Obyčejné diferenciální rovnice

Program *Mathematica* umí řešit diferenciální rovnice, které jsou probrány v příslušném textu skript. Podíváme se i na numerické řešení složitějších diferenciálních rovnic. Přesná řešení se získají pomocí příkazu DSolve, přibližná pomocí NDSolve.

ODR obecně

Základní zápis je následjící. Je nutné psát dvojí rovnost ==, protože se jedná o zjišťování rovnosti dvou funkcí. Výsledek je dán jako seznam přiřazení (rules). Výsledek ve tvaru funkce lze z přiřazení dostat, ale ověření spravnosti řešení rovnice je složitější.

```
Clear[y]; DSolve[ y'[x] y[x] + x == 0, y[x], x]
```

$$\left\{\left\{y(x)\to -\sqrt{2\,c_1-x^2}\right\},\left\{y(x)\to \sqrt{2\,c_1-x^2}\right\}\right\}$$

Pro další postup je lepší si rovnici označit. Dosazením zjistíme, že program ví, co je y[x], ale už neví, co je `y'[x]`.

```
Clear[y]; odr = y'[x] y[x] + x == 0
```

$$y(x)\,y'(x)+x=0$$

```
reseni = DSolve[odr, y[x], x]
```

$$\left\{\left\{y(x)\to -\sqrt{2\,c_1-x^2}\right\},\left\{y(x)\to \sqrt{2\,c_1-x^2}\right\}\right\}$$

Zkusme dosadit řešení do rovnice.

```
odr /. reseni
```

$$\left\{x-\sqrt{2\,c_1-x^2}\;\;y'(x)=0,\;\sqrt{2\,c_1-x^2}\;\;y'(x)+x=0\right\}$$

Zkusme tedy vzít jen jedno řešení a znovu dosadit.

```
reseni1 = reseni[[1]]
```

$$\left\{y(x) \to -\sqrt{2\,c_1 - x^2}\right\}$$

```
odr /. reseni1
```

$$x - \sqrt{2\,c_1 - x^2}\;\; y'(x) = 0$$

Místo přiřazení vezmeme řešení jako funkci a ještě dejme za konstantu číslo. Opět nedostaneme ověření správnosti řešení.

```
y1[x_] = y[x] /. reseni1
```

$$-\sqrt{2\,c_1 - x^2}$$

```
y1[x_] = y1[x] /. C[1] → 2
```

$$-\sqrt{4 - x^2}$$

```
odr /. y[x] → y1[x]
```

$$x - \sqrt{4 - x^2}\;\; y'(x) = 0$$

Všimněme si, že z y(x) nelze získat hodnoty ani derivaci. Z funkce y1(x) už ano.

```
y[2]
```

