. 2\pi$$

> `plot([1*(t-sin(t)), 1*(1-cos(t)), t=0..2*Pi], scaling=constrained);`



# Vykreslení grafu pro pevně zvolené  $r=1$ .

Opět uijeme příkazu pro matematický zápis křivkového integrálu I. druhu a to

> **i:=Lineint(1,par);**

$$i := \int_0^{2\pi} \sqrt{\left(\frac{\partial}{\partial t}(r(-\cos(t)+1))\right)^2 + \left(\frac{\partial}{\partial t}(r(t-\sin(t)))\right)^2} dt$$

Spočteme výslednou hodnotu

> **value(i);**

$$\frac{8r^2}{\sqrt{r^2}}$$

Doplňme ještě podmínku na  $r$  a spočteme výsledek

> **assume(r>0);**

> **value(i);**

$$8r$$

Následující příkazem podmínku na  $r$  opět zrušíme

```
> r:='r';
```

$r := r$

Výpočet pomocí *PathInt*.

```
> PathInt(1, [x,y] = Path(<r*(t-sin(t)), r*(1-cos(t))>,
 t=0..2*Pi));
```

$$\frac{8r^2}{\sqrt{r^2}}$$

>

>

### Příklad 8

Spočítejte délku kardioidy

```
> restart:with(student):with(plots):with(VectorCalculus):
```

Warning, the name changecoords has been redefined

Warning, the assigned names <,> and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: \*, +, .., Vector, diff, int, limit, series

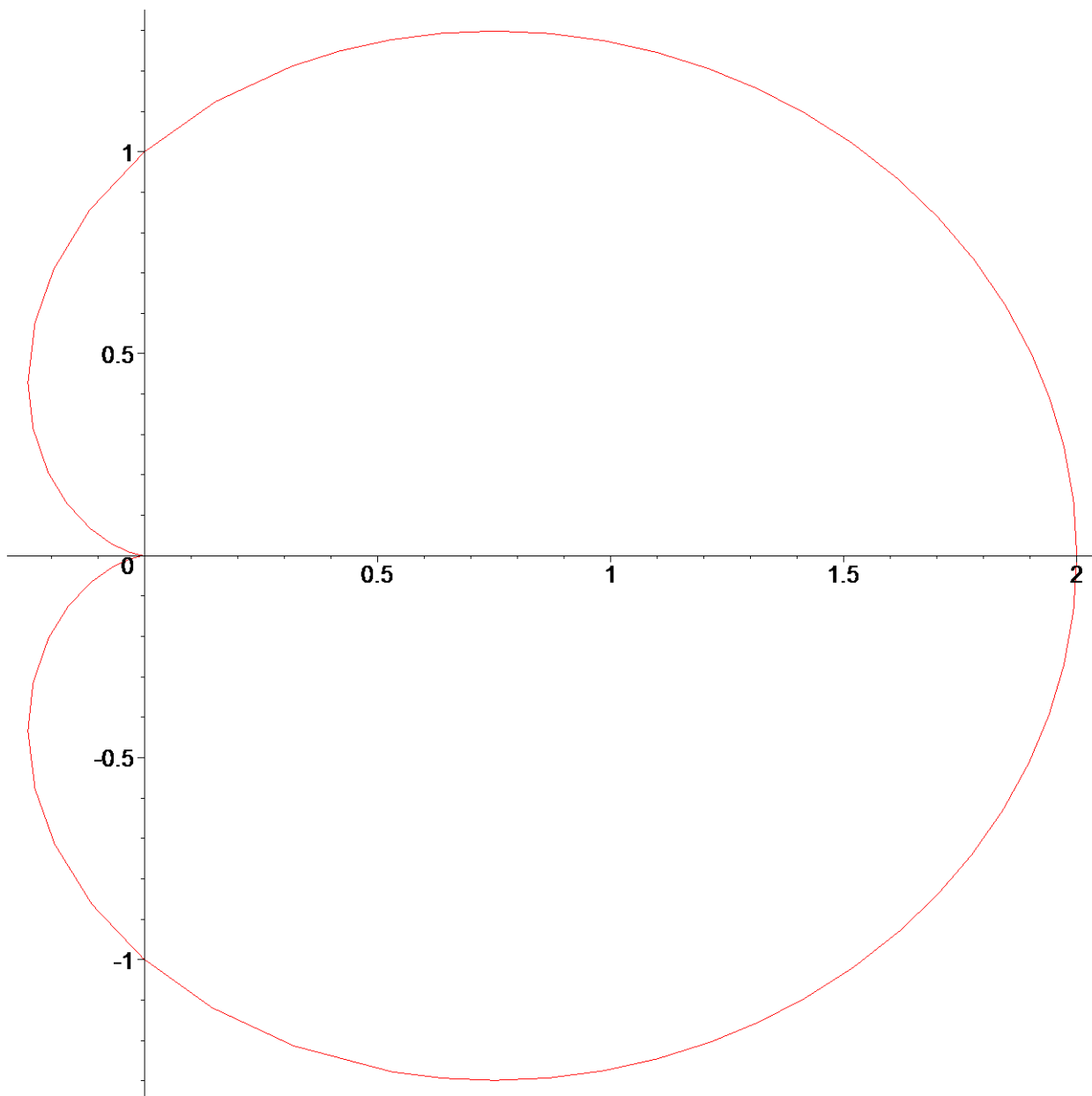
Parametrické rovnice jsou

```
> par:=x=r*(1+cos(t))*cos(t),y=r*(1+cos(t))*sin(t),
 t=0..2*Pi;
```

$$par := x = r (\cos(t) + 1) \cos(t), y = r (\cos(t) + 1) \sin(t), t = 0 .. 2 \pi$$

Nakreslíme graf křivky pro  $r=1$

```
> plot([1*(1+cos(t))*cos(t), 1*(1+cos(t))*sin(t), par[3]]);
```



# Křivka zadaná parametricky, proto je nutné užít [...].

Definice křivkového integrálu předpokládá křivku hladkou, popř. po částech hladkou. Je tedy nutné rozdělit kardioidu na dvě hladké křivky. Proto budeme počítat integrál např. jen v horní polorovině. Počítáme tedy křivkový integrál

> **i := Lineint(1, par[1], par[2], t=0..Pi) ;**

$$i := \int_0^{\pi} \sqrt{\left(\frac{\partial}{\partial t}(r(\cos(t)+1)\cos(t))\right)^2 + \left(\frac{\partial}{\partial t}(r(\cos(t)+1)\sin(t))\right)^2} dt$$

# Zde jsme museli užít volání jednotlivých členů posloupnosti, protože byl zvolen jiný rozsah parametru  $t$ .

Podmínka na  $r$

> **assume (r>0) ;**

Délka křivky v horní polorovině je

> **value(i) ;**



Protože jsme uvažovali pouze horní polorovinu, výsledná délka křivky je

```
> vysledek:=2*value(i);
```

*vysledek := 8 r~*

Výpočet pomocí *PathInt*.

```
> PathInt(1, [x,y] =
 Path(<r*(1+cos(t))*cos(t), r*(1+cos(t))*sin(t)>,
 t=0..2*Pi));
```

*8 r~*

# Tento příkaz provede výpočet i při zadání po částech hladké křivky

```
>
```

```
>
```

### Příklad 9

Spočítejte hmotnost části elipsy v I. kvadrantu, je-li délková hustota  $\mu$  v každém bodě elipsy rovna y-ové souřadnici daného bodu.

```
> restart:with(student):with(plots):with(VectorCalculus):
```

*Warning, the name changecoords has been redefined*

*Warning, the assigned names <,> and <|> now have a global binding*

*Warning, these protected names have been redefined and unprotected: \*, +, .., Vector, diff, int, limit, series*

Hmotnost křivky  $\Gamma$  je dána integrálem

```
> m(Gamma)=Int(mu(x,y), s=Gamma..` `);
```

$$m(\Gamma) = \int_{\Gamma} \mu(x, y) ds$$

Parametrické rovnice části elipsy v I. kvadrantu jsou

```
> par:=x=a*cos(t), y=b*sin(t), t=0..Pi/2;
```

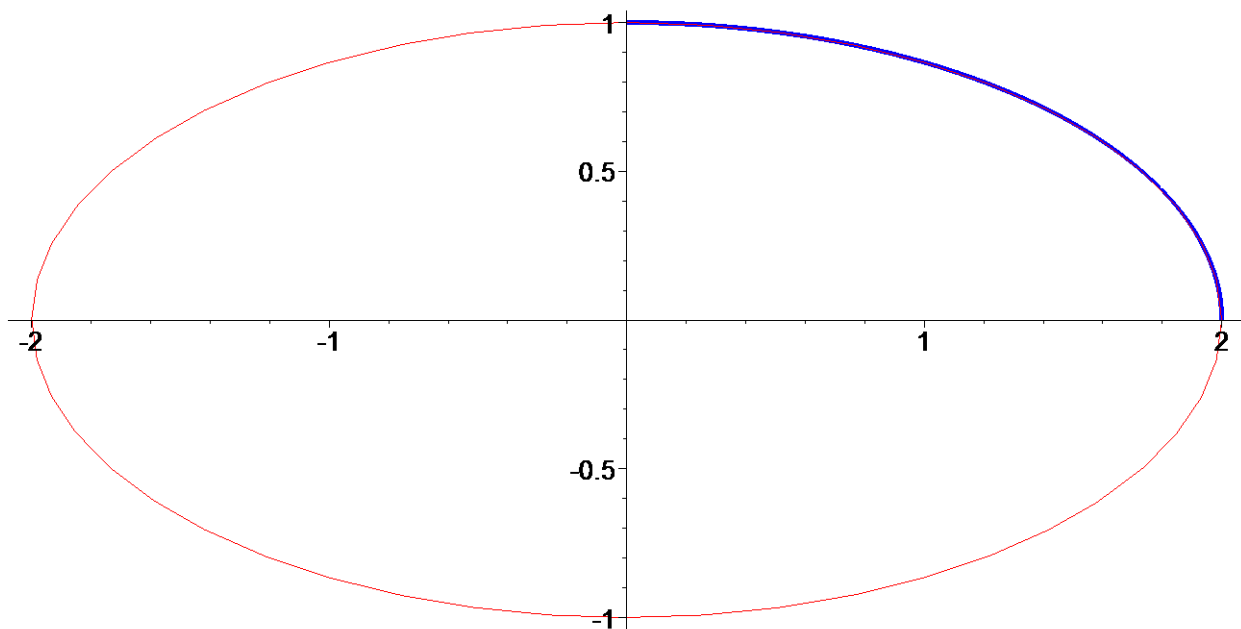
$$par := x = a \cos(t), y = b \sin(t), t = 0 .. \frac{\pi}{2}$$

Nakresleme obrázek, zvolme  $a=2, b=1$

```
> p:=plot([2*cos(t), 1*sin(t), t=0..2*Pi]):
 p1:=plot([2*cos(t), 1*sin(t), t=0..Pi/2], thickness=5,
 color=blue):
```

# Parametr *thickness* určuje šířku pera pro vykreslení křivky.

```
> display({p,p1}, scaling=constrained);
```



Zapišme křivkový integrál

> **i := Lineint(y, par) ;**

$$i := \int_0^{\frac{\pi}{2}} b \sin(t) \sqrt{\left(\frac{\partial}{\partial t}(a \cos(t))\right)^2 + \left(\frac{\partial}{\partial t}(b \sin(t))\right)^2} dt$$

# První parametr příkazu zastupuje integrand, kterým v tomto případě je délková hustota, dle zadání y.

Pro upravení integrálu je třeba zadat podmínky pro parametry a, b

> **assume(a > 0) ; assume(b > 0) ;**

Výsledná hmotnost je

> **value(i) ;**

$$\frac{1}{2} \frac{b \left( a^2 \ln(a) - b \sqrt{-a^2 + b^2} - a^2 \ln(\sqrt{-a^2 + b^2} + b) \right)}{\sqrt{-a^2 + b^2}}$$

Ukažme, jak bychom sami postupovali při výpočtu integrálu i

> `i1:=simplify(i);`

$$i1 := b \int_0^{\frac{\pi}{2}} \sin(t) \sqrt{a^2 - a^2 \cos(t)^2 + b^2 \cos(t)^2} dt$$

Zavedeme substituci

> `changevar (u=cos (t) , i1 , u) ;`

$$b \int_0^1 \sqrt{a^2 - a^2 u^2 + b^2 u^2} du$$

# Příkaz je obsažen ve *student* package. Prvním parametrem je rovnice pro substituci, druhým integrál nebo sumace, do které substituci chceme dosadit, třetím parametrem je nová proměnná.

Tento integrál je tabulkovým integrálem. Spočteme ho pomocí systému Maple

> `value(%);`

$$b a \operatorname{hypergeom}\left(\left[\frac{-1}{2}, \frac{1}{2}\right], \left[\frac{3}{2}\right], -\frac{-a^2 + b^2}{a^2}\right)$$

Výpočet pomocí *PathInt*.

> `PathInt(y, [x,y] = Path(<a*cos (t) , b*sin (t)>, t=0..Pi/2));`

$$\frac{\frac{1}{2} b (a^2 \ln(a) - b \sqrt{-a^2 + b^2} - a^2 \ln(\sqrt{-a^2 + b^2} + b))}{\sqrt{-a^2 + b^2}}$$

>

### Příklad 10

Vypočtete těžiště části asteroidy v I. kvadrantu

> `restart:with(student):with(plots):with(VectorCalculus):`

Warning, the name `changecoords` has been redefined

Warning, the assigned names `<, >` and `<|>` now have a global binding

Warning, these protected names have been redefined and unprotected: `*`, `+`, `..`, `Vector`, `diff`, `int`, `limit`, `series`

Souřadnice těžiště křivky jsou dány následujícími vztahy

> `x[T]:=M[x](gamma)/s(gamma)=`

`Int(y, s=gamma..`)/ (Int(1, s=gamma..`));`

`y[T]:=M[y](gamma)/s(gamma)=`

`Int(x, s=gamma..`)/ (Int(1, s=gamma..`));`

$$x_T := \frac{M_x(\gamma)}{s(\gamma)} = \frac{\int_{\gamma} y ds}{\int_{\gamma} 1 ds}$$

$$y_T := \frac{M_y(\gamma)}{s(\gamma)} = \frac{\int_{\gamma} x ds}{\int_{\gamma} 1 ds}$$

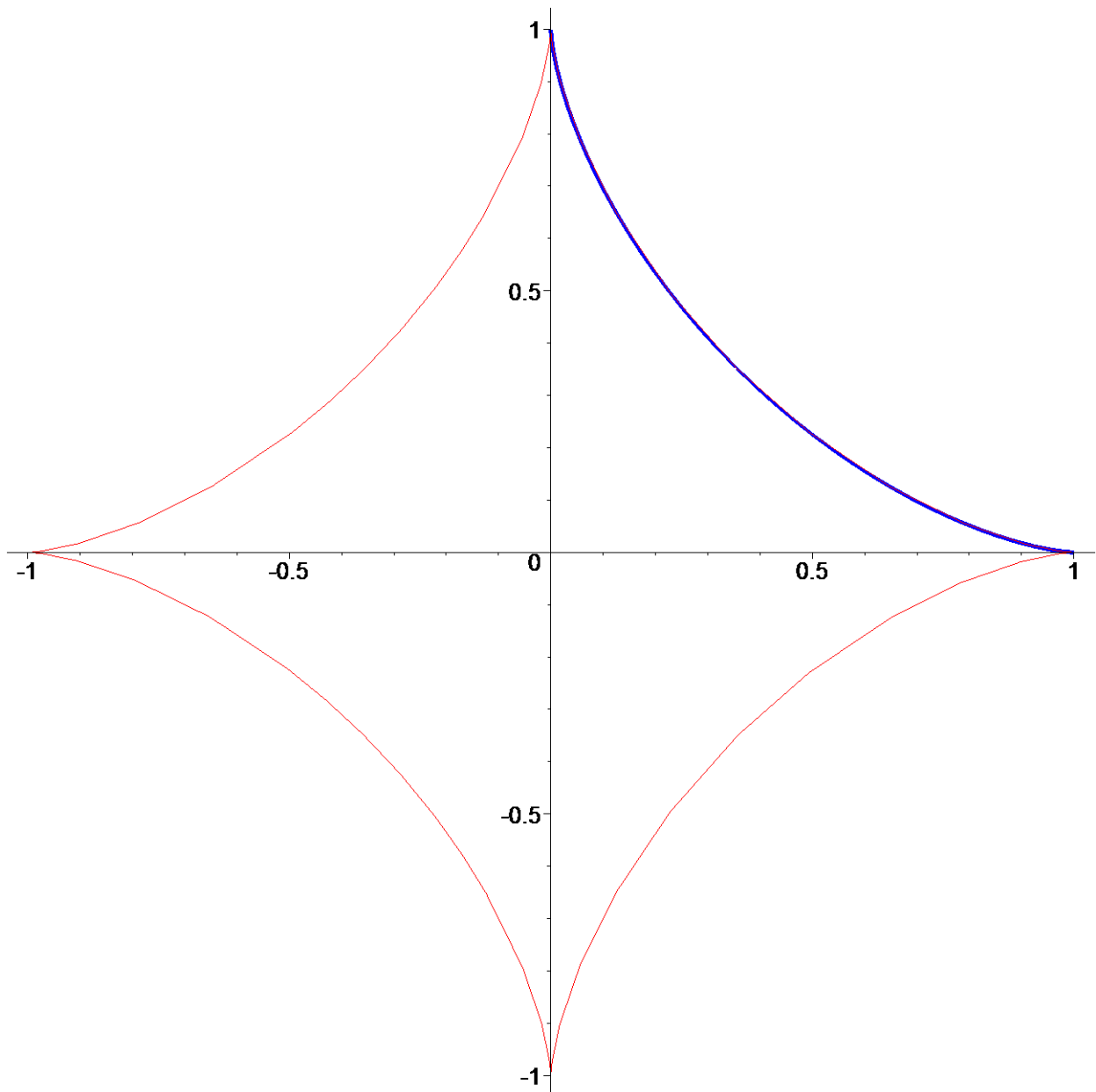
Parametrické rovnice asteroidy v I. kvadrantu jsou

> **par:=x=r\*cos(t)^3,y=r\*sin(t)^3,t=0..Pi/2;**

$$par := x = r \cos(t)^3, y = r \sin(t)^3, t = 0 .. \frac{\pi}{2}$$

Nakresleme graf

> **plot([cos(t)^3,sin(t)^3,t=0..2\*Pi]):**  
**plot([cos(t)^3,sin(t)^3,t=0..Pi/2],color=blue,thickness=5):**  
**display({%,%});**



Spočteme délku křivky

```
> s:=Int(1,s=gamma..` `);
```

$$s = \int_{\gamma} 1 ds$$

Dosazením parametrických rovnic do křivkového integrálu dostaneme

```
> s:=Lineint(1,par);
```

$$s := \int_0^{\frac{\pi}{2}} \sqrt{\left(\frac{\partial}{\partial t}(r \cos(t)^3)\right)^2 + \left(\frac{\partial}{\partial t}(r \sin(t)^3)\right)^2} dt$$

Přidáme ještě podmínku pro  $r$  a spočteme tento integrál

```
> assume(r>0);
```

```
> sv:=simplify(value(%));
```

$$sv := \frac{3 r \sim}{2}$$

Dále spočteme odpovídající momenty vzhledem k souřadným osám

```
> Mx:=Lineint(y,par);
```

```
My:=Lineint(x,par);
```

# Zadání integrálů

Určeme jejich hodnoty

```
> Mxv:=simplify(value(Mx));
```

```
Myv:=simplify(value(My));
```

$$Mxv := \frac{3 r \sim^2}{5}$$

$$Myv := \frac{3 r \sim^2}{5}$$

Nakonec dosadíme do vzorců pro výpočet souřadnic těžiště

```
> x[T]:=Mxv/sv;
```

```
y[T]:=Myv/sv;
```

$$x_T := \frac{2 r \sim}{5}$$

$$y_T := \frac{2 r \sim}{5}$$

Výpočet pomocí **PathInt**.

```
> _s:=simplify(PathInt(1,[x,y] = Path(<r*cos(t)^3,r*sin(t)^3>,t=0..Pi/2)));
```

$$_s := \frac{3 r \sim}{2}$$

```
> _Mx:=simplify(PathInt(y,[x,y] = Path(<r*cos(t)^3,r*sin(t)^3>,t=0..Pi/2)));
```

$$_Mx := \frac{3 r \sim^2}{5}$$

```
> _My:=simplify(PathInt(x,[x,y] = Path(<r*cos(t)^3,r*sin(t)^3>,
t=0..Pi/2)));
```

$$_My := \frac{3 r^2}{5}$$

```
>
>
```

### Příklad 11

Určete statický moment 1. závitu kuželové šroubovice vzhledem k souřadnicové rovině  $xy$ , je-li délková hustota v každém bodě křivky rovna kvadrátu vzdálenosti od roviny  $xy$ .

```
> restart:with(student):with(plots):with(VectorCalculus):
Warning, the name changecoords has been redefined
```

```
Warning, the assigned names <,> and <|> now have a global binding
```

```
Warning, these protected names have been redefined and unprotected: *, +, ..,
Vector, diff, int, limit, series
```

První statický moment vzhledem k souřadnicové rovině  $xy$  je definován jako

```
> M[xy]=Int(z*mu(x,y,z),s=gamma..` `);
```

$$M_{xy} = \int_{\gamma} z \mu(x, y, z) ds$$

kde  $\mu$  je délková hustota. Analogicky definujeme zbylé dva momenty.

Parametrické rovnice kuželové šroubovice

```
> x=t*cos(t);y=t*sin(t);z=v*t;t=0..2*Pi;
```

$$x = t \cos(t)$$

$$y = t \sin(t)$$

$$z = v t$$

$$t = 0 \dots 2 \pi$$

Pro jednoduchost položme  $v=1$

```
> par:=x=t*cos(t),y=t*sin(t),z=t,t=0..2*Pi;
```

$$par := x = t \cos(t), y = t \sin(t), z = t, t = 0 \dots 2 \pi$$

Nakresleme obrázek

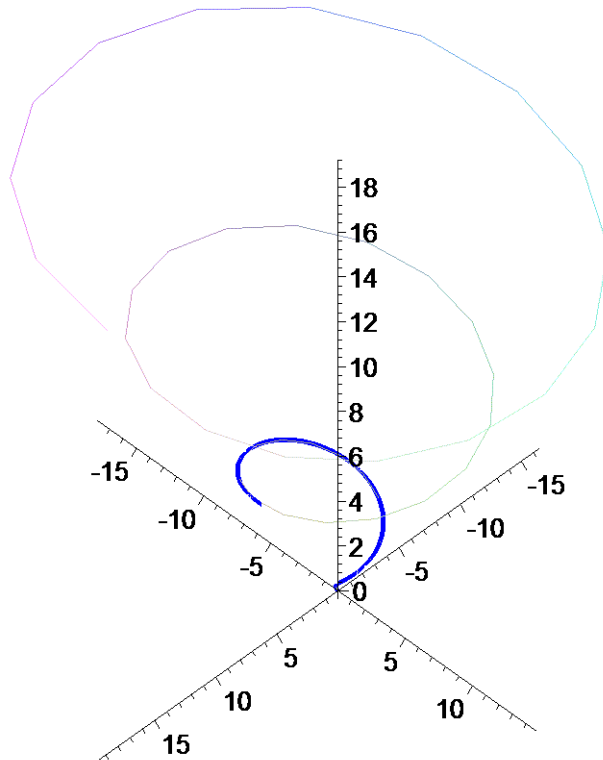
```
> p:=spacecurve([t*cos(t),t*sin(t),t],t=0..6*Pi,
axes=normal):
```

# Kuželová šroubovice je křivka v prostoru, proto byl použit tento příkaz. Parametr *axes* určuje zobrazení souřadnicového systému.

```
> p1:=spacecurve([t*cos(t),t*sin(t),t],t=0..2*Pi,
numpoints=10000,axes=normal,color=blue,thickness=5):
```

# Zvýraznění prvního závitu šroubovice, parametr *numpoints* uveden z důvodu "hladšího" zobrazení křivky.

```
> display({p,p1});
```



Zapišme křivkový integrál s přihlédnutím na zadanou délkovou hustotu, která je dána kvadrátem vzdálenosti bodu od roviny  $xy$ .

