RownaniaRozniczkowe

May 20, 2022

[8]:
```
import sympy  #import biblioteki sympy
#import sympy as sp #można zdefiniować skrót polecenia (zamiast sympy będzie␣
 ↪można pisać sp):
sympy.init_printing #ładniejszy wygląd symboli matematycznych podobnie jak w␣
 ↪latexu
from sympy import * #po tej komendzie zamiast sympy.log lub sp.log wystarczy␣
 ↪napisać log
from sympy.abc import x #x jest zmienną niezależną
```

[9]:
```
y=Function("y")(x) # y jest funkcją zmiennej x
```

[10]:
```
y
```

[10]:

$$y(x)$$

[24]:
```
dy1=Derivative(y,x) # dy1 jest 1 pochodną funkcji f
dy2=Derivative(Derivative(y,x),x) # dy2_ jest 2 pochodną funkcji f
dy3=Derivative(y,x,x,x) # dy2_ jest 3 pochodną funkcji f
```

[25]:
```
dy1
```

[25]:

$$\frac{d}{dx}y(x)$$

[26]:
```
dy2
```

[26]:

$$\frac{d^2}{dx^2}y(x)$$

[28]:
```
dy3
```

[28]:

$$\frac{d^3}{dx^3}y(x)$$

[33]:
```
dsolve(Derivative(y,x)-2*x*y**2,y)
```

[33]:

$$y(x) = -\frac{1}{C_1 + x^2}$$

0.0.1 Jak rozwiązać równanie różniczkowe?

[59]:
```
y = symbols("y", cls=Function)
x = symbols("x")
print(y(x))
y(x).diff(x)
```

```
y(x)
```

[59]: $\frac{d}{dx}y(x)$

[ ]:
```
# sposób 1
```

[77]:
```
x*y(x).diff(x)-2*y(x)
```

[77]: $x\frac{d}{dx}y(x) - 2y(x)$

[58]:
```
dsolve(x*y(x).diff(x)-2*y(x),y(x)) #rozwiązanie równania różniczkowego -
 ↪bezpośrednio
```

[58]: $y(x) = C_1x^2$

[ ]:
```
# sposób 2
```

[63]:
```
rownanie=Eq(x*y(x).diff(x),2*y(x)) #tworzymy równanie różniczkowe
rownanie
```

[63]: $x\frac{d}{dx}y(x) = 2y(x)$

[64]:
```
dsolve(rownanie,y(x)) #rozwiązujemy równanie
```

[64]: $y(x) = C_1x^2$

[ ]:
```
#rozwiązanie zagadnienia brzegowego
```

[61]:
```
dsolve(x*y(x).diff(x)-2*y(x),y(x),ics={y(1):2}) #rozwiązanie zagadnienia
 ↪Cauchy'ego z zadanymi war. początkowymi
```

[61]: $y(x) = 2x^2$

[ ]:
```
#nie zawsze otrzymamy rozwiązanie
```

[67]:
```
sol=dsolve(y(x).diff(x)-x-(x+1)*y(x)+1,y(x))
sol
```

[67]: $y(x) = C_1e^{x\left(\frac{x}{2}+1\right)} - \sqrt{2}\sqrt{\pi}e^{\frac{x^2}{2}+x+\frac{1}{2}}\operatorname{erf}\left(\frac{\sqrt{2}\,(x+1)}{2}\right) - 1$

[ ]:
```
#dodanie warunków poczatkowych też nie poprawia sytuacji
```

[73]:
```
soli=dsolve(y(x).diff(x)-x-(x+1)*y(x)+1,y(x),ics={y(0):0})
soli
```

[73]:

$$y(x) = \left(1 + \sqrt{2}\sqrt{\pi}e^{\frac{1}{2}} \operatorname{erf}\left(\frac{\sqrt{2}}{2}\right)\right) e^{x\left(\frac{x}{2}+1\right)} - \sqrt{2}\sqrt{\pi}e^{\frac{x^2}{2}+x+\frac{1}{2}} \operatorname{erf}\left(\frac{\sqrt{2}\,(x+1)}{2}\right) - 1$$

[ ]:
```
#możemy wyznaczyć wartość przybliżoną
```

[76]:
```
sol.subs(x,0.2).evalf(10) #bez uwzględnienia war. poczt. w punkcie x=0.2
```

[76]:

$$y(0.2) = 1.246076731C_1 - 4.964551727$$

[75]:
```
soli.subs(x,0.2).evalf(10) #z uwzględnieniem war. poczt. w punkcie x=0.2
```

[75]:

$$y(0.2) = -0.2028286628$$

[ ]:
```
#Jak sprawdzić typ równania?
```

[80]:
```
y(x).diff(x)+y(x)-y(x)**2 #równanie
```

[80]:

$$-y^2(x) + y(x) + \frac{d}{dx}y(x)$$

[81]:
```
classify_ode(y(x).diff(x)+y(x)-y(x)**2, y(x))
```

[81]:
```
('separable',
 '1st_exact',
 'Bernoulli',
 '1st_power_series',
 'lie_group',
 'separable_Integral',
 '1st_exact_Integral',
 'Bernoulli_Integral')
```

[79]:
```
classify_ode(soli, y(x))
```

[79]:
```
('nth_algebraic', 'nth_algebraic_Integral')
```

[ ]:
```

```