

Kapitola 1

POJEM FUNKCE VÍCE PROMĚNNÝCH

Zde se využití počítače přímo nabízí. Tato část matematické analýzy se probírá v době, kdy nejsou probrány odpovídající partie z geometrie (zimní semestr v 2. ročníku učitelského studia). Studenti proto často postrádají geometrickou představu v prostoru, a tak jsou vizualizační schopnosti počítačových systémů velmi vítány.

Připomeňme, že v celé práci se zabýváme reálnou funkcí dvou, event. tří reálných proměnných.

1.1. Graf funkce dvou proměnných

Všimněme si podrobněji problematiky tvorby grafů reálné funkce dvou reálných proměnných pomocí programu Maple V. Zaměříme se zejména na případy, kdy počítačem získaný výstup (v dalším nazývaný PC-graf), neodpovídá grafu funkce.

Definice 1.1. Nechť f je funkce n reálných proměnných definovaná na množině $M \subseteq \mathbb{R}^n$, $n \geq 2$. *Grafem funkce f* nazýváme množinu bodů

$$G(f) = \{[x, y] \in \mathbb{R}^{n+1} : x = [x_1, \dots, x_n] \in M, y = f(x)\}.$$

Pro funkci dvou proměnných, tj. $n = 2$, je grafem funkce množina bodů v třírozměrném prostoru.

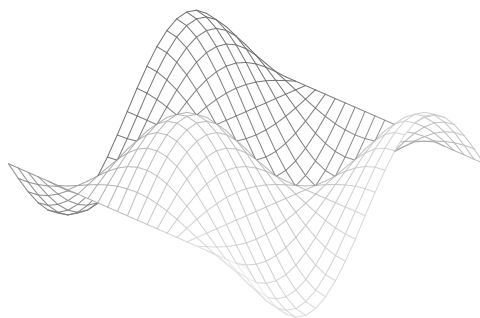
Definujme funkci $f(x, y) = \sin(x) \cos(y)$:

```
> f := (x, y) -> sin(x) * cos(y);
```

$$f := (x, y) \rightarrow \sin(x) \cos(y)$$

a sestrojme PC-graf funkce f (obr. 1.1):

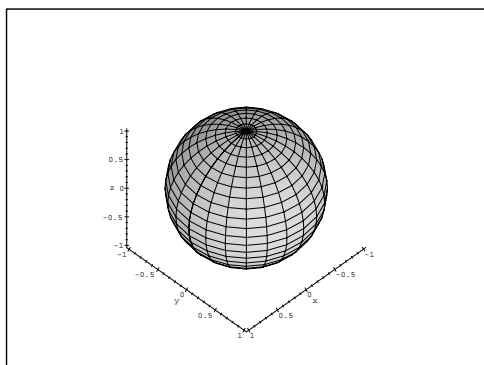
```
> plot3d(f, -Pi..Pi, -Pi..Pi);
```



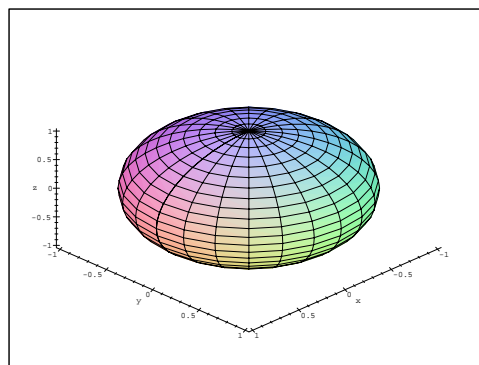
obr. 1.1

Stejně jednoduše je možno získat i PC-graf plochy dané parametricky, např. $x = \sin u \cos v$, $y = \sin u \sin v$, $z = \cos u$, $u \in [0, \pi]$, $v \in [0, 2\pi]$ (obr. 1.2):

```
> with(plots):
> plot3d([sin(u)*cos(v), sin(u)*sin(v), cos(u)], u=0..Pi, v=0..2*Pi,
> style=patch, scaling=constrained, axes=framed, labels=[x,y,z]);
```



obr. 1.2



obr. 1.3

Parametrem `scaling=constrained` jsme dosáhli stejného měřítka na osách výsledného PC-grafu. Porovnejme PC-graf na obrázku 1.2 s PC-grafem na obrázku 1.3, na kterém je tatáž koule generovaná bez nastavení parametru `scaling=constrained`:

```
> plot3d([sin(u)*cos(v), sin(u)*sin(v), cos(u)], u=0..Pi, v=0..2*Pi,
> style=patch, axes=framed, labels=[x,y,z]);
```

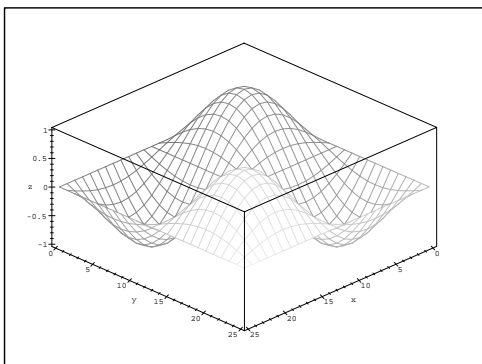
Jakým způsobem probíhá konstrukce PC-grafu? Zadáme funkční předpis a množinu bodů $[x, y]$, pro které chceme funkci zobrazit. Tato množina je typu $\langle x_{\min}, x_{\max} \rangle \times \langle y_{\min}, y_{\max} \rangle$. Na ní pak program vytvoří síť, v jejích uzlových bodech numericky spočítá funkční hodnoty (tyto jsou uloženy do objektu **PLOT3D**). Hustotu sítě regulujeme pomocí parametru `grid=[m,n]`, kde m a n udává počet uzlových bodů ve směru os x a y . Implicitní nastavení tohoto parametru je `[25,25]`. Funkční hodnoty jsou poté podle interpolačních pravidel pospojovány a PC-graf zobrazen na výstupním zařízení.

