Дифференциальные уравнения

In [1]:
```
from sympy import *
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt
%matplotlib inline
```

Пример 1

In [2]:
```
x = symbols('x')
y = Function('y')
```

In [3]:
```
eq = diff(y(x),x) - (exp(sqrt(x)-2)/sqrt(x))
dsolve(eq,y(x)).simplify()
```

Out[3]: $y(x) = C_1 + 2e^{\sqrt{x}-2}$

Пример 2

In [4]:
```
eq = (x+1)*diff(y(x),x) + x*y(x)
dsolve(eq, y(x))
```

Out[4]: $y(x) = C_1\left(x+1\right)e^{-x}$

Пример 3

In [5]:
```
eq = x*diff(y(x),x) - y(x) - sqrt(y(x)**2-x**2)
dsolve(eq, y(x))
```

Out[5]: $y(x) = x\cosh\left(C_1 - \log\left(x\right)\right)$

Пример 4

In [6]:
```
z = symbols('z')
fz = (1+z**2)/z**3
I1 = integrate(fz)
I1
```

Out[6]: $\log\left(z\right) - \frac{1}{2z^2}$

In [7]:
```
I2 = -integrate(1/x)
I2
```

Out[7]: $-\log\left(x\right)$

Пример 5

In [8]:
```
u = symbols('u')
v = Function('v')
eq = diff(v(u), u) - v(u)/u - 1/2
dsolve(eq, v(u))
```

Out[8]: $v(u) = u\left(C_1 + 0.5\log\left(u\right)\right)$

Пример 6

In [9]:
```
eq = x*diff(y(x),x) - y(x) - x**3
dsolve(eq, y(x))
```

Out[9]:

$$y(x) = x\left(C_1 + \frac{x^2}{2}\right)$$

Пример 7

In [10]:
```
eq = diff(y(x),x)+4*x*y(x)
dsolve(eq,y(x))
```

Out[10]: $y(x) = C_1 e^{-2x^2}$

In [11]:
```
integrate(6*x*exp(x**2),x)
```

Out[11]: $3e^{x^2}$

Пример 8

In [12]:
```
y = symbols('y')
x = Function('x')
eq = diff(x(y),y) - 2*x(y)/y - y**2
dsolve(eq, x(y))
```

Out[12]: $x(y) = y^2\left(C_1 + y\right)$

Пример 9

In [13]:
```
x = symbols('x')
y = Function('y')
eq = diff(y(x),x) - y(x)/x - (x**4)*(y(x)**2)
dsolve(eq,y(x))
```

Out[13]:

$$y(x) = \frac{6x}{C_1 - x^6}$$

Пример 10

In [14]:
```
z = Function('z')
eq = diff(z(x),x)/4 + x*z(x) - x
dsolve(eq, z(x))
```

Out[14]: $z(x) = C_1 e^{-2x^2} + 1$

Пример 11

In [15]:
```
u = Function('u')
eq = diff(u(x),x) + 4*u(x)/x + u(x)**2
des = dsolve(eq, u(x))
des.simplify()
```

Out[15]: $u(x) = \frac{3}{C_1 x^4 - x}$

Пример 12

In [16]:
```
eq = (2*x-3)*diff(y(x),x) + 3*x**2+2*y(x)
dsolve(eq,y(x))
```

Out[16]: $y(x) = \frac{C_1 - x^3}{2x - 3}$

Пример 13

In [17]:
```
x,y = symbols('x y')
Q = x**2 - y**2
I2 = integrate(Q, (y,0,y))
I2
```

Out[17]: $x^2 y - \frac{y^3}{3}$

Пример 14

In [18]:
```
x = symbols('x')
y = Function('y')
eq = (x**2-1)*diff(y(x), x) + 2*x*y(x)**2
dsolve(eq, y(x))
```

Out[18]: $y(x) = -\frac{1}{C_1 - \log(x^2 - 1)}$

Пример 15

In [19]:
```
eq = x*diff(y(x),x) - y(x) + y(x)**2*(log(x)+2)*log(x)
des = dsolve(eq, y(x))
des
```

Out[19]: $y(x) = \frac{x}{C_1 + x \log(x)^2}$

Пример 16

In [20]:
```
eq = (1+x**2)*diff(y(x),x) + y(x)
des = dsolve(eq, y(x))
des
```

Out[20]: $y(x) = C_1 e^{-\operatorname{atan}(x)}$

Пример 17

In [21]:
```
def Lin_homogen_2(a,yl):
    x = symbols('x')
    u = Function('u')
    z = Function('z')

    y1d = diff(y1,x)

    eq = y1*diff(u(x),x)+(2*y1d+a*y1)*u(x)
    u0 = dsolve(eq,u(x))

    eq = diff(z(x),x)-u0.rhs
    z0 = dsolve(eq,z(x))

    y = y1*z0.rhs
    return y.simplify()
```

Пример 18

In [22]:
```
a = -(2*x+1)/x
y1 = exp(x)
Lin_homogen_2(a,y1)
```

Out[22]: $\left(\frac{C_1 x^2}{2} + C_2\right) e^x$

Пример 19

In [23]:
```
a = -2/x
y1 = x
Lin_homogen_2(a,y1)
```

Out[23]: $x\,(C_1 x + C_2)$

Пример 20

In [24]:
```
integrate(log(x),x)
```

Out[24]: $x\log(x) - x$

Пример 21

In [25]:
```
eq = diff(y(x),x,5) - x*cos(2*x)
des = dsolve(eq, y(x))
des
```

Out[25]:

$$y(x) = C_1 + C_2x^2 + C_3x^3 + C_4x^4 + x\left(C_5 + \frac{\sin(2x)}{32}\right) + \frac{5\cos(2x)}{64}$$

Пример 22

In [26]:
```
eq = x*diff(y(x),x,2) - 3*diff(y(x),x)
des = dsolve(eq, y(x))
des
```

Out[26]: $y(x) = C_1 + C_2x^4$

Пример 23

In [27]:
```
z = Function('z')
eq = 2*x*z(x)*diff(z(x),x)-z(x)**2+1
des = dsolve(eq, z(x))
des
```

Out[27]:
```
[Eq(z(x), -sqrt(C1*x + 1)), Eq(z(x), sqrt(C1*x + 1))]
```

In [28]:
```
C1 = symbols('C1')
eq = diff(y(x),x)-sqrt(C1*x+1)
des = dsolve(eq, y(x))
des
```