$$y(2)$$

```
D[y[x], x]
```

$$y'(x)$$

```
y1[2]
```

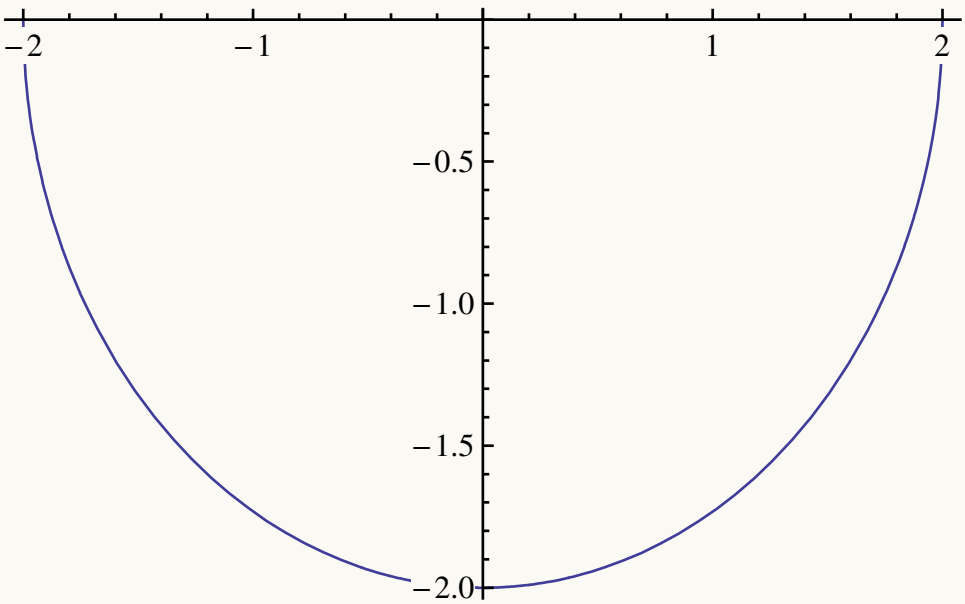
$$0$$

```
D[y1[x], x]
```

$$-\frac{x}{\sqrt{4-x^2}}$$

Graf řešení nakreslit tedy můžeme s pomocí funkce y1(x).

```
Plot[y1[x], {x, -2, 2}]
```



Řešení rovnice s počáteční podmínkou lze získat přímo dodáním podmínky do rovnice. Program nás upozorní, že druhý tvar obecného řešení nelze použít.

```
DSolve[{odr, y[0] == -2}, y[x], x]
```

— *DSolve::bvnul : For some branches of the general solution, the given boundary conditions lead to an empty solution.* ≫

$$\left\{\left\{y(x) \to -\sqrt{4-x^2}\right\}\right\}$$

Někdy bývá vhodnější řešit rovnici nikoli pro y[x], ale jen pro y. Výsledkem jsou tzv. “pure” funkce. V tomto případě lze snadno ověřit správnost řešení.

```
res = DSolve[odr, y, x]
```

$$\left\{\left\{y \to \left(\{x\} \mapsto -\sqrt{2\,c_1-x^2}\right)\right\}, \left\{y \to \left(\{x\} \mapsto \sqrt{2\,c_1-x^2}\right)\right\}\right\}$$

```
odr /. res
```

{True, True}

Řešení s počáteční podmínkou získáme stejným způsobem jako dříve. Zadáním substituce do funkce lze získat její hodnoty.

```
res1 = DSolve[{odr, y[0] == -2}, y, x]
```

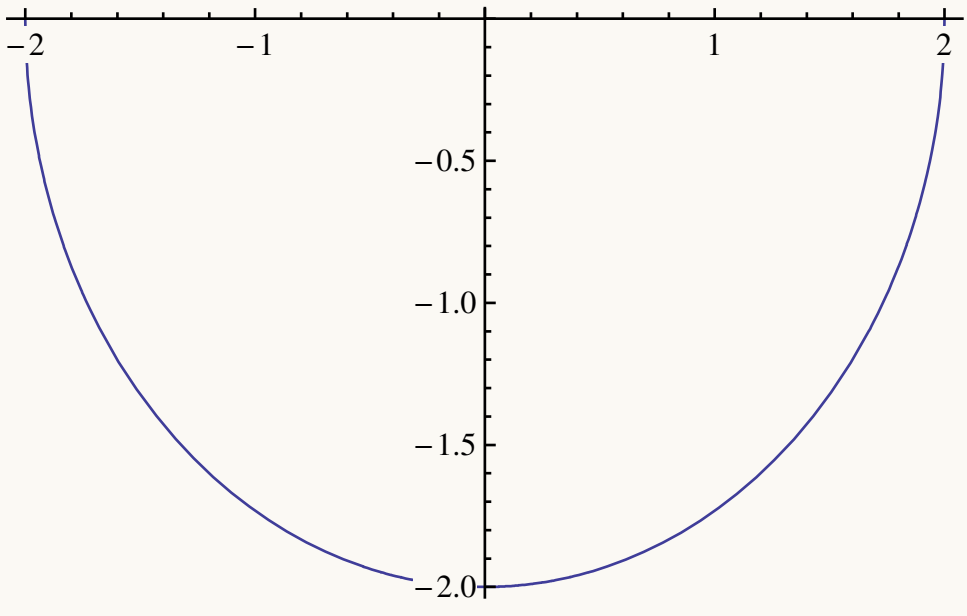
— *DSolve::bvnul : For some branches of the general solution, the given boundary conditions lead to an empty solution.* ≫

