Funkce a její vlastnosti

Zadávání funkce a její obory

Zadávání funkcí více proměnných je stejné jako u jedné proměnné

In[1]:=
```
f[x_, y_] := Sqrt[x y]
```

In[2]:=
```
f[3, 8]
```

Out[2]=
$$2\sqrt{6}$$

In[3]:=
```
f[2, a]
```

Out[3]=
$$\sqrt{2}\ \sqrt{a}$$

Také zjišťování definičního oboru a oboru hodnot probíhá obdobně jako u jedné proměnné.

In[4]:=
```
Reduce[z == f[x, y], z, Reals]
```

Out[4]=
$$((y < 0 \land x \leq 0) \lor y = 0 \lor (y > 0 \land x \geq 0)) \bigwedge z = \sqrt{x\,y}$$

In[5]:=
```
Reduce[z == f[x, y], x, y, Reals]
```

Out[5]=
$$z = 0 \lor (z > 0 \land (x < 0 \lor x > 0))$$

Pro hodnoty funkce ve více bodech najednou se používá místo příkazu Map (který se používá pro jednu proměnnou) příkaz MapThread nebo Thread. Body, ve kterých se hledají hodnoty se však zadávají nikoli jako posloupnost bodů, ale jako posloupnost prvních souřadnic bodů následovaná posloupností druhých souřadnic, atd. V následujícím příkladě tedy dostáváme hodnoty v bodech (2,4), (8,16), (5,5).

In[6]:=
```
MapThread[f, {{2, 8, 5}, {4, 16, 5}}]
```

Out[6]=
$$\left\{2\sqrt{2}, 8\sqrt{2}, 5\right\}$$

In[7]:=
```
MapThread[f, {{a, b, c}, {x, y, z}}]
```

Out[7]=
$$\left\{\sqrt{a\,x}\,,\ \sqrt{b\,y}\,,\ \sqrt{c\,z}\right\}$$

In[8]:=
```
MapThread[g, {{a, b, c}, {x, y, z}, {u, v, w}}]
```

Out[8]=
$$\{g(a,\ x,\ u),\ g(b,\ y,\ v),\ g(c,\ z,\ w)\}$$

Trochu jinak pracuje příkaz Thread. V předchozích příkladech musí být délka jednotlivých posloupností stejná, u Thread může být jedna konstantní.

In[9]:=
```
MapThread[f, {{2, 3}, 4}]
```

— *MapThread::mptd : Object 4 at position {2, 2} in MapThread[f, {{2, 3}, 4}] has only 0 of required 1 dimensions.* ≫

Out[9]=
$$\text{MapThread}[f,\ \{\{2,\ 3\},\ 4\}]$$

In[10]:=
```
Thread[f[{2, 3, 4}, 4]]
```

Out[10]=
$$\left\{2\sqrt{2}\,,\ 2\sqrt{3}\,,\ 4\right\}$$

Limita

Mathematica neumí počítat limity funkcí více proměnných. Normální zadání výpočtu limity vede k výpočtu podle jednotlivých souřadnic.

In[11]:=
```
Limit[x^4 Sin[x y^2], {x → 0, y → 1}]
```

Out[11]=
$$\left\{0,\ x^4 \sin(x)\right\}$$

In[12]:=
```
Limit[Sin[x y^2] / (x y^2), {x → 0, y → 0}]
```

Out[12]=
$$\{1,\ 1\}$$

In[13]:=
```
Limit[x ^2 / (x^2 + y^2), { x → 0, y → 0}]
```
Out[13]=
$$\{0,\ 1\}$$

Poslední příklad alespoň implikuje, že příslušná limita neexistuje. Použijeme-li výpočet limity dané podmínkou, program spočte limitu z limity podle pořadí proměnných v podmínce.