Tento postup však zřejmě může vést k zavádějícím výsledkům. Pro ilustraci např. vytvoříme PC-graf funkce $g(x, y) = \sin(2\pi x) \sin(2\pi y)$, pro x a y z intervalu $\langle 0, 25 \rangle$ beze změny implicitního nastavení parametrů:

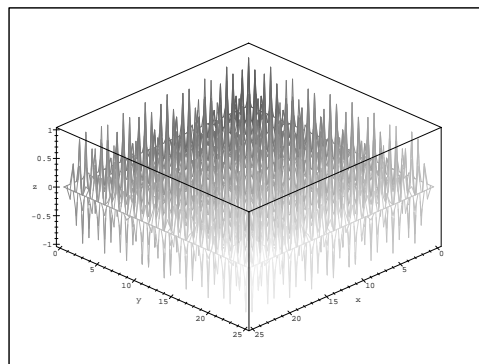
```
> plot3d(sin(2*Pi*x)*sin(2*Pi*y), x=0..25, y=0..25, axes=boxed,
> labels=[x,y,z]);
```

Podrobnější analýzou zadané funkce však zjistíme, že získaný PC-graf (obr. 1.4) neodpovídá skutečnosti, funkce $\sin(2\pi x)$ a $\sin(2\pi y)$ jsou periodické s periodou 1 a tomu PC-graf na obrázku 1.4 neodpovídá. Zhuštěním sítě dostáváme výsledek bližší skutečnému chování uvažované funkce (obr. 1.5):

```
> plot3d(sin(2*Pi*x)*sin(2*Pi*y), x=0..25, y=0..25, axes=boxed,
> grid=[60,60], labels=[x,y,z]);
```



obr. 1.4



obr. 1.5

Další problémy vznikají při tvorbě grafů nespojitých funkcí. Nejjednodušší situace nastává v případě, kdy studovaná funkce není v bodě $[x_0, y_0]$ spojitá, ale v tomto bodě existuje konečná limita. Pak můžeme buď změnit hustotu uzlových bodů nebo funkci vhodným způsobem dodefinovat.

Příklad 1.1. Vytvořte PC-graf funkce

$$f(x, y) = \frac{x^2 y}{x^2 + y^2}.$$

Příkazem:

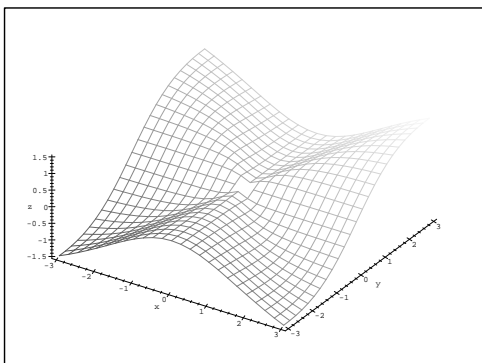
```
> f := (x, y) -> (x^2*y) / (x^2+y^2);
```

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

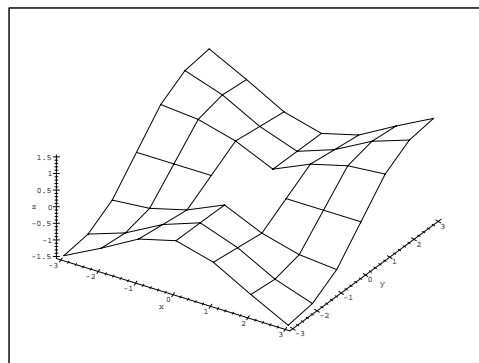
zadáme funkci a příkazem:

```
> plot3d(f, -3..3, -3..3, orientation=[-57,38], axes=framed, labels=[x,y,z]);
```

získáme PC-graf zobrazený na obr. 1.6.



obr. 1.6



obr. 1.7

Bod, ve kterém vyšetřovaná funkce není spojitá, je při této hustotě sítě totožný s uzlovým bodem a program v něm nemůže spočítat funkční hodnotu. Při zobrazování na výstupním zařízení je funkční hodnota v bodě nespojitosti vynechána a zobrazený PC-graf neodpovídá v okolí bodu $[0, 0]$ grafu funkce. Všimněme si u tohoto příkladu podrobně struktury Mapleovské grafiky. Generujme graf zkoumané funkce pro x a y z intervalu $\langle -3, 3 \rangle$ při hustotě sítě $[7, 7]$ a podívejme se, jaká je struktura vytvořeného objektu **PLOT3D**:

```
> p:=plot3d(f, -3..3, -3..3, orientation=[-57,38], axes=framed,
> grid=[7,7], color=black, labels=[x,y,z]);p;
```

```
p := PLOT3D(GRID(-3...3., -3...3., [[-1.5000000000000000,
-1.384615384615385, -.9000000000000000, 0,
.9000000000000000, 1.384615384615385, 1.5000000000000000]
, [-.9230769230769231, -1., -.8000000000000000, 0,
.8000000000000000, 1., .9230769230769231], [
-.3000000000000000, -.4000000000000000,
-.5000000000000000, 0, .5000000000000000,
.4000000000000000, .3000000000000000],
[0, 0, 0, FAIL, 0, 0, 0], [-.3000000000000000,
-.4000000000000000, -.5000000000000000, 0,
.5000000000000000, .4000000000000000, .3000000000000000]
, [-.9230769230769231, -1., -.8000000000000000, 0,
.8000000000000000, 1., .9230769230769231], [
-1.5000000000000000, -1.384615384615385,
-.9000000000000000, 0, .9000000000000000,
1.384615384615385, 1.5000000000000000]],
COLOR( RGB, 0, 0, 0 )), AXESLABELS( x, y, z ), TITLE( ),
AXESSTYLE( FRAME ), PROJECTION( -57., 38., 1 ))
```

V objektu **PLOT3D** jsou uloženy funkční hodnoty v uzlových bodech sítě, které jsou počítány numericky postupně po řadách. Všimněme si funkčních hodnot pro body $[0, -3]$, $[0, -2]$, \dots . Zde skutečně výpočet funkční hodnoty pro bod $[0, 0]$ „havaruje“ (*FAIL*). Ná-

sledně je tento objekt použit při zobrazování na výstupním zařízení a chybějící funkční hodnota v bodě $[0,0]$ je vynechána (obr. 1.7).