> **i:=Lineint(z\*z^2,par);**

$$i := \int_0^{2\pi} t^3 \sqrt{\left(\frac{d}{dt}(t \cos(t))\right)^2 + \left(\frac{d}{dt}t\right)^2 + \left(\frac{d}{dt}(t \sin(t))\right)^2} dt$$

Výsledný moment vzhledem k souřadnicové rovině  $xy$  je roven

> **value(i);**

$$\frac{4\pi^2(2+4\pi^2)^{(3/2)}}{5} - \frac{4(2+4\pi^2)^{(3/2)}}{15} + \frac{8\sqrt{2}}{15}$$

Numericky spočteme integrál pomocí

> **evalf(i);**

2038.743287

Výpočet pomocí *PathInt*.

```
> PathInt(z^3, [x, y, z] = Path(<t*cos(t), t*sin(t), t>, t=0..2*Pi));
```

$$\frac{4\pi^2(2+4\pi^2)^{(3/2)}}{5} - \frac{4(2+4\pi^2)^{(3/2)}}{15} + \frac{8\sqrt{2}}{15}$$

```
>
```

```
>
```

### Příklad 12

Vypočtěte plošný obsah části válcové plochy  $x^2+y^2=r^2$  omezené rovinou  $z=0$ , a plochou  $z=r-x^2/r$ .

```
> restart:with(student):with(plots):with(VectorCalculus):
```

```
Warning, the name changecoords has been redefined
```

```
Warning, the assigned names <,> and <|> now have a global binding
```

```
Warning, these protected names have been redefined and unprotected: *, +, .., Vector, diff, int, limit, series
```

Nakresleme obrázek

```
> p:=implicitplot3d(x^2+y^2=1,x=-1..2,y=-1..2, z=-1..2,color=red,style=hidden):
```

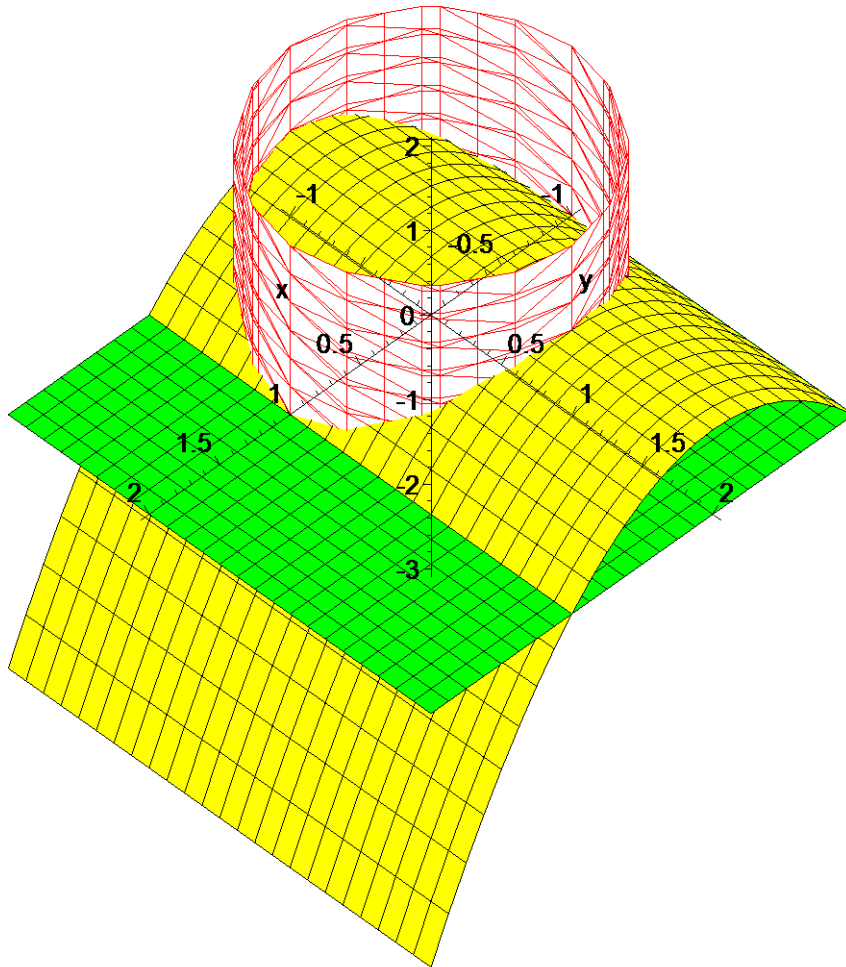
```
Parametr style určuje grafické zobrazení plochy, hodnota hidden skryje mřížku.
```

```
> p1:=plot3d(0,x=-1..2,y=-1..2,color=green):
```

```
> p2:=plot3d(1-x^2,x=-1..2,y=-1..2,color=yellow):
```

```
> display({p,p1,p2},axes=normal);
```





# Parametr *axes* určuje zobrazení souřadnicového systému.

V tomto případě je řídicí křivkou kružnice s parametrickým vyjádřením

```
> par:=x=r*cos(t), y=r*sin(t), t=0..2*Pi;
```

```
par := x = r cos(t), y = r sin(t), t = 0 .. 2 pi
```

Počítáme **plošný obsah válcové plochy**, tedy integrál

```
> S:=Int(h(x,y)-g(x,y), s=gamma..` `);
```

$$S := \int_{\gamma} h(x, y) - g(x, y) ds$$

kde  $h(x,y)$  je "horní" a  $g(x,y)$  "dolní" plocha. V našem případě je  $g(x,y)=0$ ,  $h(x,y)=r-x^2/r$ . Horní plochu musíme převést pomocí zvolené parametrizace. Výsledkem je

```
> z:=r-r^2*cos(t)^2/r=r*sin(t)^2;
```

```
z:=r*sin(t)^2;
```

$$z := r - r \cos(t)^2 = r \sin(t)^2$$

$$z := r \sin(t)^2$$

Dosaďme nyní do integrálu

> `i:=Lineint(z,par);`

$$i := \int_0^{2\pi} r \sin(t)^2 \sqrt{\left(\frac{\partial}{\partial t}(r \sin(t))\right)^2 + \left(\frac{\partial}{\partial t}(r \cos(t))\right)^2} dt$$

Výsledkem je

> `i1:=value(i);`

$$i1 := \int_0^{2\pi} r \sin(t)^2 \sqrt{r^2 \cos(t)^2 + r^2 \sin(t)^2} dt$$

Musíme upravit výraz pod odmocninou a výsledkem je

> `assume(r>0);`

> `simplify(i1,{sin(t)^2+cos(t)^2=1});`

$$\pi r^2$$

# Vymezení konstanty a zjednodušení pomocí vlastního vzorce.

Výpočet pomocí *PathInt*.

> `PathInt(z,[x,y]=Path(<r*cos(t),r*sin(t)>,t=0..2*Pi));`

$$\pi r^2$$

>

>

### Příklad 13

Nalezněte těžiště 1. závitu šroubovice

> `restart:with(student):with(plots):with(VectorCalculus):`

Warning, the name changecoords has been redefined

Warning, the assigned names <,> and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: \*, +, .., Vector, diff, int, limit, series

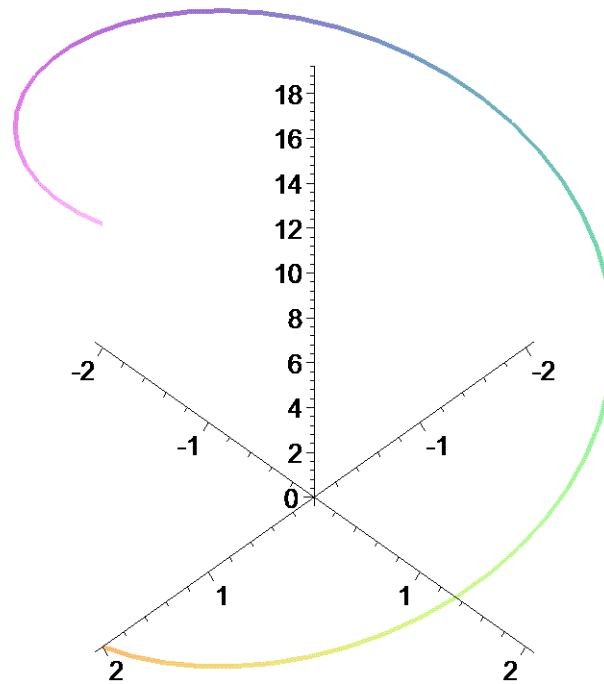
Parametrické rovnice jsou

> `par:=x=r*cos(t),y=r*sin(t),z=v*t,t=0..2*Pi;`

$$par := x = r \cos(t), y = r \sin(t), z = v t, t = 0 .. 2 \pi$$

Vykreslíme tuto křivku, položíme  $v=3$ ,  $r=2$ .

> `spacecurve([2*cos(t),2*sin(t),3*t],t=0..2*Pi,  
axes=normal,thickness=5);`



Potřebné integrály jsou:

- *délka křivky*

> `s[gamma]:=Lineint(1,par);`

$$s_\gamma := \int_0^{2\pi} \sqrt{\left(\frac{\partial}{\partial t}(r \sin(t))\right)^2 + \left(\frac{\partial}{\partial t}(vt)\right)^2 + \left(\frac{\partial}{\partial t}(r \cos(t))\right)^2} dt$$

# Pomocí [...] je zapsán pouze index.

> `value(s[gamma]);`

$$\int_0^{2\pi} \sqrt{r^2 \cos(t)^2 + v^2 + r^2 \sin(t)^2} dt$$

Potřebujeme upravit výraz pod odmocninou

> `sg:=simplify(%,{cos(t)^2+sin(t)^2=1});`

$$sg := 2 \sqrt{v^2 + r^2} \pi$$

# Druhý parametr určuje rovnice, pomocí kterých se výraz zjednoduší.

Výpočet pomocí *PathInt*.

```
> _s:=PathInt(1, [x,y,z] = Path(<r*cos(t), r*sin(t), v*t>,
t=0..2*Pi));
```

$$_s := \int_0^{2\pi} \sqrt{r^2 \cos(t)^2 + v^2 + r^2 \sin(t)^2} dt$$

# Dále lze stejně upravit jako při výpočtu pomocí *Lineint*

- *první statické momenty*

```
> M[x,y]:=Lineint(z,par);
```

$$M_{x,y} := \int_0^{2\pi} v t \sqrt{\left(\frac{\partial}{\partial t}(r \sin(t))\right)^2 + \left(\frac{\partial}{\partial t}(v t)\right)^2 + \left(\frac{\partial}{\partial t}(r \cos(t))\right)^2} dt$$

```
> M[y,z]:=Lineint(x,par);
```

$$M_{y,z} := \int_0^{2\pi} r \cos(t) \sqrt{\left(\frac{\partial}{\partial t}(r \sin(t))\right)^2 + \left(\frac{\partial}{\partial t}(v t)\right)^2 + \left(\frac{\partial}{\partial t}(r \cos(t))\right)^2} dt$$

```
> M[x,z]:=PathInt(y, [x,y,z] = Path(<r*cos(t), r*sin(t), v*t>,
t=0..2*Pi));
```

$$M_{x,z} := 0$$

# Pomocí příkazu *PathInt* dostáváme hned výsledek.

>

Vypočítejme hodnoty uvedených integrálů

```
> Mxy:=value(M[x,y]);
```

```
Myz:=value(M[y,z]);
```

```
Mxz:=value(M[x,z]);
```

$$M_{xy} := 2 \pi^2 v \sqrt{v^2 + r^2}$$

$$M_{yz} := 0$$

$$M_{xz} := 0$$

Z výsledků je patrné, že těžiště leží na souřadnicové ose z, což je očekávaný výsledek. Zbývá určit jeho z-tovou souřadnici.

```
> z[T]:=Mxy/sg;
```

$$z_T := \pi v$$

>

>

>

## 6.2 Křivkový integrál II. druhu

System Maple obsahuje pro výpočet křivkového integrálu II. druhu příkaz **LineInt** z rozšiřující knihovny *VectorCalculus*.

Syntaxe:

**LineInt(F, dom)** , kde  $F$  je vektorové pole určené danou soustavou souřadnic,  $dom$  je jednou z následujících konstrukcí, pomocí kterých definujeme křivku (např. *Path*, *Circle*, *Ellipse*, *LineSegments* ...).

Křivkový integrál II. druhu můžeme pro orientovanou křivku danou parametrickými rovnicemi  $x = x(t), y = y(t), z = z(t), t \in \langle a, b \rangle$  zapsat ve tvaru

$$\int_{\Gamma} P(x, y, z)dx + Q(x, y, z)dy + R(x, y, z)dz = \pm \int_a^b (P(x(t), y(t), z(t))\dot{x}(t) + Q(x(t), y(t), z(t))\dot{y}(t) + R(x(t), y(t), z(t))\dot{z}(t)) dt, \quad (1)$$

kde znaménko +, resp. - platí pro souhlasné, resp. nesouhlasné parametrické vyjádření křivky  $\Gamma$ . Přesnou definici lze nalézt např. v [1].

Tento integrál lze také vyjádřit pomocí skalárního součinu

$$\int_{\Gamma} \vec{F}(X) \cdot d\vec{S} = \int_{\Gamma} P(x, y, z)dx + Q(x, y, z)dy + R(x, y, z)dz, \quad \text{kde}$$

$$\vec{F}(X) = \vec{F}(x, y, z) = \vec{F}(P(x, y, z), Q(x, y, z), R(x, y, z)) \quad \text{a} \quad d\vec{S} = (dx, dy, dz).$$

### Příklad 1

Vypočítejte křivkový integrál druhého druhu  $\int_{\Gamma} (x - y)dx + (x + y)dy$  podél křivky  $\Gamma = AB$ .

> **restart:**

Integrál spočteme třemi způsoby.

Zadáme úsečku

> **b := [[2, 3], [3, 5]];**

$b := [[2, 3], [3, 5]]$

Úsečku  $AB$  můžeme parametrizovat například takto

> **\_par := x=2+t, y=3+2\*t, t=0..1;**

**par := 2+t, 3+2\*t, t=0..1;**

$_par := x = 2 + t, y = 3 + 2 t, t = 0 .. 1$

$par := 2 + t, 3 + 2 t, t = 0 .. 1$

# Druhý příkaz není zadán rovnicemi, protože parametrizace v tomto tvaru je lépe použitelná v následujících příkazech.

Dále určíme, zda je křivka  $\Gamma$  orientována souhlasně s parametrickým vyjádřením, tj. jde z bodu  $A$  do bodu  $B$ . Dosadíme meze parametru  $t$  do rovnic parametrizace.

> **subs (t=0, {\_par [1], \_par [2] });**

$\{x = 2, y = 3\}$

# Jde o bod A. Použili jsme zadání parametrických rovnic s rovnicemi pro větší přehlednost

> **subs (t=1, {\_par [1], \_par [2] });**

$\{x = 3, y = 5\}$

# Jde o bod B

Můžeme konstatovat, že křivka je orientována souhlasně s danou parametrizací. Ještě poznamenejme, že u všech dalších příkladů bude křivka souhlasně orientována s daným parametrickým vyjádřením. Pokud tomu bude jinak, bude na to upozorněno. V dalších příkladech už nebudeme orientaci ověřovat.

Nakreslíme nyní obrázek.

Nejprve pojmenujeme vektorové pole

```
> vpole := [x-y, x+y];
```

```
vpole := [x - y, x + y]
```

```
Integrand jako seznam
```

```
> vplot := plots[fieldplot](vpole, x=-1..6, y=-1..6, grid=[10,10]):
```

```
Nakreslení vektorového pole pomocí funkce z rozšiřující knihovny plots.
```

```
> par_p := plot([par], -1..6, -1..6, thickness=5):
```

```
Vykreslení křivky. Užili jsme druhý zápis parametrických rovnic ve tvaru seznamu, protože přesně odpovídá požadavkům příkazu plot na vykreslení křivky dané parametrickými rovnicemi.
```

```
> p1 := plots[arrow]([2,3], [3,5], color=blue, difference=true):
```

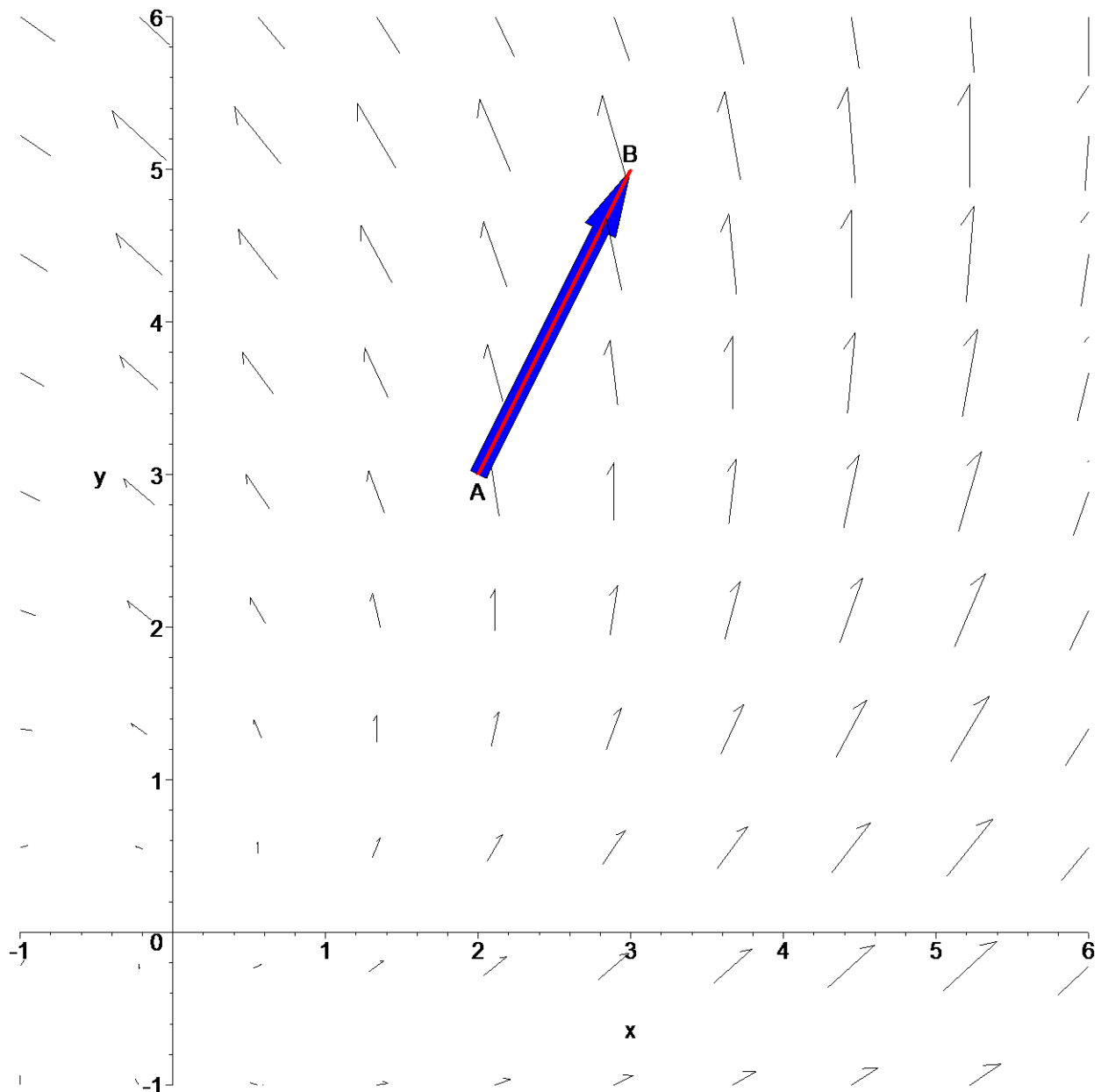
```
Nakreslení orientace křivky
```

```
> t1 := plots[textplot]([2, 2.95, `A`], align=BELOW):
```

```
> t2 := plots[textplot]([3, 5.05, `B`], align=ABOVE):
```

```
Zápis textu do obrázku
```

```
> plots[display]({vplot, par_p, t1, t2, p1});
```



# Vykreslení do jednoho obrázku

Dosadíme parametrické rovnice do integrandu

```
> parpole := [subs(_par[1], _par[2], vpole[1]), subs(_par[1], _par[2], vpole[2])];
```

```
parpole := [-1 - t, 5 + 3 t]
```

# Příkaz *subs* provádí vícenásobnou substituci, substituční rovnice musí být zadány jako posloupnost, poslední parametr určuje místo použití.

Spočteme derivace

```
> dpar := diff([par[1], par[2]], t);
```

```
dpar := [1, 2]
```

Výpočet provedeme

- dosazením do vztahu (1) dostaneme

```
> Int(parpole[1]*dpar[1]+parpole[2]*dpar[2], t=0..1);
```

$$\int_0^1 9 + 5 t dt$$

> **value (%) ;**

$$\frac{23}{2}$$

- **pomocí skalárního součinu**

Tento křivkový integrál II. druhu můžeme po dosazení parametrických rovnic zapsat pomocí skalárního součinu takto

> **Int ( (parpole.Diff ([par[1],par[2]] , t)) , par[3]) ;**

$$\int_0^1 [-1-t, 5+3t] \cdot \left( \frac{d}{dt} [2+t, 3+2t] \right) dt$$

Počítáme tedy integrál

> **Int (parpole.dpar , par[3]) ;**

$$\int_0^1 [-1-t, 5+3t] \cdot [1, 2] dt$$

Jde o skalární součin dvou vektorů. Použijeme tedy příkaz **dotprod** z rozšiřující knihovny *linalg*, který počítá skalární součin dvou vektorů.

> **arg:=linalg[dotprod] (parpole , dpar) ;**

$$arg := 9 + 5 t$$

# Argument integrálu

Počítáme integrál a výsledek je

> **Int (arg , par[3]) =int (arg , par[3]) ;**

$$\int_0^1 9 + 5 t dt = \frac{23}{2}$$

- **s využitím příkazů knihovny *VectorCalculus***

Knihovna *VectorCalculus* obsahuje pro výpočet skalárního součinu příkaz **DotProduct**, který však vyžaduje zadání vektorového pole, tj. pomocí  $\langle \cdot \rangle$ .

> **with (VectorCalculus) :**

Warning, the assigned names  $\langle , \rangle$  and  $\langle | \rangle$  now have a global binding

Warning, these protected names have been redefined and unprotected: \*, +, .., Vector, diff, int, limit, series

> **p\_pole := <parpole[1], parpole[2]>;**

$$p\_pole := (-1-t) \mathbf{e}_x + (5+3t) \mathbf{e}_y$$

> **d\_par := <dpar[1], dpar[2]>;**

$$d\_par := \mathbf{e}_x + 2 \mathbf{e}_y$$

> **DotProduct (p\_pole , d\_par) ;**

$$9 + 5 t$$

Tento způsob zápisu je nepříjemný, protože musí být nastavena správná soustava souřadnic. V našem případě kartézská, což zjistíme pomocí příkazu (více viz. příloha *Knihovna VectorCalculus*)

> **GetCoordinates () ;**

*cartesian*



Nyní přejdeme k výpočtu pomocí příkazu *LineInt*. Nejprve musíme pracovat ve správné soustavě souřadnic, proto použijeme příkaz *SetCoordinate*, pomocí kterého danou soustavu nastavíme. V našem případě požaduje kartézskou soustavu proměnných  $x$  a  $y$ .

```
> SetCoordinates (cartesian[x,y]);
```

*cartesian* <sub>$x,y$</sub>

# Připomeňme, že systém Maple má několik předdefinovaných soustav souřadnic. Více viz. příloha Knihovna *VectorCalculus*, popř. nápověda.

