```
> A2:=augment(submatrix(A,1..3,[1]),b,submatrix(A,1..3,[3]));
```

Potom vypočítáme hodnotu x_2:

```
> x2:=det(A2)/det(A);
```

Zadání matice. Matici M typu $(2,3)$ zadáme jedním z příkazů:

```
> M:=linalg[matrix](2,3,[1,2,3,4,5,6]);
> M:=linalg[matrix]([[1,2,3],[4,5,6]]);
```

Operace s vektory. Vektory, např. $\vec{u}=(1,-2,3)$ a $\vec{v}=(0,-5,4)$, zadáme příkazy „`u:=[1,-2,3];`" a „`v:=[0,-5,4];`"
Skalární, respektive vektorový součin vektorů $\vec{u}$, $\vec{v}$ provedeme následující aplikací příkazu `dotprod`, resp. `crossprod`:

```
> linalg[dotprod](u,v,'orthogonal');
> linalg[crossprod](u,v);
```

Normu (eukleidovskou) vektoru $\vec{u}$ spočítáme příkazem „`linalg[norm](u,2);`". Pozor na záměnu s výpočtem absolutní hodnoty výrazu x. Ta se určí příkazem „`abs(x);`". Úhel vektorů $\vec{u}$, $\vec{v}$ spočítáme následující posloupností příkazů. První z nich otevře balíček příkazů `linalg`, druhý uvede velikost úhlu α v radiánech a pomocí třetího příkazu potom tento údaj převedeme na velikost úhlu ve stupních:

```
> with(linalg):
> alpha:=arccos(dotprod(u,v)/(norm(u,2)*norm(v,2)));
> evalf(convert(alpha,degrees));
```

3.2 3D grafy v Maple

V této příloze se budeme nejprve podrobně věnovat možnostem znázornění trojrozměrných křivek a ploch pomocí prostředků rozhraní *Classic worksheet* programu Maple. V závěru přílohy potom zmíníme dvě konkrétní aplikace programu Maple, které dovolují uživatelsky nenáročným způsobem, bez znalosti jakéhokoliv příkazu, konstruovat grafické znázornění ploch a provádět analýzu ploch druhého stupně. Jedná se o takzvané *maplety.* První z těchto mapletů, *Interactive Plot Builder*, je asistentem pro kreslení 3D grafů dodávaným spolu s programem. Takovýchto asistentů má uživatel programu Maple k dispozici více, pouze však v prostředí *Standard worksheet* (pro více informací stačí zadat „`?assistants`"). Druhý z představených mapletů je příkladem uživatelsky naprogramované aplikace. Byl vytvořen v roce 2008 studentem Padagogické fakulty JU Markem Dvorožňákem. Pojetí této aplikace přesně odpovídá jejímu účelu, kterým je provedení kompletní analýzy zadané křivky spolu s jejím grafickým znázorněním.

1. Křivka
Křivka zvaná **šroubovice** je dána parametrickými rovnicemi $x=r\cos\omega,\ y=r\sin\omega,\ z=v_0\omega,$ kde r je poloměr příslušného otáčení, ω je úhel tohoto otáčení

a v_0 je parametr šroubového pohybu, tzv. redukovaná výška závitu (posunutí podél osy rotace příslušné otočení o jeden radián). Zadáme parametrické rovnice, zvolíme si hodnoty r a v_0 a odpovídající šroubovici zobrazíme.

```
> H:=omega->[r*cos(omega),r*sin(omega),v0*omega];
> r:=5: v0:=0.1:
```

Pro grafické znázornění křivky `H(omega)` můžeme použít buď příkaz `spacecurve` z balíčku `plots`, nebo příkaz `plot3d`. Zde jsou obě možnosti:

```
> plots[spacecurve](H(t),t=0..4*Pi,axes=frame);
> plot3d(H(t),t=0..4*Pi,s=0..1,numpoints=10000);
```

V druhém případě používáme kvůli dané syntaxi příkazu `plot3d` (očekává dva parametry) „klamný“ parametr s, který s křivkou nijak nesouvisí. Výslednou podobu grafů můžeme opět ovlivnit prostřednictvím nepovinných parametrů, tzv. voleb (`options`).