Out[28]:

$$y(x) = C_2 + \frac{2(C_1x+1)^{\frac{3}{2}}}{3C_1}$$

Пример 24

In [29]:
```
u = Function('u')
eq = diff(u(x),x) + 4*u(x)/x + u(x)**2
des = dsolve(eq, u(x))
des
```

Out[29]:

$$u(x) = \frac{3}{x(C_1x^3 - 1)}$$

Пример 25

In [30]:
```
y = symbols('y')
z = Function('z')
eq = 2*y*diff(z(y),y) + z(y)
des = dsolve(eq, z(y))
des
```

Out[30]:

$$z(y) = \frac{C_1}{\sqrt{y}}$$

Пример 26

In [31]:
```
x = symbols('x')
y = Function('y')
eq = diff(y(x),x,3)+diff(y(x),x,2)-2*diff(y(x),x)
des = dsolve(eq, y(x))
des
```

Out[31]: $y(x) = C_1 + C_2e^{-2x} + C_3e^{x}$

Пример 27

In [32]:
```
lamda=symbols('lamda')
roots(lamda**5-2*lamda**4+9*lamda**3-18*lamda**2)
```

Out[32]:
```
{2: 1, -3*I: 1, 3*I: 1, 0: 2}
```

Пример 28

In [33]:
```
roots(lamda**6+12*lamda**5+61*lamda**4+336*lamda**3 +2016*lamda**2+6400*lamda+7424
```

Out[33]:
```
{-4: 4, 2 - 5*I: 1, 2 + 5*I: 1}
```

Пример 29

In [34]:
```
x = symbols('x')
y = Function('y')
eq = diff(y(x),x,2)+8*diff(y(x),x)+16*y(x)-4*x**2*exp(3*x)
des = dsolve(eq, y(x))
des
```

Out[34]: $$y(x) = \frac{4x^2e^{3x}}{49} - \frac{16xe^{3x}}{343} + (C_1 + C_2x)\,e^{-4x} + \frac{24e^{3x}}{2401}$$

Пример 30

In [35]:
```
x,C3,C4 = symbols('x C3 C4')
ych = exp(-x)*(C3*sin(2*x)+C4*cos(2*x))
ych1 = diff(ych,x)
ych1
```

Out[35]: $$-\left(C_3\sin(2x) + C_4\cos(2x)\right)e^{-x} + \left(2C_3\cos(2x) - 2C_4\sin(2x)\right)e^{-x}$$

Пример 31

In [36]:
```
lamda=symbols('lamda')
roots(lamda**3-5*lamda**2+6*lamda)
```

Out[36]:
```
{3: 1, 2: 1, 0: 1}
```

In [37]:
```
x,C1d,C2d,C3d = symbols('x Cd C2d C3d')
y1 = 1
y2 = exp(2*x)
y3 = exp(3*x)

y1d = diff(y1,x)
y2d = diff(y2,x)
y3d = diff(y3,x)

y1dd = diff(y1,x,2)
y2dd = diff(y2,x,2)
y3dd = diff(y3,x,2)

eq1 = C1d*y1+C2d*y2+C3d*y3
eq2 = C1d*y1d+C2d*y2d+C3d*y3d
eq3 = C1d*y1dd+C2d*y2dd+C3d*y3dd

solve([eq1,eq2,eq3-2**x], [C1d,C2d,C3d])
```

Out[37]:
```
{Cd: 2**x/6, C2d: -2**x*exp(-2*x)/2, C3d: 2**x*exp(-3*x)/3}
```

Пример 32

In [38]:
```
x = symbols('x')
y = Function('y')
eq = diff(y(x),x,4)-3*diff(y(x),x,2)+2*diff(y(x),x)
des = dsolve(eq, y(x))
des
```

Out[38]:

$$y(x) = C_1 + C_4 e^{-2x} + (C_2 + C_3 x)\, e^{x}$$

Пример 33

In [39]:
```
eq = diff(y(x),x,2) + 2*diff(y(x),x) - exp(x)
des = dsolve(eq,y(x))
des
```

Out[39]:

$$y(x) = C_1 + C_2 e^{-2x} + \frac{e^{x}}{3}$$

Пример 34

In [40]:
```
x = symbols('x')
y = Function('y')
eq = x*diff(y(x),x,2)+diff(y(x),x)-sqrt(x)
des = dsolve(eq,y(x))
des
```

Out[40]:

$$y(x) = C_1 + C_2 \log(x) + \frac{4x^{\frac{3}{2}}}{9}$$

Пример 35

In [41]:
```
x = symbols('x')
y = Function('y')
eq = diff(y(x),x,4)-2*diff(y(x),x,3)+diff(y(x),x,2)-2*diff(y(x),x)
des = dsolve(eq,y(x))
des
```

Out[41]: $y(x) = C_1 + C_2 e^{2x} + C_3 \sin(x) + C_4 \cos(x)$

Пример 36

In [42]:
```
t = symbols('t')
x = Function('x')
eq = diff(x(t),t,4) - x(t)
des = dsolve(eq,x(t))
des
```

Out[42]: $x(t) = C_1 e^{-t} + C_2 e^{t} + C_3 \sin(t) + C_4 \cos(t)$

In [43]:
```
diff(des.rhs,t,2)
```

Out[43]: $C_1 e^{-t} + C_2 e^{t} - C_3 \sin(t) - C_4 \cos(t)$

Пример 37

In [44]:
```
C1 = symbols('C1')
eq = diff(x(t),t) - x(t)/(2*t+C1)
des = dsolve(eq,x(t))
des
```

Out[44]: $x(t) = C_2 \sqrt{C_1 + 2t}$

Пример 38

In [45]:
```
x = symbols('x')
y1 = Function('y1')
eq = diff(y1(x),x,2)-2*diff(y1(x),x)+5*y1(x)
des = dsolve(eq,y1(x))
des
```

Out[45]: $y_1(x) = (C_1 \sin(2x) + C_2 \cos(2x))\, e^{x}$

In [46]:
```
C1,C2 = symbols('C1,C2')
y3 = Function('y3')
eq = diff(y3(x), x)-y3(x)-3*exp(x)*(C1*sin(2*x)+C2*cos(2*x))
des = dsolve(eq,y3(x))
des
```

Out[46]: $y_3(x) = \left(-\frac{3C_1 \cos(2x)}{2} + \frac{3C_2 \sin(2x)}{2} + C_3\right) e^{x}$