In[14]:=
```
Limit[x ^2 / (x^2 + y^2) /. y → 0, x → 0]
```
Out[14]=
$$1$$

In[15]:=
```
Limit[x ^2 / (x^2 + y^2) /. x → 0, y → 0]
```
Out[15]=
$$0$$

Asi jediný doporučený (matematicky ale nevhodný) způsob výpočtu limit je nakreslit graf v okolí limitního bodu a po optickém ,,zjištění" spojitosti funkce v onom bodě spočítat limitu některým z výše uvedených způsobů. Pokud se nám z grafu bude zdát, že limita neexistuje, musíme hledat aspoň dvě přiblížovací cesty k limitnímu bodu, která dají různé limity. Občas se tyto cesty dají z grafu odhadnout. Někdy pomůže zadat polární souřadnice (řekněme pro limitu v (0,0)) a zjistit, že limita pro r jdoucí k 0 závisí na úhlech. Ne vždy to pomůže, jak ukazuje druhý příklad.

In[16]:=
```
g[r_, t_] := x ^2 / (x^2 + y^2) /. { x → r Cos[t], y → r Sin[t]}
```
In[17]:=
```
Limit[g[r, t], r → 0]
```
Out[17]=
$$\cos^2(t)$$

In[18]:=
```
Clear[g];
g[r_, t_] := x ^2 y / (x^4 + y^2) /. { x → r Cos[t], y → r Sin[t]}
```
In[19]:=
```
Limit[g[r, t], r → 0]
```
Out[19]=
$$0$$

In[20]:=
```
Limit[x ^2 y / (x^4 + y^2) /. y → x^2, x → 0]
```
Out[20]=
$$\frac{1}{2}$$

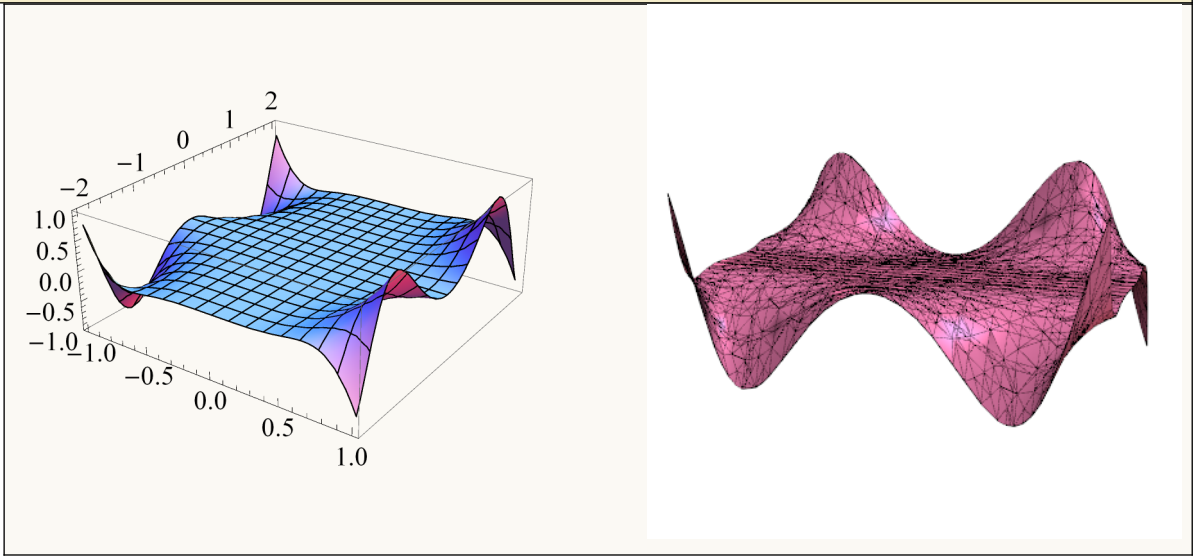
Grafy

Pro kreslení grafů funkce dvou proměnných se používá příkaz Plot3D. Většina voleb zůstává stejná jako u příkazu Plot. Výhodou trojdimenzionálních grafů v dobrých programech bývá možnost jejich otáčení. Podíváme se na grafy funkcí použité výše pro limity.