Vytvoříme vektorové pole, které získáme ze zadaného integrálu

```
> vp:=VectorField(<x-y,x+y>);
```

$$vp := (x - y) \mathbf{e}_x + (x + y) \mathbf{e}_y$$

Nyní užijeme příkaz *LineInt* spolu s konstrukcí *Line*, která vytvoří úsečku pomocí dvou zadaných bodů.

```
> LineInt(vp,Line(<2,3>,<3,5>));
```

$$\frac{23}{2}$$

V případě nesouhlasné orientace křivky s danou parametrizací dostáváme, v souladu s teorií, výsledek s opačným znaménkem.

```
> LineInt(vp,Line(<3,5>,<2,3>));
```

$$\frac{-23}{2}$$

```
>
```

```
>
```

### Příklad 2

Spočítejte zadaný křivkový integrál podél kladně orientovaného obvodu trojúhelníku určeného souřadnicovými osami a přímkou

```
> restart;
```

```
> Int([0,x]*`.`,S=gamma..`);
```

$$\int_{\gamma} [0, x] \cdot dS$$

```
> with(VectorCalculus):with(plots):
```

Warning, the assigned names <, > and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: \*, +, .., Vector, diff, int, limit, series

Warning, the name changecoords has been redefined

```
> vpole:=[0,x];
```

$$vpole := [0, x]$$

# Pojmenujeme integrand. První prvek je nulový, protože zde nemáme integraci přes  $dx$ .

Rovnice zadané přímkou je

```
> pr:=x/2+y/3=1;
```

$$pr := \frac{x}{2} + \frac{y}{3} = 1$$

Nakresleme obrázek

```
> vplot := fieldplot(vpole,x=-1..3,y=-1..3, grid=[10,10]):
```

```
Nakreslení vektorového pole pomocí příkazu z rozšiřující knihovny plots.
```

```
Průsečíky přímky se souřadnicovými osami jsou
```

```
> b:=[[2,0],[0,3]];
```

```
b := [[2, 0], [0, 3]]
```

```
Nakreslíme obrázek
```

```
> par_p:=polygonplot([[0,0],op(1..2,b)],thickness=3):
```

```
Vykreslení trojúhelníka pomocí příkazu polygonplot. Druhý parametr thickness určuje šířku pera.
```

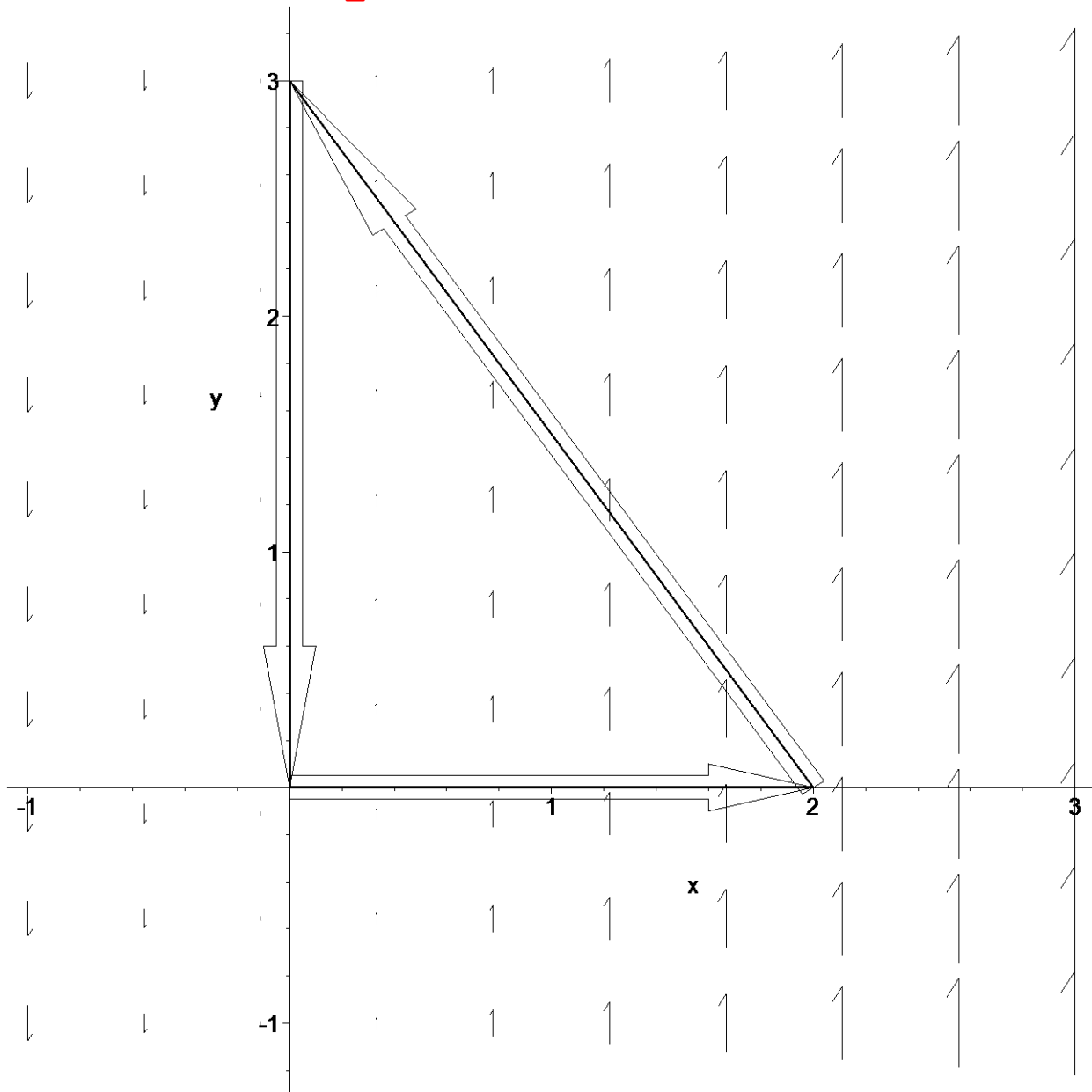
```
> p1:=arrow([0,0],[2,0],difference=true,width=0.1):
```

```
 p2:=arrow([2,0],[0,3],difference=true,width=0.1):
```

```
 p3:=arrow([0,3],[0,0],difference=true,width=0.1):
```

```
Nakreslení orientace křivky
```

```
> display({vplot, par_p,p1,p2,p3});
```



```
Pro vykreslení do jednoho obrázku
```

Jde o křivku po částech hladkou, což se při výpočtu projeví v rozdělení křivkového integrálu na tři dílčí integrály podél jednotlivých stran trojúhelníku. V našem případě ale použijeme k výpočtu

příkaz *LineInt*.

```
> LineInt(VectorField(<0,x>), LineSegments(<0,0>, <2,0>, <0,3>, <0,0>))
);
```

Error, (in VectorCalculus:-VectorField) no coordinate names were given

Jak již bylo řečeno v prvním příkladu, musíme nejprve určit výchozí soustavu souřadnic.

```
> SetCoordinates(cartesian[x,y]);
```

*cartesian*<sub>x,y</sub>

# Tento příkaz nastaví souřadnicovou soustavu na kartézskou se souřadnicovými osami  $x, y$ .

Jde o kartézskou soustavu se souřadnicovými osami  $x$  a  $y$ . Nyní spočteme požadovaný integrál

```
> LineInt(VectorField(<0,x>), LineSegments(<0,0>, <2,0>, <0,3>, <0,0>))
);
```

3

Jinou možností je určení soustavy souřadnicové již při zadávání vektorového pole pomocí příkazu *VectorField* následujícího tvaru

```
> VectorField(<0,x>, 'cartesian'[x,y]);
```

*cartesian*<sub>x,y</sub>

Pak má systém Maple informaci o použité soustavě souřadnic a výpočet provede. Tento druhý postup se zdá výhodnější při výpočtu několika integrálů v jednom souboru, protože každému vektorovému poli můžeme přiřadit jinou soustavu souřadnic a nebudou se nijak ovlivňovat.

>

>

### Příklad 3

Spočtete následující integrál podél prvního oblouku cykloidy

```
> restart:with(plots):
```

Warning, the name changecoords has been redefined

```
> Int([2*r-y, -(r-y)]*` . ` , S=Gamma . . ` `);
```

$$\int_{\Gamma} [2r - y, -r + y] \cdot dS$$

Parametrické rovnice prvního oblouku cykloidy jsou

```
> _par:=x=r*(t-sin(t)), y=r*(1-cos(t)), t=0..2*Pi;
```

```
par:=r*(t-sin(t)), r*(1-cos(t)), t=0..2*Pi;
```

$$\_par := x = r(t - \sin(t)), y = r(1 - \cos(t)), t = 0 .. 2\pi$$

$$par := r(t - \sin(t)), r(1 - \cos(t)), t = 0 .. 2\pi$$

Nakresleme obrázek

```
> vpole:= [2*r-y, -r+y];
```

$$vpole := [2r - y, -r + y]$$

```
> vplot := fieldplot(subs(r=1, vpole), x=-1..8, y=-1..3,
grid=[10,10]):
```

# Pro vykreslení obrázku je nutné pevně zvolené  $r$ , v našem případě jsme do obecného předpisu dosadili pomocí příkazu *subs* jedničku.

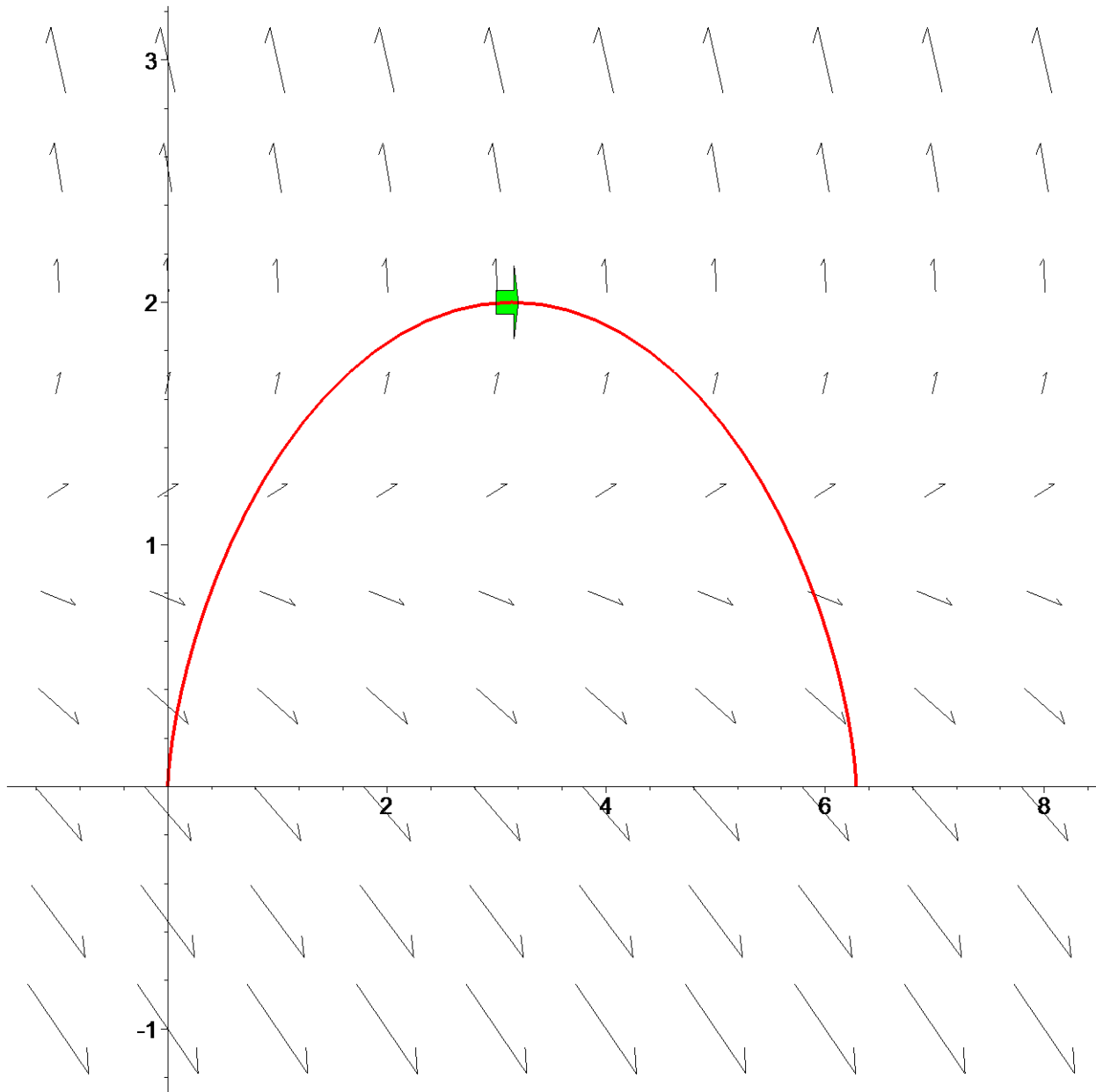
```
> p:=plot(subs(r=1, [par]), thickness=4):
```

# Graf prvního oblouku cykloidy

```

> p1:=arrow([3,2],[3.2,2],difference=true,color=green,width=0.1,height_width=.3):
Nakreslení orientace křivky
> display({vplot,p,p1});

```



```
Zobrazení do jednoho obrázku
```

- **výpočet pomocí skalárního součinu**

Dosazením parametrických rovnic do integrandů dostáváme

```

> parpole:=[subs(_par[1],_par[2],vpole[1]),subs(_par[1],_par[2],vpole[2])];

```

$$\text{parpole} := [2r - r(1 - \cos(t)), -r + r(1 - \cos(t))]$$

# Příkaz *subs* provádí vícenásobnou substituci, tj. substituční rovnice musí být zadány jako posloupnost, poslední parametr určuje místo použití.

Přesný zápis integrálu se skalárním součinem tedy je

```

> parpole:=simplify(parpole);

```

$$\text{parpole} := [r(1 + \cos(t)), -r \cos(t)]$$

# Užili jsme funkci *simplify* pro zjednodušení výrazů v identifikátoru *parpole*.

> `Int(simplify(parpole.Diff([par[1],par[2]],t)), par[3]);`

$$\int_0^{2\pi} [r + r \cos(t), -r \cos(t)] \cdot \left( \frac{d}{dt} [-r(-t + \sin(t)), r - r \cos(t)] \right) dt$$

Nyní provedeme naznačenou derivaci

> `dpar:=diff([par[1],par[2]],t);`

$$dpar := [r(1 - \cos(t)), r \sin(t)]$$

# Provedli jsme derivaci prvních dvou prvků seznamu *par*

Počítáme tedy integrál

> `Int(parpole.dpar, par[3]);`

$$\int_0^{2\pi} [r(1 + \cos(t)), -r \cos(t)] \cdot [r(1 - \cos(t)), r \sin(t)] dt$$

Jde o skalární součin dvou vektorů. Použijeme příkaz *dotprod* z rozšiřující knihovny *linalg*.

> `#assume(r>0);`

`arg:=linalg[dotprod](parpole,dpar,'orthogonal');`

$$arg := r^2(1 - \cos(t))(1 + \cos(t)) - r^2 \sin(t) \cos(t)$$

# Argument integrálu

Počítáme tedy integrál a výsledkem je

> `Int(arg,par[3])=int(arg,par[3]);`

$$\int_0^{2\pi} r^2(1 - \cos(t))(1 + \cos(t)) - r^2 \sin(t) \cos(t) dt = r^2 \pi$$

#### • výpočet pomocí knihovny *VectorCalculus*

> `with(VectorCalculus):`

Warning, the assigned names <, > and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: \*, +, .., Vector, diff, int, limit, series

> `vp:=VectorField(<vpole[1],vpole[2]>,cartesian[x,y]);`

$$vp := (2r - y) \bar{e}_x + (-r + y) \bar{e}_y$$

# Vytvoření vektorového pole

> `LineInt(vp,Path(<r*(t-sin(t)),r*(1-cos(t))>,t=0..2*Pi));`

$$r^2 \pi$$

>

>

#### Příklad 4

Spočítejte křivkový integrál podél sjednocení dvou křivek

> `restart:with(plots):`

Warning, the name changecoords has been redefined

> `Int([y,z,x]*`..`,S=Gamma..`);`

$$\int_{\Gamma} [y, z, x] \cdot dS$$

```
> vpole:= $[y, z, x]$;
```

```
vpole := $[y, z, x]$
```

```
Vektorové pole
```

```
Parametrické rovnice šroubovice
```

```
> par:= $r*\cos(t), r*\sin(t), v*t, t=0..2*Pi$;
```

```
_par:= $x=r*\cos(t), y=r*\sin(t), z=v*t, t=0..2*Pi$;
```

```
par := $r \cos(t), r \sin(t), v t, t = 0 .. 2 \pi$
```

```
_par := $x = r \cos(t), y = r \sin(t), z = v t, t = 0 .. 2 \pi$
```

```
Druhou křivkou je úsečka zadaná body
```

```
> body:= $[[r, 0, 0], [r, 0, 2*Pi*v]]$;
```

```
body := $[[r, 0, 0], [r, 0, 2 \pi v]]$
```

```
Nakreslíme obrázek pro pevně zvolené parametry, $r=1, v=r$.
```

```
> p:=fieldplot3d(vpole, x=-1..7, y=-1..7, z=-1..7, grid=[6,6,6], thickness=2);
```

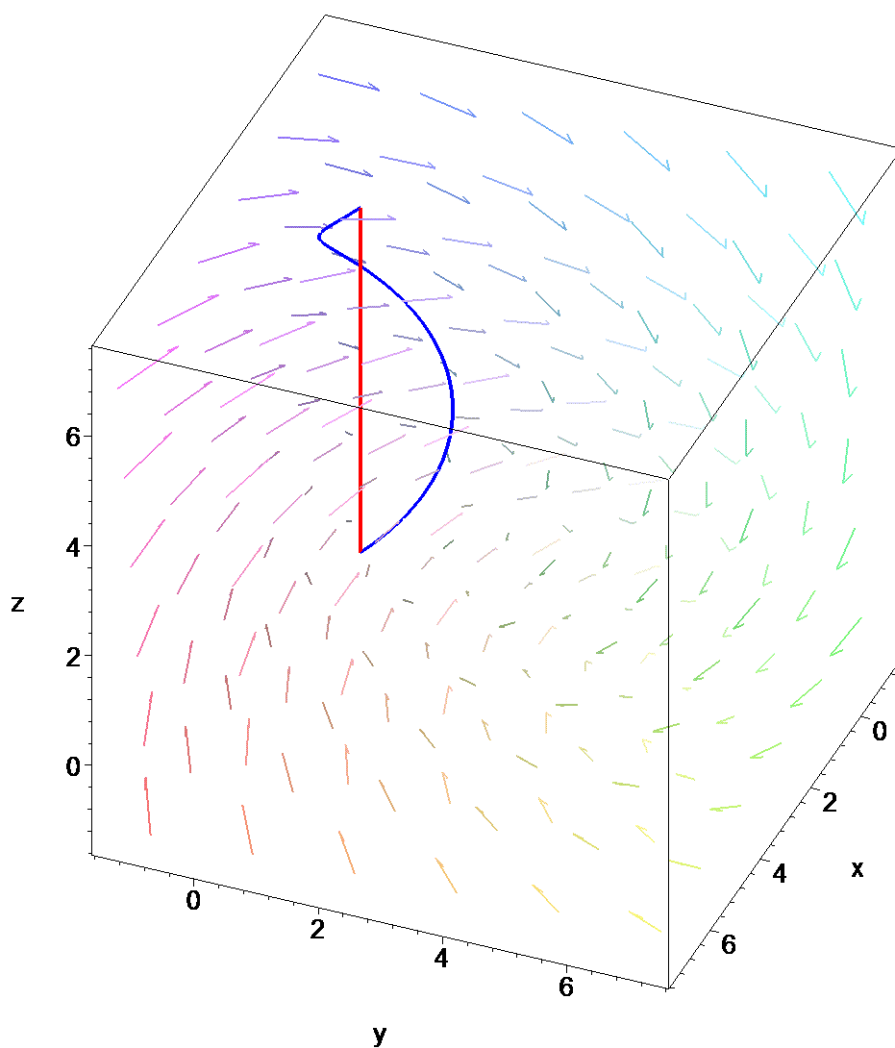
```
Vykreslení vektorového pole ve 3D.
```

```
> p1:=spacecurve(subs(r=1, v=1, [par]), thickness=4, color=blue);
```

```
Křivka v prostoru, proto příkaz spacecurve.
```

```
> p2:=spacecurve(subs(r=1, v=1, body), thickness=4, color=red);
```

```
> display({p, p1, p2}, axes=boxed, orientation=[22, 55]);
```



- výpočet pomocí skalárního součinu

Nejprve spočteme křivkový integrál podél šroubovice. Dosazením parametrických rovnic do integrandů dostáváme

```
> parpole := [subs(_par[1], _par[2], _par[3], vpole[1]), subs(_par[1], _par[2], _par[3], vpole[2]),
subs(_par[1], _par[2], _par[3], vpole[3])];
parpole := [r sin(t), v t, r cos(t)]
```

# Příkaz *subs* provádí vícenásobnou substituci, tj. substituční rovnice musí být zadány jako posloupnost, poslední parametr určuje místo použití.

Přesný zápis integrálu se skalárním součinem tedy je

```
> Int(parpole.Diff([par[1], par[2], par[3]], t), par[4]);
```

$$\int_0^{2\pi} [r \sin(t), v t, r \cos(t)] \cdot \left( \frac{d}{dt} [r \cos(t), r \sin(t), v t] \right) dt$$

Nyní provedeme naznačenou derivaci

```
> dpar:=diff([par[1],par[2],par[3]],t);
```

$$dpar := [-r \sin(t), r \cos(t), v]$$

Počítáme tedy integrál

```
> Int(parpole.dpar, par[4]);
```

$$\int_0^{2\pi} [r \sin(t), v t, r \cos(t)] \cdot [-r \sin(t), r \cos(t), v] dt$$

Pro výpočet skalárního součinu použijeme příkazu **dotprod** z knihovny *linalg*.

```
> arg:=linalg[dotprod](parpole,dpar,'orthogonal');
```

$$arg := -r^2 \sin(t)^2 + r \cos(t) v t + v r \cos(t)$$

# Integrand

Počítáme tedy integrál a výsledkem je

```
> i:=Int(arg,par[4])=int(arg,par[4]);
```

$$i := \int_0^{2\pi} -r^2 \sin(t)^2 + r \cos(t) v t + v r \cos(t) dt = -r^2 \pi$$

Dále spočteme integrál podél úsečky

Jednou z možných parametrizací úsečky je

```
> _par1:=[x=r,y=0,z=2*Pi*v-2*Pi*v*t,t=0..1];
```

```
par1:=[r,0,2*Pi*v-2*Pi*v*t,t=0..1];
```

$$_par1 := [x = r, y = 0, z = 2 \pi v - 2 \pi v t, t = 0 .. 1]$$

$$par1 := [r, 0, 2 \pi v - 2 \pi v t, t = 0 .. 1]$$

Dosazením parametrických rovnic do integrandů dostáváme

```
> parpole1:=[subs(op(1..3,_par1),vpole[1]),subs(op(1..3,_par1),vpole[2]),
subs(op(1..3,_par1),vpole[3])];
```

$$parpole1 := [0, 2 \pi v - 2 \pi v t, r]$$

# Příkaz *subs* provádí vícenásobnou substituci, tj. substituční rovnice musí být zadány jako posloupnost, poslední parametr určuje místo použití.

Přesný zápis integrálu se skalárním součinem tedy je

```
> Int(parpole1.Diff([op(1..3,par1)],t), par1[4]);
```

$$\int_0^1 [0, 2 \pi v - 2 \pi v t, r] \cdot \left( \frac{d}{dt} [r, 0, 2 \pi v - 2 \pi v t] \right) dt$$

Nyní provedeme naznačenou derivaci

```
> dpar1:=[op(1..3,diff(par1,t))];
```

$$dpar1 := [0, 0, -2 \pi v]$$

Počítáme tedy integrál

```
> Int(parpole1.dpar1, par1[4]);
```

$$\int_0^1 [0, 2 \pi v - 2 \pi v t, r] \cdot [0, 0, -2 \pi v] dt$$

```
> arg1:=linalg[dotprod](parpole1,dpar1,'orthogonal');
```

$$arg1 := -2 \pi v r$$

# Výpočet integrandu



Počítáme tedy integrál a výsledkem je

```
> i1:=Int(arg1,par1[4])=int(arg1,par1[4]);
```

$$i1 := \int_0^1 -2 \pi v r dt = -2 \pi v r$$

Výsledný integrál je roven součtu jednotlivých integrálů

```
> rhs(i+i1);
```

$$-r^2 \pi - 2 \pi v r$$

- výpočet pomocí příkazu *LineInt*

```
> with(VectorCalculus):
```

Warning, the assigned names <, > and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: \*, +, .., Vector, diff, int, limit, series

Nyní spočítáme jednotlivé integrály a výsledkem bude jejich součet.

```
> LineInt(VectorField(<y,z,x>,cartesian[x,y,z]),
Path(<r*cos(t),r*sin(t),v*t>,t=0..2*Pi));
```

$$-r^2 \pi$$

# Po šroubovici.

```
> LineInt(VectorField(<y,z,x>,cartesian[x,y,z]),Line(<r,0,2*Pi*v>,
<r,0,0>));
```

$$-2 \pi v r$$

# Po úsečce. Pozor jdeme z koncového bodu do počátečního!

Výsledkem je součet těchto integrálů

```
> %+%%;
```

$$-r^2 \pi - 2 \pi v r$$

```
>
```

### Příklad 5

Následující příklad postupně spočítáme čtyřmi postupy: pomocí skalárního součinu, pomocí *VectorCalculus*, užitím *Greenova* vzorce, pomocí potenciálu.

```
> restart:with(plots):
```

Warning, the name changecoords has been redefined

```
> Int([(x+y)^2,-(x-y)^2]*`.\`,`S=Gamma..\`);
```

$$\int_{\Gamma} [(x+y)^2, -(x-y)^2] \cdot dS$$

Křivka je dána sjednocením části paraboly a úsečky zadané body  $[0,0],[1,1]$ .

```
> _par:=[x=t,y=t^2,t=0..1];
```

```
par:=[t,t^2,t=0..1];
```

$$\_par := [x = t, y = t^2, t = 0 .. 1]$$

$$par := [t, t^2, t = 0 .. 1]$$

# Parametrizace paraboly.

```
> _par1:=[x=1-t,y=1-t,t=0..1];
```

```
par1:=[1-t,1-t,t=0..1];
```

```
_par1 := [x = 1 - t, y = 1 - t, t = 0 .. 1]
```

```
par1 := [1 - t, 1 - t, t = 0 .. 1]
```

```
Parametrické rovnice úsečky.
```

```
> vpole := [(x+y)^2, -(x-y)^2];
```

```
vpole := [(x + y)2, -(x - y)2]
```

```
Zadání pole
```

```
Nakreslení obrázku
```

```
> p:=plot(t^2,t=-1..1,color=WHITE):
```

```
_p:=plot(t^2,t=0..1,color=blue,thickness=3):
```

```
Graf části paraboly. První příkaz je v tomto tvaru, protože jinak by obrázek zahrnoval pouze interval <0,1> na ose x, což není vhodné kvůli lepší představě tvaru vektorového pole. Proto také byla zvolena křivka bílé barvy, protože splyne s pozadím..
```

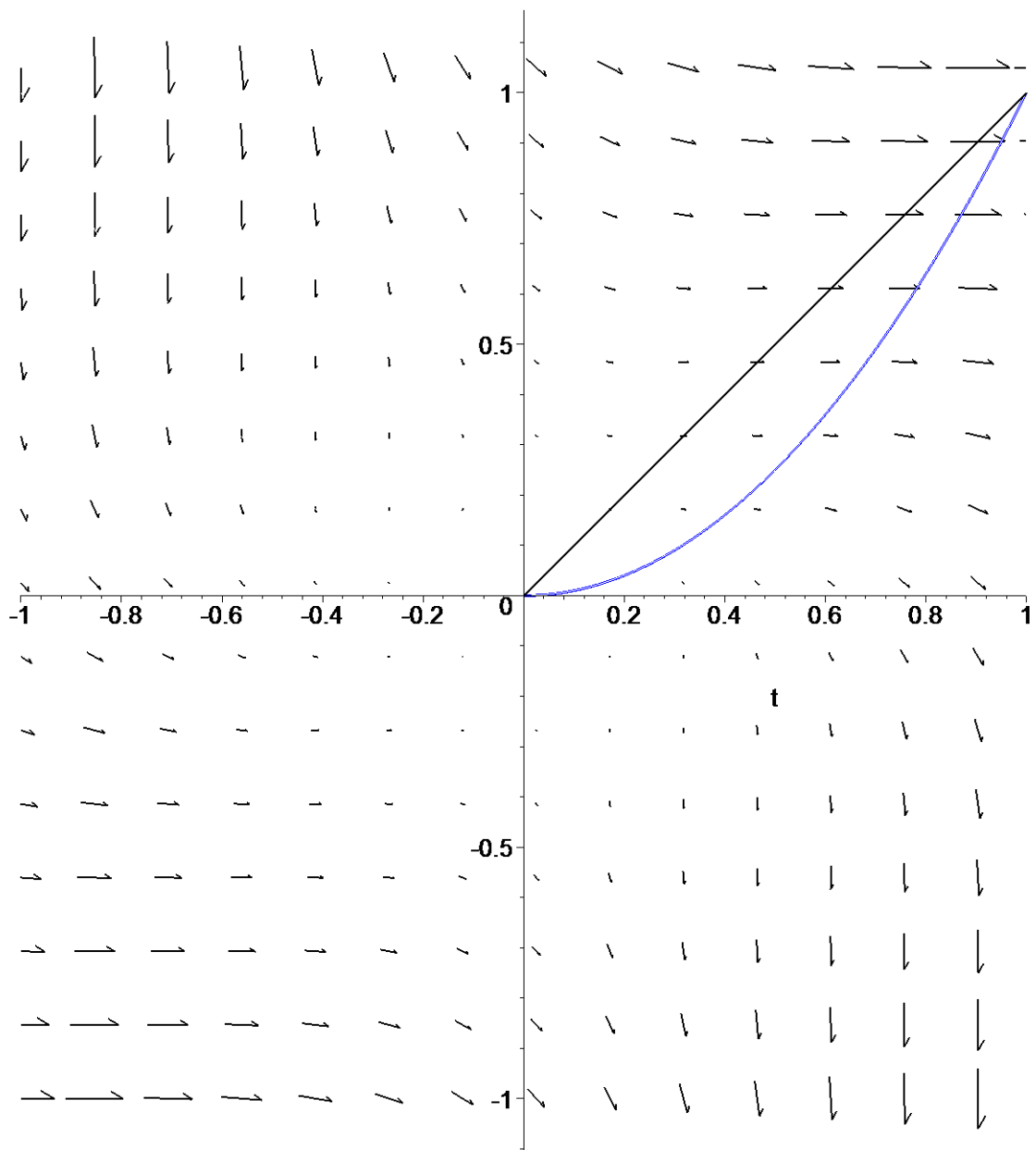
```
> p1:=polygonplot([[0,0],[1,1]],thickness=3):
```

```
Vykreslení úsečky zadáním počátečního a koncového bodu.
```

```
> p2:=fieldplot(vpole,x=-1..1.05,y=-1..1.05,grid=[15,15],thickness=2):
```

```
Obrázek vektorového pole
```

```
> display({p,_p,p1,p2},scaling=constrained);
```



- Výpočet pomocí skalárního součinu

Dosazením parametrických rovnic do integrandů obdržíme

```
> parpole := [subs (op (1..2, _par), vpole[1]), subs (op (1..2, _par), vpole[2])];
parpole1 := [subs (op (1..2, _par1), vpole[1]), subs (op (1..2, _par1), vpole[2])];
```

$$\text{parpole} := [(t + t^2)^2, -(t - t^2)^2]$$

$$\text{parpole1} := [(2 - 2t)^2, 0]$$

# Použili jsme příkaz *op*, jehož první parametr určuje počet prvků seznamu (popř. množiny) zadané ve druhém parametru, který bude vypsán jako posloupnost. Příkaz *subs* provádí vícenásobnou substituci, tj. substituční rovnice musí být zadány jako posloupnost, poslední parametr určuje místo použití.