Пример 39

In [47]:
```
A = Matrix([[1,2], [4,3]])
A.eigenvects()
```

Out[47]:
```
[(-1,
  1,
  [Matrix([
   [-1],
   [ 1]])]),
 (5,
  1,
  [Matrix([
   [1/2],
   [  1]])])]
```

In [48]:
```
x,C1,C2 = symbols('t C1 C2')
y1 = C1*exp(-x)+(C2/2)*exp(5*x)
y2 = -C1*exp(-x)+C2*exp(5*x)
print(diff(y1,x)-y1-2*y2, diff(y2,x)-4*y1-3*y2)
```

```
0 0
```

Пример 40

In [49]:
```
A = Matrix([[2,1], [-2,4]])
A.eigenvects()
```

Out[49]:
```
[(3 - I,
  1,
  [Matrix([
   [1/2 + I/2],
   [        1]])]),
 (3 + I,
  1,
  [Matrix([
   [1/2 - I/2],
   [        1]])])]
```

In [50]:
```
x,C1,C2 = symbols('x C1 C2')
y1 = exp(3*x)*((C1+C2)*cos(x)+(C1-C2)*sin(x))
y2 = exp(3*x)*(2*C1*cos(x)-2*C2*sin(x))
(diff(y1,x)-2*y1-y2).simplify()
```

Out[50]: 0

Пример 41

In [51]:
```
A = Matrix([[0,1], [-1,0]])
A.eigenvects()
```

Out[51]:
```
[(-I,
  1,
  [Matrix([
   [I],
   [1]])]),
 (I,
  1,
  [Matrix([
   [-I],
   [ 1]])])]
```

In [52]:
```
C1d,C2d,x = symbols('C1d,C2d x')
eq1 = C1d*sin(x)-C2d*cos(x)-x
eq2 = C1d*cos(x)+C2d*sin(x)-3
des = solve([eq1,eq2], [C1d,C2d])
des
```

Out[52]:
```
{C1d: x*sin(x)/(sin(x)**2 + cos(x)**2) + 3*cos(x)/(sin(x)**2 + cos(x)**2),
 C2d: -x*cos(x)/(sin(x)**2 + cos(x)**2) + 3*sin(x)/(sin(x)**2 + cos(x)**2)}
```

Пример 42

In [53]:
```
t = symbols('t')
x = Function('x')
y = Function('y')
eq1 = diff(x(t),t) - x(t) + y(t)
eq2 = diff(y(t),t) - x(t) - 3*y(t)

dsolve((eq1,eq2))
```

Out[53]:
```
[Eq(x(t), -C2*t*exp(2*t) - (C1 - C2)*exp(2*t)),
 Eq(y(t), C1*exp(2*t) + C2*t*exp(2*t))]
```

Пример 43

In [54]:
```
t = symbols('t')
x = Function('x')
y = Function('y')
eq1 = diff(x(t),t)-x(t)-y(t)
eq2 = diff(y(t),t)-y(t)
dsolve((eq1,eq2))
```

Out[54]:
```
[Eq(x(t), C1*exp(t) + C2*t*exp(t)), Eq(y(t), C2*exp(t))]
```

Пример 44

In [55]:
```
t = symbols('t')
x = Function('x')
y = Function('y')
z = Function('z')
eq1 = diff(x(t),t)-2*x(t)-y(t)
eq2 = diff(y(t),t)-x(t)-2*y(t)
eq3 = diff(z(t),t)-x(t)-y(t)-2*z(t)
des = dsolve((eq1,eq2,eq3))
des
```

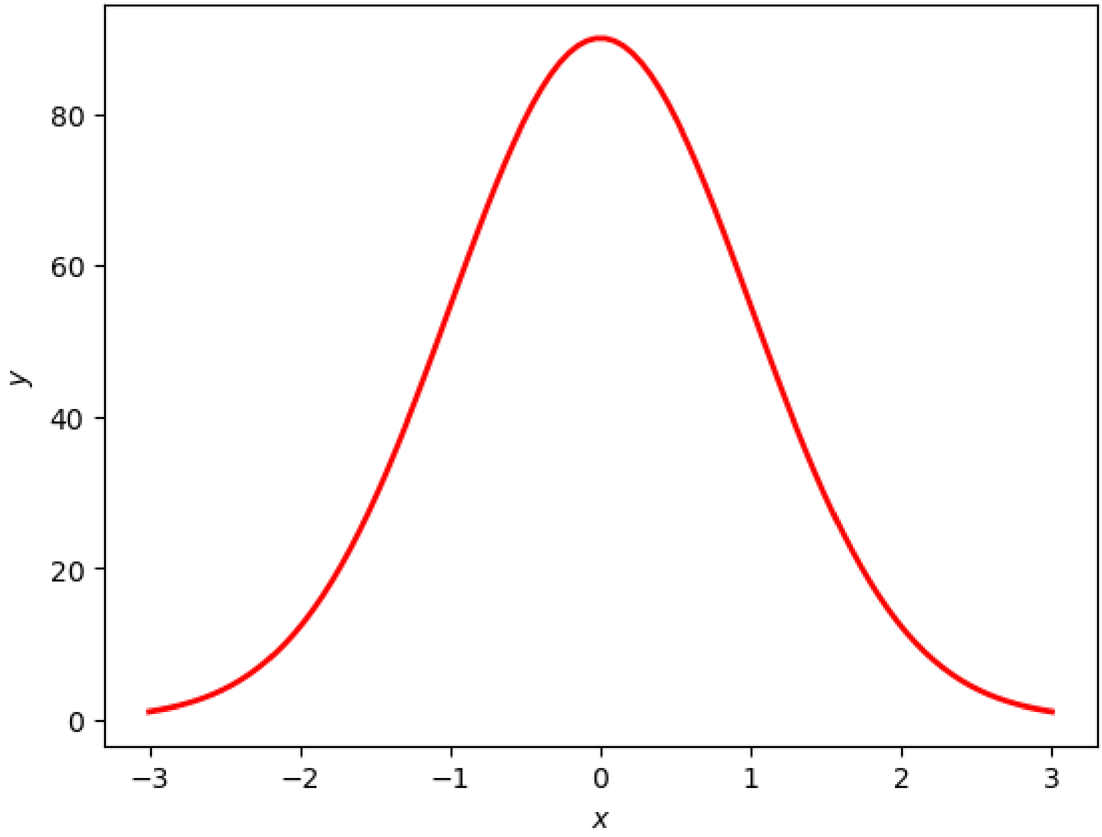
Out[55]:
```
[Eq(x(t), -C1*exp(t) + C2*exp(3*t)/2),
 Eq(y(t), C1*exp(t) + C2*exp(3*t)/2),
 Eq(z(t), C2*exp(3*t) + C3*exp(2*t))]
```