Následující trojdimenzioální (ve zkratce 3D) obrázky grafů bývají v pdf souboru uváděny až ve třech možnostech. Na prvním místě bývá graf automaticky převedný do pdf souboru jako 2D obrazek (nelze jej otáčet, zvětšovat apod.). Ten přesně vizuálně odpovídá grafu vytvořenému programem *Mathematica*.
Druhou možností je export obrazu z programu *Mathematica* do nějakého 3D formátu vhodného pro 3D vložení do pdf souboru. Zde nastanou potíže, protože při exportu se některé prvky grafu ztratí (např. osy souřadnic, barvy). Různými úpravami se něco dá napravit v 3D editorech, ale i tyto úpravy se mohou ztratit při importu do pdf. Výsledné 3D obrazy v pdf bývají proto trochu odlišné od těch původních vytvořených programem *Mathematica*. Nicméně, lze s nimi otáčet, zvětšovat je, upravovat barvy apod.
Předchozí situace je důvodem, proč bývá v pdf souboru ještě tlačítko s názvem "Web 3D obraz", které po stisknutí vytvoří na internetovém prohlížeči 3D obrázek podobnější tomu původnímu. Ale i v tomto případě je nutné počítat se změnami závisejícími na způsobu exportu 3D obrazu z programu *Mathematica*. Pokud se podaří vytvořit zdařilý export pomocí Javaview programu, je obrázek věrnější. Pokud je nutné použít export do java 3D obrazů, ztratí se např. osy souřadnic.

In[21]:=
```
Plot3D[x^4 Sin[x y^2], {x, -1, 1}, {y, -2, 2},
 PlotRange → {-1, 1}, ImageSize → {190, 190}]
```