Přesný zápis integrálů se skalárním součinem je

```
> Int (parpole.Diff ([par[1], par[2]], t), par[3]);
```

$$\int_0^1 [(t+t^2)^2, -(t-t^2)^2] \cdot \left(\frac{d}{dt}[t, t^2]\right) dt$$

> `Int(parpole1.Diff([par1[1], par1[2]], t), par1[3]);`

$$\int_0^1 [(2-2t)^2, 0] \cdot \left(\frac{d}{dt}[1-t, 1-t]\right) dt$$

Nyní provedeme naznačenou derivaci

> `dpar:=op(1..2,diff(par,t));`  
`dpar1:=op(1..2,diff(par1,t));`

$$dpar := [1, 2t]$$

$$dpar1 := [-1, -1]$$

Počítáme tedy integrály

> `Int(parpole.dpar, par[3]);`

$$\int_0^1 [(t+t^2)^2, -(t-t^2)^2] \cdot [1, 2t] dt$$

> `Int(parpole1.dpar1, par[3]);`

$$\int_0^1 [(2-2t)^2, 0] \cdot [-1, -1] dt$$

Jde o skalární součin dvou vektorů, proto použijeme příkaz *dotprod* z rozšiřující knihovny *linalg*.

> `arg:=linalg[dotprod](parpole,dpar,'orthogonal');`

$$arg := (t+t^2)^2 - 2t(t-t^2)^2$$

> `arg1:=linalg[dotprod](parpole1,dpar1);`

$$arg1 := -4(-1+t)^2$$

# Integrand

Počítáme tedy integrály

> `Int(arg,par[3])=int(arg,par[3]);`  
`Int(arg1,par1[3])=int(arg1,par1[3]);`

$$\int_0^1 (t+t^2)^2 - 2t(t-t^2)^2 dt = 1$$

$$\int_0^1 -4(-1+t)^2 dt = \frac{-4}{3}$$

Výsledkem je jejich součet

> `rhs(%)+rhs(%%);`

$$\frac{-1}{3}$$

### • Užitím knihovny *VectorCalculus*

> `with(VectorCalculus):`

Warning, the assigned names <, > and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: \*, +, ., Vector, diff, int, limit, series

Vytvoříme vektorové pole v kartézské soustavě (z integrandů)

```
> vf:=VectorField(<(x+y)^2,-(x-y)^2>,cartesian[x,y]);
```

$$vf := (x+y)^2 \bar{e}_x - (x-y)^2 \bar{e}_y$$

```
> i1:=LineInt(vf,Path(<t,t^2>,t=0..1));
```

$$i1 := 1$$

# Výpočet po parabolu

```
> i2:=LineInt(vf,Line(<1,1>,<0,0>));
```

$$i2 := \frac{-4}{3}$$

# POZOR - křivka je orientována z koncového bodu do počátečního.

Výsledkem je součet jednotlivých integrálů

```
> i1+i2;
```

$$\frac{-1}{3}$$

### • Užitím Greenova vzorce

Nejprve uveďme Greenův vzorec  $\iint_{\Omega} \left( \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) dx dy = \int_{\partial\Omega} P dx + Q dy$ . Předpokladem je normální

oblast s po částech hladkou hranicí, dále spojitost  $P(x,y)$ ,  $Q(x,y)$  a jejich derivací, a kladná orientace hranice oblasti vzhledem k oblasti. Přesnou definici Greenovy věty lze nalézt např. v [1].

Z obrázku je patrné, že oblast ohraničená oběma křivkami je konvexní, tzn. je i normální.

Orientace hranice vzhledem k oblasti je kladná (viz. obrázek).

```
> with(student):
```

```
> P:=(x+y)^2;
```

```
Q:=- (x-y)^2;
```

$$P := (x+y)^2$$

$$Q := -(x-y)^2$$

# Určení funkcí  $P$ ,  $Q$  z původního integrálu

Provedeme naznačené derivace

```
> dQ:=Diff(Q,x)=diff(Q,x);
```

```
dP:=Diff(P,y)=diff(P,y);
```

$$dQ := \frac{\partial}{\partial x} (-(x-y)^2) = -2x + 2y$$

$$dP := \frac{\partial}{\partial y} ((x+y)^2) = 2x + 2y$$

Dosadíme do Greenova vzorce

```
> Doubleint(rhs(dQ)-rhs(dP),x,y,Omega);
```

$$\iint_{\Omega} -4x \, dx \, dy$$

# Pomocí příkazu *rhs* získáme pravou stranu rovnice.

Oblast  $\Omega$  je omezena zdola parabolou a shora úsečkou. Podle obrázku lze snadno určit meze pro dvojný integrál.

```
> Doubleint (rhs (dQ) - rhs (dP) , y=x^2 .. x, x=0 .. 1) ;
```

$$\int_0^1 \int_{x^2}^x -4x \, dy \, dx$$

Spočteme hodnotu integrálu

```
> value (%) ;
```

$$\frac{-1}{3}$$

### • Potenciál (hledání kmenové funkce)

V kapitole o knihovně *VectorCalculus* byl uveden příkaz pro výpočet potenciálu. Jde o příkaz *ScalarPotential*. Na tomto příkladu si ukážeme jeho použití. Ještě připomeňme, že výsledný integrál je roven rozdílu funkční hodnoty potenciálu v koncovém bodu a funkční hodnoty potenciálu v počátečním bodu. V tomto případě také mluvíme o nezávislosti křivkového integrálu na integrační cestě.

```
> restart:with(VectorCalculus) ;
```

Warning, the assigned names <, > and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: \*, +, .., Vector, diff, int, limit, series

```
> v:=VectorField(<(x+y)^2, -(x-y)^2>, cartesian[x,y]) ;
```

$$v := (x+y)^2 \bar{e}_x - (x-y)^2 \bar{e}_y$$

# Vektorové pole získané z integrandů

Výpočet potenciálu

```
> ScalarPotential(v) ;
```

Je vidět, že potenciál neexistuje, což je v souladu s tím, že v tomto případě není v oblasti splněna podmínka

```
> Diff(P(x,y), y) = Diff(Q(x,y), x) ;
```

$$\frac{\partial}{\partial y} P(x, y) = \frac{\partial}{\partial x} Q(x, y)$$

která je ekvivalentní s výrokem o nezávislosti křivkového integrálu na integrační cestě, viz.[1, V.7.6].

Pro kontrolu naznačené derivace spočteme

```
> diff((x+y)^2, y) = diff(-(x-y)^2, x) ;
```

$$2x + 2y = -2x + 2y$$

```
>
```

```
>
```

### Příklad 6

Tento příklad slouží k výpočtu křivkového integrálu s využitím potenciálu.

Vypočtěte křivkový integrál podél části paraboly dané body  $[0,0]$ ,  $[1,1]$

```
> restart:
```

```
with(plots):with(VectorCalculus):
```

Warning, the name changecoords has been redefined

Warning, the assigned names <, > and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: \*, +, .., Vector, diff, int, limit, series

```
> vpole := [4*(x^2+y^2)*x, 4*(x^2+y^2)*y];
vf := VectorField(<4*(x^2+y^2)*x, 4*(x^2+y^2)*y>, 'cartesian' [x, y]);
```

$$vpole := [(4x^2 + 4y^2)x, (4x^2 + 4y^2)y]$$
$$vf := (4x^2 + 4y^2)x \bar{e}_x + (4x^2 + 4y^2)y \bar{e}_y$$

# Získání vektorového pole

Parametrizace paraboly (křivka je s ní souhlasně orientována)

```
> _par := [x=t, y=t^2, t=0..1];
par := [t, t^2, t=0..1];
```

$$_par := [x = t, y = t^2, t = 0 .. 1]$$
$$par := [t, t^2, t = 0 .. 1]$$

Nakreslení obrázku

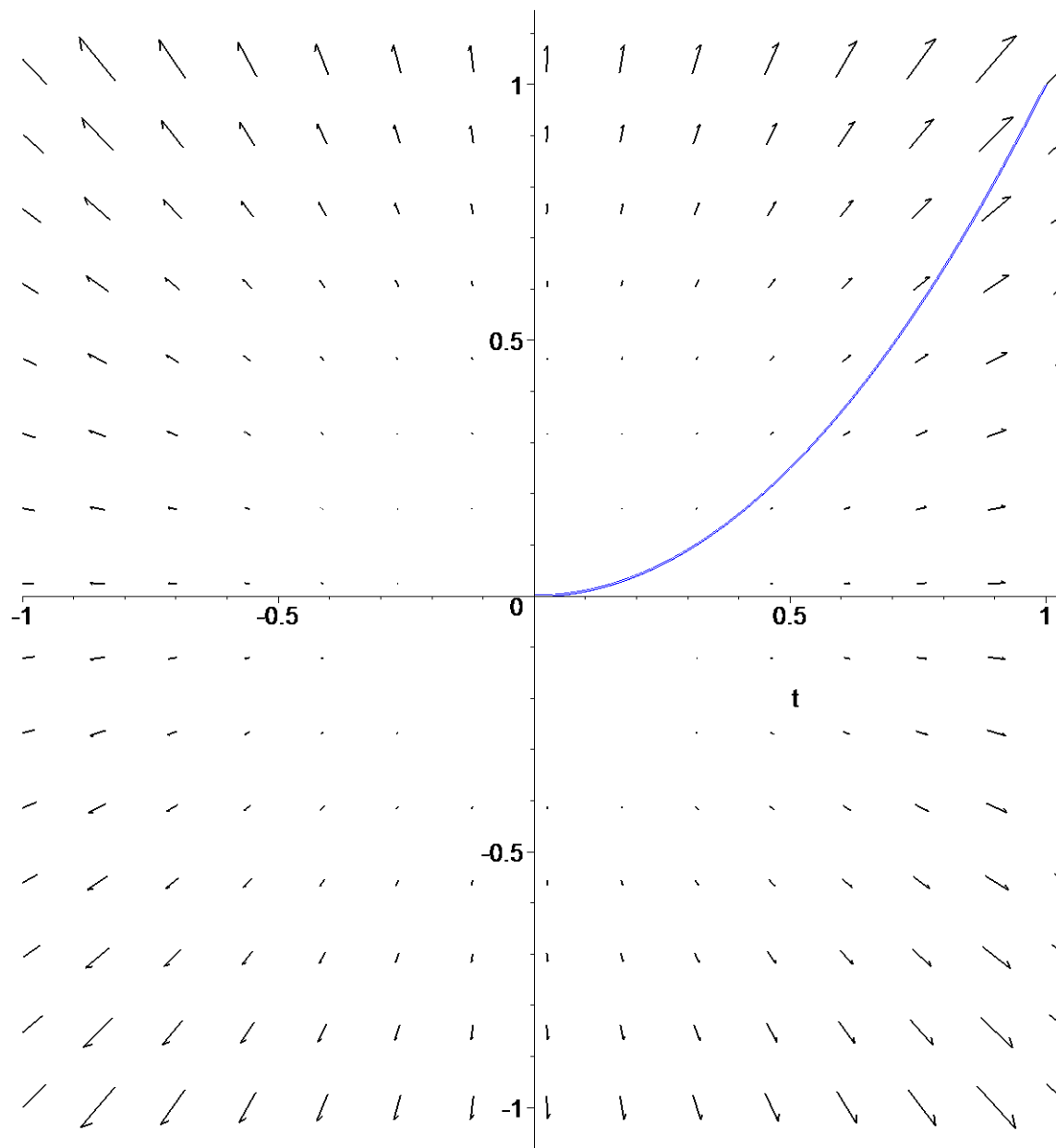
```
> p := plot(t^2, t=-1..1.02, color=WHITE);
_p := plot(t^2, t=0..1, color=blue, thickness=3);
```

# Graf části paraboly. První příkaz je v tomto tvaru, protože jinak by obrázek zahrnoval pouze interval <0,1> na ose x, což není vhodné kvůli lepší představě tvaru vektorového pole. Proto také byla zvolena křivka bílé barvy, protože splyne s pozadím..

```
> p1 := fieldplot(vpole, x=-1..1.05, y=-1..1.05, grid=[15,15], thickness=2);
```

# Obrázek vektorového pole

```
> display({p, _p, p1}, scaling=constrained);
```



K výpočtu užijeme potenciálu. Nutnou podmínkou pro získání potenciálu je nulová rotace, tj. podmínka

>  $\text{Diff}(P(x,y), y) = \text{Diff}(Q(x,y), x)$  ;

$$\frac{\partial}{\partial y} P(x,y) = \frac{\partial}{\partial x} Q(x,y)$$

kde  $P$  je první prvek seznamu  $vpole$  a  $Q$  je druhý prvek. Ověříme platnost této podmínky

>  $\text{Diff}(vpole[1], y) = \text{diff}(vpole[1], y)$  ;

$\text{Diff}(vpole[2], x) = \text{diff}(vpole[2], x)$  ;

$$\frac{\partial}{\partial y} ((4x^2 + 4y^2)x) = 8yx$$

$$\frac{\partial}{\partial x} ((4x^2 + 4y^2)y) = 8yx$$

Přístupme k výpočtu potenciálu.

>  $\text{ScalarPotential}(vf)$  ;

$$x^4 + 2y^2x^2 + y^4$$



V tomto případě můžeme konstatovat, že křivkový integrál nezávisí na integrační cestě. To znamená, že křivkový integrál podél libovolné křivky spojující tyto dva body bude mít stejnou hodnotu.

>

>

## 7. Plošný integrál

Pojem plošného integrálu stejně jako pojem křivkového integrálu probíraného v předchozí kapitole je nedílnou součástí integrálního počtu funkcí více proměnných. Systém Maple obsahuje několik příkazů, které dovedou v některých případech tyto integrály spočítat. Nejprve připomeňme, že se plošné integrály dělí stejně jako křivkové na dva druhy. Přesné definice těchto integrálů lze najít např. v [1].

### • Plošný integrál I. druhu

Pro připomenutí uvedeme pouze zápis tohoto druhu integrálu v případě silné regularity úseku  $S$  vzhledem k  $xy$  daného rovnicí  $z = f(x, y)$ .

$$\iint_{\bar{S}} F(x, y, z) d\sigma = \iint_{\bar{S}_{xy}} F(x, y, f(x, y)) \sqrt{1 + f_x^2(x, y) + f_y^2(x, y)} dx dy$$

Předpokladem pro tento zápis je silná regularita úseku plochy  $S$  vzhledem k  $xy$ , dále spojitost funkce  $F$  na  $S$  a spojitost funkce  $f(x, y)$  a příslušných derivací.

V případě, že úsek je zadán parametrickými rovnicemi, potom

$$\iint_{\bar{S}} F(x, y, z) d\sigma = \iint_M F(X(u, v), Y(u, v), Z(u, v)) H(u, v) du dv$$

Přesné definice naleznete např. v [1].

>

### • Plošný integrál II. druhu

Základem je vektorové pole ve tvaru

$$> \mathbf{F}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = (\mathbf{P}(\mathbf{x}, \mathbf{y}, \mathbf{z}), \mathbf{Q}(\mathbf{x}, \mathbf{y}, \mathbf{z}), \mathbf{R}(\mathbf{x}, \mathbf{y}, \mathbf{z}));$$

$$\mathbf{F}(x, y, z) = (\mathbf{P}(x, y, z), \mathbf{Q}(x, y, z), \mathbf{R}(x, y, z))$$

a normála ve tvaru

$$> \mathbf{n} = (\cos(\alpha), \cos(\beta), \cos(\gamma));$$

$$\mathbf{n} = (\cos(\alpha), \cos(\beta), \cos(\gamma))$$

Pak můžeme tento integrál zapsat

$$\iint_{\bar{S}} (P \cos \alpha + Q \cos \beta + R \cos \gamma) d\sigma = \iint_{\bar{S}} P(x, y, z) dy dz + \iint_{\bar{S}} Q(x, y, z) dx dz + \iint_{\bar{S}} R(x, y, z) dx dy.$$

kde  $\bar{S}$  je regulární ke všem třem souřadnicovým rovinám,  $\mathbf{n}(x, y, z)$  je jeho orientovaná jednotková normála a  $P, Q, R$  jsou funkce spojité na  $\bar{S}$ .

Přesné definice naleznete např. v [1].

>

## 7.1 Plošný integrál I. druhu

System Maple obsahuje pro výpočet plošného integrálu I. druhu příkaz **SurfaceInt**, který je součástí rozšiřující knihovny *VectorCalculus*.

Syntaxe je

**SurfaceInt(fce, [x,y,z] = dom)** , kde *fce* je integrand a *dom* je úsek plochy daný

jednou z následujících konstrukcí:

*Box(r1, r2, r3)*, *Sphere(cen, rad)*, *Surface(v)*. Více o použití těchto konstrukcí naleznete v příloze *Knihovna VectorCalculus* nebo v následujících příkladech, popř. v nápovědě.

>

### Příklad 1

Vypočtěte plošný integrál  $\iint_S z\sqrt{x^2 + y^2 + 1}d\sigma$  po části ležící v I. oktantu hyperbolického paraboloidu ohraničeného válcem.

> **restart:with(student):with(plots):with(VectorCalculus):**

Warning, the name changecoords has been redefined

Warning, the assigned names <, > and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: \*, +, .., Vector, diff, int, limit, series

Rovnice paraboloidu

> **z=x\*y;**

$$z = x y$$

Rovnice válce

> **val:=x^2+y^2=r^2;**

$$val := x^2 + y^2 = r^2$$

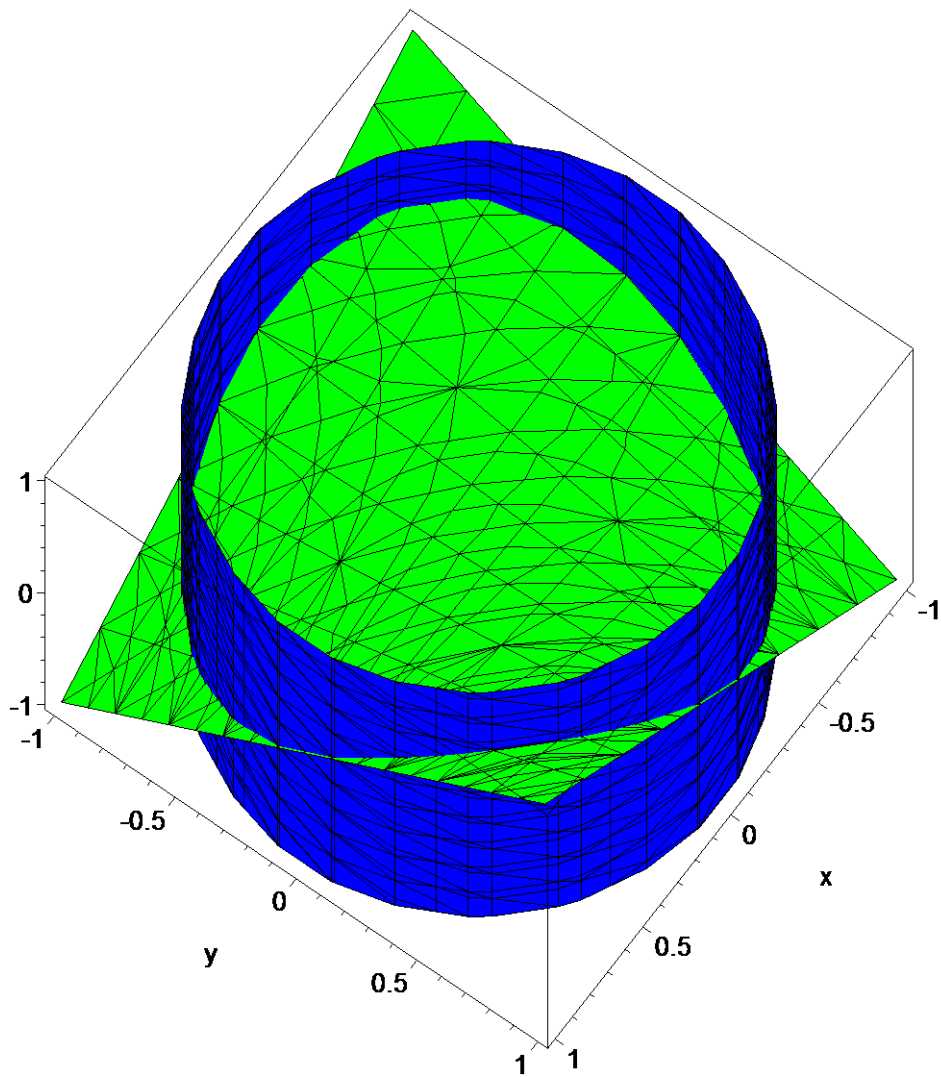
Nakreslíme obrázek

> **p:=implicitplot3d(z=x\*y, x=-1..1, y=-1..1, z=-1..1, color=green):**

> **p1:=implicitplot3d(x^2+y^2=1, x=-1..1, y=-1..1, z=-1..1, color=blue):**

# Vykreslení funkce dané implicitně ve 3D.

> **display({p,p1}, orientation=[36,22], axes=boxed);**



Oblast je silně regulární vzhledem k  $xy$ , tedy můžeme použít vzorec

$$\iint_{\bar{S}} F(x, y, z) d\sigma = \iint_{\bar{S}_{xy}} F(x, y, f(x, y)) \sqrt{1 + f_x^2(x, y) + f_y^2(x, y)} dx dy, \text{ kde funkce } z=f(x, y)=x*y.$$

Zapišeme funkci  $f(x, y)$  a spočteme příslušné derivace.

> **f:=x\*y;**

$$f := x y$$

> **fx:=diff(f, x);**

$$fx := y$$

> **fy:=diff(f, y);**

$$fy := x$$

Po dosazení dostáváme dvojný integrál přes průmět plochy  $S$  do roviny  $xy$ , tj. přes čtvrtkruh.

> **i:=Doubleint(f\*sqrt(x^2+y^2+1)\*sqrt(1+fx^2+fy^2), x, y, S[xy]);**

$$i := \iint_{S_{xy}} x y (x^2 + y^2 + 1) dx dy$$

Zavedeme polární souřadnice

```
> assume (u>0) ;
```

```
> changevar ({x=u*cos (v) , y=u*sin (v) } , subs (S [xy]=S [uv] , i) , [u,v]) ;
```

$$\iint_{S_{uv}} (u^3 + u^5) \sin(v) \cos(v) du dv$$

# Příkaz *changevar* neumí transformovat meze integrálu, proto nebyly v předchozím příkazu uvedeny a bylo použito jiného označení pro oblast integrace.