$$\left\{\left\{y \to \left(\{x\} \mapsto -\sqrt{4-x^2}\right)\right\}\right\}$$

```
y[1] /. res1
```

$$\left\{-\sqrt{3}\right\}$$

```
Plot[y[x] /. res1, {x, -2, 2}]
```

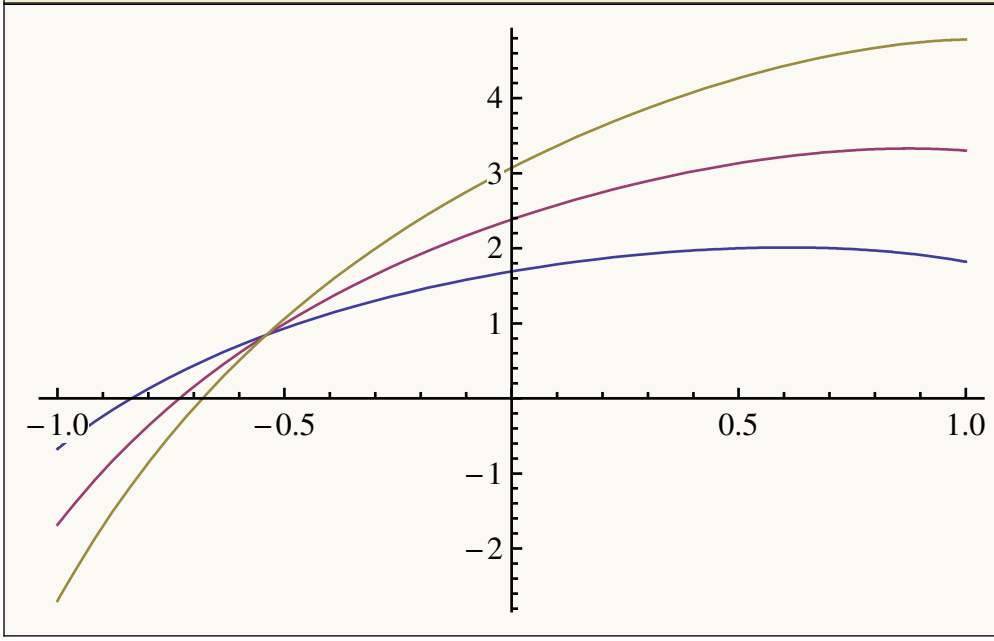


V případě složitějších rovnic může být výsledek popsán pomocí ,,neelemntárních funkcí". Průběh řešení lze nakreslit stejným způsobem jak bylo popsáno výše. Ukážeme několik řešení najednou, nejdříve pro volbu první konstanty, pak druhé a pak obě dohromady.

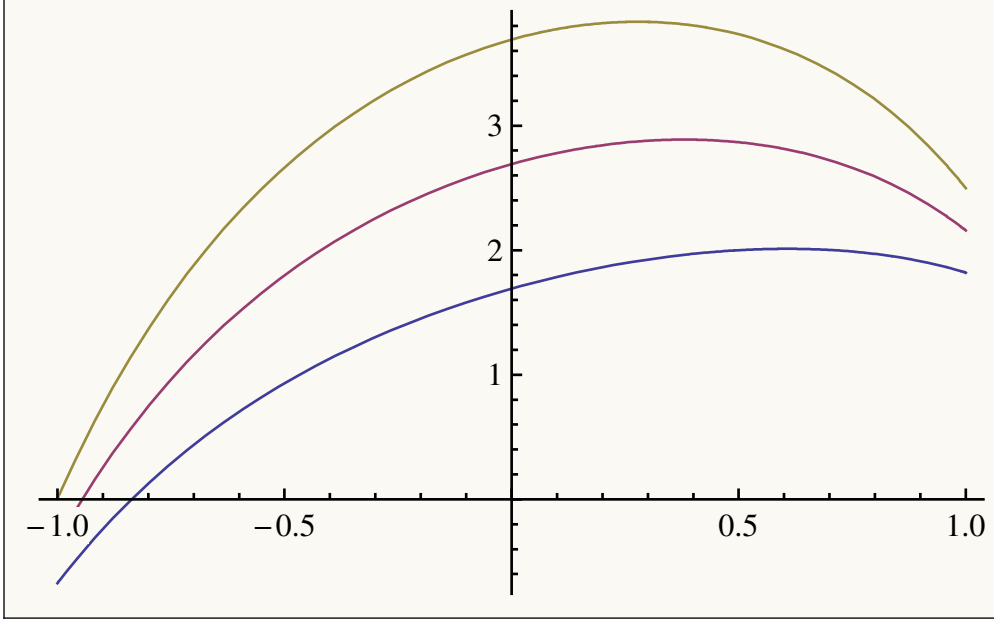
```
rr = DSolve[{y''[x] - 2 x y'[x] + y[x] == 0}, y, x]
```