Out[21]=



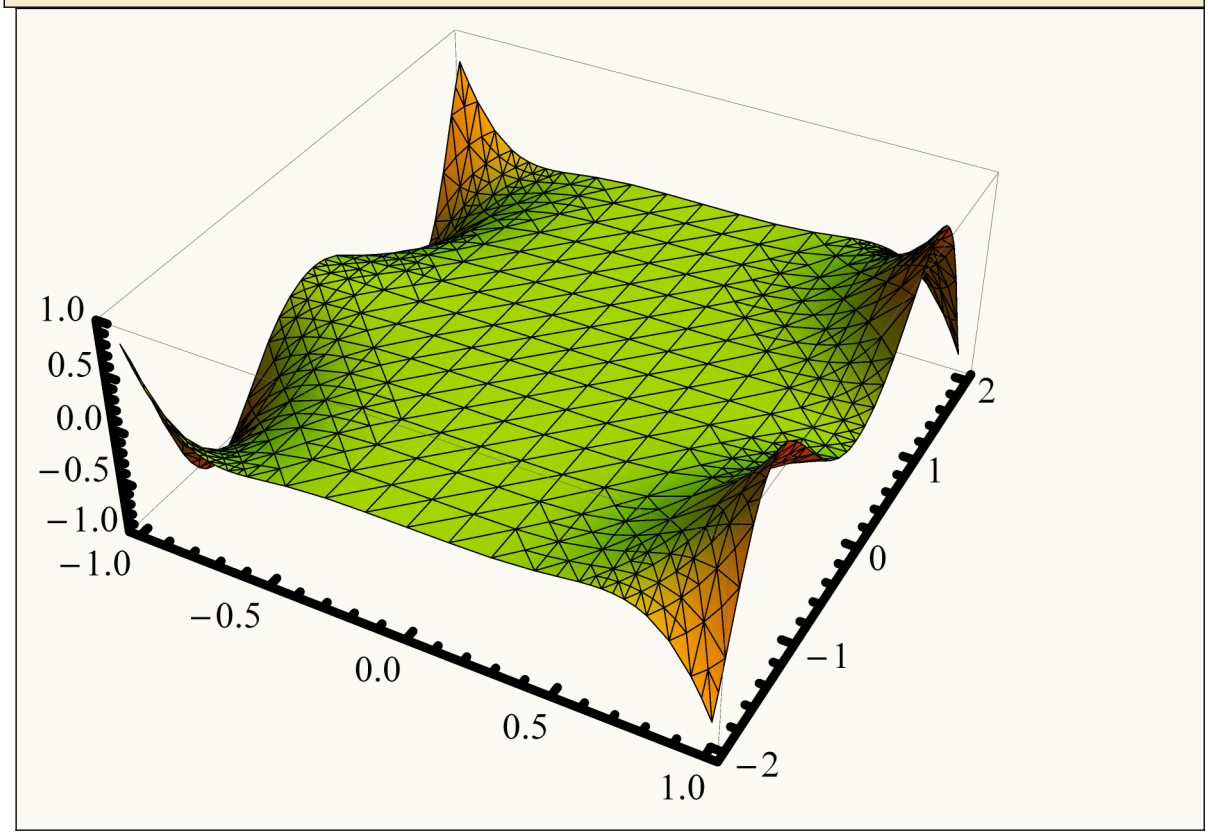
Web 3D obraz

Přidáme některé volby (jsou stejné nebo podobné těm pro dvoudimenzionální obrazy získané pomocí Plot).

In[30]:=

```
Plot3D[{x^4 Sin[x y^2]}, {x, -1, 1}, {y, -2, 2},
 PlotRange → {-1, 1}, Axes → True,
 AxesStyle → Directive[Black, 14, Thickness[.01]],
 PlotStyle → {Yellow}, Mesh → All]
```

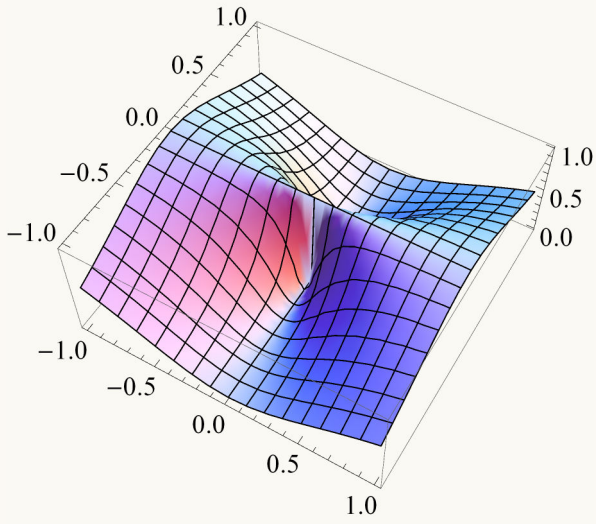
Out[30]=

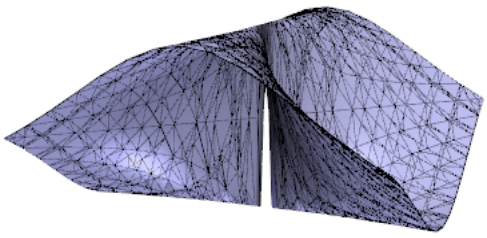


Z následujícího grafu bude vidět, že limita v (0,0) neexistuje. U dalšího grafu také limita v (0,0) neexistuje, ale dobře to vidět není. Teprve při pohledu svrchu si všimneme dvou parabol, které mají každá stejnou výšku (kromě bodů blízko 0, což je dáno délkou kroku při numerickém výpočtu hodnot funkce).

In[31]:=
```
Plot3D[x ^2 / (x^2 + y^2), {x, -1, 1}, {y, -1, 1},
 ImageSize → {220, 220}]
```

