Diferenciální rovnice prvního řádu

Robert Mařík

27. listopadu 2010

1 Diferenciální rovnice

Zde naleznete příklady a příkazy

- na nakreslení směrového pole diferenciální rovnice
- na numerickou aproximaxci metodou Runge Kutta
- na hledání obecného řešení diferenciální rovnice analytickou cestou
- na hledání řešení počáteční úlohy analytickou cestou

1.1 Směrové pole a numerické řešení

Nadefinujeme proměnné, meze pro kreslení a pravou stranu diferenciální rovnice

Sage code

```
x,y = var('x y')
(x_min, x_max, y_min,y_max)=(0,2,0,4)
f(x,y)=y*(3-y)
html('$y\,^\prime=%s$'%(latex(f(x,y))))
```

$y' = -(y-3)y$

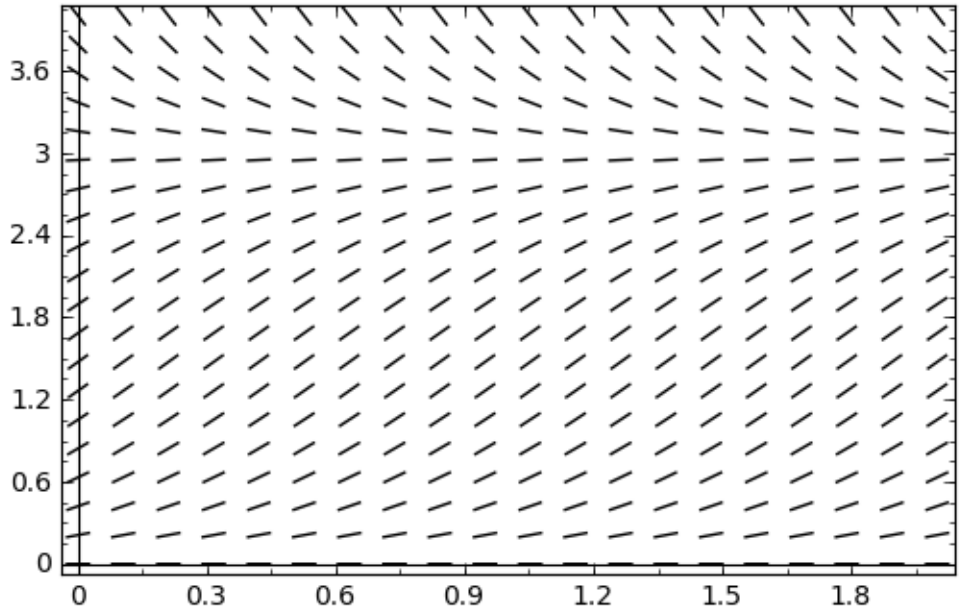
Následující příkazy uloží směrové pole do proměnné A a poté toto pole nakreslí.

Sage code

```
A=plot_slope_field(f(x,y),(x, x_min, x_max), (y, y_min, y_max))
html(r'$y\,^{\prime}=%s$'%latex(f(x,y)))
show(A)
```

$y' = -(y-3)y$

[0]Podporováno grantem FRVŠ 131/2010.



1.2 Numerická aproximace

Definujeme numerickou aproximaci metodou Runge Kutta

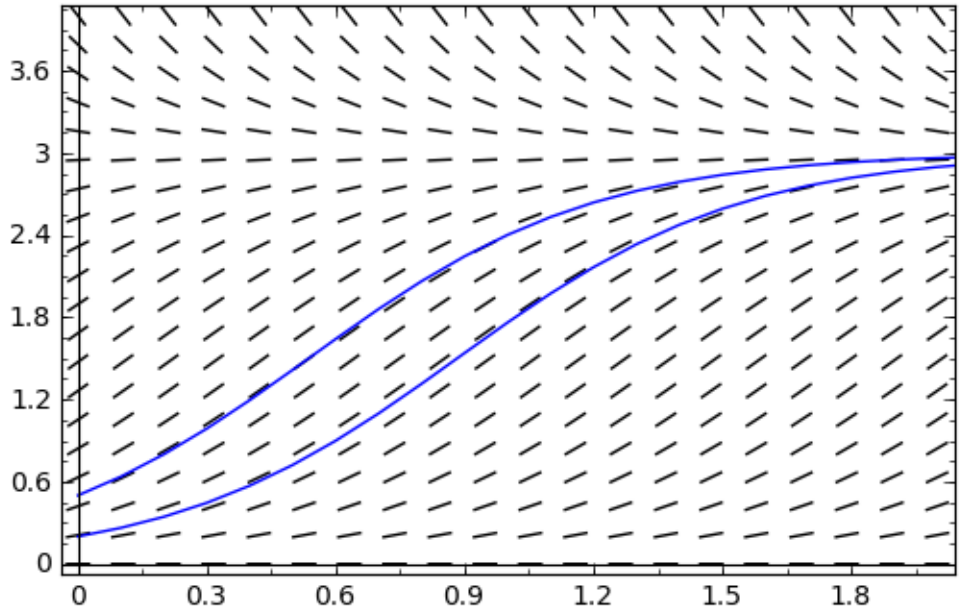
Sage code

```
x,y = var('x y')
def rk(fnc,x0,y0,krok,x1):
    g=fast_float(fnc,'x','y')
    n=int((1.0)*(x1-x0)/krok)
    x00=x0; y00=y0
    soln = [[x00,y00]]
    for i in range(n+1):
        k1 = g(x00,y00)
        k2 = g(x00+krok/2,y00+k1*krok/2)
        y00 = y00+krok*k2
        x00 = x00+krok
        soln.append([x00,y00])
    return soln
```