$$\left\{\left\{y \to \left(\{x\} \longmapsto c_1 H_{\frac{1}{2}}(x) + c_2 \, {}_1F_1\left(-\frac{1}{4}; \frac{1}{2}; x^2\right)\right)\right\}\right\}$$

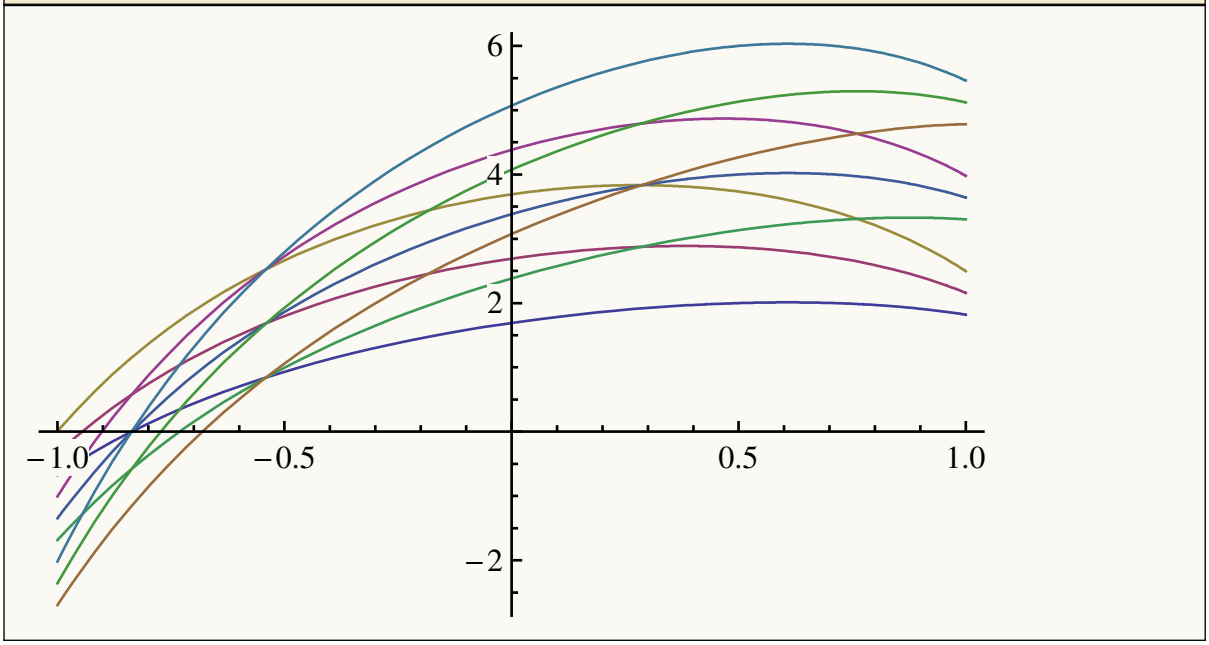
```
Plot[Evaluate[Table[y[x] /. rr /. {C[1] → k, C[2] → 1},
    {k, 1, 3, 1}, {l, 1, 1, 1}]], {x, -1, 1}]
```




```
Plot[Evaluate[Table[y[x] /. rr /. {C[1] → k, C[2] → 1},
    {k, 1, 1, 1}, {l, 1, 3}]], {x, -1, 1}]
```