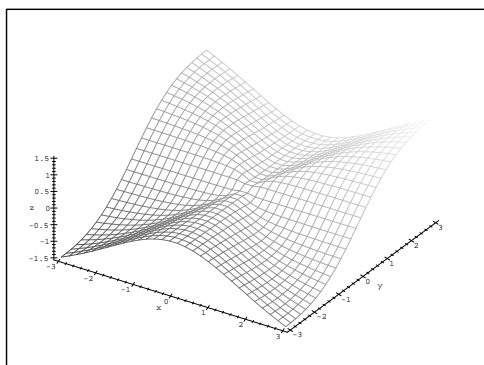
Změníme tedy hustotu uzlových bodů tak, aby bod $[0,0]$ (bod nespojitosti) nebyl uzlovým bodem (obr. 1.8):

```
> plot3d(f, -3..3, -3..3, orientation=[-57,38], axes=framed,
> grid=[30,30], labels=[x,y,z]);
```

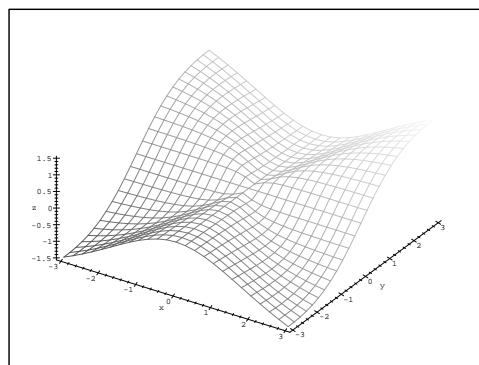
Jinou možností je dodefinovat funkční hodnotu v bodě $[0,0]$ tak, aby funkce f v tomto bodě byla spojitá. Poté generujeme PC-graf získané spojitě funkce:

```
> g:=proc(x,y) if x=0 and y=0 then 0 else (x^2*y)/(x^2+y^2) fi end:
> plot3d(g, -3..3,-3..3,orientation=[-57,38],axes=framed,labels=[x,y,z]);
```

Obdržíme výsledek znázorněný na obr. 1.9.



obr. 1.8



obr. 1.9

Příklad 1.2. Funkce

$$f(x, y) = \frac{\sin xy}{xy}$$

není spojitá v bodech ležících na osách x a y , ale má zde konečnou limitu rovnu jedné.

Při pokusu o tvorbu PC-grafu příkazem:

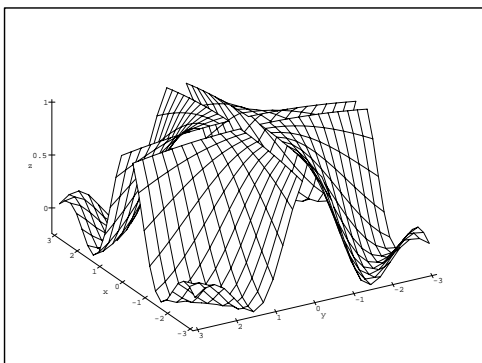
```
> plot3d(sin(x*y)/(x*y), x=-3..3, y=-3..3, axes=framed, color=black,
> orientation=[150,50], labels=[x,y,z], tickmarks=[7,7,3]);
```

dostáváme PC-graf na obrázku 1.10. Zde jsou opět patrné nespojené body, ve kterých výpočet funkčních hodnot „havaroval“ (body nespojitosti na osách x a y opět vycházejí do uzlových bodů sítě).

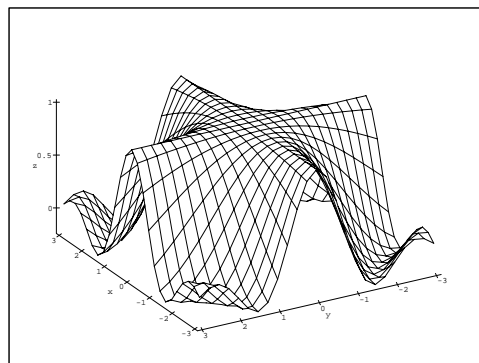
Vytvořme tedy PC-graf spojitě funkce (obr. 1.11) (dodefinujeme funkci tak, aby byla spojitá)

$$g(x, y) = \begin{cases} 1 & \text{pro } x = 0 \text{ nebo } y = 0 \\ f(x, y) & \text{jinak.} \end{cases}$$

```
> g:=proc(x,y) if x=0 or y=0 then 1 else sin(x*y)/(x*y) fi end:
> plot3d(g, -3..3, -3..3, axes=framed, orientation=[150,50],
> color=black, labels=[x,y,z], tickmarks=[7,7,3]);
```



obr. 1.10



obr. 1.11

Jinou možností je opět vhodně změnit hustotu sítě tak, aby body nespojitosti nebyly totožné s uzlovými body sítě.

Pokud v bodech nespojitosti neexistuje konečná limita, je znázornění chování takové funkce pomocí počítače obtížnější.

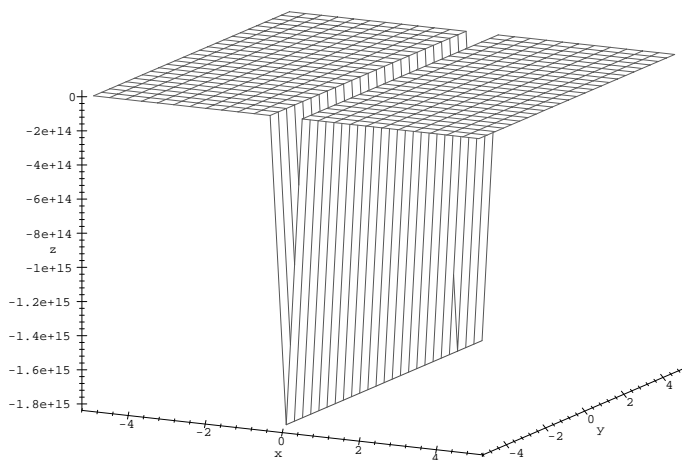
Příklad 1.3. Generujte PC-graf funkce $f(x, y) = 1/x$.