Vypočteme aproximace a nakreslíme dvě partikulární řešení

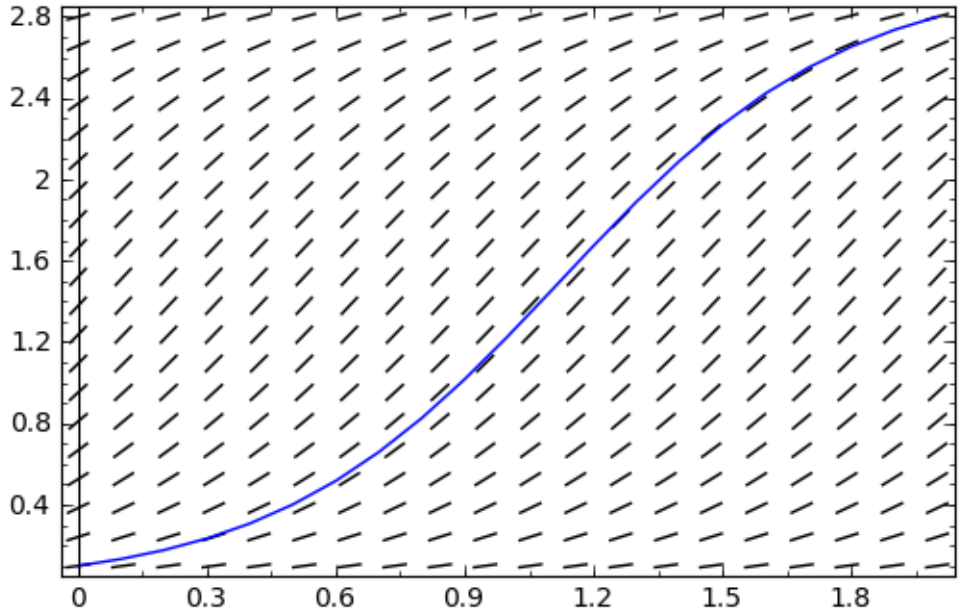
Sage code

```
f(x,y)=y*(3-y)
A=plot_slope_field(f(x,y),(x, x_min, x_max), (y, y_min, y_max))
B=list_plot(rk(f(x,y), 0, 0.2, 0.1, x_max), plotjoined=True)
C=list_plot(rk(f(x,y), 0, 0.5, 0.1, x_max), plotjoined=True)
show(A+B+C,ymin=y_min,ymax=y_max,xmin=x_min,xmax=x_max)
```



Sage code

```
f(x,y)=y*(3-y)
desolve_rk4(f(x,y),y,ics=[0,0.1],ivar=x,end_points=[0,2],output='slope_field')
```



1.3 Analytické řešení diferenciálních rovnic

Proměnná je x a proměnnou y chápeme jako funkci proměnné x.
K nalezení obecného řešení slouží příkaz `desolve`

Sage code

```
x = var('x')
y = function('y', x)
desolve(diff(y,x) == (2-y)*sin(x), y)
```

$(c + 2\,e^{-\cos(x)})e^{\cos(x)}$
Upravíme výsledek (funkce `expand` roznásobuje závorky a podtržnítko `_` značí předchozí výsledek)

Sage code

```
expand(_)
```

$ce^{\cos(x)} + 2$

Pokud nainstalujeme patch, můžeme si nechat vypsat typ diferenciální rovnice.

