

Gradient, derivace ve směru

Gradient

Příkaz Gradient je v programu *Mathematica* používán trochu v jiném smyslu, než jsme zavedli (i když základ pojmu je stejný). Gradient námi definovaný se počítá příkazem [Grad](#).

```
Needs["VectorAnalysis`"]
```

```
f[x_, y_, z_] := x^3 y^2 - 5 y^4 + x y z
```

```
Grad[f[x, y, z], Cartesian[x, y, z]]
```

```
{3 x^2 y^2 + y z, 2 x^3 y + x z - 20 y^3, x y}
```

```
Grad[f[x, y, z], Cartesian[x, y, z]] /. {x -> 1, y -> 2, z -> 3}
```

```
{18, -153, 2}
```

Velikost gradientu je odmocnina ze skalárního součinu gradientu se sebou.

```
Grad[f[x, y, z], Cartesian[x, y, z]].
```

```
Grad[f[x, y, z], Cartesian[x, y, z]]
```

```
Sqrt[%] /. {x -> 1, y -> 2, z -> 3}
```

```
(2 x^3 y + x z - 20 y^3)^2 + (3 x^2 y^2 + y z)^2 + x^2 y^2
```

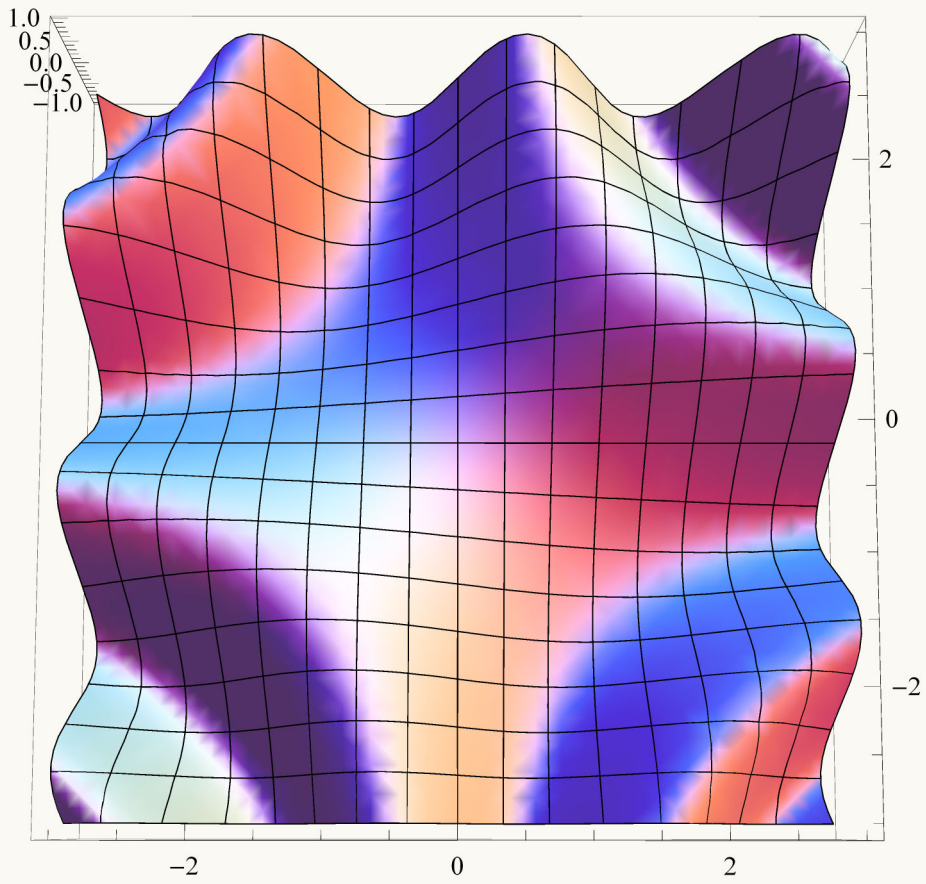
```
 $\sqrt{23737}$ 
```

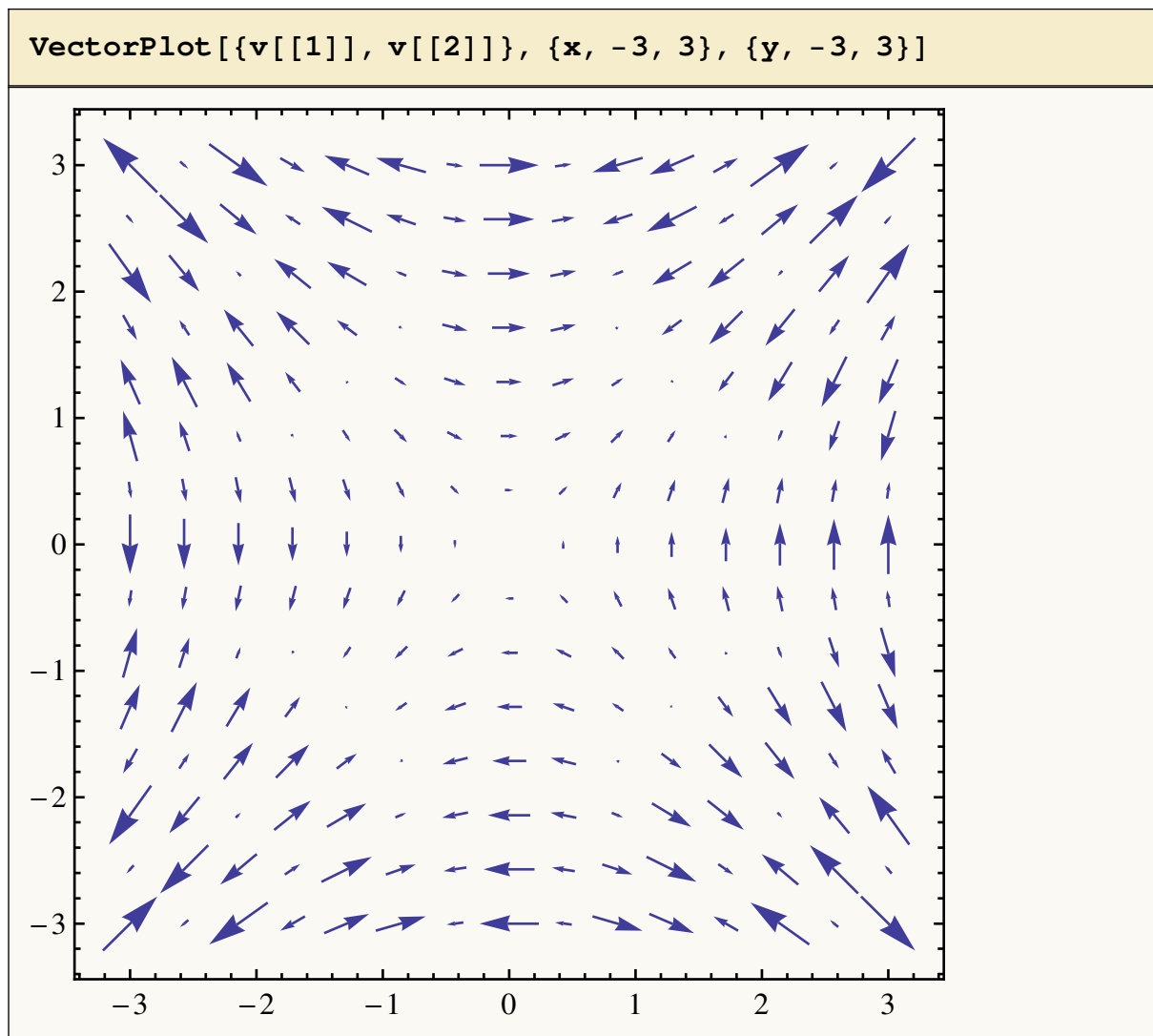
Někdy je vhodné nakreslit vektorové pole gradientu. V našem případě udává gradientový vektor směr největšího růstu grafu funkce v daném bodě. Musíme vzít funkci dvou proměnných a upravit gradient ze tří na dvě proměnné (příkaz [Cartesian](#) pracuje jen pro tři proměnné).

```
h[x_, y_] := Sin[x y]
```

```
v = Grad[h[x, y], Cartesian[x, y, z]]
```

```
graf = Plot3D[h[x, y], {x, -3, 3}, {y, -3, 3}]
```





Derivace ve směru

Derivace funkce více proměnných ve směru daném vektorem je skalární součin gradientu dané funkce a jednotkového vektoru v daném směru.

```
f[x_, y_, z_] := x^3 y^2 - 5 y^4 + x y z
u = {1, -1, 2};
bod = {2, 1, -1};
```