Integrujeme přes čtvrtkruh, počítáme tedy

```
> Doubleint (integrand (%) , u=0..r , v=0..Pi/2) ;
```

$$\int_0^{\frac{\pi}{2}} \int_0^r (u^3 + u^5) \sin(v) \cos(v) du dv$$

# Pomocí příkazu *integrand* získáme integrand z předchozího integrálu.

Vypočteme integrál

```
> value (%) ;
```

$$\frac{1}{12} r^6 + \frac{1}{8} r^4$$

Pro zrychlení výpočtů lze užít příkazy z *VectorCalculus*.

```
> assume (r>0) ;
```

```
> SurfaceInt (z*(x^2+y^2+1)^(1/2) , [x,y,z]=Surface (<s,t,s*t> ,
s=0..r , t=0..sqrt(r^2-s^2))) ;
```

$$-\frac{r^6}{24} - \frac{r^4}{8} + \frac{\left(\frac{1}{4}r^4 + \frac{1}{2}r^2\right)r^2}{2}$$

```
> simplify (%) ;
```

$$\frac{1}{12} r^6 + \frac{1}{8} r^4$$

# Příkaz *Surface* určí plochu, přes kterou se bude integrovat. Využilo se rovnice válce.

```
>
```

## Příklad 2

Vypočtete plošný integrál  $\iint_S xyz d\sigma$

```
> restart:with (plots) :with (student) :with (VectorCalculus) :
```

```
Warning, the name changecoords has been redefined
```

```
Warning, the assigned names <,> and <|> now have a global binding
```

```
Warning, these protected names have been redefined and unprotected: *, +, ..
Vector, diff, int, limit, series
```

po části roviny

```
> rov:=z=1-x-y;
```

$$rov := z = -x + 1 - y$$

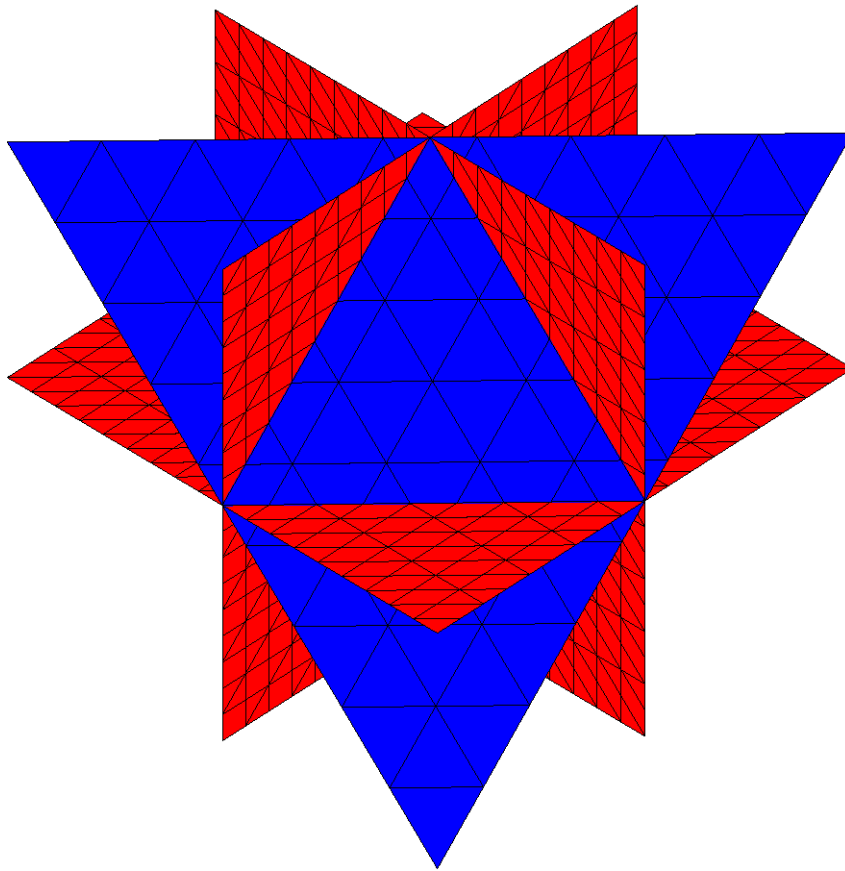
ležící v prvním oktantu.

Nakreslíme obrázek

```

> p:=implicitplot3d(rov,x=-1..1,y=-1..1,z=-1..1,color=blue):
Nakreslení roviny pomocí příkazu, který vykreslí funkci danou implicitně ve 3D.
> p1:=implicitplot3d(x>0,x=-1..1,y=-1..1,z=-1..1,color=red):
> p2:=implicitplot3d(y>0,x=-1..1,y=-1..1,z=-1..1,color=red):
> p3:=implicitplot3d(z>0,x=-1..1,y=-1..1,z=-1..1,color=red):
Nakreslení prvního oktantu pomocí rovin.
> display({p,p1,p2,p3},orientation=[44,52]);

```



Oblast je silně regulární vzhledem ke všem souřadnicovým rovinám. Můžeme tedy použít vzorec uvedený na začátku této kapitoly, kde funkce  $z = f(x, y) = 1 - x - y$ .

Zapišeme funkci  $f(x, y)$  a spočteme příslušné derivace.

```
> f:=1-x-y;
```

$$f := -x + 1 - y$$

```
> fx:=diff(f,x);
```

$$fx := -1$$

```
> fy:=diff(f,y);
```

$$f_y := -1$$

Po dosazení dostáváme dvojný integrál

```
> Doubleint(x*y*f*sqrt(1+fx^2+fy^2),
 y=0..1-x,x=0..1);
```

$$\int_0^1 \int_0^{1-x} xy(-x+1-y)\sqrt{3} dy dx$$

Spočteme integrál

```
> value(%);
```

$$\frac{\sqrt{3}}{120}$$

Pro zrychlení výpočtů lze užít příkazů z *VectorCalculus*

```
> SurfaceInt(x*y*z, [x,y,z] =
 Surface(<s,t,1-s-t>, [s,t]=Triangle(<0,0>,<1,0>,<0,1>)));
```

$$\frac{\sqrt{3}}{120}$$

# Příkaz *Surface* určí plochu, přes kterou se bude integrovat. Využilo se konstrukce *Triangle* pro vytvoření trojúhelníka ze zadaných bodů. Trojúhelník jsme získali průmětem zadané roviny do roviny *xy*.

```
>
>
>
```

### Příklad 3

Spočtěte plošný obsah rotačního paraboloidu omezeného rovinou  $z=0$ .

```
> restart:with(plots):with(student):with(VectorCalculus):
```

```
Warning, the name changecoords has been redefined
```

```
Warning, the assigned names <,> and <|> now have a global binding
```

```
Warning, these protected names have been redefined and unprotected: *, +, ..,
Vector, diff, int, limit, series
```

Obsah plochy se vypočte pomocí následujícího integrálu  $\iint_S 1 d\sigma$

Rovnice rotačního paraboloidu

```
> rp:=z=r^2-x^2-y^2;
```

$$rp := z = r^2 - x^2 - y^2$$

Nakreslíme obrázek

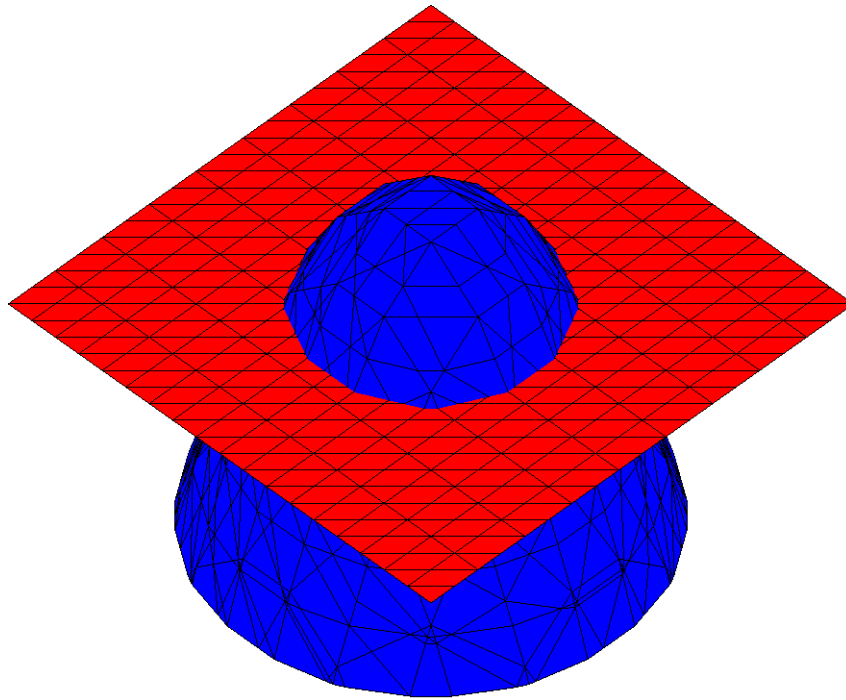
```
> p:=implicitplot3d(subs(r=1,rp),x=-2..2,y=-2..2,z=-2..2,
 color=blue):
```

```
Pevně jsme zvolili r=1.
```

```
> p1:=implicitplot3d(z,x=-2..2,y=-2..2,z=-2..2,color=red):
```

```
Nakreslení roviny.
```

```
> display({p,p1});
```



Vypočítáme příslušné derivace

> **f:=r^2-x^2-y^2;**

$$f := r^2 - x^2 - y^2$$

> **fx:=diff(f,x);**

$$fx := -2x$$

> **fy:=diff(f,y);**

$$fy := -2y$$

Dosadíme do integrálu, který budeme počítat přes průmět plochy do souřadnicové roviny  $xy$  - jde o kruh.

> **i:=Doubleint(1\*sqrt(1+fx^2+fy^2),x,y,S[xy]);**

$$i := \iint_{S_{xy}} \sqrt{4x^2 + 1 + 4y^2} \, dx \, dy$$

Zavedeme polární souřadnice

> **assume(u>0);**



```
> changevar ({x=u*cos(v), y=u*sin(v)}, subs(S[xy]=S[uv], i), [u, v]);
```

$$\iint_{S_{uv}} \sqrt{1+4u^2} u \, du \, dv$$

# Systém Maple neumí transformovat meze integrace!

Integrujeme přes kruh, počítáme tedy integrál

```
> Doubleint(integrand(%), u=0..r, v=0..2*Pi);
```

$$\int_0^{2\pi} \int_0^r \sqrt{1+4u^2} u \, du \, dv$$

Výsledkem je

```
> value(%);
```

$$\frac{(1+4r^2)^{3/2} \pi}{6} - \frac{\pi}{6}$$

Systém Maple nabízí rychlejší řešení s využitím příkazu *SurfaceInt*.

```
> SurfaceInt(1, [x, y, z] = Surface(<s, t, r^2-s^2-t^2>, [s, t] =
Circle(<0, 0>, r)));
```

$$\frac{(1+4r^2)^{3/2} \pi}{6} - \frac{\pi}{6}$$

# Využili jsme konstrukce *Circle*, která popíše kruh pomocí středu a poloměru.

```
>
>
```

#### Příklad 4

Následující příklad ukazuje výpočet plošného integrálu pomocí jeho *parametrického vyjádření*.

```
> restart:with(student):with(plots):
with(VectorCalculus):with(linalg):
```

Warning, the name changecoords has been redefined

Warning, the assigned names <, > and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: \*, +, .., Vector, diff, int, limit, series

Warning, the previous binding of the name Wronskian has been removed and it now has an assigned value

Warning, the protected names norm and trace have been redefined and unprotected

Určete hmotnost kulové plochy, je-li plošná hustota v každém bodě dané plochy rovna vzdálenosti bodu od osy *z*.

Rovnice kulové plochy (není regulární)

```
> kp:=x^2+y^2+z^2=r^2;
```

$$kp := x^2 + y^2 + z^2 = r^2$$

Plošná hustota

```
> mu:=sqrt(x^2+y^2);
```

$$\mu := \sqrt{x^2 + y^2}$$

Hmotnost plochy se spočítá pomocí  $\iint_S \mu d\sigma$ , kde  $\mu$  je plošná hustota.

Parametrické rovnice koule

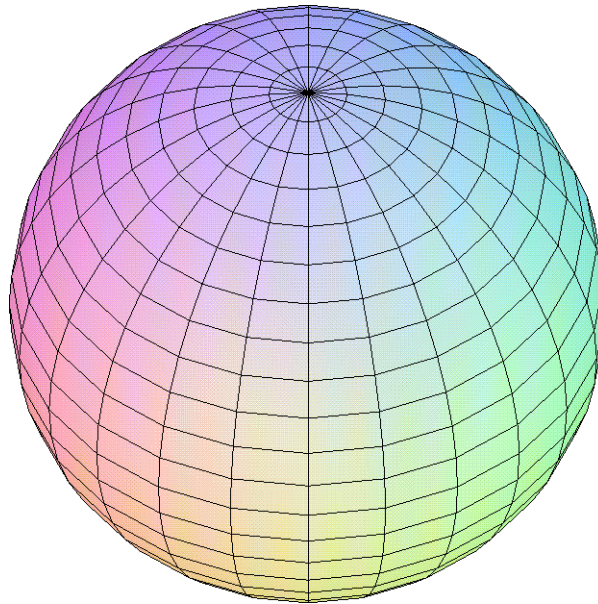
>  $X:=r*\cos(u)*\sin(v)$  :

>  $Y:=r*\sin(u)*\sin(v)$  :

>  $Z:=r*\cos(v)$  :

Pro kontrolu nakreslíme obrázek

> `plot3d([cos(u)*sin(v), sin(u)*sin(v), cos(v)], u=0..2*Pi, v=0..Pi);`



Dále určíme normálu

>  $n:=[A,B,C]$  ;

$n := [A, B, C]$

Jednotlivé složky určíme pomocí příkazů z rozšiřující knihovny *linalg*. Nejprve určíme příslušné derivace.

>  $N:=\text{Matrix}([\text{diff}([X,Y,Z],u), \text{diff}([X,Y,Z],v)])$  ;

$$N := \begin{bmatrix} -r \sin(u) \sin(v) & r \cos(u) \sin(v) & 0 \\ r \cos(u) \cos(v) & r \sin(u) \cos(v) & -r \sin(v) \end{bmatrix}$$

# Matice derivací funkcí  $X, Y, Z$ .

```
> submatrix(N, 1..2, 2..3);
A:=det(%);
```

$$\begin{bmatrix} r \cos(u) \sin(v) & 0 \\ r \sin(u) \cos(v) & -r \sin(v) \end{bmatrix}$$

$$A := -r^2 \cos(u) \sin(v)^2$$

# Výpočet determinantu submatice, která charakterizuje  $A$ .

```
> submatrix(N, [1, 2], [1, 3]);
B:=det(%);
```

$$\begin{bmatrix} -r \sin(u) \sin(v) & 0 \\ r \cos(u) \cos(v) & -r \sin(v) \end{bmatrix}$$

$$B := r^2 \sin(u) \sin(v)^2$$

# Druhá složka normály

```
> submatrix(N, 1..2, 1..2);
C:=simplify(det(%));
```

$$\begin{bmatrix} -r \sin(u) \sin(v) & r \cos(u) \sin(v) \\ r \cos(u) \cos(v) & r \sin(u) \cos(v) \end{bmatrix}$$

$$C := -r^2 \sin(v) \cos(v)$$

# Třetí složka normály

Nyní určíme velikost normály

```
> H:=sqrt(A^2+B^2+C^2);
```

$$H := \sqrt{r^4 \cos(u)^2 \sin(v)^4 + r^4 \sin(u)^2 \sin(v)^4 + r^4 \sin(v)^2 \cos(v)^2}$$

Zjednodušíme pomocí známého vzorce

```
> simplify(H, {sin(u)^2+cos(u)^2=1, sin(v)^2+cos(v)^2=1});
```

$$\sqrt{-r^4 \cos(v)^2 + r^4}$$

Využijeme tentýž vzorec

```
> simplify(%, {1-cos(v)^2=sin(v)^2});
```

$$\sqrt{r^4 \sin(v)^2}$$

```
> assume(r>0, sin(v)>0);
```

```
> H:=simplify(%, {1-cos(v)^2=sin(v)^2});
```

$$H := r^2 \sin(v)$$

# Mohli jsme omezit  $\sin$  kvůli rozsahu  $v$ .

Dosazením do  $\iint_{\tilde{S}} F(x, y, z) d\sigma = \iint_M F(X(u, v), Y(u, v), Z(u, v)) H(u, v) du dv$  dostaneme dvojný

integrál

```
> Doubleint(subs(x=X, y=Y, sqrt(x^2+y^2))*H, u, v, M);
```

$$\iint_M \sqrt{r^2 \cos(u)^2 \sin(v)^2 + r^2 \sin(u)^2 \sin(v)^2} r^2 \sin(v) du dv$$

```
> i:=Doubleint(integrand(%), u=0..2*Pi, v=0..Pi);
```

$$i := \int_0^\pi \int_0^{2\pi} \sqrt{r^2 \cos(u)^2 \sin(v)^2 + r^2 \sin(u)^2 \sin(v)^2} r^2 \sin(v) du dv$$

# Získání integrandu a přidání integračních mezí.

> **simplify(i, {sin(u)^2+cos(u)^2=1, sin(v)^2+cos(v)^2=1});**

$$r^2 \int_0^\pi \sin(v) \int_0^{2\pi} \sqrt{r^2 \sin(v)^2} du dv$$

# Zjednodušení známými rovnicemi.

> **simplify(%, {1-cos(v)^2=sin(v)^2});**

$$r^3 \int_0^{2\pi} 1 du \int_0^\pi \sin(v)^2 dv$$

Výsledná hodnota je

> **simplify(value(%));**

$$r^3 \pi^2$$

Nyní se podívejme na postup výpočtu pomocí příkazů z knihovny *VectorCalculus*.

> **SurfaceInt(mu, [x,y,z]=Sphere(<0,0,0>,r));**

$$r^3 \pi^2$$

# Prvním parametrem je plošná hustota. Plochu určíme přímo pomocí konstrukce *Sphere*.

>

>

### Příklad 5

Určete plošný obsah závitu šroubové konoidy

> **restart:with(student):with(plots):**

**with(VectorCalculus):with(linalg):**

Warning, the name changecoords has been redefined

Warning, the assigned names <, > and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: \*, +, .., Vector, diff, int, limit, series

Warning, the previous binding of the name Wronskian has been removed and it now has an assigned value

Warning, the protected names norm and trace have been redefined and unprotected

Parametrické rovnice konoidy

> **X:=u\*cos(v);**

**Y:=u\*sin(v);**

**Z:=v;**

$$X := u \cos(v)$$

$$Y := u \sin(v)$$

$$Z := v$$

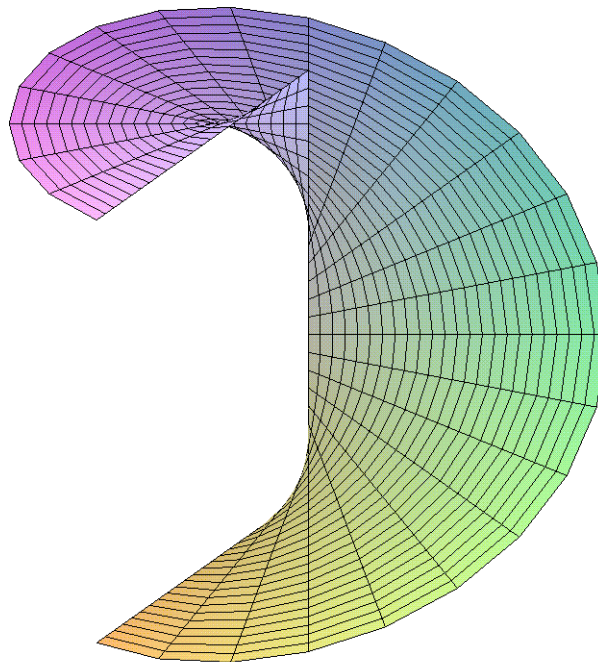
Rozsahy parametrů

```
> ru:=u=0..r;
rv:=v=0..2*Pi;
```

```
ru := u = 0 .. r
rv := v = 0 .. 2 pi
```

Nakreslíme obrázek pro pevně zvolené  $r=1$

```
> plot3d([X,Y,Z],u=0..1,rv);
```



Výpočet integrálu pomocí jeho parametrického vyjádření.

Určíme normálu

```
> n:=[A,B,C];
> M:=Matrix([diff([X,Y,Z],u),diff([X,Y,Z],v)]);
```

$$M := \begin{bmatrix} \cos(v) & \sin(v) & 0 \\ -u \sin(v) & u \cos(v) & 1 \end{bmatrix}$$

# Matice derivací jednotlivých funkcí  $X, Y, Z$ .

```
> submatrix(M,1..2,2..3);
A:=det(%);
```

$$\begin{bmatrix} \sin(v) & 0 \\ u \cos(v) & 1 \end{bmatrix}$$

$$A := \sin(v)$$

# Výpočet determinantu submatice, která charakterizuje  $A$ .

> **submatrix(M, [1, 2], [1, 3]);**  
**B:=det(%);**

$$\begin{bmatrix} \cos(v) & 0 \\ -u \sin(v) & 1 \end{bmatrix}$$

$$B := \cos(v)$$

# Druhá složka normály

> **submatrix(M, 1..2, 1..2);**  
**C:=simplify(det(%));**

$$\begin{bmatrix} \cos(v) & \sin(v) \\ -u \sin(v) & u \cos(v) \end{bmatrix}$$

$$C := u$$

# Třetí složka normály

> **n;**

$$[\sin(v), \cos(v), u]$$

Nyní určíme velikost normály

> **H:=sqrt(A^2+B^2+C^2);**

$$H := \sqrt{\cos(v)^2 + \sin(v)^2 + u^2}$$

Zjednodušíme pomocí známého vzorce

> **simplify(H, {sin(v)^2+cos(v)^2=1});**  

$$\sqrt{1+u^2}$$

Počítáme integrál

> **Doubleint(% , sigma, `` , S);**

$$\iint_S \sqrt{1+u^2} \, d\sigma \, d$$

Po dosazení obdržíme

> **Doubleint(integrand(%), ru, rv);**

$$\int_0^{2\pi} \int_0^r \sqrt{1+u^2} \, du \, dv$$

Výsledek vypočteme pomocí

> **value(%);**

$$r\sqrt{1+r^2} \pi + \operatorname{arcsinh}(r) \pi$$

> **convert(% , ln);**

$$r\sqrt{1+r^2} \pi + \ln(r + \sqrt{1+r^2}) \pi$$

# Při tradičním výpočtu dostáváme tento výsledek. Pomocí předchozího příkazu můžeme výsledek vypočtený systémem Maple převést na požadovaný tvar.

Výpočet pomocí knihovny *VectorCalculus*

> **SurfaceInt(1, [x,y,z]=Surface(<X,Y,Z>, ru, rv));**

$$\pi (r\sqrt{1+r^2} + \operatorname{arcsinh}(r))$$

>  
>  
>

## 7.2 Plošný integrál II. druhu

Pro výpočet plošného integrálu II. druhu obsahuje systém Maple příkaz *Flux*, který je součástí již zmiňované rozšiřující knihovny *VectorCalculus*.

Syntaxe je

$Flux(fce, dom, norm)$ , kde *fce* je zadané vektorové pole pomocí příkazu *VectorField* v dané souřadnicové soustavě, *dom* určuje plochu pomocí jedné z následujících možností: *Box(r1, r2, r3, [dir])*, *Sphere(cen, rad, [dir])*, *Surface(v)*. Třetí nepovinný parametr určuje směr normály, implicitně směřuje ven (ozn. *outward*). V opačném případě je nutné třetí parametr uvést, a to pomocí slova *inward*.

>

### Příklad 1

Vypočtete plošný integrál přes danou plochu.

```
> restart:with (plots) :with (VectorCalculus) :with (student) :
```

```
Warning, the name changecoords has been redefined
```

```
Warning, the assigned names <, > and <|> now have a global binding
```

```
Warning, these protected names have been redefined and unprotected: *, +, .., Vector, diff, int, limit, series
```

```
> Doubleint (y, y, z, S) + Doubleint (x, x, z, S) + Doubleint (z, x, y, S) ;
```

$$\iint_S y \, dy \, dz + \iint_S x \, dx \, dz + \iint_S z \, dx \, dy$$

Určíme vektorové pole pomocí integrandů jednotlivých integrálů.

```
> vpole := [y, x, z] ;
```

```
vpole := [y, x, z]
```

Plocha je zadána pomocí trojúhelníka ležícího v rovině

```
> rov1 := x - y + z = 1 ;
```

```
rov1 := x - y + z = 1
```

který je vymezen rovinami

```
> rov2 := x >= 0, y <= 0, z >= 0 ;
```

```
rov2 := 0 <= x, y <= 0, 0 <= z
```

Trojúhelník je orientován vektorem svírajícím ostrý úhel s kladnou poloosou z.