Пример 45

In [56]:
```
def f(y,x):
    return -y*x

x = np.linspace(-3, 3, 100)
y0 = 1
y = odeint(f, y0, x)
```

```
plt.plot(x,y,c='r',linewidth=2)
plt.xlabel('$x$')
plt.ylabel('$y$')
plt.show()
```



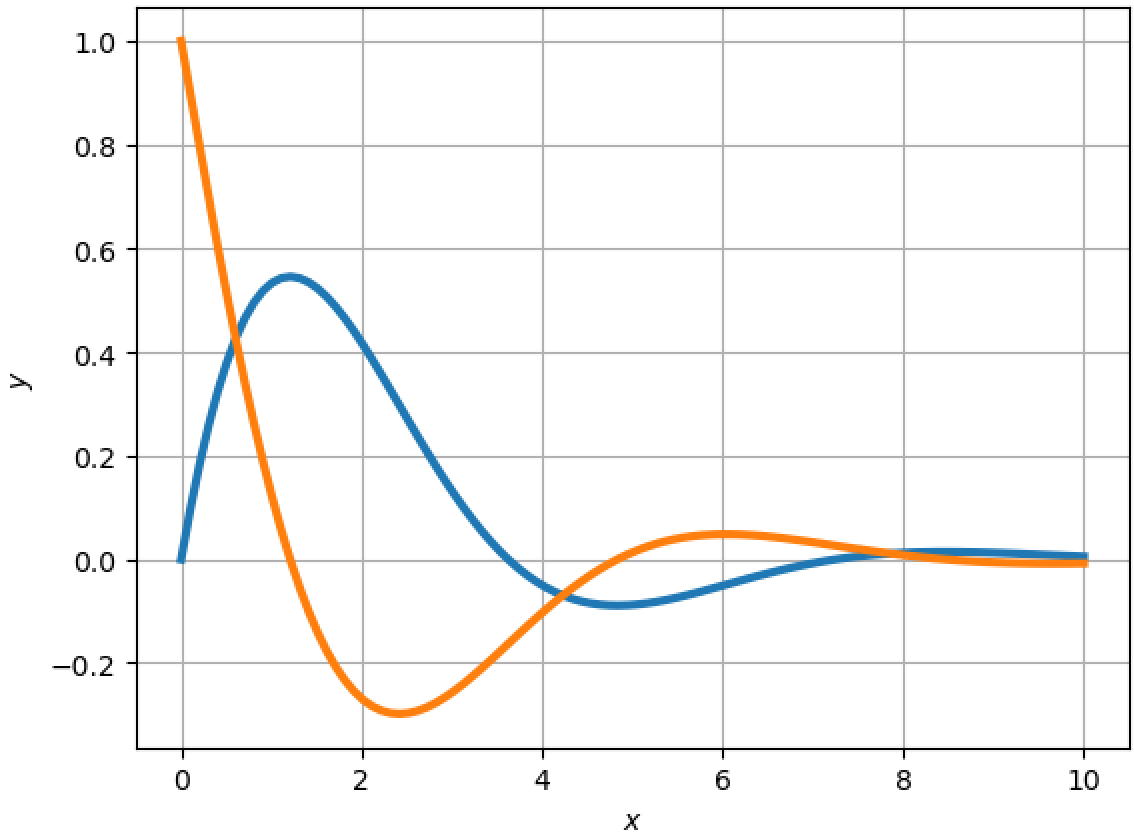
Пример 46

In [57]:
```
def f(y, x):
    y1, y2 = y
    return [y2, -y1-y2]

x = np.linspace(0,10,100)
y0 = [0, 1]
w = odeint(f, y0, x)

y1 = w[:,0]
y2 = w[:,1]
```

In [58]:
```
fig = plt.figure(facecolor='white')
plt.plot(x,y1,x,y2,linewidth=3)
plt.ylabel("$y$")
plt.xlabel("$x$")
plt.grid(True)
plt.show()
```