Protože $\lim_{x \rightarrow 0^+} 1/x = +\infty$, $\lim_{x \rightarrow 0^-} 1/x = -\infty$, není funkce f na přímce $x = 0$ spojitá.

Příkazem:

```
> plot3d(1/x, x=-5..5, y=-5..5, orientation=[-63,73], axes=framed,
> labels=[x,y,z]);
```

dostáváme PC-graf z obr. 1.12.



obr. 1.12

Vidíme, že PC-graf neodpovídá grafu zkoumané funkce. Podívejme se opět na objekt **PLOT3D**, pro zjednodušení zvolme `grid=[7,3]` (obr. 1.13):

```
> p:=plot3d(1/x, x=-5..5, y=-5..5, orientation=[-63,73], axes=framed,
```

```

> grid=[7,3], color=black, labels=[x,y,z]);p;
p := PLOT3D(GRID(-5...5., -5...5., [[-.2000000000000000,
-.2000000000000000, -.2000000000000000], [
-.3000000000000000, -.3000000000000000, [
-.3000000000000000], [-.5999999999999999,
-.5999999999999999, -.5999999999999999], [
-.2251799813685248 1016, -.2251799813685248 1016,
-.2251799813685248 1016],
[.6000000000000002, .6000000000000002, .6000000000000002]
,
[.3000000000000000, .3000000000000000, .3000000000000000]
,
[.2000000000000000, .2000000000000000, .2000000000000000]
), COLOR( RGB, 0, 0, 0 ), AXESLABELS( x, y, z ),
AXESSTYLE( FRAME ), TITLE( ), PROJECTION( -63., 73., 1 ))

```

Maple volí rozsah zobrazovaných hodnot a měřítka na osách sám tak, aby se výsledný PC-graf co nejlépe „vešel“ na výstupní zařízení. To zejména u funkcí, jejichž limita v některém bodě je rovna ∞ , způsobuje problémy (odlišnost grafu a PC-grafu funkce). Z algoritmu realizace PC-grafu na výstupním zařízení plyne i spojení těch funkčních hodnot, které by neměly být spojeny (v okolí bodů nespojitosti, body nespojitosti v tomto případě nejsou totožné s uzlovými body). Z objektu **PLOT3D** je také vidět, že při této hustotě sítě a stanovené přesnosti aproximace jsou v PC-grafu potlačeny funkční hodnoty blízké $+\infty$. Stačí však změnit přesnost aproximace (změnou hodnoty proměnné `Digits`, implicitní nastavení je `Digits:=9`), a dostáváme jinou síť uzlových bodů a také jiný PC-graf (obr. 1.14):

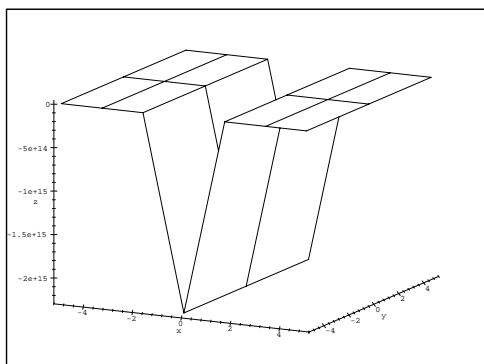
```
> Digits:=18;
```

Digits := 18

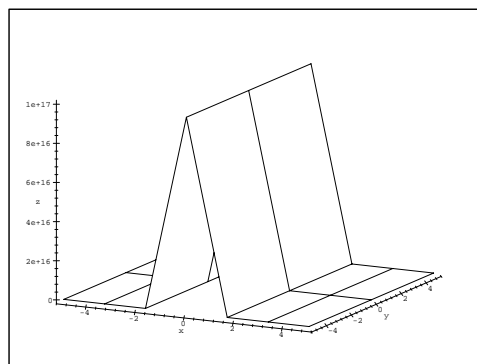
```

> plot3d(1/x, x=-5..5, y=-5..5, orientation=[-63,73], axes=framed,
> grid=[7,3], color=black, labels=[x,y,z]);

```



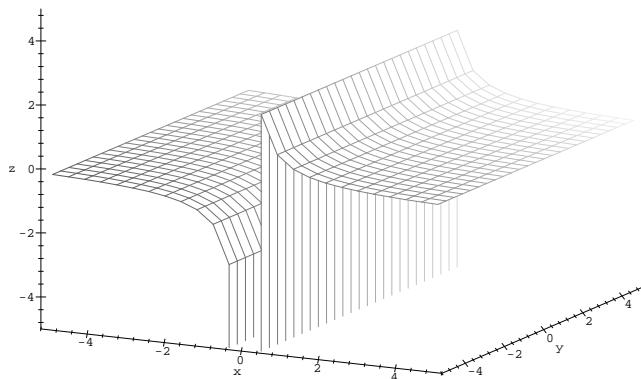
obr. 1.13



obr. 1.14

Omezíme tedy rozsah zobrazovaných hodnot (`view=-5..5`) při původní přesnosti aproximace (obr. 1.15):

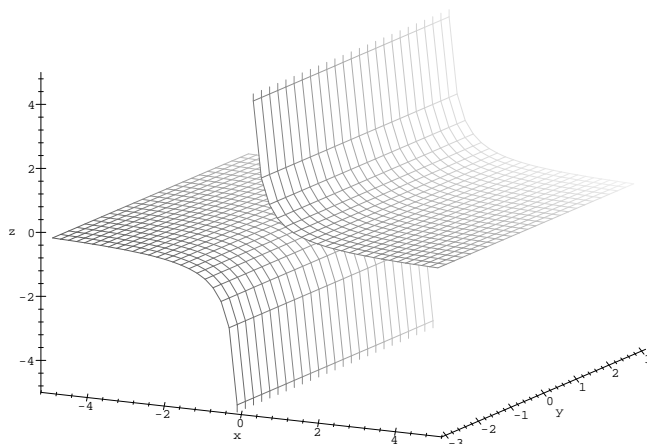
```
> Digits:=9:
> plot3d(1/x, x=-5..5, y=-5..5, view=-5..5,
> orientation=[-63,73], axes=framed, labels=[x,y,z]);
```