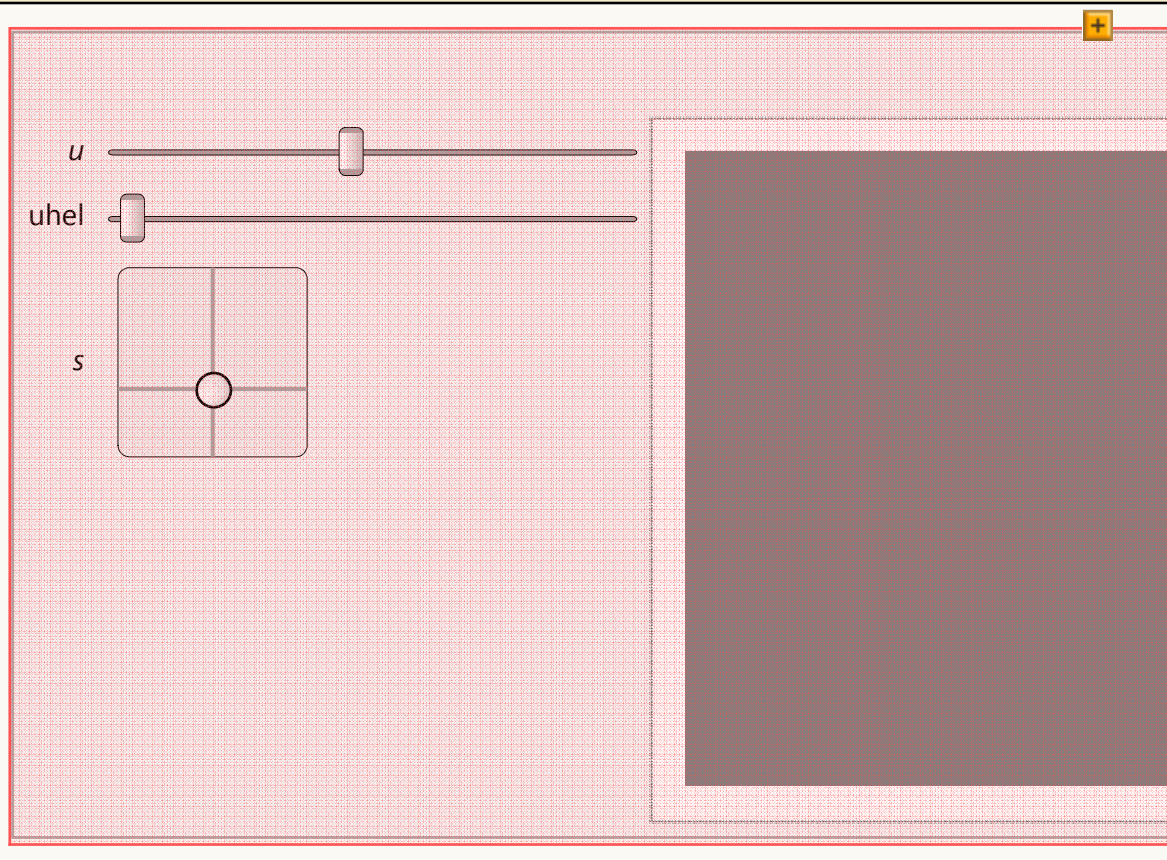
smerder =

```
{Grad[f[x, y, z], Cartesian[x, y, z]] /.  
  {x -> 2, y -> 1, z -> -1} } . u / Sqrt[u.u]
```

$$\left\{ 7 \sqrt{\frac{3}{2}} \right\}$$

Znázorníme situaci na grafech, ale musíme použít funkci dvou proměnných.

```
f[x_, y_] := Sin[x y];
Manipulate[{a, b} = {Cos[uhel], Sin[uhel]};
  smernice =
    ({D[f[x, y], x], D[f[x, y], y]}.{a, b} /.
      {x → a*u + s[[1]], y → b*u + s[[2]]});
  Show[{Plot3D[f[x, y], {x, -3, 3}, {y, -3, 3},
    PlotStyle → {LightBlue, Opacity[.5]}, Mesh → None,
    PlotRange → {-1, 1}, AxesLabel → {x, y, z}},
    ParametricPlot3D[{a t + s[[1]], b t + s[[2]],
      f[a t + s[[1]], b t + s[[2]]}], {t, -10, 10},
    PlotStyle → {Thick, Darker[Green, 1]}],
    ParametricPlot3D[{a t, b t, smernice t} +
      {a u + s[[1]], b u + s[[2]], f[a u + s[[1]], b u + s[[2]]}],
      {t, -1, 1}, PlotStyle → {Red, Thick}]]],
  {{u, -.5}, -6, 6}, {{uhel, .6}, 0, 2 Pi}
, {{s, {0, -1}}, {-3, -3}, {3, 3}},
  ControlType → {Slider, Slider, Slider2D},
  ControlPlacement → Left]
```



Gradient implicitní funkce

Gradient funkce $z=f(x,y)$ určuje směr největšího růstu grafu funkce, kdežto gradient implicitní funkce určuje normálu ke grafu funkce (vezmeme stejnou plochu určenou rovnicí $f(x,y)-z=0$). Směr největšího růstu znázorníme tečnou ke grafu nad gradientem (ten leží v rovině xy).

```
h[x_, y_, z_] := Sin[x y]; bod0 = {0.5, 1};  
bod = {bod0[[1]], bod0[[2]], h[bod0[[1]], bod0[[2]], 0] // N};
```

```
v = Grad[h[x, y, z], Cartesian[x, y, z]]; vj = v / Sqrt[v.v];  
w = Grad[h[x, y, z] - z, Cartesian[x, y, z]]; wj = w / Sqrt[w.w];  
p = {x -> bod[[1]], y -> bod[[2]]};  
v0 = v /. p // N; v1 = vj /. p // N; w1 = wj /. p // N;  
konec1 = bod + v1 + {0, 0, v1.v0};  
konec2 = bod + w1;
```

```

tecna = Graphics3D[Arrow[{bod, konec1}]];
normala = Graphics3D[Arrow[{bod, konec2}]];
graf = Plot3D[h[x, y], {x, -3, 3}, {y, -3, 3},
  PlotStyle → Opacity[.7], Mesh → 2, PlotRange → {-2, 2},
  BoxRatios → {1, 1, .7}];
tecka = Graphics3D[{PointSize[Large], Red, Point[bod]}];
Show[graf, tecna, normala, tecka]

```