2. Plocha

Následují příklady různých způsobů zobrazení plochy v závislosti na jejím zadání. Použité příkazy jsou ve většině případů uvedeny v základním tvaru. Tomu může odpovídat kvalita výsledných grafů. Jejich podobu lze dále ovlivnit, a tím i vylepšit, řadou volitelných parametrů, jejichž kompletní přehled získáme zadáním `?plot3d,options`. Za vyzkoušení určitě stojí například volby `grid`, `numpoints`, `style`, `caption` a `lightmodel`.

Plocha daná rovnicí ve tvaru $z = f(x,y)$. Plocha zvaná *Plückerův konoid* je dána rovnicí $z = \dfrac{x^2 - y^2}{x^2 + y^2}$. K jejímu zobrazení můžeme použít buď příkaz `plot3d`:

```
> z:=(x,y)->(x^2-y^2)/(x^2+y^2);
> plot3d(z(x,y),x=-10..10,y=-10..10);
```

nebo příkaz `implicitplot3d` z balíčku `plots`:

```
> plots[implicitplot3d](z=(x^2-y^2)/(x^2+y^2),x=-2..2,y=-2..2,
z=-2..2, grid=[40,40,10]);
```

Plocha daná rovnicí ve tvaru $F(x,y,z) = 0$. Trojosý elipsoid, daný rovnicí $x^2 + 2y^2 + 5z^2 - 1 = 0$, zobrazíme například pomocí příkazů

```
> Kv:=x^2+2*y^2+5*z^2-1=0;
> plots[implicitplot3d](Kv,x=-1..1,y=-1..1,z=-1..1);
```

Jak bylo řečeno v úvodu, můžeme příkazy pro kreslení grafu doplnit různými nepovinnými parametry ovlivňujícími kvalitu obrázku. Vyzkoušejte:

```
> plots[implicitplot3d](Kv,x=-1..1,y=-1..1,z=-1..1,
grid=[20,20,20],style=patchcontour,color=red,lightmodel=light1,
scaling=constrained);
```

Plocha daná parametricky. K zobrazení parametrického vyjádření plochy použijeme příkaz `plot3d`. Například kulovou plochu o středu $S = [1,2,3]$ a poloměru $r = 2$ zobrazíme užitím parametrického grafu takto:

```
> S:=[1,2,3]; r:=2;
```

```
> plot3d([S[1]+r*cos(u)*cos(t),S[2]+r*cos(u)*sin(t),S[3]+r*sin(u)],
t=0..2*Pi,u=-Pi/2..Pi/2,axes=frame);
```

Plocha daná rozměry a souřadnicemi středu/vrcholu. Balíček `plottools` obsahuje příkazy pro zobrazení různých rovinných křivek (mimo jiné elipsy a hyperboly) a trojrozměrných těles (např. koule, válec, rovnoběžnostěn, osmistěn apod.). Nás z nich na tomto místě mohou zajímat kužel, válec a koule, jejichž povrchy jsou částmi příslušných kvadratických ploch.
Kulovou plochu o středu $S = [1, 2, 3]$ a poloměru $r = 2$ tak můžeme zobrazit pomocí příkazu `sphere` ze zmíněného balíčku `plottools` takto:

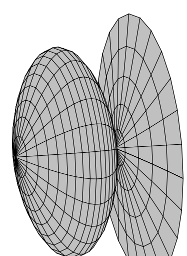
```
> Koule:=plottools[sphere]([1,2,3],2):
> plots[display](Koule,color=green,lightmodel=light2,axes=frame);
```

Příkazy z balíčku `plottools` negenerují obrázky, ale pouze data potřebná pro jejich vykreslení, které potom musíme provést příkazem `plots[display]`.
Podobně jako kulovou plochu bychom zobrazili také válec a kužel. Použili bychom příkazy `plottools[cylinder]` a `plottools[cone]`. Pro více informací viz „?plottools“.

3. Rotační plochy - parametrický graf

Na straně 103 jsme počítali objem tělesa, které vznikne rotací grafu funkce $f(x) = x^2 - 4x$ kolem osy x na intervalu $\langle 0, 5\rangle$. Toto těleso lze snadno zobrazit (viz obrázek 3.1) pomocí parametrického grafu takto:

```
> f:=x->x^2-4*x;
> P:=[x,f(x)*cos(u),f(x)*sin(u)];
> plot3d(P,x=0..5,u=-Pi..Pi,scaling=constrained);
```



Obrázek 3.1: Rotační plocha

4. Průnik ploch

Průniková křivka kulové plochy s válcovou plochou, které jsou dány rovnicemi $x^2 + y^2 + z^2 = 4r,\ (x - r)^2 + y^2 = r^2$, se nazývá **Vivianiho křivka** nebo též **Vivianiho okno** (viz obrázek 3.2).
Nejprve zobrazíme příslušnou kulovou a válcovou plochu pro konkrétní hodnotu r. Průnikovou křivku těchto dvou ploch potom vyjádříme parametricky a zobrazíme ve stejném obrázku. Výsledek by měl zhruba odpovídat obrázku 3.2.