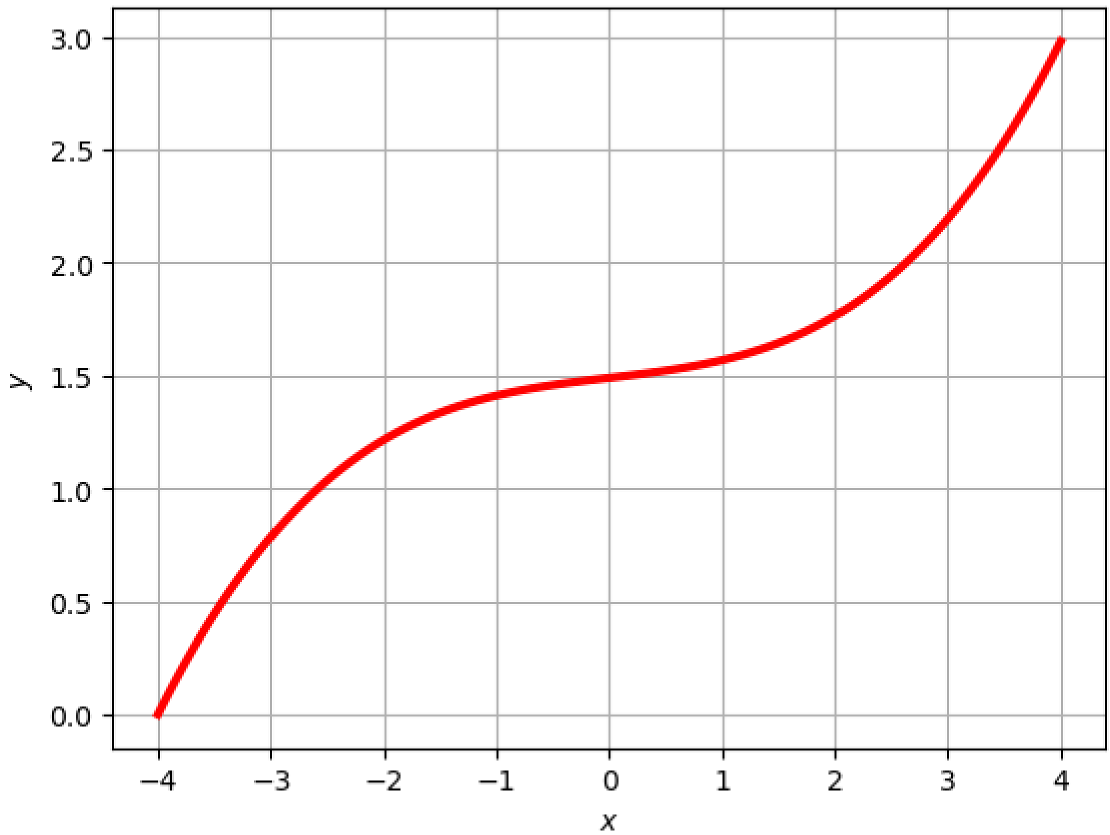
Пример 47

In [59]:
```
def f(y, x):
    y1, y2 = y
    return [y2, 2*x*y2/(x**2+1)]

x = np.linspace(-4, 4, 100)
y0 = [-75, 51]
w = odeint(f, y0, x)

y1 = w[:,0]

fig = plt.figure(facecolor='white')
plt.plot(x,y1,c='r',linewidth=3)
plt.ylabel("$y$")
plt.xlabel("$x$")
plt.grid(True)
plt.show()
```

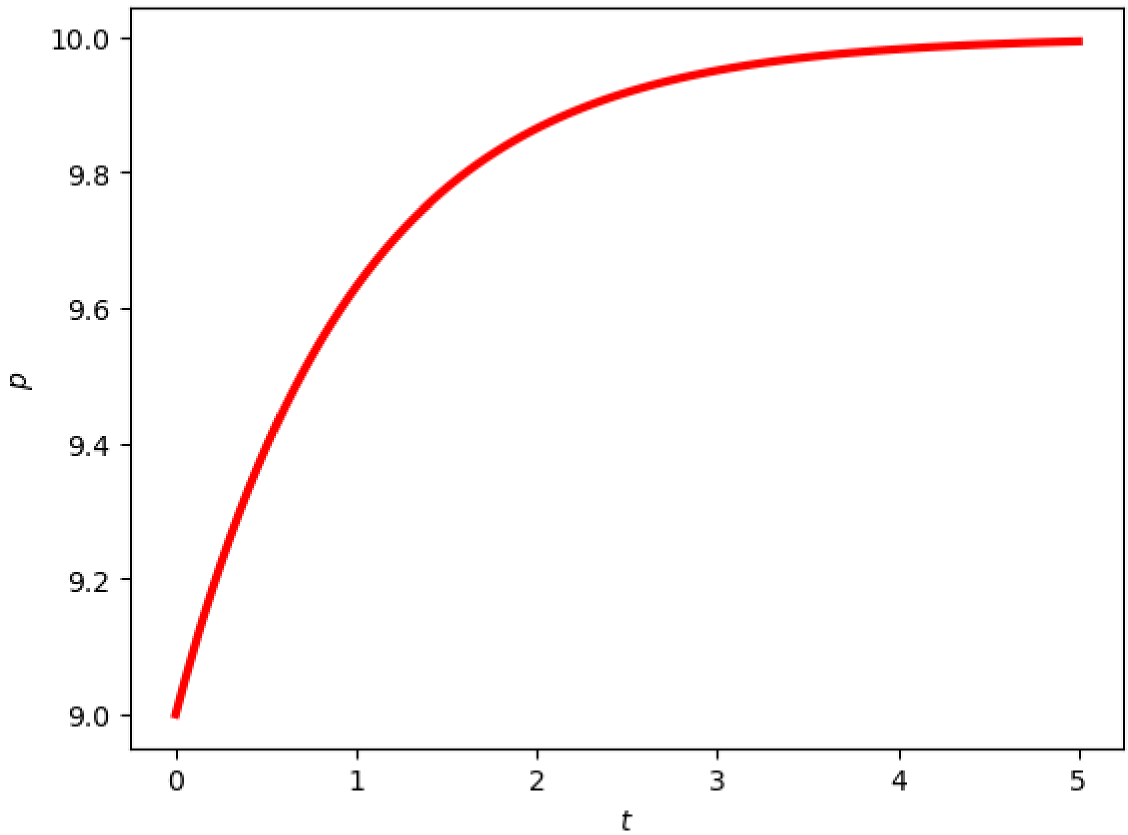


Пример 48

In [60]:
```
t = symbols('t')
p = Function('p')
eq = diff(p(t),t)+p(t)-10
des = dsolve(eq,p(t))
des
```

Out[60]: $p(t) = C_1 e^{-t} + 10$

In [61]:
```
t = np.linspace(0,5,100)
p = 10-np.exp(-t)
plt.plot(t,p,c='r',linewidth=3)
plt.ylabel('$p$')
plt.xlabel("$t$")
plt.show()
```