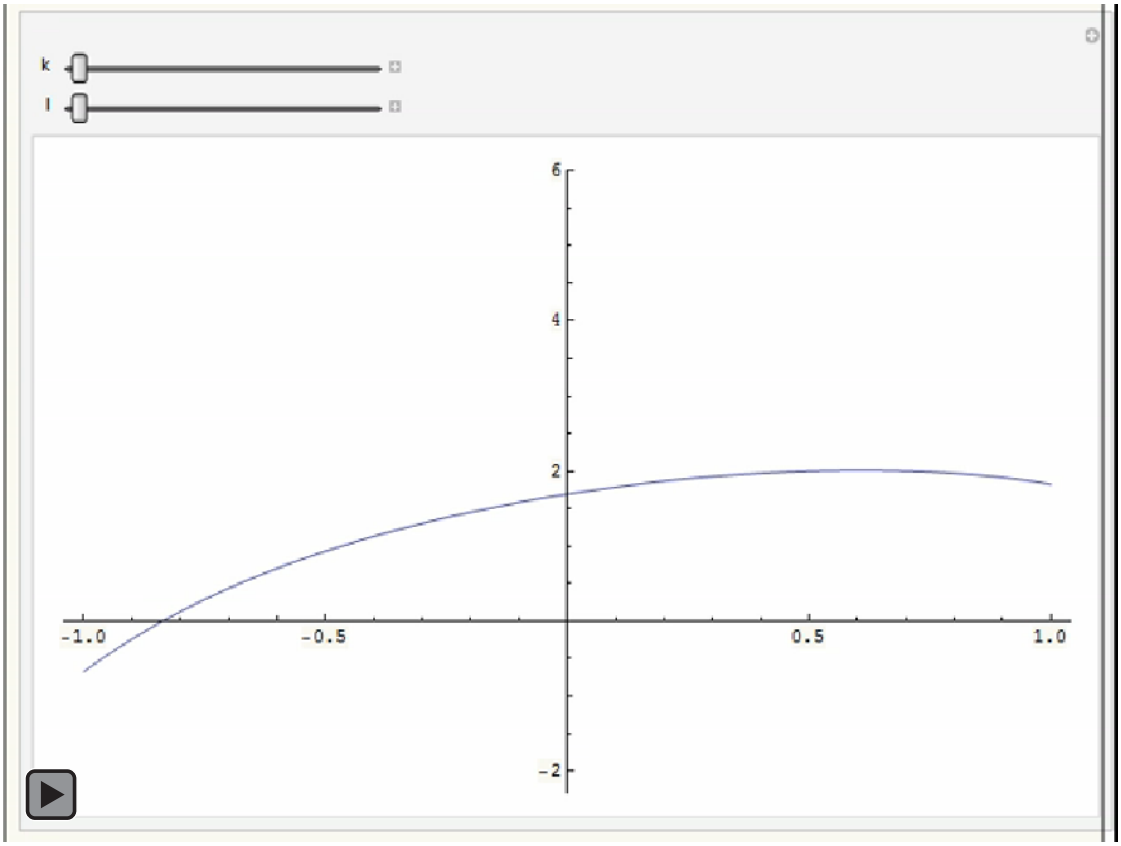


```
Plot[Evaluate[Table[y[x] /. rr /. {C[1] → k, C[2] → l},
    {k, 1, 3, 1}, {l, 1, 3, 1}]], {x, -1, 1}]
```



Po,ocí animace lze zobrazit pohyb řešení při změně jeho konstant.

```
Manipulate[Plot[Evaluate[y[x] /. rr /. {C[1] → k, C[2] → l}],
  {x, -1, 1}, PlotRange → {-2.3, 6}], {k, 1, 3, .5},
 {l, 1, 3, .5}]
```



Numerické řešení

Pokud nedovedeme najít přesné řešení (seznam rovnic, které umí *Mathematica* řešit přesně, tj. napíše výsledek pomocí známých funkcí, se najde v oddělení Overview of Ordinary Differential Equations (ODEs) v Help), musíme hledat přibližné řešení pomocí NDSolve. Výsledkem je tzv. interpolační funkce, která je vypočtena jen v některých bodech a vhodně doplněna v ostatních bodech na hladkou funkci (mající derivace).