Nakreslíme obrázek

```
> p1 := fieldplot3d (vpole, x = -2..2, y = -2..2, z = -2..2,
 color = red, thickness = 2, grid = [5, 5, 5]) :
```

```
Vektorové pole ve 3D
```

```
> p2 := implicitplot3d (x >= 0, x = -2..2, y = -2..2, z = -2..2,
 color = yellow) :
```

```
> p3 := implicitplot3d (y <= 0, x = -2..2, y = -2..2, z = -2..2,
 color = yellow) :
```

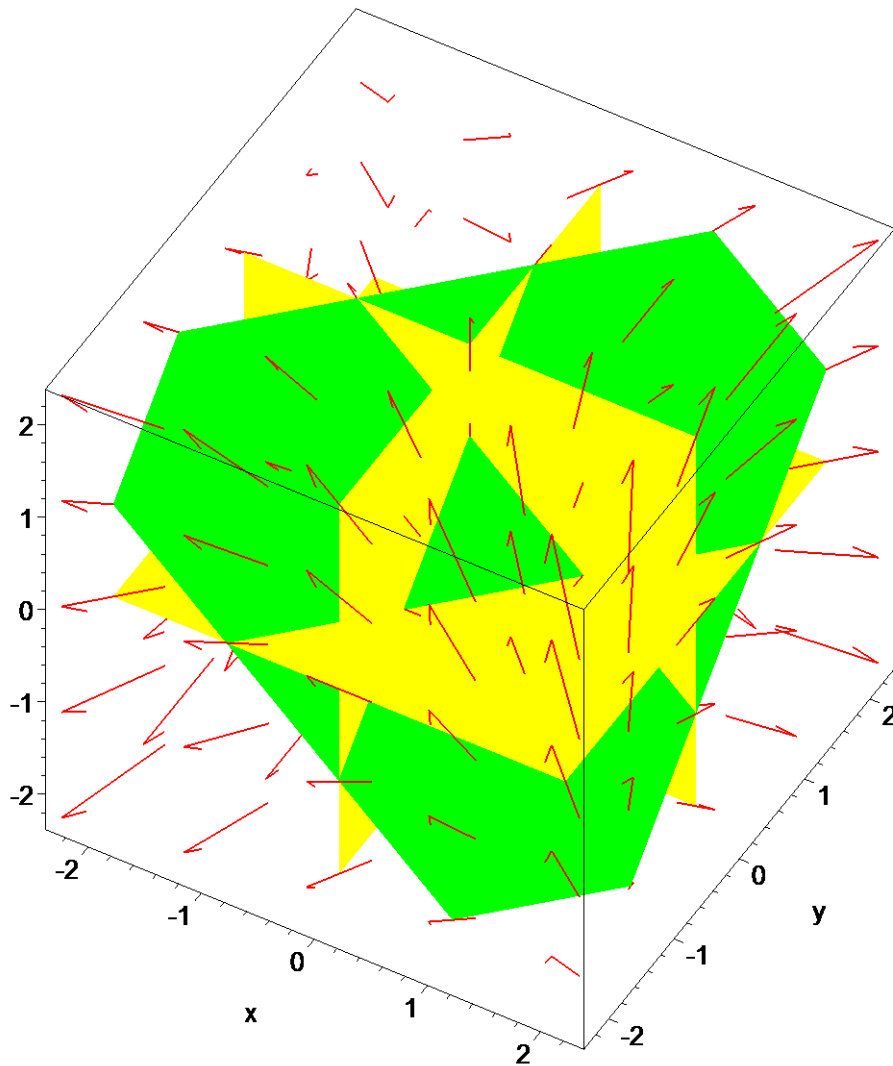
```
> p4 := implicitplot3d (z >= 0, x = -2..2, y = -2..2, z = -2..2,
 color = yellow) :
```

```
> p5 := implicitplot3d (x - y + z = 1, x = -2..2, y = -2..2, z = -2..2,
 color = green) :
```



# Nakreslení daných rovin v implicitním tvaru.

```
> display({p1,p2,p3,p4,p5}, axes=boxed,
orientation=[-60,45], style=patchnogrid);
```



# Parametr *patchnogrid* zakazuje vykreslení triangulace na zadané ploše

Nyní přistupme k výpočtu integrálu. Rozdělme výše uvedený integrál na tři dílčí. Výpočet ukážeme na jednom z nich, ostatní budou spočteny analogicky.

Počítejme tedy integrál

```
> epsilon[x]*Doubleint(y,y,z,S[yz]);
```

$$\epsilon_x \iint_{S_{yz}} y \, dy \, dz$$

Plocha je v tomto případě silně regulární vzhledem k rovině  $yz$ .

Z rovnice roviny vyjádříme proměnnou  $x$ .

```
> h:=isolate(rov1,x);
```

$$h := x = 1 + y - z$$

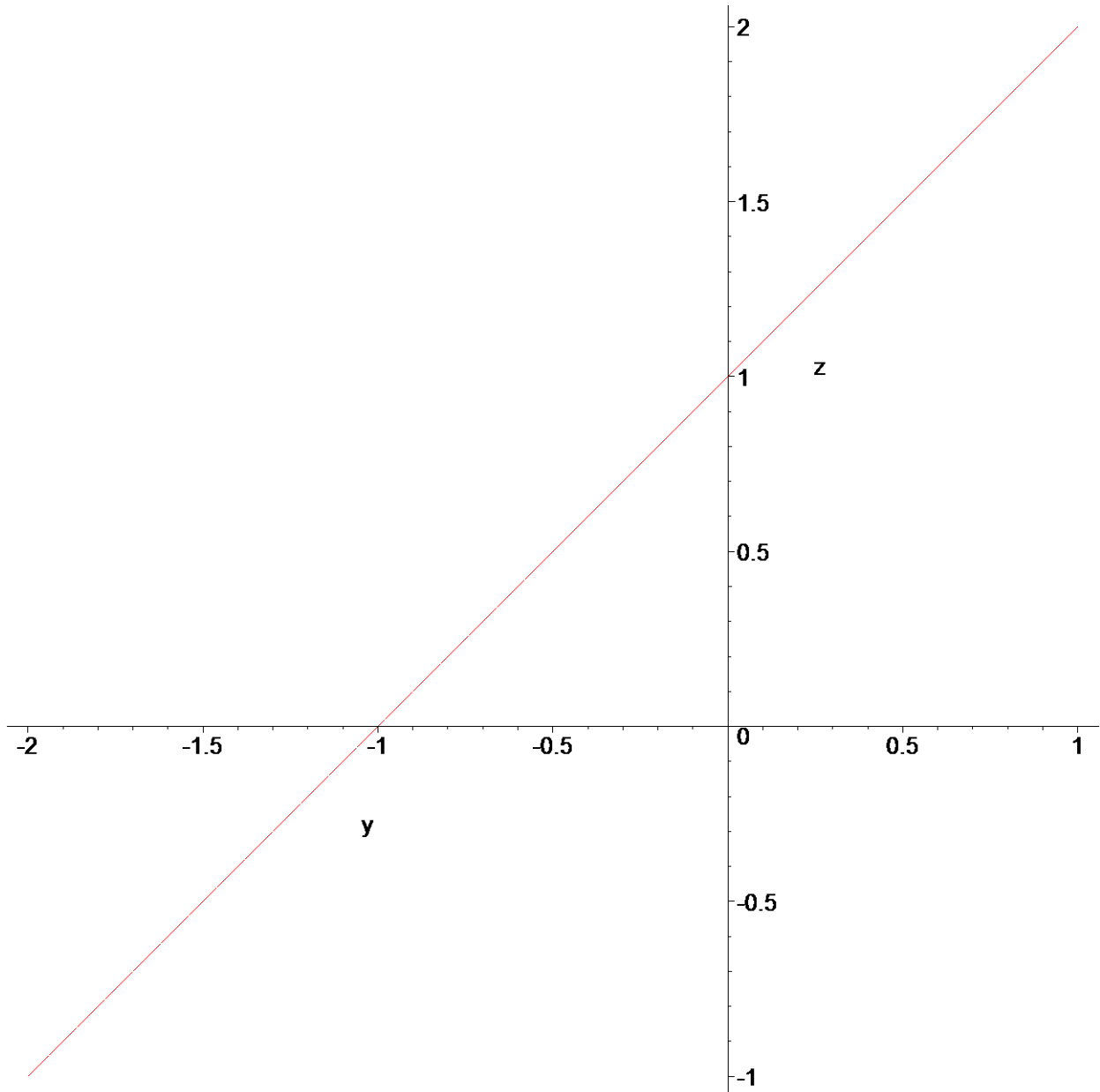
Průmětem dané roviny do roviny  $yz$  dostaneme trojúhelník a tím i meze pro integraci.

```
> ry:=y=-1..0;
rz:=z=0..y+1;
```

```
ry := y = -1 .. 0
rz := z = 0 .. 1 + y
```

Pro lepší představu nakreslíme obrázek

```
> implicitplot(y-z+1,y=-2..2,z=-2..2);
```



Dále určíme normálový vektor, který má tvar  $n = (1, -h'_y, -h'_z)$ .

Spočteme požadované derivace

```
> n := [1, -diff(rhs(h), y), -diff(rhs(h), z)];
n := [1, -1, 1]
```

Protože normálový vektor svírá s kladnou poloosou  $z$  ostrý úhel (určeno pomocí třetí souřadnice normálového vektoru  $n$ ), má tato normála stejný směr jako je směr předepsané normály, proto klademe

```
> epsilon[x]=1;
```

$$\varepsilon_x = 1$$

Počítáme tedy integrál

> **il:=1\*Doubleint(y,rz,ry);**

$$il := \int_{-1}^0 \int_0^{1+y} y \, dz \, dy$$

> **value(%);**

$$\frac{-1}{6}$$

Zbývající dva integrály spočteme analogicky

> **epsilon[y]\*Doubleint(x,x,z,S[xz]);**

**epsilon[z]\*Doubleint(z,x,y,S[xy]);**

$$\varepsilon_y \int \int_{S_{xz}} x \, dx \, dz$$

$$\varepsilon_z \int \int_{S_{xy}} z \, dx \, dy$$

Plocha je v tomto případě silně regulární vzhledem k rovině  $xz$ , popř.  $xy$ .

Z rovnice roviny vyjádříme proměnnou  $y$ , popř.  $z$ .

> **q:=isolate(rov1,y);**

**f:=isolate(rov1,z);**

$$q := y = -1 + x + z$$

$$f := z = 1 - x + y$$

Průmětem dané roviny do roviny  $xz$  dostaneme trojúhelník a tím i meze pro integraci.

> **qx:=x=0..1;**

**qz:=z=0..1-x;**

$$qx := x = 0 \dots 1$$

$$qz := z = 0 \dots 1 - x$$

Průmětem dané roviny do roviny  $xy$  dostaneme trojúhelník a tím i meze pro integraci.

> **fx:=x=0..1;**

**fy:=y=-1+x..0;**

$$fx := x = 0 \dots 1$$

$$fy := y = x - 1 \dots 0$$

Normálové vektory jsou tvaru  $n_q = (-q'_x, 1, -q'_z)$ ,  $n_f = (-f'_x, -f'_y, 1)$ .

Spočteme požadované derivace

> **nq:=[-diff(rhs(q),x),1,-diff(rhs(q),z)];**

**nf:=[-diff(rhs(f),x),-diff(rhs(f),y),1];**

$$nq := [-1, 1, -1]$$

$$nf := [1, -1, 1]$$

Protože normálový vektor svírá s kladnou poloosou  $z$  tupý, resp. ostrý, úhel (určeno pomocí třetí souřadnice normálového vektoru  $n$ ), má normála opačný, resp. stejný směr, jako je směr předepsané normály, proto klademe

> **epsilon[y]=-1;**

**epsilon[z]=1;**

$$\varepsilon_y = -1$$

$$\varepsilon_z = 1$$

Počítáme tedy integrály

```
> i2:=-1*Doubleint(x,qz,qx);
i3:=1*Doubleint(rhs(f),fy,fx);
```

$$i2 := - \int_0^1 \int_0^{1-x} x \, dz \, dx$$

$$i3 := \int_0^1 \int_{x-1}^0 (1-x+y) \, dy \, dx$$

Výsledný integrál je dán součtem dílčích integrálů

```
> value(i1+i2+i3);
```

$$\frac{-1}{6}$$

Výpočet pomocí knihovny *VectorCalculus*

```
> Flux(VectorField(<y,x,z>,'cartesian[x,y,z]'),
Surface(<s,t,1-s+t>,s=0..1,t=-1+s..0));
```

$$\frac{-1}{6}$$

# Orientace normály je implicitně nastavena.

```
>
```

```
>
```

### Příklad 2

Předchozí integrál spočtěme pomocí skalárního součinu.

V případě silné regularity k jedné ze souřadnicových rovin, je možné plošný integrál přepsat na integrál se skalárním součinem dvou vektorů. Pro silnou regularitu vzhledem k rovině  $xy$  můžeme integrál zapsat

$$\iint_S P \, dy \, dz + \iint_S Q \, dx \, dz + \iint_S R \, dx \, dy = \varepsilon_z \iint_{S_{xy}} [-f'_x, -f'_y, 1] \cdot [P(x, y, f(x, y)), Q(x, y, f(x, y)), R(x, y, f(x, y))]$$

kde  $f(x,y)$  získáme vyjádřením proměnné  $z$  ze zadané roviny.

```
> restart:with(plots):with(VectorCalculus):with(student):
```

```
Warning, the name changecoords has been redefined
```

```
Warning, the assigned names <,> and <|> now have a global binding
```

```
Warning, these protected names have been redefined and unprotected: *, +, ..
Vector, diff, int, limit, series
```

```
> f:=isolate(x-y+z-1,z);
```

$$f := z = -x + y + 1$$

Zadání integrálu

```
> P:=y;
```

```
Q:=x;
```

```
R:=1-x+y;
```

$$P := y$$

$$Q := x$$

$$R := -x + y + 1$$

Rozsah proměnných  $x, y$  získáme průmětem plochy do roviny  $xy$ .

```
> rx:=x=0..1;
 ry:=y=x-1..0;
```

$$rx := x = 0 .. 1$$

$$ry := y = x - 1 .. 0$$

Spočteme požadované derivace

```
> n:=[-diff(rhs(f),x),-diff(rhs(f),y),1];
```

$$n := [1, -1, 1]$$

Vzhledem k souhlasné orientaci této normály se zadanou normálou, položíme

```
> epsilon[z]=1;
```

$$\varepsilon_z = 1$$

Spočteme skalární součin pomocí příkazu *dotprod* z knihovny *linalg*.

```
> v:=simplify(linalg[dotprod](n,[P,Q,R],'orthogonal'));
```

$$v := 2y - 2x + 1$$

# Parametr *orthogonal* je uveden pro tradiční výpočet skalárního součinu.

Po dosazení počítáme následující integrál

```
> Doubleint(v,ry,rx);
```

$$\int_0^1 \int_{x-1}^0 (2y - 2x + 1) dy dx$$

```
> value(%);
```

$$\frac{-1}{6}$$

```
>
```

```
>
```

### Příklad 3

Určete tok polohového vektoru přes horní stranu úseku hyperbolického paraboloidu definovaného nad čtvercem  $\langle 0, a \rangle \times \langle 0, a \rangle$  ležícím v souřadnicové rovině  $xy$ .

```
> restart:with(plots):with(VectorCalculus):with(student):
```

Warning, the name *changecoords* has been redefined

Warning, the assigned names  $\langle, \rangle$  and  $\langle | \rangle$  now have a global binding

Warning, these protected names have been redefined and unprotected:  $*$ ,  $+$ ,  $..$ , *Vector*, *diff*, *int*, *limit*, *series*

```
> Doubleint(x,y,z,S)+Doubleint(y,x,z,S)
 +Doubleint(z,x,y,S);
```

$$\iint_S x dy dz + \iint_S y dx dz + \iint_S z dx dy$$

Úsek je silně regulární vzhledem k rovině  $xy$ , proto můžeme použít postup z předchozího příkladu.

Užijeme tedy následujícího vyjádření

$$\iint_S P dy dz + \iint_S Q dx dz + \iint_S R dx dy = \varepsilon_z \iint_{S_{xy}} [-f'_x, -f'_y, 1] \cdot [P(x, y, f(x, y)), Q(x, y, f(x, y)), R(x, y, f(x, y))]$$

.

Rovnice hyperbolického paraboloidu

```
> par:=z=x*y+a^2;
```

$$par := z = x y + a^2$$

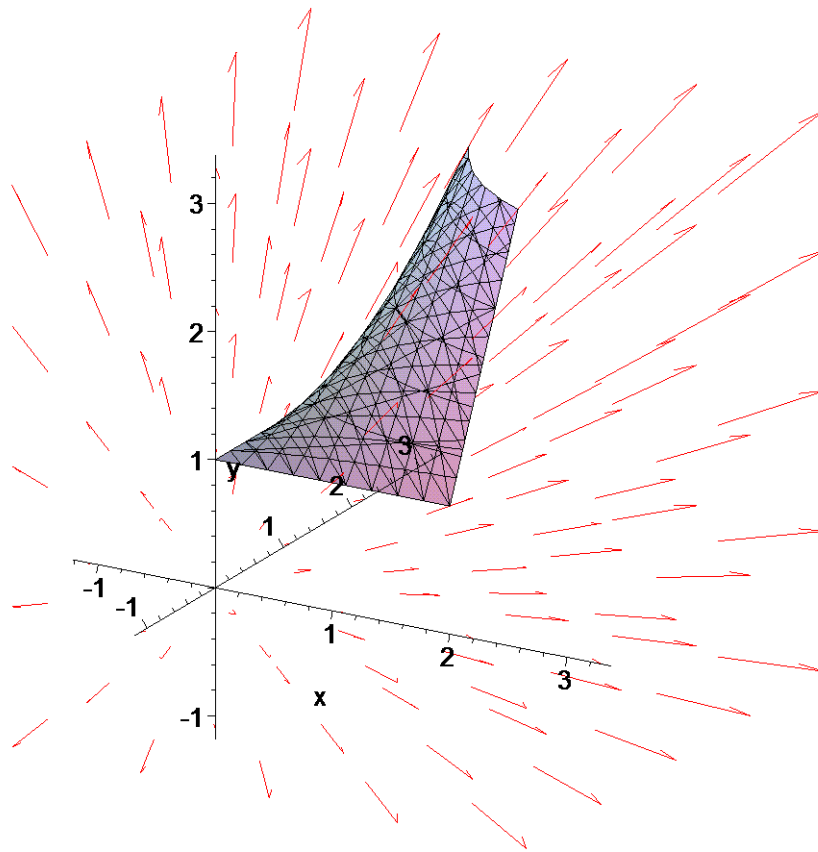
Nakreslíme obrázek pro  $a=1$ .

```
> p:=fieldplot3d([x,y,z],x=-1..3,y=-1..3,z=-1..3,
 grid=[5,5,5],color=red):
```

```
> p1:=implicitplot3d(subs(a=1,par),x=0..2,y=0..2,
 z=0..3,axes=normal,orientation=[-60,70]):
```

# Parametr *axes* určuje zobrazení souřadnicové soustavy.

```
> display({p,p1});
```



Určíme normálu

```
> n:=[-diff(rhs(par),x),-diff(rhs(par),y),1];
```

$$n := [-y, -x, 1]$$

Podle z-ové složky určíme

```
> epsilon[z]=1;
```

$$\varepsilon_z = 1$$

Zápis integrandů

```
> P:=x;Q:=y;R:=x*y+a^2;
```

$$P := x$$

$$Q := y$$

$$R := x y + a^2$$

Spočteme příslušný skalární součin

```
> linalg[dotprod](n, [P,Q,R], 'orthogonal');
```

$$-x y + a^2$$

Plocha je omezena čtvercem v rovině  $xy$ . Tedy počítáme integrál

```
> Doubleint(% , x=0..a, y=0..a);
```

$$\int_0^a \int_0^a -x y + a^2 dx dy$$

```
> value(%);
```

$$\frac{3 a^4}{4}$$

Výpočet pomocí knihovny *VectorCalculus*

```
> Flux(VectorField(<x,y,z>, 'cartesian[x,y,z]'),
Surface(<s,t,s*t+a^2>, s=0..a, t=0..a));
```

$$\frac{3 a^4}{4}$$

```
>
>
>
>
```

## Příloha 1 : Knihovna *student*

Tato kapitola se zabývá jednou z rozšiřujících knihoven, která je primárně určena pro užití při výuce matematiky. Obsahuje mnoho příkazů, které zefektivňují výpočty pomocí systému Maple. Jak již bylo řečeno, jde o rozšiřující knihovnu, proto je nutné nahrát ji do paměti pomocí příkazu *with(student)*, popř. nahrát jen jeden příkaz pomocí *with(student, příkaz)*. V této knihovně jsou obsaženy procedury pro násobení a sloučení výrazů, substituce proměnných, integrování per partes, doplnění na čtverec, výpočet vzdálenosti, atd.

Následující tabulka obsahuje výpis příkazů knihovny *student*.

|                |                                                                                              |
|----------------|----------------------------------------------------------------------------------------------|
| combine        | spojuje členy do výrazu                                                                      |
| completesquare | doplnění výrazu na kvadrát                                                                   |
| D              | operátor derivace                                                                            |
| Diff           | derivace funkce                                                                              |
| distance       | výpočet vzdálenosti                                                                          |
| Doubleint      | dvojný integrál                                                                              |
| equate         | vytvoří množinu rovnic ze seznamů, polí nebo tabulek                                         |
| extrema        | vypočítá extrém funkce                                                                       |
| changevar      | substituce                                                                                   |
| Int            | integrál                                                                                     |
| integrand      | určí integrand                                                                               |
| intercept      | průsečík dvou křivek                                                                         |
| intparts       | metoda výpočtu integrálu - Per Partes                                                        |
| isolate        | vyjádření výrazu(proměnné) z rovnice                                                         |
| leftbox        | aproximace integrálu - graf dolního součtu                                                   |
| leftsum        | aproximace integrálu - dolní součet                                                          |
| Limit          | výpočet limity                                                                               |
| Lineint        | křivkový integrál                                                                            |
| makeproc       | z výrazu vytvoří funkci                                                                      |
| maximize       | maximum                                                                                      |
| middlebox      | aproximace integrálu - graf k funkci                                                         |
| middlesum      | aproximace integrálu - výška obdélníka je dána střední hodnotou funkce v krajních bodech int |
| midpoint       | spočítá střední hodnotu                                                                      |
| minimize       | minimum                                                                                      |
| Point          | kontroluje, zda jde o bod                                                                    |
| powsubs        | substituce členů ve výrazech                                                                 |
| Product        | součin                                                                                       |
| rightbox       | aproximace integrálu - graf horního součtu                                                   |
| rightsum       | aproximace integrálu - horní součet                                                          |
| showtangent    | kreslí tečnu v bodě                                                                          |
| simpson        | integrování pomocí Simpsonovy formule                                                        |
| slope          | počítá směrnici přímky                                                                       |
| Sum            | součet - posloupnost, řada,...                                                               |
| summand        | argument v součtu(sumě)                                                                      |
| trapezoid      | numerická aproximace integrálu - lichoběžníková metoda                                       |
| Tripleint      | matematický zápis trojného integrálu                                                         |
|                |                                                                                              |

Některé z výše uvedených příkazů nyní vysvětlíme.

>

### Operátor derivace $D(f)$

Operátor derivace  $D(f)$ , popř.  $D[i](f)$ , kde  $f$  je výraz (může být aplikován i jako funkce),  $i$  je



kladné číslo nebo výraz (popř. kombinace).

Operátor derivace  $D$  aplikovaný na funkci  $f$ , tj.  $D(f)$ , vypočte derivaci funkce  $f$  (např.  $D(\sin) = \cos$ ).

Derivaci funkce jedné proměnné lze zapsat jako  $D(f)(x) = \text{diff}(f(x), x)$ , kde příkaz *diff* počítá derivaci dané funkce dle dané proměnné. Přesné vysvětlení příkazu *diff* naleznete v kapitole *Diferenciální počet funkcí jedné a více reálných proměnných*.

Pro zápis derivace funkce ve tvaru funkce se používá konstrukce  $f := \text{unapply}(D(f)(x), x)$ .

Pro funkci  $n$  proměnných lze spočítat parciální derivaci podle  $i$ -té proměnné pomocí  $D[i](f)$ .

Následující příklady ukazují použití tohoto operátoru.

```
> restart:with(student) :
> D(sin) ; # základní užití operátoru derivace
D
cos
> D(ln+sin^(1/2)) ;
(a → 1/a) + cos / (2*sqrt(sin))
> D(ln+sin^(1/2))(x) ; # užití operátoru jako funkce
1/x + 1/2 * cos(x) / sqrt(sin(x))
> D(exp)(x) = diff(exp(x), x) ; # přesný význam operátoru D
e^x = e^x
> f := x -> x^2 + 2*x + 1 ;
f := x -> x^2 + 2*x + 1
> f1 := x -> D(f) ; # zápis nové funkce pomocí operátoru
D
f1 := x -> D(f)
jde o funkci
x -> 2*x + 2
> f0 := unapply(D(f)(x), x) ; # jiný možný postup k vytvoření
funkce
f0 := x -> 2*x + 2
jde o funkci
2*x + 2
> D(f) ; # první derivace funkce f
x -> 2*x + 2
> D(f)(2) ; # první derivace funkce f v bode 2
6
> D(D(f)) ; # druhá derivace
2
> (D@@n)(g) ; # n-tá derivace obecné funkce g
(D^(n))(g)
```

```

> D(cos(x)); # špatný zápis derivace funkce
cos(x)
D(cos(x))
> D(cos)(x); # správný zápis derivace cos(x)
-sin(x)
> f2:=(x,y,z)->x^2+y^2+z^2; # funkce proměnných x,y,z
f2 := (x, y, z) → x2 + y2 + z2
> D[1](f2); # parciální derivace funkce f2 podle
první proměnné
(x, y, z) → 2x
> D[2](f2); # parciální derivace funkce f2 podle
druhé proměnné
(x, y, z) → 2y
> D[3](f2)(x,y,z); # parciální derivace funkce f2 podle třetí proměnné v
bodě (x,z,y)
2z
> D[3](f2)(0,0,1); # parciální derivace funkce f2 v
bodě [0,0,1]
2
> D[1,1](f2); # parciální derivace druhého řádu
podle první proměnné, tj.x
2
> D[1,2](f2); # parciální derivace druhého řádu
podle x a pak podle y
0
>

```

## Doubleint

je příkaz pro matematický zápis dvojného integrálu funkce  $g(x,y)$ .