obr. 1.15

Na PC-grafu je vidět pozitivní vliv změny rozsahu zobrazovaných hodnot, nadále však přetrvává spojování i těch bodů PC-grafu, které spojeny být neměly. Skutečnosti odpovídající PC-graf získáme následujícím způsobem. Tvorbu PC-grafu rozdělíme do dvou částí tím, že definiční obor rozdělíme na dvě oblasti: $\langle -5, -0.001 \rangle \times \langle -3, 3 \rangle$ a $\langle 0.001, 5 \rangle \times \langle -3, 3 \rangle$. Jednotlivé samostatně vytvářené části PC-grafu v závěru interpretujeme v jediném (obr. 1.16) pomocí příkazu `display3d` z knihovny `plots`:

```
> o1:=plot3d(1/x, x=-5..-0.001, y=-3..3, view=-5..5):
> o2:=plot3d(1/x, x=0.001..5, y=-3..3, view=-5..5):
> display3d({o1,o2}, orientation=[-63,73], axes=framed, labels=[x,y,z]);
```



obr. 1.16

Poznámka 1.1. Tvorba PC-grafu nespojité funkce jedné reálné proměnné je zjednodušena parametrem `discont=true`. Při použití tohoto parametru program nejprve určí body nespojitosti zadané funkce a poté rozdělí horizontální osu na intervaly, na kterých je tato funkce spojitá, takže nedojde ke spojení těch bodů PC-grafu, které spojeny být neměly.

V některých případech je vhodnější nezobrazovat funkci ve tvaru explicitním, ale provést parametrizaci funkce ($x = \phi(u, v)$, $y = \psi(u, v)$, $z = \chi(u, v)$, kde u a v jsou parametry). Výhodné je to zejména u funkcí, které vykazují středovou nebo osovou symetrii.

Příklad 1.4. Vytvořte PC-graf funkce

$$f(x, y) = \frac{2}{x^2 + y^2 - 9}.$$

Definičním oborem funkce f je množina $\mathbb{R}^2 - \{[x, y] : x^2 + y^2 = 9\}$, tedy rovina xy kromě bodů ležících na kružnici se středem v bodě $[0, 0]$ a poloměrem $r = 3$. V těchto bodech není funkce spojitá. Pokud se pokusíme vytvořit PC-graf funkce jednoduchým příkazem:

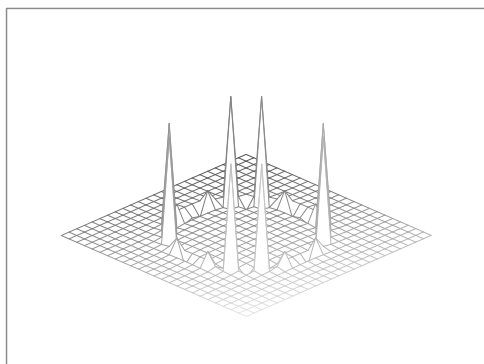
```
> f:=2/(x^2+y^2-9);
```

$$f := 2 \frac{1}{x^2 + y^2 - 9}$$

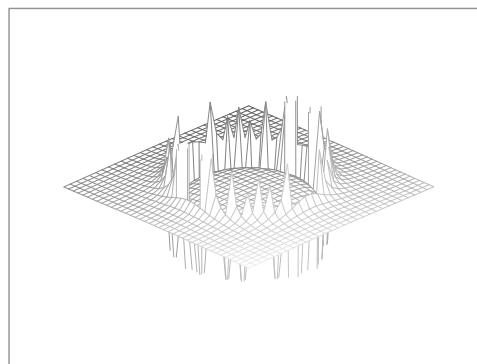
```
> plot3d(f, x=-5..5, y=-5..5);
```

dostáváme obr. 1.17. Změna hustoty uzlových bodů a omezení rozsahu zobrazovaných hodnot v tomto případě nepomáhá (obr. 1.18):

```
> plot3d(f, x=-5..5, y=-5..5, view=-5..5, grid=[30,40]);
```



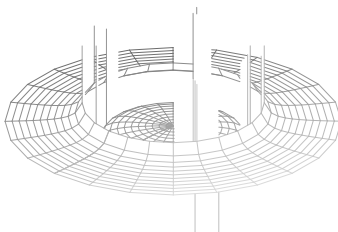
obr. 1.17



obr. 1.18

Proveďme nyní parametrizaci $x = u \cos v$, $y = u \sin v$, $z = \frac{2}{u^2 - 9}$ a generujme PC-graf této funkce:

```
> plot3d([u*cos(v), u*sin(v), subs({x=u*cos(v), y=u*sin(v)}, f)],  
> v=0..2*Pi, u=0..6, view=-5..5):
```



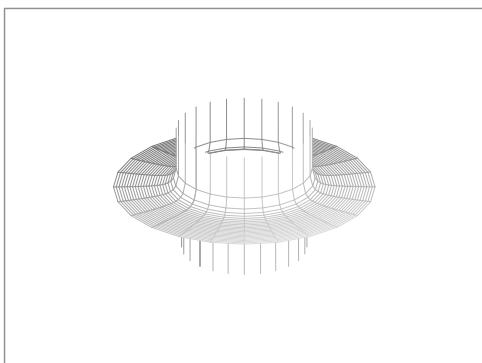
obr. 1.19

Všimněme si rozdílu mezi PC-grafem funkce f (dané explicitně, obr. 1.17 a 1.18) a PC-grafem téže funkce dané parametricky (obr. 1.19) (graf by měl být v obou případech stejný).