Následující rovnici nelze vyřešit přesně a příkaz DSolve musel být přerušen. Je nutné použít numerický přístup. Nejdříve uvedeme směrové pole, pak jedno řešení a to vložíme do směrového pole.

In[97]:=
```
odr2 = y'[x] == Sin[x] + Cos[y[x]]
```

Out[97]=

$$y'(x) = \cos(y(x)) + \sin(x)$$

In[113]:=
```
DSolve[odr2, y[x], x]
```

— Solve::ifun :

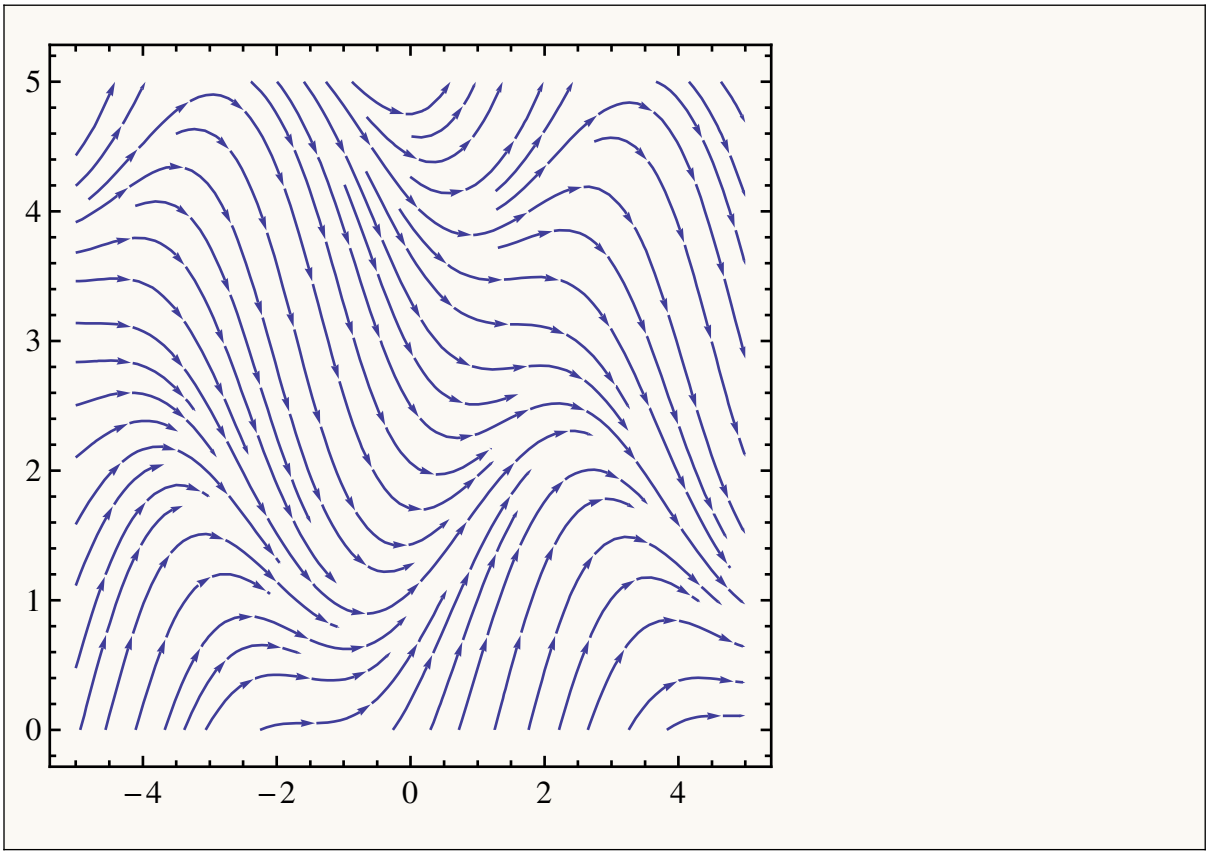
Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information. >>