Out[31]=

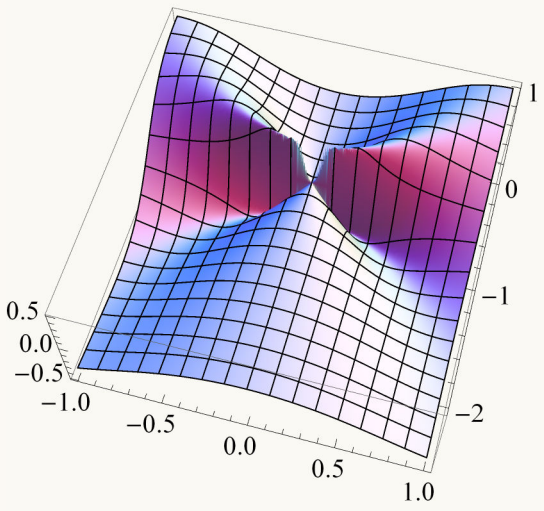


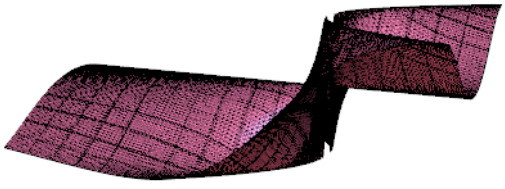


Web 3D obraz

In[23]:=
```
Plot3D[x ^2 y / (x^4 + y^2), {x, -1, 1}, {y, -2, 1},
 PlotPoints → 100, ImageSize → {200, 200}]
```

Out[23]=





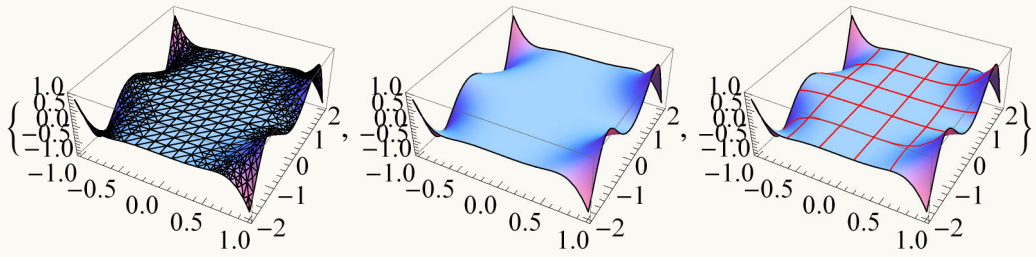
Web 3D obraz

Jednou z vhodných voleb je Mesh, která určuje ,,síťování" plochy. Lze zvolit i barvu sitě.

In[24]:=
```
{p1 = Plot3D[x^4 Sin[x y^2], {x, -1, 1}, {y, -2, 2},
    PlotRange → {-1, 1}, Mesh → All, ImageSize -> 120],

 p2 = Plot3D[x^4 Sin[x y^2], {x, -1, 1}, {y, -2, 2},
    PlotRange → {-1, 1}, Mesh → None,
    ImageSize → 120] ,
 p3 = Plot3D[x^4 Sin[x y^2], {x, -1, 1}, {y, -2, 2},
    PlotRange → {-1, 1}, Mesh → 4, MeshStyle -> Red,
    ImageSize → 120]}
```

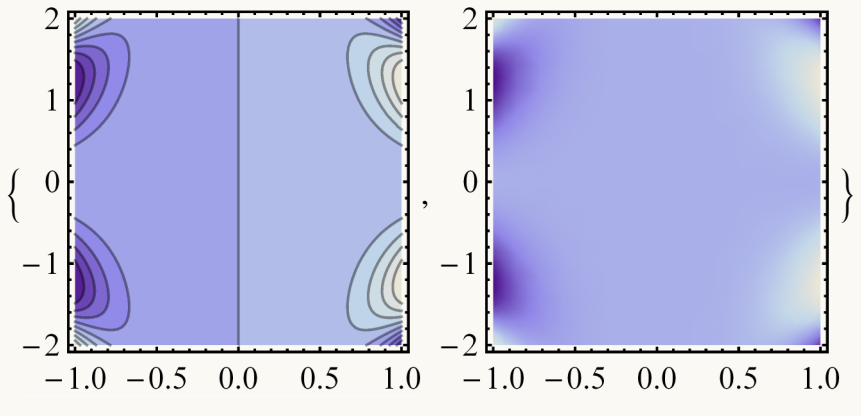
Out[24]=



Někdy se může hodit nakreslit plochu jako mapu buď s vrstevnicemi (ContourPlot) nebo pomocí ,,hustoty" (DensityPlot - tmavší znamená větší růst funkce, světleší znamena větší klesání funkce směrem od středu). Příkaz ContourPlot3D vykreslí i plochu zadanou implicitně.