Definujeme obě plochy a zvolíme hodnotu $r = 5$:

```
> Sph:=x^2+y^2+z^2=4*r^2; Cyl:=(x-r)^2+y^2=r^2;
> r:=5:
```

Výstupy příkazů `implicitplot` nezobrazíme (proto je ukončíme dvojtečkou), ale uložíme do proměnných `SphG`, `CylG`:

```
> SphG:=plots[implicitplot3d](Sph,x=-10..10,y=-10..10,z=-10..10,
color=grey):
> CylG:=plots[implicitplot3d](Cyl,x=-10..10,y=-10..10,z=-10..10,
color=pink):
```

K jejich zobrazení do jedné soustavy souřadnic pak použijeme příkaz `display` z balíčku `plots`:

```
> plots[display](SphG,CylG);
```

Průnikovou křivku získáme řešením soustavy rovnic daných ploch. Pro získání kompletního řešení použijeme příkaz `allvalues`:

```
> Sol:=allvalues(solve({Cyl,Sph},{x,y}));
```

Abychom převedli obě řešení do tvaru uspořádaných trojic, použijeme příkaz `eval` následujícím způsobem:

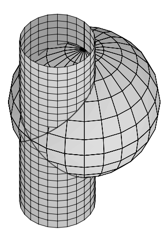
```
> VW1:=eval([x,y,z],Sol[1]); VW2:=eval([x,y,z],Sol[2]);
```

K zobrazení použijeme příkaz `spacecurve` z balíčku `plots`. Opět výstup uložíme do proměnné a ukončíme dvojtečkou:

```
> VWG:=plots[spacecurve]({VW1,VW2 },z=-10..10, color=red,
thickness=3, numpoints=1000):
```

Všechny tři objekty (dvě plochy a průnikovou křivku) zobrazíme do jedné soustavy souřadnic:

```
> plots[display3d](SphG,CylG,VWG):
```



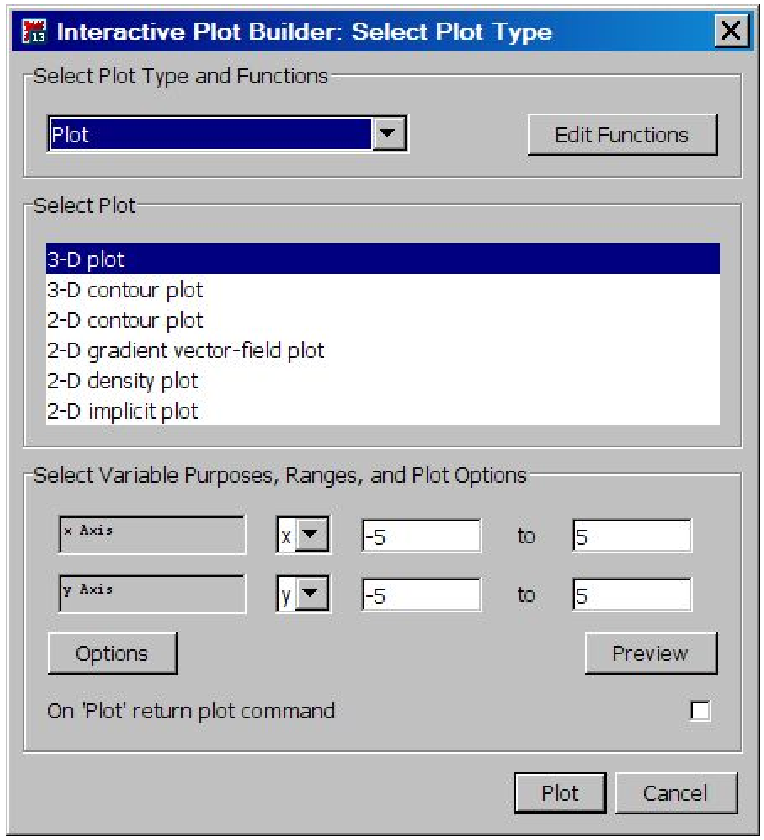
Obrázek 3.2: Vivianiho okno

Poznámka: Chceme-li dostat úplně stejný obrázek, jako je Obr.3.2, musíme příslušné plochy zadat parametricky:

```
> SphG2:=plot3d([2*r*cos(t)*cos(u),2*r*cos(t)*sin(u),2*r*sin(t)],
t=-Pi..Pi, u=0..2*Pi, scaling=constrained):
> CylG2:=plot3d([r*cos(u)+r,r*sin(u),t], t=-3*r..3*r, u=-Pi..Pi,
scaling=constrained):
> plots[display3d](SphG2,CylG2,VWG):
```