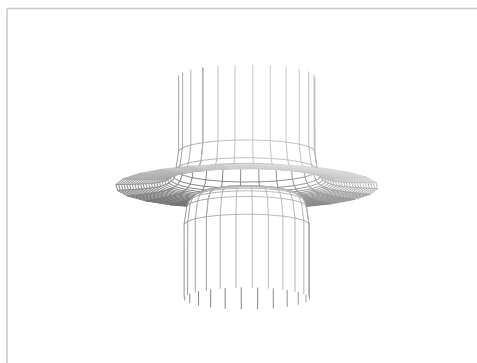
Protože získaný PC-graf stále neodpovídá grafu funkce, rozdělíme tvorbu PC-grafu opět do dvou částí, přičemž parametr u bude postupně nabývat hodnot z intervalů $\langle 0, 2.999 \rangle$ a $\langle 3.001, 6 \rangle$:

```
> s1:=plot3d([u*cos(v), u*sin(v), subs({x=u*cos(v), y=u*sin(v)}, f)],
> v=0..2*Pi, u=0..2.999):
> s2:=plot3d([u*cos(v), u*sin(v), subs({x=u*cos(v), y=u*sin(v)}, f)],
> v=0..2*Pi, u=3.001..6):
> display3d({s1,s2}, view=-8..8);
> display3d({s1,s2}, view=-8..8, orientation=[40,102]);
```

Z důvodu názornosti je funkce zobrazena ze dvou různých pohledů (obr. 1.20 a obr. 1.21).



obr. 1.20



obr. 1.21

Všimněme si nyní ještě některých parametrů příkazu `plot3d`, kterými můžeme ovlivnit vzhled výsledného PC-grafu. Doposud jsem generovali PC-graf vždy nad čtvercovou nebo obdélníkovou oblastí. Ale rozsah druhého parametru může být udán v závislosti na prvním. Například při generování PC-grafu povrchu polokoule nad čtvercovým oborem:

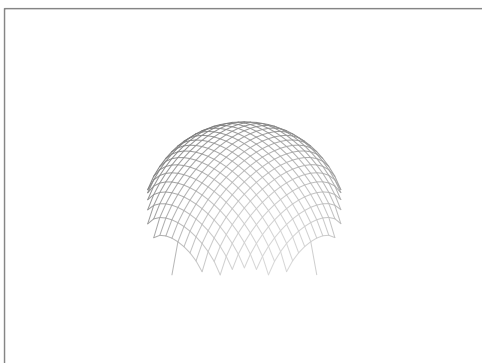
```
> plot3d(sqrt(1-x^2-y^2), x=-1..1, y=-1..1, scaling=constrained);
```

dostáváme PC-graf na obrázku 1.22. Získaný PC-graf neodpovídá na okrajích oblasti grafu funkce („zubaté okraje“ jsou opět způsobeny spojováním funkčních hodnot v uzlových bodech).

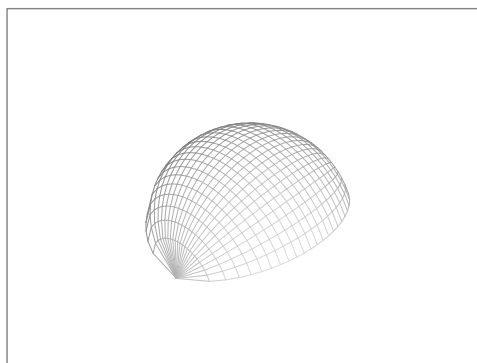
Při použití kruhové oblasti:

```
> plot3d(sqrt(1-x^2-y^2), x=-1..1, y=-sqrt(1-x^2)..sqrt(1-x^2),
> scaling=constrained);
```

(tj. proměnného rozsahu na ose y) dostáváme PC-graf odpovídající grafu funkce (obr. 1.23).



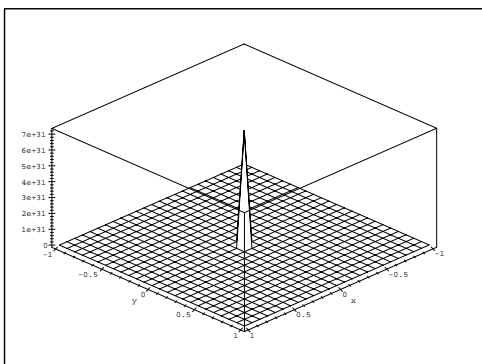
obr. 1.22



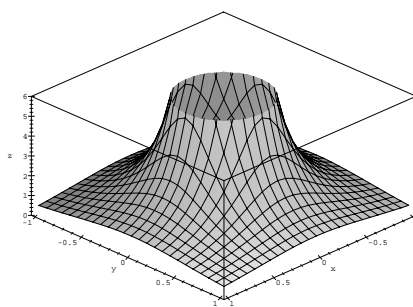
obr. 1.23

Rozsah zobrazovaných hodnot ve směru osy z měníme parametrem `view=[zmin..zmax]`. Pokud tento parametr nezadáme, volí Maple rozsah zobrazovaných hodnot sám, což opět může vést k zavádějícím výsledkům (viz. také komentář k příkladu 1.3). Porovnejme následující dva PC-grafy (obr. 1.24 a obr. 1.25), generované příkazy:

```
> plot3d(1/(x^2+y^2), x=-1..1, y=-1..1, axes=boxed, color=black,
> labels=[x,y,z]);
> plot3d(1/(x^2+y^2), x=-1..1, y=-1..1, view=0..6, style=patch,
> axes=boxed, labels=[x,y,z]);
```



obr. 1.24



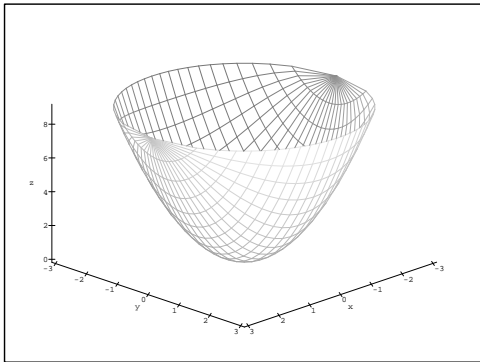
obr. 1.25

Pro zkoumanou funkci je $\lim_{(x,y) \rightarrow (0,0)} 1/(x^2 + y^2) = +\infty$, rozsah zobrazovaných hodnot a měřítka na osách v prvním případě Maple volil sám ($+\infty$ aproximoval hodnotou $7 \cdot 10^{31}$). Výsledný PC-graf (obr. 1.24) pak neodpovídá grafu funkce. Obor zobrazovaných hodnot tedy omezíme parametrem `view=0..6` na interval $\langle 0, 6 \rangle$, získaný PC-graf je znázorněn na obr. 1.25.