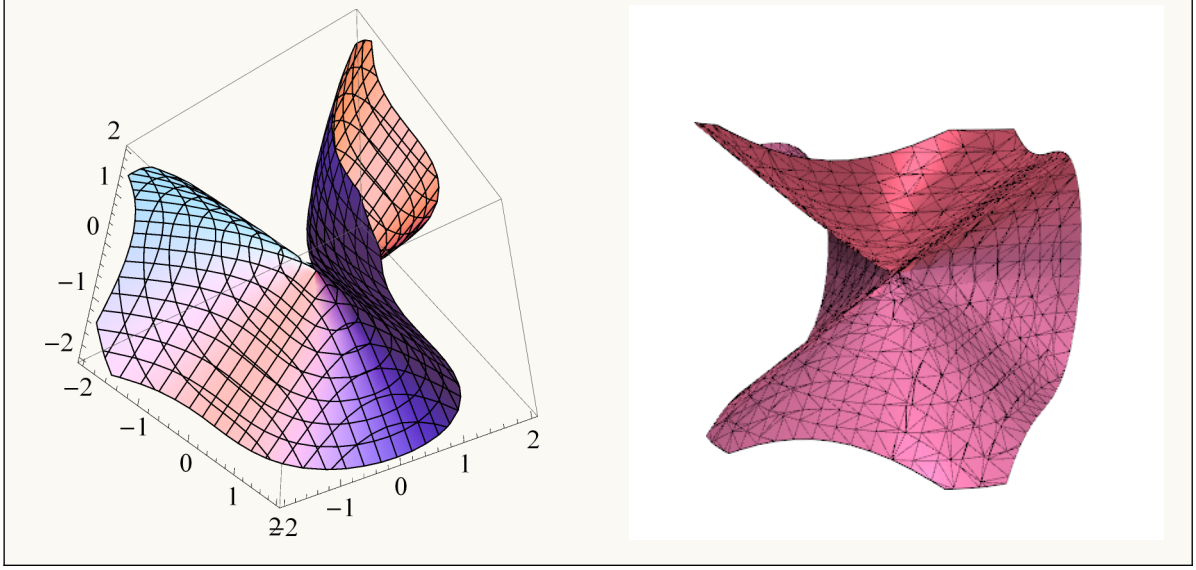
In[25]:=
```
{ContourPlot[x^4 Sin[x y^2], {x, -1, 1}, {y, -2, 2},
  PlotRange → {-1, 1}, ImageSize → 150],
 DensityPlot[x^4 Sin[x y^2], {x, -1, 1}, {y, -2, 2},
  PlotRange → {-1, 1}, ImageSize → 150]}
```

Out[25]=



In[26]:=
```
ContourPlot3D[x^3 - 2 y^2 + 3 z^2 == 0, {x, -2, 2},
 {y, -2, 2}, {z, -2, 2}, ImageSize → {200, 200}]
```