**Doubleint(g,x,y)**

**Doubleint(g,x,y,obor)**

**Doubleint(g,x=a..b,y=c..d)**

Pro výpočet hodnoty dvojného integrálu je nutné použít příkaz **value**.

```

> Doubleint(f(x,y),x=0..1,y=-Pi..Pi); # základní použití
příkazu

```

$$\int_{-\pi}^{\pi} \int_0^1 x^2 + 2x + 1 \, dx \, dy$$

```

> f1:=(x,y)->x+y; # definice funkce
f1 := (x, y) → x + y
> int1:=Doubleint(f1(x,y),x=0..1,y=0..1); # zápis dvojného
integrálu funkce f1

```

$$\text{int1} := \int_0^1 \int_0^1 x + y \, dx \, dy$$

> **value(int1);** # výpočet hodnoty  
předchozího integrálu

1

> **Doubleint(g(x,y), x, y, C);** # ukázka obecného  
zápisu integrační oblasti

$$\iint_C g(x, y) \, dx \, dy$$

>

## extrema

je příkaz určený k výpočtům extrému funkce na množině  $M$ . Výpočet je prováděn metodou *Lagrangeových multiplikátorů*.

*extrema(výraz, {omezení})*

*extrema(funkce f, {omezení}, {proměnné})*

*extrema(funkce f, {omezení}, {proměnné}, 'body')*

Výsledkem je maximální a minimální hodnota funkce  $f$  na hranici množiny  $M$  a do proměnné *'body'* jsou uloženy souřadnice bodů, v nichž extrémy nastávají.

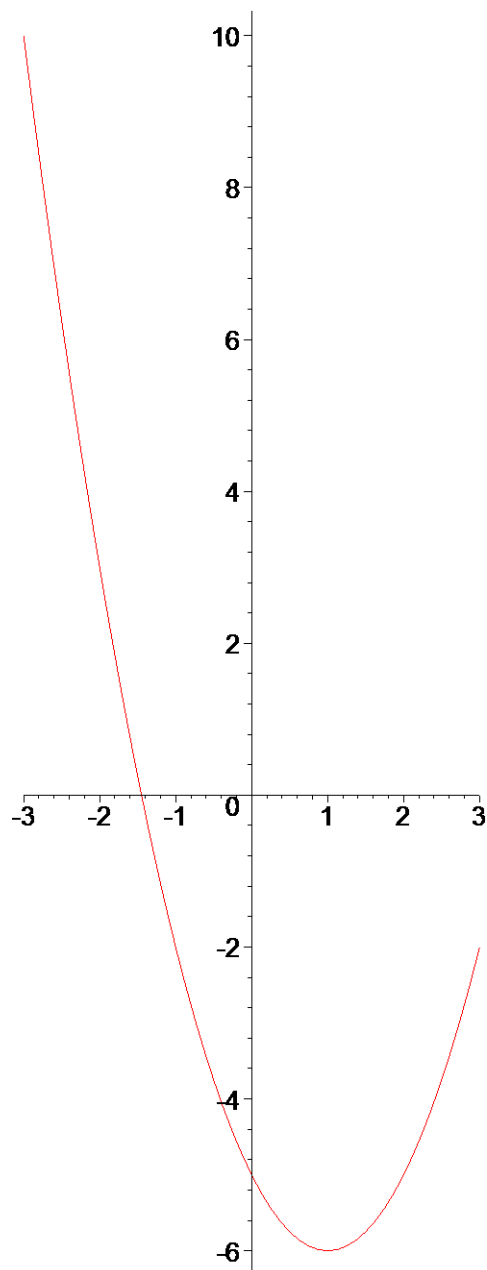
> **f:=x->x^2-2\*x-5;** # definice funkce

$$f := x \rightarrow x^2 - 2x - 5$$

> **extrema(f(x), {});** # výpočet extrémů funkce na celém  
reálném oboru

{-6}

> **plot(f, -3..3, scaling=constrained);**



System Maple nabízí ještě jinou možnost výpočtu extrémů funkce a to užití příkazů *maximize*, popř. *minimize*. Více naleznete v kapitole *Diferenciální počet funkcí jedné a více reálných proměnných*.

```
> minimize(f(x)); # výsledkem je také hodnota funkce
v bodě minima
```

-6

```
> extrema(3*x^3-4*x+1, {}, x, 'bod'); # do identifikátoru bod budou
vypsány souřadnice extrémů
```

$\left\{\frac{-7}{9}, \frac{25}{9}\right\}$

```
> bod; # výpis souřadnic
```

$\left\{\left\{x = \frac{2}{3}\right\}, \left\{x = \frac{-2}{3}\right\}\right\}$

```
>
```

## changevar

je příkaz, pomocí kterého lze provádět substituci v integrálech, sumách či limitách.

**changevar(s,f)**

**changevar(s,f,u)**

**changevar(t,g,v)**

První parametr  $s$  je relace (rovnice) definující novou proměnnou v závislosti na původní (může být definována i implicitně). Druhý parametr  $f$  je výraz, na který bude daná relace aplikována. Třetí nepovinný parametr  $u$  definuje název nové proměnné.

V případě vícenásobné substituce užijeme třetí možnou syntaxi, kde  $t$  je množina relací definující tuto substituci,  $g$  zastupuje dvojný nebo trojný integrál a  $v$  označuje seznam nových proměnných.

Zde je nutné poznamenat, že tento příkaz neumí transformovat meze při vícenásobné substituci.

Pokud provádíme substituci v integrálech, sumách či limitách, je vhodnější užívat příkazy *Int*, *Sum*, *Limit* (tedy pouze matematické zápisy) a hodnoty pak získat pomocí příkazu *value*.

> **p1 := Int(x\*sqrt(1 + x^2), x);** # zápis integrálu

$$p1 := \int x \sqrt{1+x^2} dx$$

> **p2 := changevar(u=1 + x^2, p1, u);** # aplikace substituce

$$p2 := \int \frac{\sqrt{u}}{2} du$$

> **p3:=value(p2);** # výpočet integrálu

$$p3 := \frac{u^{(3/2)}}{3}$$

> **p4 := subs(u=x^2 + 1, p3);** # návrat k původní  
proměnné pomocí příkazu *subs*

$$p4 := \frac{(1+x^2)^{(3/2)}}{3}$$

> **diff(p4, x);** # kontrola výsledku

$$x \sqrt{1+x^2}$$

> **Doubleint(x^2+y^2, x=-1..1, y=-1..1);** # zápis dvojného  
integrálu

$$\int_{-1}^1 \int_{-1}^1 x^2 + y^2 dx dy$$

> **changevar({x=r\*cos(t), y=r\*sin(t)}, %, [t, r]);**

# substituce ve dvojném

integrálu - NETRANSFORMUJÍ SE MEZE !

Warning, Computation of new ranges not implemented

$$\iint r^2 |r| dt dr$$

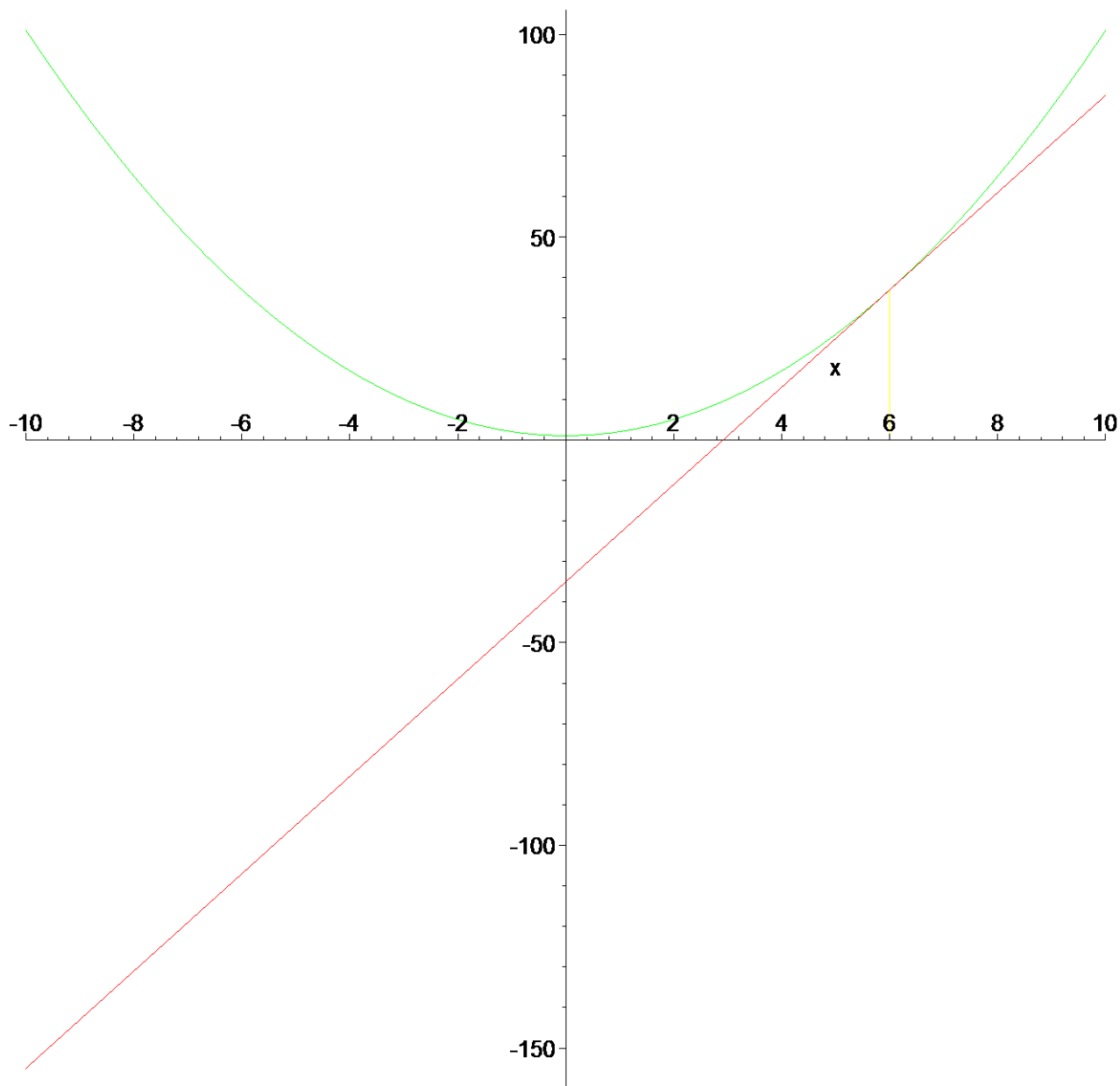
>

## showtangent

vykresluje graf a tečnu ke grafu funkce  $f(x)$  v daném bodě  $[a, f(a)]$

*showtangent(f(x),x=a)*

> **showtangent(x^2+1,x=6);**



>

## Sum

je příkaz sloužící k matematickému zápisu sumace. Pro určení hodnoty sumace lze užít příkaz **sum**, popř. *value*.

**Sum(f(i),i=a..b)**, kde  $i$  je index,  $a, b$  jsou meze.

Příkaz přímo nevyčíslí hodnotu součtu. Výraz uvnitř sumace je někdy výhodné nejprve upravit, např. pomocí *expand*, *factor*, *combine*, popř. *changevar*.

> **Sum(k\*x,k=1..n);**

$$\sum_{k=1}^n kx$$

> **sum(k\*x,k=1..n);**

**# určení hodnoty**

$$\frac{x(n+1)^2}{2} - \frac{(n+1)x}{2}$$

> **expand(%);** # ukázka možného použití úprav

$$x \left( \sum_{k=1}^n k \right)$$

Knihovna *student* obsahuje také příkazy pro numerickou aproximaci integrálu.

## simpson

slouží k aproximaci integrálu pomocí Simpsonovy formule

**simpson(f(x),x=a..b,n)**, kde  $f(x)$  je funkcí proměnné  $x$ ,  $a, b$  jsou meze integrace a  $n$  je počet použitých intervalů (nepovinný, implicitně roven 4).

> **simpson(f(x),x=a..b,n);** # výpočetní vzorec

$$\frac{1}{3}(b-a) \left( a^2 - 2a - 10 + b^2 - 2b \right. \\ \left. + 4 \left( \sum_{i=1}^{\frac{n}{2}} \left( \left( a + \frac{(2i-1)(b-a)}{n} \right)^2 - 2a - \frac{2(2i-1)(b-a)}{n} - 5 \right) \right) \right. \\ \left. + 2 \left( \sum_{i=1}^{\frac{n}{2}-1} \left( \left( a + \frac{2(b-a)i}{n} \right)^2 - 2a - \frac{4(b-a)i}{n} - 5 \right) \right) \right) / n$$

> **simpson(x,x=1..3,12);** # numerická aproximace integrálu

$$\frac{2}{9} + \frac{2}{9} \left( \sum_{i=1}^6 \left( \frac{5}{6} + \frac{i}{3} \right) \right) + \frac{1}{9} \left( \sum_{i=1}^5 \left( 1 + \frac{i}{3} \right) \right)$$

> **value(%);** # hodnota této aproximace

4

> **int(x,x=1..3);** # hodnota integrálu vypočtená přímo

4

>

>

## trapezoid

slouží k aproximaci integrálu pomocí lichoběžníkové metody.

**trapezoid(f(x),x=a..b,n)**, kde  $f$  je funkce proměnné  $x$ ,  $a, b$  jsou meze integrace a  $n$  je počet lichoběžníků (nepovinný parametr, implicitně 4).

> **trapezoid(f(x),x=a..b,n);**

$$\frac{1}{2} \frac{(b-a) \left( a^2 - 2a - 10 + 2 \left( \sum_{i=1}^{n-1} \left( \left( a + \frac{(b-a)i}{n} \right)^2 - 2a - \frac{2(b-a)i}{n} - 5 \right) \right) + b^2 - 2b \right)}{n}$$

> **trapezoid(x,x=1..3,12);**

$$\frac{1}{3} + \frac{1}{6} \left( \sum_{i=1}^{11} \left( 1 + \frac{i}{6} \right) \right)$$

> **value(%)** ;

4

>



## Příloha 2 : Knihovna *VectorCalculus*

System Maple obsahuje rozšiřující knihovnu pro práci s vektory. Jde o tzv. "*VectorCalculus package*". Knihovna obsahuje přes 40 příkazů a několik typů souřadnicových systémů. Podobně jako knihovnu *student*, je nutné nahrát ji do paměti pomocí příkazu

```
> with(VectorCalculus) :
```

```
Warning, the assigned names <, > and <|> now have a global binding
```

```
Warning, these protected names have been redefined and unprotected: *, +, .., Vector, diff, int, limit, series
```

popř. pro získání pouze jednoho příkazu (např. *LineInt*)

```
> with(VectorCalculus, LineInt) :
```

Další možností je volání příkazů pomocí předpisu

```
jmeno_fce[package](parametry_fce)
```

V tomto odstavci se zaměříme pouze na příkazy, které jsou užitečné při výpočtech křivkových a plošných integrálů.

```
>
```

### *Souřadnicové systémy definované ve VectorCalculus*

Tato knihovna obsahuje následující souřadnicové systémy:

- **2D** - bipolar, cardioid, cassinian, cartesian, elliptic, hyperbolic, invcassinian, logarithmic, logcosh, parabolic, polar, rose, tangent
- **3D** - bipolarcylindrical, bispherical, cardioidal, cardioidecylindrical, cartesian, casscylindrical, conical, cylindrical, elleycylindrical, hypercylindrical, invcasscylindrical, invproospheroidal, logcoshcylindrical, logcylindrical, oblatespheroidal, paraboloidal, paracylindrical, prolatespheroidal, rosecylindrical, sixsphere, spherical, tangencylindrical, tangentsphere, toroidal

Uvedme několik převodů jednotlivých souřadnicových systémů do kartézského

- ve 2D jde o zobrazení  $(u, v) \rightarrow (x, y)$   
kartézská:  $x = u$   
 $y = v$   
eliptická:  $x = \cosh(u) \cdot \cos(v)$   
 $y = \sinh(u) \cdot \sin(v)$   
polární:  $x = u \cdot \cos(v)$   
 $y = u \cdot \sin(v)$
- ve 3D jde o zobrazení  $(u, v, w) \rightarrow (x, y, z)$   
kartézská:  $x = u$   
 $y = v$   
 $z = w$   
sférická:  $x = u \cdot \cos(v) \cdot \sin(w)$   
 $y = u \cdot \sin(v) \cdot \sin(w)$   
 $z = u \cdot \cos(w)$

```
>
```

### *SetCoordinates*

Příkaz, pomocí kterého se mění systém souřadnic.

Syntaxe:

***SetCoordinates(v,c)*** , kde  $v$  je vektor,  $c$  je jméno příslušného systému souřadnic. Pokud není  $v$  uveden, je nastaven nový souřadnicový systém pro všechny další příkazy, jinak se nový souřadnicový systém týká pouze uvedeného vektoru.

**Příklady:**

```
> restart:with(VectorCalculus):
```

```
Warning, the assigned names <,> and <|> now have a global binding
```

```
Warning, these protected names have been redefined and unprotected: *, +, .., Vector, diff, int, limit, series
```

```
Nahrajeme knihovnu do paměti
```

```
> v := <a,b>;
```

$$v := a \mathbf{e}_x + b \mathbf{e}_y$$

```
Zapišeme vektor
```

```
> SetCoordinates(v, 'polar');
```

$$a \mathbf{e}_r + b \mathbf{e}_\theta$$

```
Změníme souřadnicový systém na polární
```

```
> GetCoordinates(v);
```

*polar*

```
Pomocí tohoto příkazu zjišťujeme, který systém souřadnic je nastaven (viz. následující odstavec)
```

```
> SetCoordinates('spherical' [r,phi,theta]);
```

*spherical*<sub>r,φ,θ</sub>

```
Zde není specifikován vektor, proto se jedná o nastavení implicitního souřadnicového systému na sférický
```

```
> v := VectorField(<r,0,0>);
```

$$v := r \mathbf{e}_r$$

```
Vytvoření vektoru pomocí VectorField. Více viz. příkaz VectorField
```

```
> GetCoordinates(v);
```

*spherical*<sub>r,φ,θ</sub>

```
Zjištění aktuálního souřadnicového systému
```

```
>
```

```
>
```

***GetCoordinates***

Vypíše aktuální systém souřadnic.

Syntaxe:

***GetCoordinates(v)*** , kde  $v$  je vektor. V případě, kdy je funkce zavolána bez parametrů, je vypsán stávající souřadnicový systém.

**Příklady:**

```
> restart:with(VectorCalculus):
```

```
Warning, the assigned names <,> and <|> now have a global binding
```

```
Warning, these protected names have been redefined and unprotected: *, +, .., Vector, diff, int, limit, series
```

```
> GetCoordinates () ;
```

*cartesian*

```
Určení typu souřadnicového systému, který je právě používán
```

```
> v := VectorField(<r,0,0>, 'spherical'[r,phi,theta]);
```

$v := r \mathbf{e}_r$

```
Nový vektor, který je zadán ve sférických souřadnicích
```

```
> GetCoordinates (v);
```

*spherical*<sub>r, φ, θ</sub>

```
Kontrola, že jde opravdu o sférický systém souřadnic - platné jen pro daný vektor.
```

```
> GetCoordinates ();
```

*cartesian*

```
Implicitní je stále kartézská soustava souřadnic
```

```
> SetCoordinates ('polar');
```

*polar*

```
Změníme implicitní soustavu souřadnic na polární.
```

```
> GetCoordinates (<a,b>);
```

*polar*

```
Pokud je vektor vytvořen bez dalších parametrů, je zadán právě v aktuální soustavě souřadnic, tj. polární.
```

```
>
```

### AddCoordinates

Tento příkaz je určen k vytvoření vlastních souřadnicových systémů.

Syntaxe:

*AddCoordinates(system[parametr1,...parametrN], rce, prepis)* , kde parametr

*system* je názvem nového systému, druhý parametr *rce* zastupuje seznam rovnic, které tento systém definují, a třetí nepovinný parametr určuje, zda nový systém přepíše už existující systém (pokud mají stejný název). Před tvorbou nových souřadnicových systémů je vhodné nejprve určit vlastnosti nových parametrů systému pomocí příkazu *assume*.

### Příklady:

```
> restart:with(VectorCalculus):
```

```
Warning, the assigned names <,> and <|> now have a global binding
```

```
Warning, these protected names have been redefined and unprotected: *, +, .., Vector, diff, int, limit, series
```

```
> assume (r>0, 0<=theta, theta<2*Pi);
AddCoordinates ('m_polar' [r,theta],
[r*cos(theta), r*sin(theta)]);
```

*m\_polar*

```
Definice vlastního souřadnicového systému s názvem m_polar. Parametry jsou: kladné číslo r a nezáporné číslo theta, které je shora omezeno konstantou 2π .
```

```
> Laplacian (f(r,theta), 'm_polar' [r,theta]);
```

$$\frac{\left(\frac{\partial}{\partial r} f(r, \theta)\right) + r \left(\frac{\partial^2}{\partial r^2} f(r, \theta)\right) + \frac{\frac{\partial^2}{\partial \theta^2} f(r, \theta)}{r}}{r}$$

# Spočte laplacián dané funkce

> **assume (u>=0, v>=0) ;**

**AddCoordinates ( 'foo' [u,v] , [u^2+v^2, u^2-v^2] );**

*foo*

# Vytvoření dalšího souřadnicového systému

> **F := VectorField ( <f(u,v), g(u,v)>, 'foo' [u,v] );**

$$F := f(u, v) \bar{\mathbf{e}}_u + g(u, v) \bar{\mathbf{e}}_v$$

# Vektor zapsaný v příslušném systému "foo".

> **Divergence ( F );**

$$\frac{1}{8} \frac{2v \left(\frac{\partial}{\partial u} f(u, v)\right) \sqrt{2} + 2u \left(\frac{\partial}{\partial v} g(u, v)\right) \sqrt{2}}{u-v}$$

# Spočtení divergence výše uvedeného vektoru funkcí F.

>

### MapToBasis

Je příkaz určený pro převod vektorů mezi jednotlivými souřadnicovými soustavami.

Syntaxe:

**MapToBasis(V, c)** , kde *V* je vektor nebo vektorová funkce, kterou chceme převést, *c* je název souřadnicového systému, který je zadán následujícím předpisem *nazev[parametr1, parametr2,...]*. Pokud není zadán druhý parametr, je vektor převeden do tvaru právě užívaného systému souřadnic.

### Příklady:

> **restart:with(VectorCalculus) :**

Warning, the assigned names <, > and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: \*, +, .., Vector, diff, int, limit, series

> **GetCoordinates () ;**

*cartesian*

# Zjistíme aktuální souřadnicový systém

> **MapToBasis ( <1,1>, 'polar' );**

$$\sqrt{2} \mathbf{e}_r + \frac{\pi}{4} \mathbf{e}_\theta$$

# Převedeme vektor z kartézských souřadnic na polární

> **MapToBasis ( %, 'cartesian' );**

$$\mathbf{e}_x + \mathbf{e}_y$$

# Zpětný převod, tj. kontrola výsledku

> **v := <r, theta>;**

$$v := r \mathbf{e}_x + \theta \mathbf{e}_y$$

# Nový vektor

> **SetCoordinates( v, 'polar' );**

$$r \mathbf{e}_r + \theta \mathbf{e}_\theta$$

# Vektor je vyjádřen v polární soustavě souřadnic

> **MapToBasis( v );**

$$r \cos(\theta) \mathbf{e}_x + r \sin(\theta) \mathbf{e}_y$$

# Převod vektoru do právě užívaného systému souřadnicového, tj. systém kartézský,

> **SetCoordinates( 'spherical' [r, phi, theta] );**

$$spherical_{r, \phi, \theta}$$

# Změna souřadnicového systému na sférický

> **v := VectorField( <r, 0, 0> );**

$$v := r \mathbf{e}_r$$

# Vektor vyjádřený ve sférickém systému souřadnic

> **MapToBasis( v, 'cartesian' [x, y, z] );**

$$x \mathbf{e}_x + y \mathbf{e}_y + z \mathbf{e}_z$$

# Převod do kartézského systému souřadnic

>

### VectorField

Definuje vektorové pole v daném souřadnicovém systému.

Syntaxe:

**VectorField(v,c)**, kde  $v$  je vektor a  $c$  specifikuje souřadnicový systém, ve

kterém je vektor zadán. Druhý parametr je nepovinný, pokud není uveden, je vytvořeno vektorové pole nad stávajícím souřadnicovým systémem. Pokud chceme vypočítat hodnotu pole v daném bodě, použijeme příkaz *evalVF*. Chceme-li zjistit, ve které soustavě souřadnic je vektor zadán, použijeme příkaz *attributes(objekt)*, který obecně vypíše typ objektu a některé jeho vlastnosti.