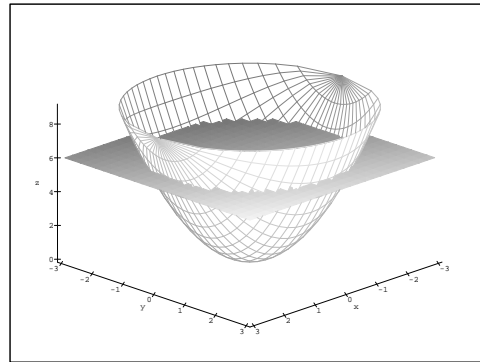
1.2. Vrstevnice

Pro vytvoření představy o tvaru a průběhu znázorňované plochy nám často pomáhají vrstevnice grafu funkce a řezy rovinami $z = 0$, $y = 0$, $x = 0$, příp. rovinami s nimi rovnoběžnými. Maple nám tak může pomoci při vysvětlování geometrického významu pojmu vrstevnice funkce a při jejich znázorňování.

Ukažme si nyní konstrukci vrstevnice funkce $f(x, y) = x^2 + y^2$ na hladině $c = 6$. Nejdříve generujme PC-graf funkce f a označme jej $P1$ (obr. 1.26). Poté vytvořme PC-graf roviny $z = 6$, označme jej $P2$, a interpretujme funkci i rovinu v jednom PC-grafu (obr. 1.27):



obr. 1.26



obr. 1.27

```
> with(plots):
> f := (x,y) -> x^2+y^2:
> P1 := plot3d(f(x,y), x=-3..3, y= -sqrt(9-x^2)..sqrt(9-x^2), axes=framed,
> tickmarks=[7,7,5], orientation=[45,60], labels=[x,y,z]): ";
> P2 := plot3d(6, x=-3..3, y= -3..3, style = patchnogrid):
> display3d({P1,P2}, axes=framed,tickmarks = [7,7,5],orientation=[45,60],
> labels=[x,y,z]);
```

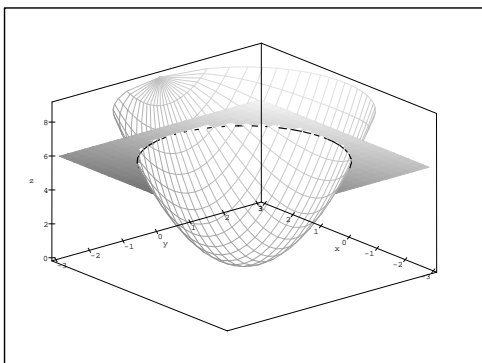
Křivka, vzniklá jako průsečnice grafu funkce f a roviny $z = 6$ je dána parametricky rovnicemi

$$x = \sqrt{6} \cos t, \quad y = \sqrt{6} \sin t, \quad z = 6$$

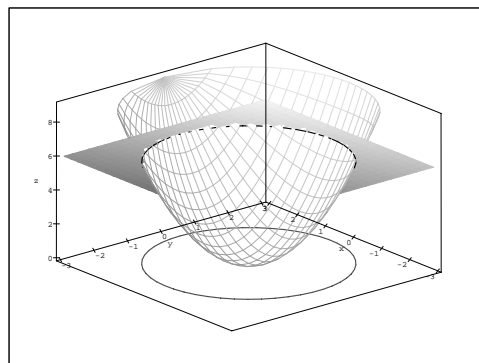
a představuje vrstevnici funkce f na hladině $c = 6$. Znázornění vrstevnice v rovině získáme průmětem do roviny xy . Situaci znázorňují následující dva obrázky (obr. 1.28, obr. 1.29).

Pro vykreslení prostorové křivky jsme použili procedury `spacecurve` z knihovny `plots`:

```
> P3 := spacecurve([sqrt(6)*cos(t), sqrt(6)*sin(t),6],t=0..2*Pi,
> color=black, thickness=3):
> P4 := spacecurve([sqrt(6)*cos(t), sqrt(6)*sin(t),0],t=0..2*Pi,
> color=red, thickness=3):
> display3d({P1,P2,P3},tickmarks=[7,7,5],orientation = [40,120],
> axes=boxed, labels=[x,y,z]);
> display3d({P1,P2,P3,P4},tickmarks=[7,7,5],orientation = [40,120],
> axes=boxed, labels=[x,y,z]);
```



obr. 1.28



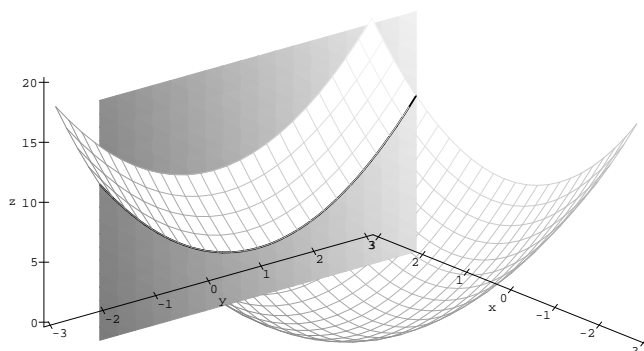
obr. 1.29

Podobným způsobem můžeme znázornit i řezy rovinami rovnoběžnými s rovinami xz a yz . Např. zobrazme průnik roviny $x = 2$ a grafu funkce f . Jako průsečnici získáme křivku, kterou můžeme popsat parametricky rovnicemi

$$x = 2, y = t, z = f(2, t) = 4 + t^2.$$

Graf funkce, rovinu i jejich průsečnici interpretujme v jednom PC-grafu (obr. 1.30):