Out[113]=

$Aborted

In[114]:=
```
sp2 = StreamPlot[{1, Sin[x] + Cos[y]}, {x, -5, 5}, {y, 0, 5}]
```

Out[114]=



In[99]:=
```
res2 = NDSolve[{odr2, y[0] == 2}, y, {x, -5, 5}]
```

Out[99]=

$\{\{y \to \text{InterpolatingFunction}[(-5.\ \ 5.\ ), <>]\}\}$

Následující příkaz First vybere ze seznamu první výraz, s kterým pak lze lépe pracovat.

In[100]:=
```
resa2 = First[y /. NDSolve[{odr2, y[0] == 2}, y, {x, -5, 5}]]
```

Out[100]=

$\text{InterpolatingFunction}[(-5.\ \ 5.\ ), <>]$

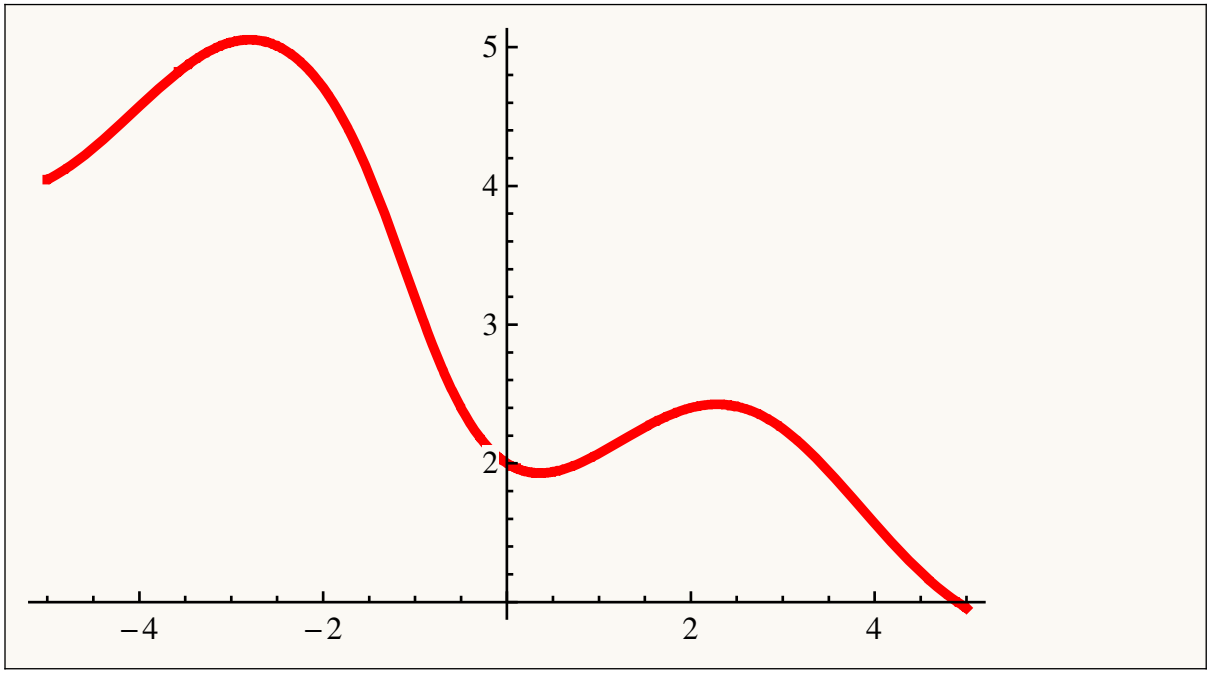
In[101]:=
```
resa2[3]
```