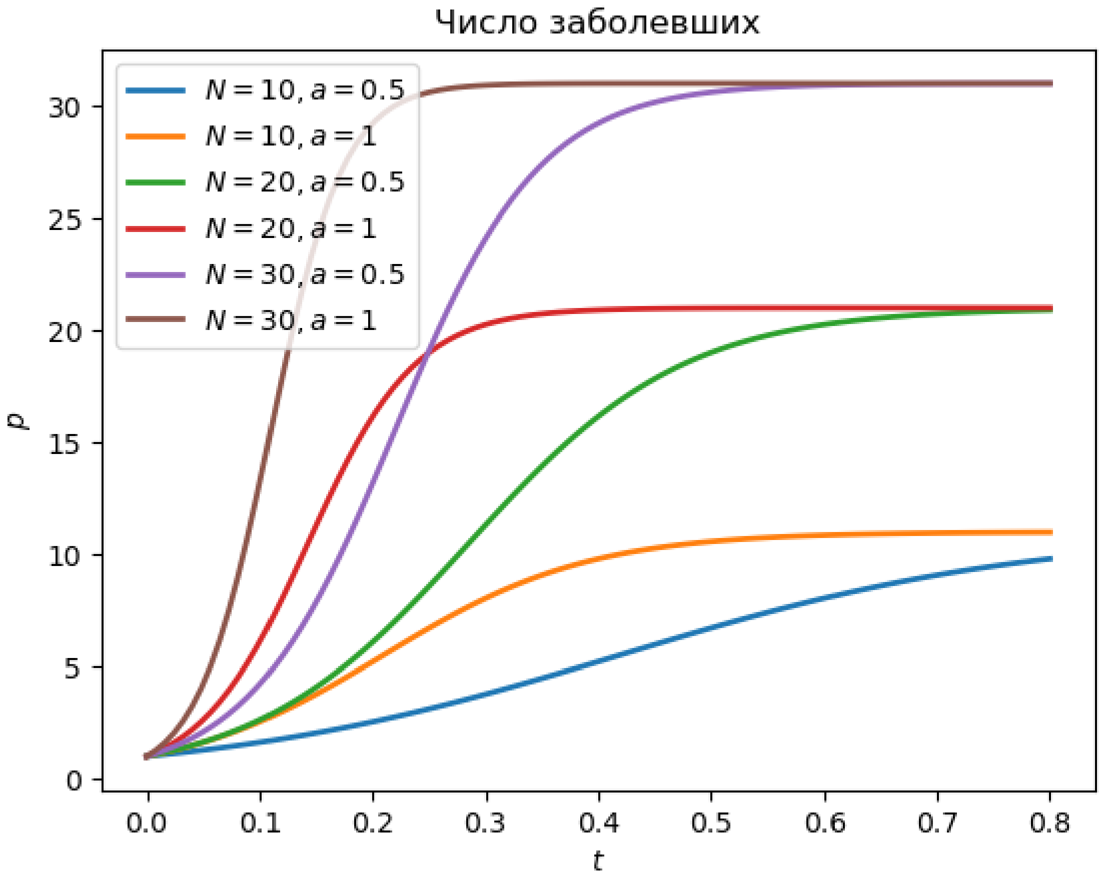


Пример 49

In [62]:
```
u = symbols('u')
N = symbols('N')
integrate(1/((N+1)*u-1),u)
```

Out[62]:
$$\frac{\log\left(u\left(N+1\right)-1\right)}{N+1}$$

In [63]:
```
t = np.linspace(0,0.8,100)
for param in [[10, 0.5],[10,1],[20,0.5],[20,1],[30,0.5],[30,1]]:
    N = param[0]
    a = param[1]
    X = (N+1)/(N*np.exp(-(N+1)*a*t)+1)
    plt.plot(t, X, lw=2, label="$N=%s, a=%s$" % (N, a))
plt.legend()
plt.ylabel('$p$')
plt.xlabel("$t$")
plt.title("Число заболевших");
```

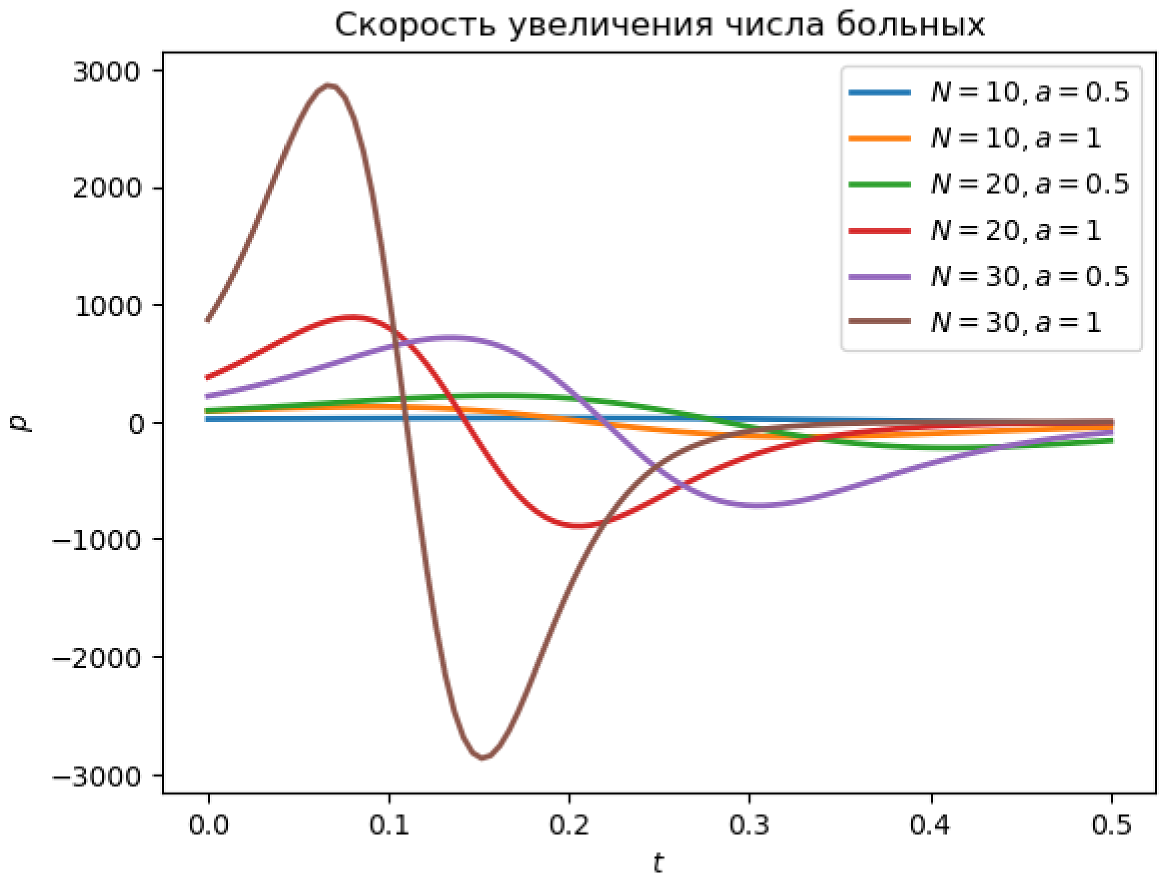


In [64]:
```
t,N,a = symbols('t N a')
X = (N+1)/(N*exp(-(N+1)*a*t)+1)
Xprim = diff(X,t,2)
Xprim.simplify()
```

Out[64]:

$$\frac{Na^2(N+1)^3\left(N-e^{at(N+1)}\right)e^{at(N+1)}}{\left(N+e^{at(N+1)}\right)^3}$$

In [65]:
```
t = np.linspace(0,0.5,100)
for param in [[10, 0.5],[10,1],[20,0.5],[20,1],[30,0.5],[30,1]]:
    N = param[0]
    a = param[1]
    Xprim = a**2*N*(N+1)**3*(N-np.exp((N+1)*a*t))*np.exp((N+1)*a*t) / (N+np.exp((N
    plt.plot(t, Xprim, lw=2, label = "$N=%s, a=%s$" % (N, a))
plt.legend()
plt.ylabel('$p$')
plt.xlabel("$t$")
plt.title("Скорость увеличения числа больных");
```