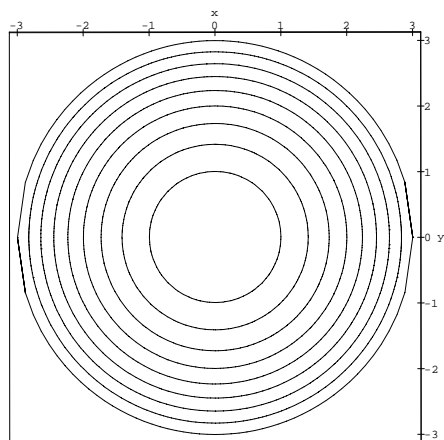
```
> P1 := plot3d(f(x,y), x=-3..3, y=-3..3):
> P2 := implicitplot3d(x=2, x=-3..3, y=-3..3, z=0..20, style=patchnogrid):
> P3 := spacecurve([2,t,j(2,t)], t=-3..3, thickness=3, color=black):
> display3d({P1,P2,P3}, tickmarks=[7,7,5], axes=framed,
> orientation=[40,120], labels=[x,y,z]);
```



obr. 1.30

Pro přímé znázorňování vrstevnic používáme příkaz `contourplot` (obr. 1.31):

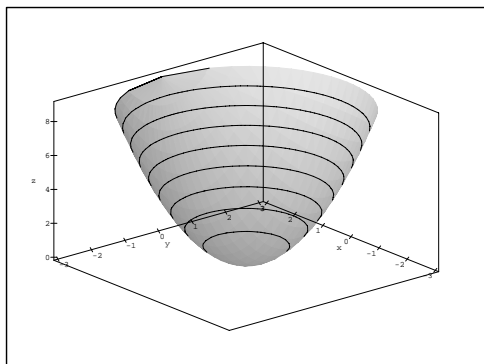
```
> plots[contourplot](f(x,y), x=-3..3, y=-sqrt(9-x^2)..sqrt(9-x^2),
> axes=boxed, color=black, contours=10, numpoints=2500,
> scaling=constrained, tickmarks=[7,7,0]);
```



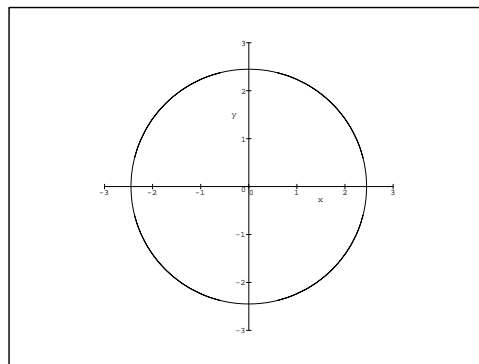
obr. 1.31

Parametr `style=patchcontour` příkazu `plot3d` slouží k zobrazení grafu funkce s vrstevnicemi (obr. 1.32) a pro zobrazení vrstevnice na dané hladině můžeme použít příkazu `levelcurve` z knihovny `mvcalp` (obr. 1.33):

```
> plot3d(f(x,y), x=-3..3, y=-sqrt(9-x^2)..sqrt(9-x^2), style=patchcontour,
> axes=boxed, orientation=[40,120], tickmarks=[7,7,5], labels=[x,y,z]);
> with(mvcalp):
> levelcurve(f(x,y),6, x=-3..3, y=-3..3, color=black, scaling=constrained,
> tickmarks=[7,7]);
```



obr. 1.32



obr. 1.33

Protože tvorba matematické grafiky není často jednoduchou záležitostí a vzhled výsledného PC-grafu můžeme ovlivňovat celou řadou parametrů, uvádíme na závěr této kapitoly i stručný přehled základních použitých příkazů a jejich parametrů. Popis všech příkazů čtenář najde buď v manuálech [11], [12] a [13] nebo přímo v systému nápovědy programu Maple V.

Přehled použitých příkazů

Generování PC-grafu funkce dvou proměnných:

`plot3d(f(x,y),x=a..b,y=c..d,volby);` pro výrazy a `plot3d(f,a..b,c..d,volby);` pro funkce.

Volitelné parametry `volby` ovlivňují vzhled výsledného PC-grafu. Nejčastěji používané parametry jsou popsány v následující tabulce:

Volba	Efekt příkazu <code>plot3d(..., volby);</code>
<code>scaling = UNCONSTRAINED</code> <code>CONSTRAINED</code>	měřítko na osách
<code>view = zmin..zmax</code>	volba rozsahu zobrazovaných hodnot
<code>orientation = [theta,phi]</code>	úhel pohledu
<code>style = POINT</code> <code>HIDDEN</code> <code>PATCH</code> <code>WIREFRAME</code> <code>CONTOUR</code> <code>PATCHCONTOUR</code> <code>PATCHNOGRID</code> <code>LINE</code>	způsob vykreslení grafu
<code>axes = BOXED</code> <code>NORMAL</code> <code>FRAME</code> <code>NONE</code>	znázornění os
<code>grid = [m,n]</code>	regulace hustoty sítě
<code>numpoints = n</code>	alternativní zadání počtu bodů sítě
<code>labels = [x,y,z]</code>	popis os
<code>tickmarks = [n,m,p]</code>	počet značek na osách

Generování PC-grafu funkce dvou proměnných dané parametricky:

`plot3d([f(s,t), g(s,t), h(s,t)], s=a..b, t=c..d, volby);` (obr. 1.2).

K rozšíření možností práce s grafikou slouží knihovna `plots`. Procedury této knihovny zpřístupníme příkazem `with(plots)`:

Vykreslení prostorové křivky:

`plots[spacecurve]([f(t), g(t), h(t)], t=a..b, volby);` (obr. 1.28 a 1.29).

Znázornění vrstevnic:

`plots[contourplot](f(x,y), x=a..b, y=c..d, volby);` (obr. 1.31).

Znázornění vrstevnice na dané hladině:

`mvcalp[levelcurve](f(x,y), hladina, x=a..b, y=c..d, volby);` (obr. 1.33).

`plots[implicitplot3d](expr1, x=a..b,y=c..d,z=p..q, volby);` (obr. 1.30 a viz. také Kapitola 7);