Out[101]=

2.25022

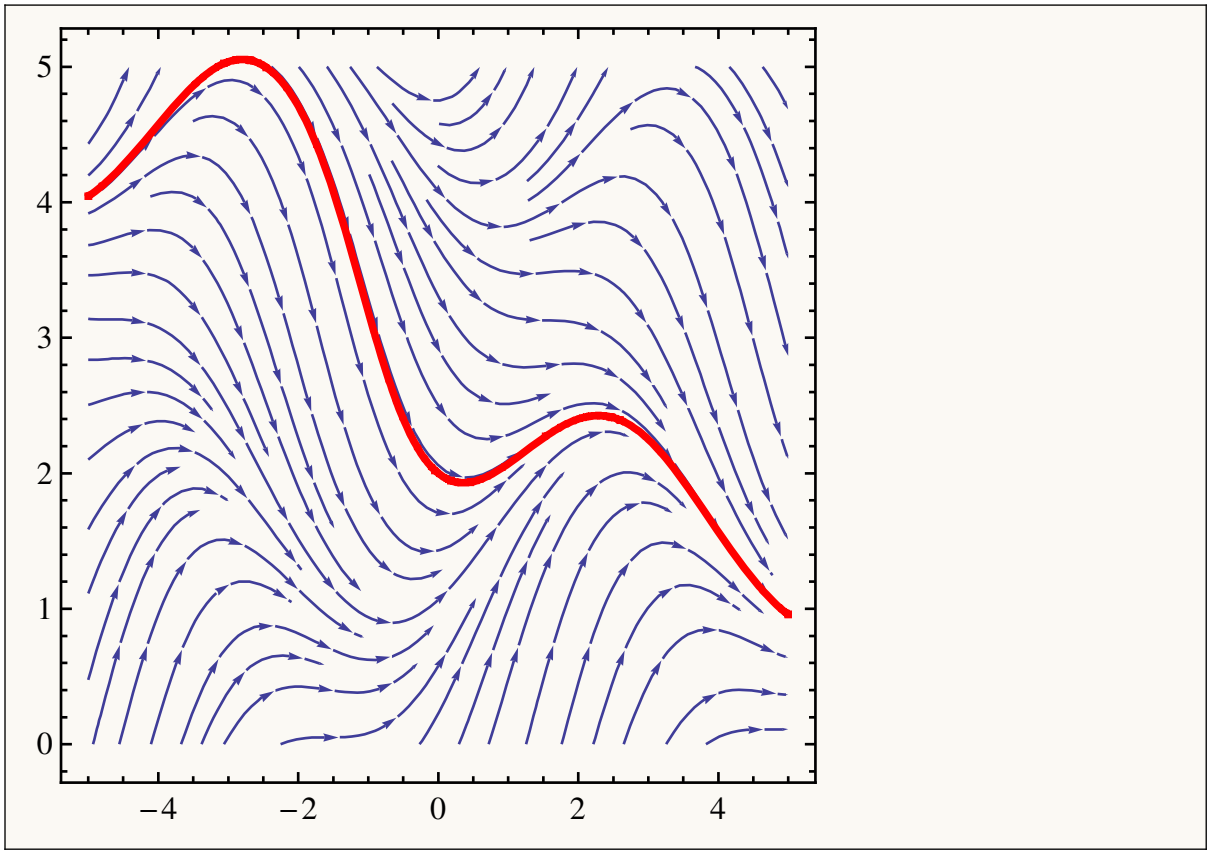
In[102]:=
```
traj2 = Plot[Evaluate[y[x] /. res2], {x, -5, 5},
  PlotStyle → {Red, Thickness[.01]}]
```

Out[102]=



In[127]:=
```
Show[sp2, traj2, ImageSize → {300, 300}]
```

Out[127]=



Animace přibližných řešení nelze provést stejně jako u přesných řešení nanoře. Lze použít následující postup.

```
Clear[re]; For [k = 1, k < 10, k++,
 re[k] = First[y /. NDSolve[{odr2, y[0] == k}, y, {x, -5, 5}]]]
```

```
Manipulate[Plot[re[k][x], {x, -5, 5}, PlotRange → {-2, 12}],
 {k, 1, 9, 1}]
```