5. Aplikace pro kreslení 3D grafů

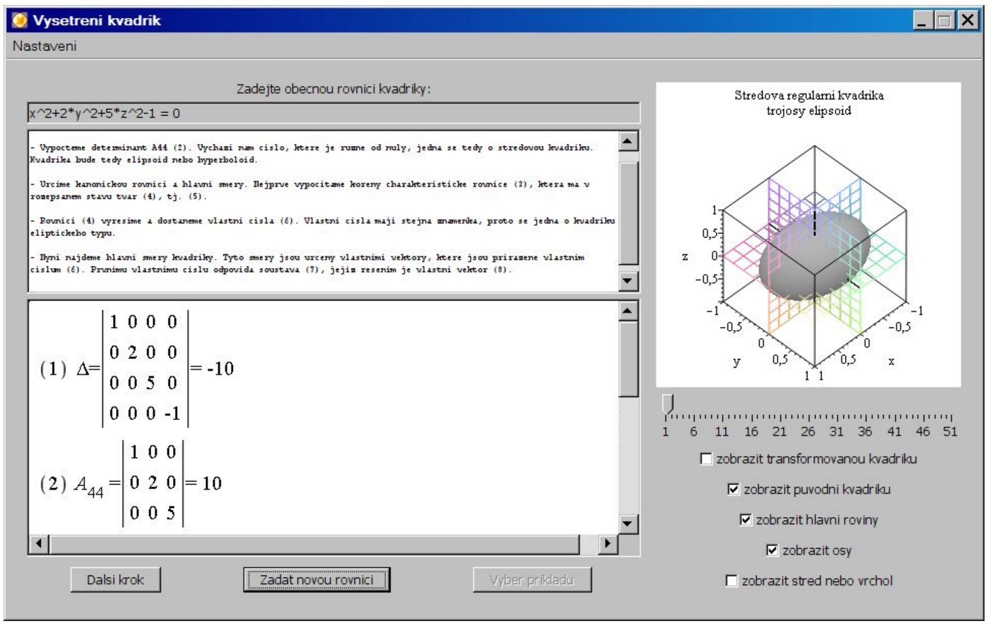
Interactive Plot Builder. V prostředí *Standard worksheet* můžeme využít řadu jednoúčelových aplikací - mapletů, které jsou součástí systému Maple. Více informací získáme po zadání `?assistants`, resp. `?tutors`.



Obrázek 3.3: *Interactive Plot Builder* - asistent pro tvorbu 3D grafů

Maplet „Vyšetření kvadrik“

Pro podporu výuky křivek druhého stupně a jako doplněk této knihy naprogramoval student Pedagogické fakulty Jihočeské univerzity Marek Dvorožňák v roce 2008 tento maplet provádějící úplnou klasifikaci kvadrik krok za krokem.



Obrázek 3.4: Maplet *Vyšetření kvadrik*

Aplikace je dostupná zdrojovém CD knihy

Maplety mohou být spuštěny nezávisle na běhu programu Maple. Bohužel je však nutné, aby byl program Maple (verze 9.5 nebo vyšší) na počítači nainstalován. Maplety využívají jeho výpočetní jádro.

3.3 Modely vybraných ploch v Maple

Tato příloha je věnována užití Maple k odvození parametrických rovnic a grafickému znázornění vybraných kvadratických ploch. Uvedená řešení v Maple jsou inspirována příklady, které najdeme v knize na následujících stranách: str. 76 (rotační jednodílný hyperboloid), str. 90 (hyperbolický paraboloid) a str. 97 (rotační kuželová plocha).

1. Rotační jednodílný hyperboloid

Příklad: *Ukažte, že rotací přímky kolem osy, která je s přímkou mimoběžná, vznikne rotační jednodílný hyperboloid (Str. 76).*