Примеры решения задач

Решить задачу Коши $y' = \frac{xy^2 - yx^2}{x^3}, y(-1) = 1$

In [66]:
```
x = symbols('x')
y = Function('y')
eq = diff(y(x),x) - (x*y(x)**2-y(x)*x**2)/x**3
des = dsolve(eq, y(x))
des
```

Out[66]:

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

Решить уравнение $(x + y - 4)y' = 2x + y + 3$

In [67]:
```
u = symbols('u')
z = Function('z')
eq = u*(1+z(u))*diff(z(u),u)-2+z(u)**2
des = dsolve(eq, z(u))
des.simplify()
```

Out[67]:

$$C_1 = \log(u) + \frac{\left(\sqrt{2} + 2\right) \log\left(z(u) - \sqrt{2}\right)}{4} + \frac{\left(2 - \sqrt{2}\right) \log\left(z(u) + \sqrt{2}\right)}{4}$$

Решить уравнение $xy' + y = y^2$.

In [68]:
```
eq = x*diff(y(x),x) + y(x) - y(x)**2
dsolve(eq, y(x))
```

Out[68]:
$$y(x) = -\frac{1}{C_1 x - 1}$$

Найти общее решение уравнения $y'' - 2\left(1 + \mathrm{tg}^2\, x\right) y = 0$, если известно одно его частное решение $y_1 = \mathrm{tg}\, x$.

In [69]:
```
a = 0
y1 = tan(x)
Lin_homogen_2(a,y1)
```

Out[69]:
$$\left(C_1 \int \frac{e^{-x}}{\tan^2(x)}\, dx + C_2\right) \tan(x)$$

Решить уравнение $\left(1 + x^2\right) y'' + 2xy' = x^3$.

In [70]:
```
x = symbols('x')
z = Function('z')
eq = (1+x**2)*diff(z(x),x)+2*x*z(x)-x**3
dsolve(eq,z(x))
```

Out[70]:
$$z(x) = \frac{C_1 + \frac{x^4}{4}}{x^2 + 1}$$

In [71]:
```
z2 = x**4/(4*(x**2+1))
integrate(z2,x)
```

Out[71]:
$$\frac{x^3}{12} - \frac{x}{4} + \frac{\operatorname{atan}(x)}{4}$$

Решить уравнение $y'' - 3y' + 2y = e^{2x} \sin x$

In [72]:
```
eq = diff (y(x),x,2)-3*diff(y(x),x)+2*y(x)-exp(2*x)*sin(x)
dsolve(eq,y(x))
```

Out[72]:
$$y(x) = \left(C_1 + \left(C_2 - \frac{\sin(x)}{2} - \frac{\cos(x)}{2}\right) e^x\right) e^x$$

Решить систему уравнений $\begin{cases} \frac{dy_1}{dx} = 2y_1 - y_2 \\ \frac{dy_2}{dx} = 2y_2 - y_1 - 5e^x \sin x \end{cases}$

In [73]:
```
x = symbols('x')
y1 = Function('y1')
y2 = Function('y2')
eq1 = diff(y1(x),x)-2*y1(x)+y2(x)
eq2 = diff(y2(x),x)-2*y2(x)+y1(x)
des = dsolve((eq1,eq2))
des
```

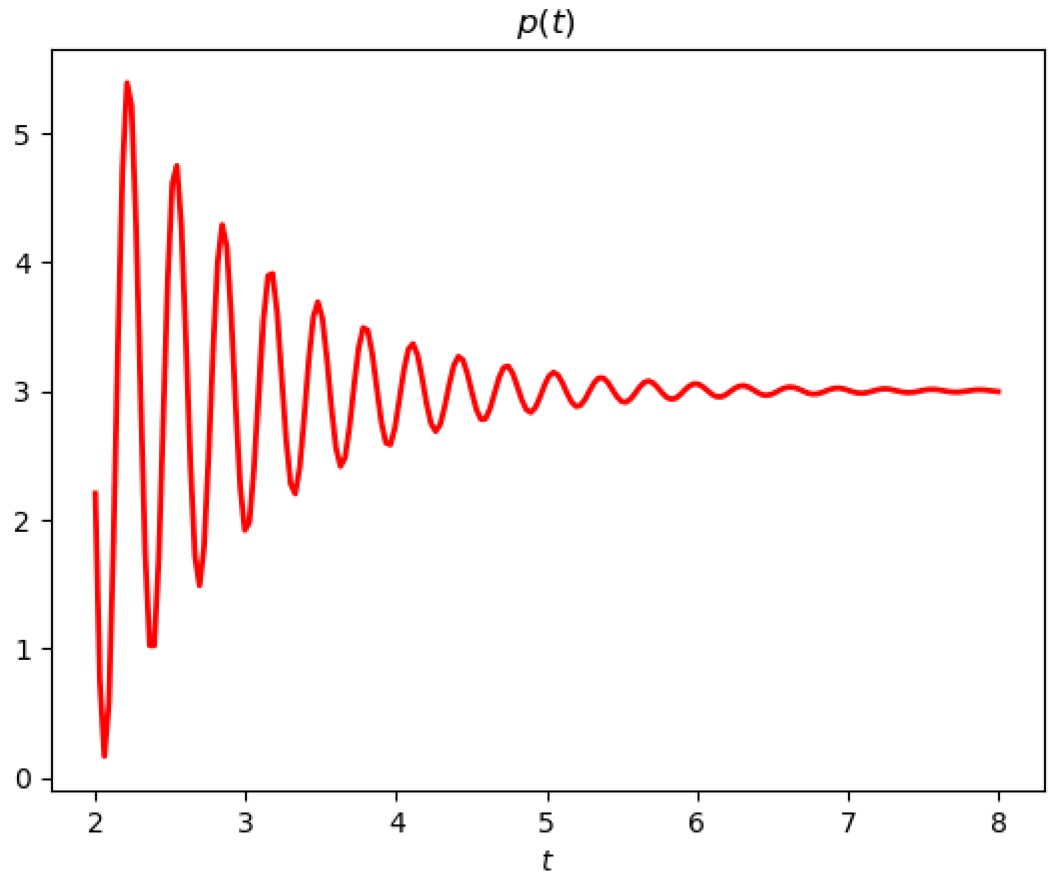
Out[73]:
```
[Eq(y1(x), C1*exp(x) - C2*exp(3*x)), Eq(y2(x), C1*exp(x) + C2*exp(3*x))]
```

Функции спроса D и предложения S, выражающие зависимость от цены p и ее производных, имеют вид:
$D(t) = 3p'' - p' - 200p + 600, \quad S(t) = 4p'' + p' + 201p - 603$. Найти зависимость равновесной цены от времени.