Sage code

```
desolve(diff(y,x) == (2-y)*sin(x), y,show_method=True)
```

$\left[(c + 2\,e^{-\cos(x)})e^{\cos(x)}, \text{linear}\right]$

K nalezení partikulárního řešení přidáme parametrics

Sage code

```
desolve(diff(y,x) == (2-y)*sin(x), y,ics=[0,3])
```

$(2\,e + e^{\cos(x)})e^{-1}$

Sage code

```
expand(_)
```

$e^{\cos(x)-1} + 2$

Diferenciální rovnice $y' = e^x e^{-y}$. Uvitř programu Sage se derivace funkce y podle x značí pro začátečníky poněkud matoucím způsobem: $D[0](y)(x)$. (Pokud si prohlížíte už vygenerované PDF tak vidíte přímo obvyklé $y'(x)$.)

Sage code

```
x = var('x')
y = function('y', x)
eq=diff(y,x) == exp(x)*exp(-y)
show(eq)
desolve(eq, y)
```

$$D[0]\left(y\right)\left(x\right) = e^{x-y(x)}$$

$-e^{x} + e^{y(x)} = c$

Převod řešení do explicitního tvaru

Sage code

```
solve(_,y)
```

$[y\left(x\right) = \log\left(c + e^{x}\right)]$

Seraprace proměnných, výsledkem je rovnice definující řešení implicitně. V některých případech jde nalézt explicitní řešení pomocí příkazu solve

Sage code

```
eq=y^2-1+y*diff(y,x)*(x^2-1)==0
eq
```

$(x^{2} - 1)y\left(x\right)D[0]\left(y\right)\left(x\right) + y\left(x\right)^{2} - 1 = 0$

Sage code

```
A=desolve(eq,y)
A
```

$-\frac{1}{2}\log\left(y\left(x\right) - 1\right) - \frac{1}{2}\log\left(y\left(x\right) + 1\right) = c + \frac{1}{2}\log\left(x - 1\right) - \frac{1}{2}\log\left(x + 1\right)$

Následující příkaz sloučí logaritmy,je ale potřeba vynásobit rovnici dvěma, abychom odstranili koeficienty u jednotlivých logaritmů

Sage code

```
A=maxima.logcontract(2*A).sage()
A
```

$-\log\left(y\left(x\right)^{2} - 1\right) = 2\,c + \log\left(\frac{(x-1)}{(x+1)}\right)$

Konstantu $2c$ zapíšeme ve tvaru $\ln(K)$.

Sage code

```
K,c=var('K c')
A.subs_expr(2*c==ln(K))
```

$$-\log\left(y\left(x\right)^{2}-1\right)=\log\left(K\right)+\log\left(\frac{\left(x-1\right)}{\left(x+1\right)}\right)$$

Explicitně vyjádříme y

Sage code

```
solve(_,y)
```

$$\left[y\left(x\right)=-\sqrt{\frac{Kx}{\left(Kx-K\right)}-\frac{K}{\left(Kx-K\right)}+\frac{x}{\left(Kx-K\right)}+\frac{1}{\left(Kx-K\right)}},y\left(x\right)=\sqrt{\frac{Kx}{\left(Kx-K\right)}-\frac{K}{\left(Kx-K\right)}+\frac{x}{\left(Kx-K\right)}+\frac{}{\left(Kx\right.}}\right.$$

Výsledek je moc dlouhý, raději každou položku v předchozím výsledku upravíme metodou `factor()` (rozklad na součin).

Sage code

```
[i.factor() for i in _]
```

$$\left[y\left(x\right)=-\sqrt{\frac{\left(\left(x-1\right)K+x+1\right)}{\left(x-1\right)K}},y\left(x\right)=\sqrt{\frac{\left(\left(x-1\right)K+x+1\right)}{\left(x-1\right)K}}\right]$$

Lineární diferenciální rovnice, vyjde jenom výraz, řešení je *y=výraz*

Sage code

```
eq=diff(y,x)+y==x^2
eq
```

$$y\left(x\right)+D[0]\left(y\right)\left(x\right)=x^{2}$$

Sage code

```
desolve(eq,y)
```

$$((x^{2}-2\,x+2)e^{x}+c)e^{-x}$$

Sage code

```
expand(_)
```

$$ce^{-x}+x^{2}-2\,x+2$$

Řešíme diferenciální rovnici $y'-\frac{2y}{x}=\frac{1}{x^{3}}$

Sage code

```
desolve(diff(y,x)-2*y/x == 1/x^3 , y)
```

$$\frac{1}{4}\,(4\,c-\frac{1}{x^{4}})x^{2}$$

Sage code

```
expand(_)
```

$$cx^{2}-\frac{1}{4}\,\frac{1}{x^{2}}$$