### Příklady:

> **restart:with(VectorCalculus):**

Warning, the assigned names <, > and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: \*, +, .., Vector, diff, int, limit, series

> **v := VectorField( <x, y, z>, 'cartesian' [x, y, z] );**

$$v := x \mathbf{e}_x + y \mathbf{e}_y + z \mathbf{e}_z$$

# Vektorové pole v kartézské soustavě souřadnic

> **attributes( v );**

*vectorfield, coords = cartesian<sub>x, y, z</sub>*

# Zjištění vlastností objektu v

> **v := VectorField( <1/r^2, 0, 0>, 'spherical' [r, phi, theta] );**

$$\mathbf{v} := \frac{1}{r^2} \mathbf{e}_r$$

# Vektorové pole zadané ve sférických souřadnicích

> **attributes( v );**

*vectorfield, coords = spherical<sub>r, φ, θ</sub>*

> **simplify(evalVF(v, <1, 0, Pi>));**

$$\frac{1}{1 + \pi^2} \mathbf{e}_r + \arctan\left(\frac{1}{\pi}\right) \mathbf{e}_\phi$$

# Vyčíslení hodnoty vektorového pole v daném bodě, pomocí *simplify* bylo provedeno pouze zjednodušení. Vektor *v* je ve sférických souřadnicích a bod je zadán v kartézské soustavě souřadnicové.

> **SetCoordinates( 'cylindrical' [r, theta, z] );**

*cylindrical<sub>r, θ, z</sub>*

# Změna pracovních souřadnic na cylindrické

> **v := VectorField( <r\*theta, theta, z^2> );**

$$\mathbf{v} := r \theta \mathbf{e}_r + \theta \mathbf{e}_\theta + z^2 \mathbf{e}_z$$

# Vektorové pole zadané v těchto souřadnicích

> **attributes( v );**

*vectorfield, coords = cylindrical<sub>r, θ, z</sub>*

# Zjištění vlastností

>

## int

Výpočet integrálu pomocí tohoto příkazu předpokládá vyjádření oblasti pomocí předem předepsaných forem, jinak se tento postup nedá užít.

Syntaxe:

**int(fce, dom)**, kde *fce* je výraz, který představuje integrand, a parametr *dom* určuje oblast integrace.

Oblast je vyjádřena pomocí jedné z následujících možností: *Circle(cen, rad)*, *Ellipse(eqn)*, *Parallelepiped(r1, [r2, r3, ..])*, *Rectangle(r1, r2)*, *Region(r1, [r2, r3, ...])*, *Sector(obj, start, finish)*, *Sphere(cen, rad)*, *Tetrahedron(v1, v2, v3, v4)*, *Triangle(v1, v2, v3)*. Více podrobností o jednotlivých vyjádřeních naleznete v nápovědě, například pomocí příkazu **?VectorCalculus[int]**.

## Příklady:

> **restart:with(VectorCalculus):**

Warning, the assigned names <, > and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: \*, +, .., Vector, diff, int, limit, series

> **int( x^2+y^2, [x,y] = Circle( <0,0>, r ) );**

$$\frac{\pi r^4}{2}$$

# Počítáme dvojný integrál přes kruh o poloměru *r*.

> **int( exp(-x^2-y^2-z^2), [x,y,z] = Sphere( <0,0,0>, r ) );**

$$\frac{\pi (-2r + \sqrt{\pi} \operatorname{erf}(r) e^{(r^2)})}{e^{(r^2)}}$$

# Trojný integrál přes sféru o poloměru  $r$ .

> `int( x*y, [x,y] = Triangle( <0,0>, <1,0>, <0,1> ) );`

$$\frac{1}{24}$$

# Dvojný integrál přes trojúhelník

> `int( sin(x)/x, [x,y] = Rectangle( 0..Pi/2, 0..Pi/2 ) );`

$$\frac{1}{2} \operatorname{Si}\left(\frac{\pi}{2}\right) \pi$$

# Oblastí je obdélník

> `int( 1, [x,y] = Region( 0..1, x^2..x ) );`

$$\frac{1}{6}$$

# Oblast je ohraničena křivkami  $y=x^2$  a  $y=x$ .

> `int( 1, [x,y] = Ellipse( x^2/4 + y^2/9 - 1 ) );`

$$6\pi$$

# Oblastí je elipsa

> `int( y^2, [x,y] = Sector( Circle( <0,0>, r ), 0, Pi ) );`

$$\frac{\pi r^4}{8}$$

# Integrál přes půlkružnici

> `int( x, [x,y] = Sector( Ellipse( x^2/4 + y^2/9 - 1 ), 0, Pi/2 ) );`

$$4$$

# Integrál přes čtvrtinu elipsy

> `int( <t,t^2,t^3>, t );`

$$\frac{t^2}{2} \mathbf{e}_x + \frac{t^3}{3} \mathbf{e}_y + \frac{t^4}{4} \mathbf{e}_z$$

# Integrace složek vektorového pole

>

>

### **PathInt**

Příkaz pro výpočet křivkového integrálu I. druhu. Více informací naleznete v kapitole Křivkový integrál I. druhu.

Syntaxe:

***PathInt(fce, dom)***, kde *fce* představuje integrand a *dom* je řídící křivka vyjádřená pomocí jedné z následujících možností: *Arc(obj, start, finish)*, *Circle(cen, rad)*, *Ellipse(eqcn)*, *Line(p1, p2)*, *LineSegments(p1, p2, [p3, p4, ...])*, *Path(v, rng, [c])*. Více informací o možných způsobech určení řídící křivky naleznete v nápovědě.

### **Příklady:**

> `restart:with(VectorCalculus):`

Warning, the assigned names <, > and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: \*, +, .., Vector, diff, int, limit, series

```
> PathInt(x^2, [x,y] = Line(<0,0>, <1,1>));
```

$$\frac{\sqrt{2}}{3}$$

# Křivkový integrál podél úsečky

```
> PathInt(y^2, [x,y] = LineSegments(<0,0>, <1,1>, <1,2>));
```

$$\frac{\sqrt{2}}{3} + \frac{7}{3}$$

# Křivkou je lomená čára zadaná třemi body

```
> PathInt(y, [x,y] = Path(<cos(t), sin(t)>, t=0..Pi));
```

$$2$$

# Křivkou je kladná část jednotkové kružnice

```
> PathInt(y, [x,y] = Path(<1,t>, 0..Pi, 'coords'='polar'));
```

$$2$$

# Jiný zápis pro tentýž integrál - je zavedena polární soustava souřadnic

```
> PathInt(x^2+y^2, [x,y] = Circle(<0,0>, 3));
```

$$54 \pi$$

# Křivkou je kružnice o poloměru 3

```
> PathInt(1, [x,y] = Ellipse(x^2/4+y^2/9-1));
```

$$12 \operatorname{EllipticE}\left(\frac{\sqrt{5}}{3}\right)$$

# Křivkou je elipsa o zadané rovnici. Jedná se o eliptický integrál.

```
> PathInt(x, [x,y] = Arc(Circle(<0,0>, 1), 0, Pi/2));
```

$$1$$

# Integrál počítán po čtvrtkružnici, která je určena pomocí *Arc*.

```
>
```

```
>
```

## **LineInt**

Příkaz pro výpočet křivkového integrálu II. druhu. Více informací naleznete v kapitole Křivkový integrál II. druhu.

Syntaxe:

**LineInt**(*F*, *dom*) , kde *F* je vektor v dané soustavě souřadnic, *dom* je jednou z následujících možností, pomocí kterých definujeme křivku: *Arc*(*obj*, *start*, *finish*), *Circle*(*cen*, *rad*), *Ellipse*(*eqn*), *Line*(*p1*, *p2*), *LineSegments*(*p1*, *p2*, [*p3*, *p4*, ...]), *Path*(*v*, *rng*, [*c*]).

## **Příklady:**

```
> restart:with(VectorCalculus):
```

Warning, the assigned names <, > and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: \*, +, .., Vector, diff, int, limit, series

```
> SetCoordinates(cartesian[x,y]);
```



*cartesian*<sub>x,y</sub>

# Nejprve je nutno určit soustavu souřadnicovou, ve které budeme pracovat

```
> LineInt(VectorField(<x,y>), Line(<1,2>, <3,-4>));
```

10

# Počítáme křivkový integrál II. druhu po úsečce dané dvěma body

```
> LineInt(VectorField(<x,y>), LineSegments(<0,0>, <1,1>, <1,-1>));
```

1

# Integrál po lomené čáře.

```
> LineInt(VectorField(<x^2,y^2>), Path(<t,t^2>, t=0..2));
```

24

```
> LineInt(VectorField(<y,-x>), Circle(<0,0>, r));
```

$-2r^2\pi$

# Kružnice o poloměru  $r$ .

```
> LineInt(VectorField(<y,-x>), Ellipse(x^2/4+y^2/9-1));
```

$-12\pi$

```
> LineInt(VectorField(<y,-x>), Arc(Circle(<0,0>, r), 0, Pi));
```

$-r^2\pi$

```
> LineInt(VectorField(<y,-x>), Arc(Ellipse(x^2/4+y^2/9-1), 0, Pi/2));
```

$-3\pi$

>

>

## Flux

Příkaz pro výpočet plošného integrálu II. druhu. Více o teorii a několik ukázek naleznete v kapitole Plošné integrály.

Syntaxe:

**Flux**(*fce*, *dom*) , kde *fce* je vektor a *dom* je jedna z následujících možností, které definují požadovanou plochu: *Box*( $r_1, r_2, r_3, [dir]$ ), *Sphere*(*cen*, *rad*, [*dir*]), *Surface*(*v*). Více informací naleznete v helpu.

## Příklady:

```
> restart:with(VectorCalculus):
```

Warning, the assigned names <, > and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: \*, +, .., Vector, diff, int, limit, series

```
> Flux(VectorField(<x,y,z>, cartesian[x,y,z]), Surface(<r,s,t>, s=0..Pi, t=0..2*Pi, coords=spherical));
```

$4r^3\pi$

# Integrál přes kouli definovanou ve sférických souřadnicích.

```
> Flux(VectorField(<x,y,z>, cartesian[x,y,z]), Sphere(<0,0,0>, r));
```

$$4r^3\pi$$

# Jiný možný způsob zápisu.

```
> Flux(VectorField(<y,-x,0>, cartesian[x,y,z]),
 Box(1..2, 3..4, 5..6));
```

0

# Integrál přes kvádr

```
>
```

### ScalarPotential

Příkaz, který slouží ke stanovení potenciálu. Více naleznete v kapitole Křivkový integrál II. druhu.

Syntaxe:

**ScalarPotential(v)**, kde  $v$  je vektorové pole zadané pomocí funkce *VectorField*. V případě, že potenciál neexistuje, příkaz vrací NULL. Připomínáme, že potenciál vektorového pole  $v$  je skalární funkce  $f$  taková, že  $\text{grad } f = v$ .

### Příklady:

```
> restart:with(VectorCalculus):
```

Warning, the assigned names <,> and <|> now have a global binding

Warning, these protected names have been redefined and unprotected: \*, +, .., Vector, diff, int, limit, series

```
> SetCoordinates('cartesian'[x,y,z]);
```

*cartesian*<sub>x,y,z</sub>

# Nastavení "pracovní" soustavy souřadnic.

```
> v := VectorField(<x,y,z>);
```

$$v := x \mathbf{e}_x + y \mathbf{e}_y + z \mathbf{e}_z$$

# Vytvoření vektorového pole

```
> ScalarPotential(v);
```

$$\frac{x^2}{2} + \frac{y^2}{2} + \frac{z^2}{2}$$

# Výpočet potenciálu

```
> v := VectorField(<y,-x,0>);
```

$$v := y \mathbf{e}_x - x \mathbf{e}_y$$

# Jiné vektorové pole

```
> ScalarPotential(v);
```

# Potenciál vektorového pole neexistuje, proto systém Maple nic nevypíše

```
> SetCoordinates('spherical'[r,phi,theta]);
```

*spherical*<sub>r,φ,θ</sub>

# Zavedení nového souřadnicového systému, jde o sférický systém

```
> v := VectorField(<r,0,0>);
```

$$v := r \mathbf{e}_r$$

# Vytvoření vektorového pole.

```
> ScalarPotential(v);
```

$$\frac{r^2}{2}$$

```
Vypočítali jsme potenciál
> Gradient(%);
```

$$-\frac{r}{r^2}$$

```
Pro kontrolu jsme vypočítali gradient.
>
>
```

### SurfaceInt

Příkaz je určen pro výpočet plošného integrálu II. druhu. Více o tomto druhu integrálu najdete v kapitole Plošné integrály.

Syntaxe:

**Surface(fce, dom)**, kde *fce* je integrand a *dom* určuje plochu, přes kterou se bude integrovat. Opět je možné tuto plochu zadat pouze pomocí následujících možností: *Box(r1, r2, r3)*, *Sphere(cen, rad)*, *Surface(v)*. Více o těchto možnostech naleznete v nápovědě.

### Příklady:

```
> restart:with(VectorCalculus):
```

```
Warning, the assigned names <, > and <|> now have a global binding
```

```
Warning, these protected names have been redefined and unprotected: *, +, ..,
Vector, diff, int, limit, series
```

```
> SurfaceInt(1, [x,y,z] = Surface(<r,s,t>, s=0..Pi,
t=0..2*Pi, coords=spherical)) assuming r>0;
```

$$4 \pi r^2$$

# Plošný integrál I. druhu přes sféru danou ve sférických souřadnicích. Parametr *assuming* určuje vlastnosti na *r*.

```
> SurfaceInt(x+y+z, [x,y,z] = Surface(<s,t,4-2*s-t>,
[s,t] = Triangle(<0,0>,<1,0>,<0,1>)));
```

$$\frac{11\sqrt{6}}{6}$$

# Ukázka použití konstrukce *Surface* spolu s definicí trojúhelníku pomocí *Triangle*.

```
> SurfaceInt(y^2, [x,y,z] = Sphere(<0,0,0>, r));
```

$$\frac{4 r^4 \pi}{3}$$

# Určení sféry pomocí jejího středu a poloměru

```
> SurfaceInt(x*y*z, [x,y,z] = Box(1..2, 3..4, 5..6));
```

$$\frac{693}{4}$$

# Určení plochy pomocí stran kváдру.

```
>
>
```

## Příloha 3 : Programování - jazyk MPJ

Pro vytváření vlastních procedur v systému Maple je vytvořen speciální jazyk. Tato kapitola není úplným referenčním textem, je jen základem pro pochopení struktur vlastních procedur. Při programování se velmi často používají cykly a podmínky, a proto i systém Maple tyto řídicí struktury obsahuje. Zopakujme syntaxi pro vytvoření vlastní funkce (procedury).

Struktura vypadá následovně

```
název_fce := proc(param1,param2,...)
 tělo funkce
end proc;
```

Nejprve zvolíme *název funkce* a pak pomocí klíčového slova **proc** vytvoříme funkci s parametry, které jsou uzavřeny do kulatých závorek. Dále následuje *tělo funkce*, ve kterém se většinou vyhodnotí zadané parametry a jejich typy, provedou se příslušné příkazy a konstrukce se ukončí pomocí *end proc*.

Nyní se podíváme na řízení toku programu pomocí struktury **if-then(-elif-then)-else-end if**. Přesná syntaxe je

```
if podmínka1 then příkazy1 else příkazy2 end if
```

Následující obrázek ukazuje, jak tato struktura pracuje.

| <b>if podmínka ?</b> |             |
|----------------------|-------------|
| <b>+</b>             | <b>-</b>    |
| <b>then</b>          | <b>else</b> |
| příkazy 1            | příkazy 2   |
| <b>end if</b>        |             |

Je-li podmínka za *if* splněna (tj. + ), jsou provedeny příkazy uvedené za *then*. Pokud splněna není ( tj. - ), provedou se příkazy uvedené za *else*. V obou případech se hned po provedení příkazů pokračuje v programu. Existuje ještě jiná možnost zápisu podmínky s *if*, a to

```
if podmínka1 then příkazy1 elif podmínka2 then příkazy2 else příkazy3 end if
```

Následující obrázek ukazuje, jak tato struktura pracuje.

|                        |                          |                          |
|------------------------|--------------------------|--------------------------|
| <b>if</b> podmínka 1 ? |                          |                          |
| <b>+</b>               | <b>-</b>                 |                          |
| <b>then</b>            | <b>elif</b> podmínka 2 ? |                          |
|                        | <b>+</b>                 | <b>-</b>                 |
|                        | <b>then</b><br>příkazy 2 | <b>else</b><br>příkazy 3 |
| příkazy 1              |                          |                          |
| <b>end if</b>          |                          |                          |

Nyní si na příkladu hledání maxima ze dvou zadaných čísel ukažme použití těchto dvou struktur. Z pedagogických důvodů budeme požadovat i informaci o případné rovnosti čísel. Příklad později rozšíříme na hledání maxima z libovolného počtu zadaných čísel, a to pomocí cyklů typu *for-do* nebo *while-do*.

```
> max1:=proc(a,b) # užití if-else-end if
 if a<>b then
 if a>b then
 return a;
 else
 return b;
 end if;
 else
 return a=b;
 end if;
end proc:
> max1(5,-6);
max1(-5,6);
max1(-6,-6);
```

5

6

-6 = -6

Výše uvedenou proceduru lze samozřejmě zapsat i bez použití *else*, ale jako ukázka je vhodnější výše uvedený kód. Nyní s výhodou uijeme struktury s *elif*.

```
> max2:=proc(a,b) # užití if-elif-else-end if
 if a>b then return a;
 elif a<b then return b;
 else return a=b;
 fi;
end proc:
> max2(5,-6);
```

```
max2 (-5, 6) ;
max2 (-6, -6) ;
```

```
5
6
-6 = -6
```

V předchozím zdrojovém kódu si hlavně všimněte předposledního řádku, kde je uvedena alternativa ukončení k *end if*, tj. *fi*.

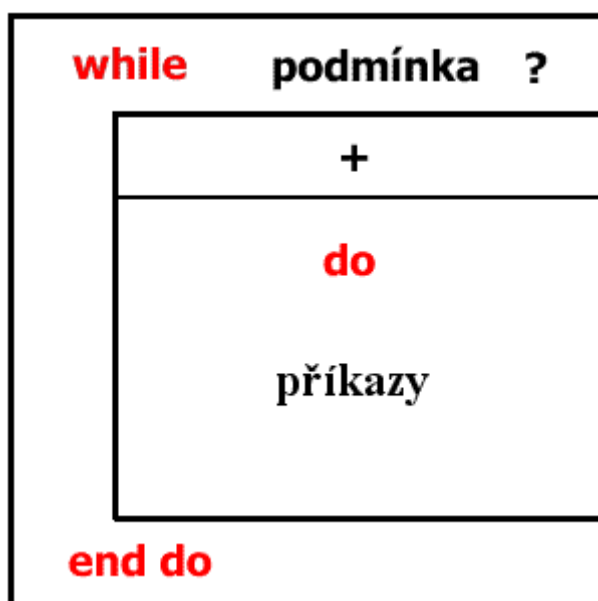
Nyní přistupme k dalším strukturám, půjde o cykly.

Začneme s cyklem **while-do-end do**. Přesnější syntaxe je

```
while podmínka do příkazy end do
```

Jde o cyklus s podmínkou, je-li pravdivá, jsou provedeny příkazy za klíčovým slovem *do*.

Následující obrázek ukazuje, jak tento cyklus pracuje.



Vraťme se k proceduře hledání maxima. Nyní ji změníme pro libovolný počet zadaných parametrů s využitím cyklu *while-do*.

```
> max3:=proc ()
 local i,n,max;
 n:=nargs;
 fce max3
 max:=args[1];
 parametr
 i:=2;
 druhého parametru
 while i<=n do
 if args[i]>max then
 max:=args[i];
 přepíšeme max
 end if;
 i:=i+1;
 # podmínka while-do-end do
 # definice lokálních proměnných
 # počet vložených parametrů do
 # jako maximum nastavíme 1.
 # začneme porovnávat až od
 # je-li parametr větší než max,
 # přejdeme k následujícímu
```

```

parametru
 end do;
 return max; # vypíšeme maximum
end proc:
> max3(5,-6);
max3(1);
max3(0,-65,76);
max3(0,-6,76,-98.25,78,78);

5
1
76
78

```

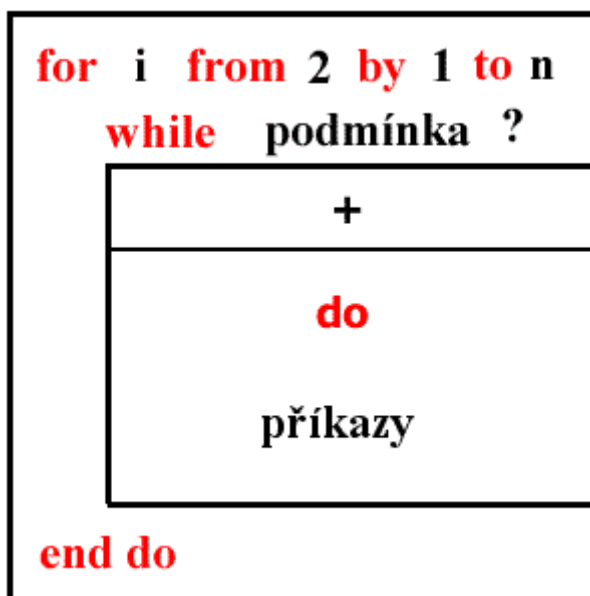
Přistupme k vysvětlení předchozí procedury. Pokud potřebujeme nějaké vlastní (lokální) proměnné, které budou pouze v těle procedury, je nutné užít klíčového slova *local*. V případě, že tyto proměnné nejsou takto předem deklarovány, Maple ohlásí varování. Podíváme-li se znovu na předchozí zdrojový kód, zjistíme, že nejsou zadány žádné parametry. To je ale omyl. Procedura počítá minimálně s jedním, a to při inicializaci proměnné *max*. K zadaným parametrům se v Maplu přistupuje pomocí pole *args[i]* a celkový počet zadaných parametrů je obsažen v *nargs*. Vraťme se k předchozímu kódu. Dále následuje inicializace maxima prvním zadaným parametrem a pak pomocí cyklu *while-do* procházíme a porovnáváme zbylé parametry s hodnotou uloženou v *max*. Nakonec je maximum vypísáno.

Pokud znáte jiný programovací jazyk, např. jazyk C, pravděpodobně byste v tomto případě použili jiný cyklus a to cyklus *for*. I zde je to možné, protože i Maple nabízí tento často užívaný cyklus. Podívejme se blíže na jeho dvě možné syntaxe.

*for* proměnná *from* odkud *by* krok *to* kam *while* podmínka *do* příkazy *end do*  
nebo

*for* proměnná *in* struktura *while* podmínka *do* příkazy *end do*

V prvním příkazu se proměnná inicializuje hodnotou uvedenou za klíčovým slovem *from*, zvyšovat se bude o hodnotu *kroku* dokud nedosáhne hodnoty *kam*. Dále se vyhodnotí podmínka příkazu *while* a pokud je splněna, jsou provedeny *příkazy*. Následující obrázek vše vysvětlí lépe.



Ukažme si to na dalším možném zápisu naší známé procedury hledání maxima.

```
> max4:=proc ()
 local i,max; # definice lokálních
 proměnných
 max:=args[1]; # jako maximum nastavíme 1.
 parametr
 for i from 2 to nargs do # začneme porovnávat až od
 druhého parametru
 if args[i]>max then # je-li parametr větší než
 max, přepíšeme max
 max:=args[i];
 end if;
 end do;
 return max; # vypíšeme maximum
end proc:
> max4 (5, -6);
```

5

```
> max4 ();
 # procedura zde hlásí chybu, protože se snaží přistupovat k
 prvnímu prvku pole
 argumentů, ale tento prvek neexistuje
```

Error, (in max4) invalid subscript selector

```
> max4 (0, -65, 76);
> max4 (0, -6, 76, -98.25, 78, 78);
```

76

78

Nyní se podívejme na cyklus *for-in-do*. Mějme seznam čísel, které chceme postupně sčítat, dokud nedosáhneme určité hodnoty.

```
> cisla:=[1,4,6,7,12,15,20]: mmax:=0:
> for z in cisla while mmax<=30 do
 mmax:=mmax+z;
end do:
> mmax;
```

45

Výše uvedený zdrojový kód je snáze čitelný. Všimněme si, že místo proměnné *max* je užitá proměnná *mmax*, a to proto, že *max* je klíčovým slovem systému Maple. V předchozích zdrojových kódech mohla být proměnná *max* používána bez nebezpečí, protože byla deklarována jako lokální, pomocí *local*.

```
>
>
```



## Literatura

- [1] Ženíšek A. : Křivkový a plošný integrál, PC-DIR Real, s.r.o. Brno, 1999
- [2] Nedoma J. : Matematika I, část 1., 2., 3., PC-DIR (1998,1999,2000)
- [3] Nedoma J. : Matematika I, Akademické nakladatelství CERM, s.r.o. Brno, 2001
- [4] Suchomel J., Plačková J., Knoflíček F. : Matematika II. Integrální počet, Rektorát Vysokého učení technického v Brně, 1985
- [5] Polcerová M., Polcer J. : Sběrka příkladů z matematiky II, VUTIUM, Brno, 1999
- [6] J. Karásek : Matematika II, Nakladatelství VUT Brno, 1995
- [7] Heal K.M., Hansen M.L., Rickard K.M. : Maple V. Learning Guide, Springer, 1996
- [8] Buchar J., Hřebíček J., Hřebíčková J., Slaběňáková J. : Úvod do programu Maple V, skripta VŠZL Brno, 1993
- [9] Došlá Z., Plch R., Sojka P. : Matematická analýza s programem Maple - Diferenciální počet funkcí více proměnných, Masarykova univerzita v Brně, Brno 1999
- [10] <http://www.maplesoft.com/> - stránky Waterloo Maple Inc.

>