In [74]:
```
t = symbols('t')
p = Function('p')
eq = diff(p(t),t,2)+2*diff(p(t),t)+401*p(t)-1203
des = dsolve(eq,p(t))
des
```

Out[74]: $p(t) = (C_1 \sin(20t) + C_2 \cos(20t))\, e^{-t} + 3$

In [75]:
```
t = np.linspace(2,8,200)
y = 3 + np.exp(-t)*(10*np.sin(20*t)+20*np.cos(20*t))
plt.plot(t, y, c = 'r', lw=2)
plt.xlabel("$t$")
plt.title("$p(t)$")
plt.show()
```



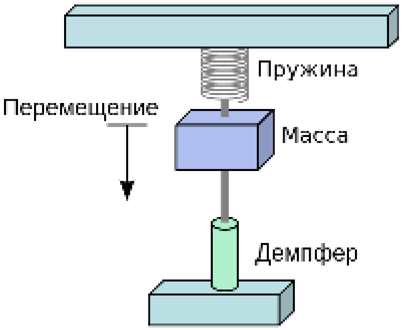
Решить задачу Коши $\left(x^2 y - y\right)^2 y' = x^2 y - y + x^2 - 1, \quad y(\infty) = 0$

In [76]:
```
x = symbols('x')
y = Function('y')
eq = (x**2*y(x)-y(x))**2*diff(y(x),x)-x**2*y(x)+y(x)-x**2+1
des = dsolve(eq, y(x))
des
```

Out[76]: $\frac{y^2(x)}{2} - y(x) - \frac{\log(x-1)}{2} + \frac{\log(x+1)}{2} + \log(y(x)+1) = C_1$

Решение задачи с использованием дифференциальных уравнений

Одной из распространенных инженерных задач, в которой используются дифференциальные уравнения, является моделирование поведения механической системы. Например, рассмотрим простую систему масса-пружина-демпфер, которая состоит из массы, прикрепленной к пружине и демпферу.



Положение массы как функция времени, x(t), может быть описано следующим дифференциальным уравнением второго порядка:

$$mx''(t) + cx'(t) + k * x(t) = 0$$

где m - масса, c - коэффициент демпфирования, k - постоянная пружины, а $x''(t)$ и $x'(t)$ - вторая и первая производные $x(t)$ по времени, соответственно.

In [77]:
```
def f(x, t):
    return np.array([x[1], -(c/m)*x[1] - (k/m)*x[0]])
```

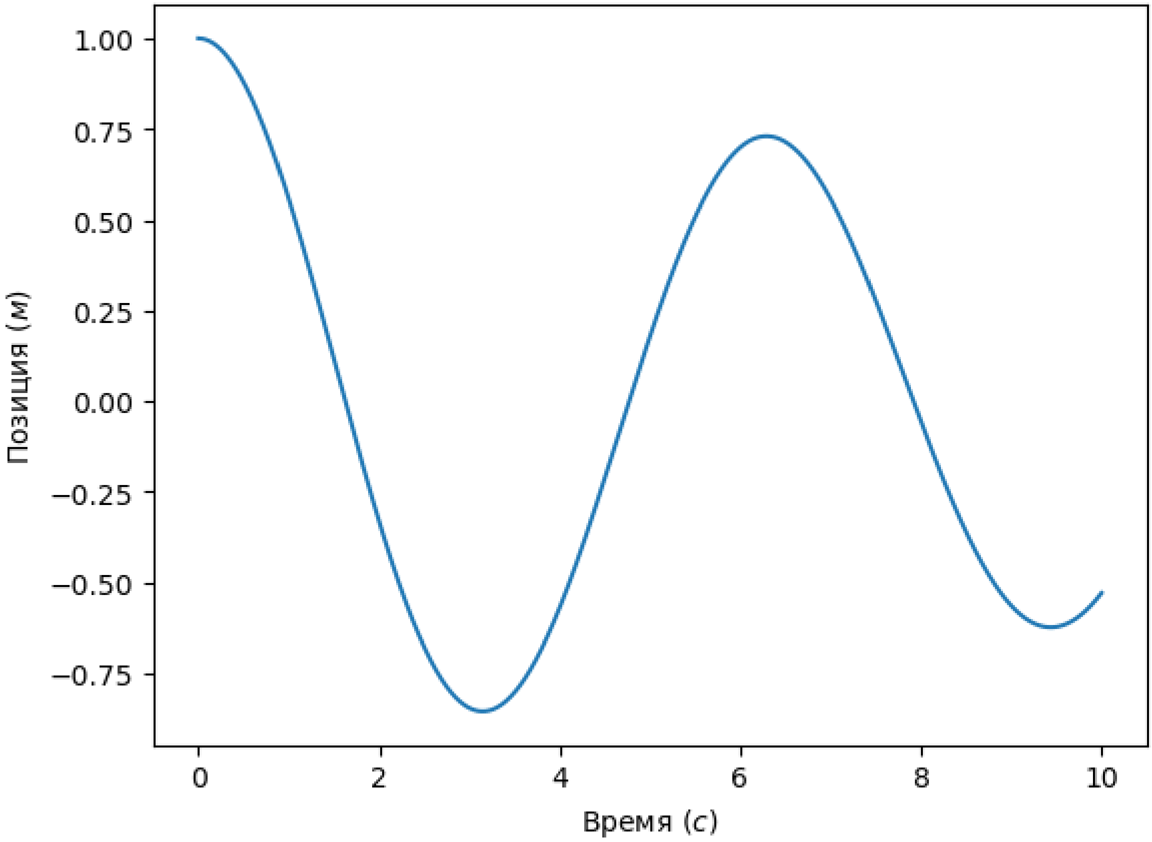
In [78]:
```
m = 1.0
c = 0.1
k = 1.0

x0 = 1.0     # начальное перемещение (м)
v0 = 0.0     # начальная скорость (м/с)

t = np.linspace(0, 10, 1000)

sol = odeint(f, [x0, v0], t)

plt.plot(t, sol[:, 0])
plt.xlabel('Время ($c$)')
plt.ylabel('Позиция ($м$)')
plt.show()
```



1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
0
2
4
6
8
10
Позиция (м)
Время (с)