Out[26]=

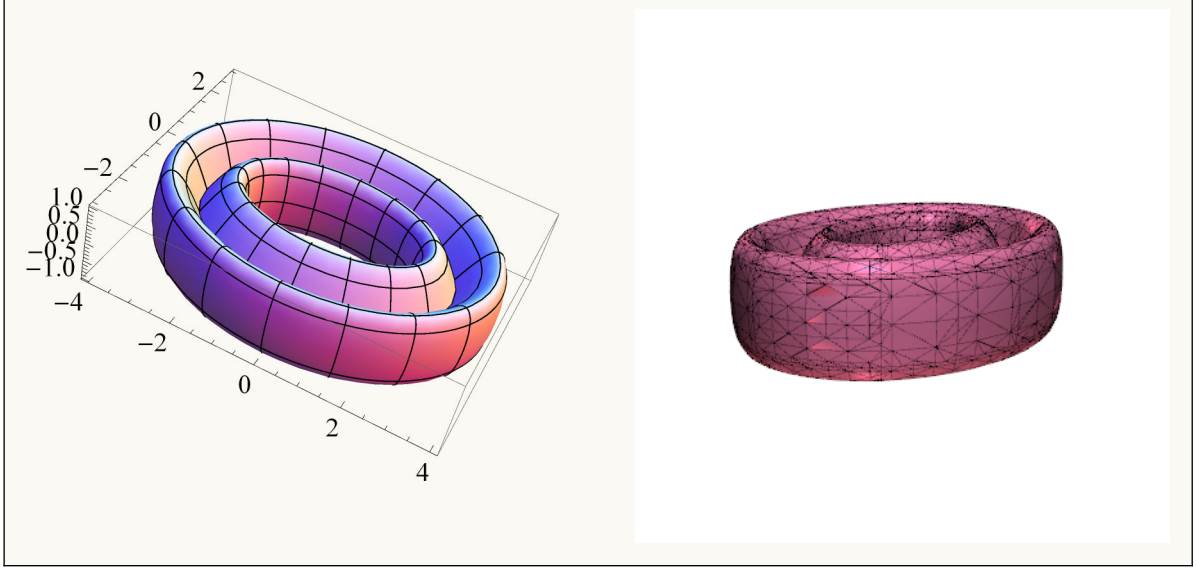


Web3D obraz

Větší bohatství ploch se získá pomocí parametrického zadání. V příkazech programu se najdou i RevolutionPlot3D, SphericalPlot3D, RegionPlot3D a další možnosti.

In[27]:=
```
ParametricPlot3D[{(3 + Cos[v]) Cos[u], (2 + Cos[v]) Sin[u],
  Sin[2 v]}, {u, 0, 2 Pi}, {v, 0, 2 Pi}, ImageSize → {200, 200}]
```

Out[27]=

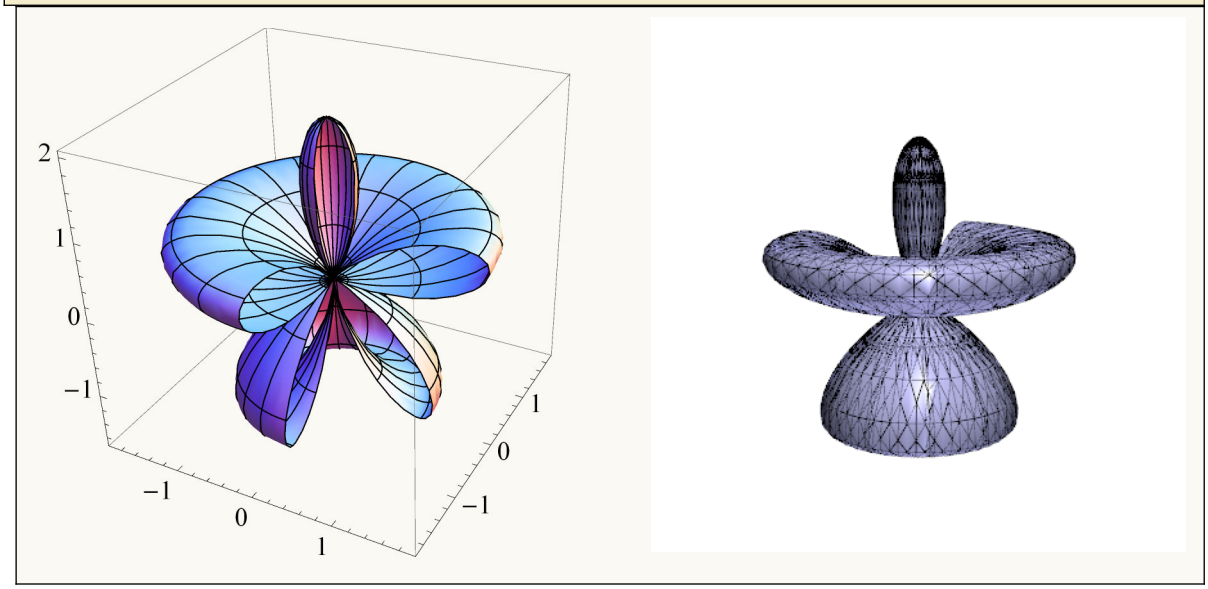


Web 3D obraz

In[28]:=

```
SphericalPlot3D[1 + Cos[5 θ], {θ, 0, Pi}, {ϕ, 0, 3 / 2 Pi},
 ImageSize → {200, 200}]
```

Out[28]=



Web 3D obraz