

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ  
АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО  
ОБРАЗОВАНИЯ «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчёт о лабораторной работе №3 по дисциплине основы программной  
инженерии**

Выполнила:  
Нестеренко Тамара Антоновна,  
3 курс, группа ПИЖ-б-о-20-1,  
Проверил:  
Доцент кафедры инфокоммуникаций,  
Воронкин Р.А.

Ставрополь, 2023 г.

## ВЫПОЛНЕНИЕ

### 1. Практическая часть

# Производная

```
In [4]: # !/usr/bin/env python3
# -*- coding: utf-8 -*-

from sympy import *
```

```
In [5]: x = symbols('x')
y = x*cos(x)
diff(x*cos(x), x)
```

Out[5]:  $-x \sin(x) + \cos(x)$

Рисунок 1 – Пример выполнения программы

```
In [6]: diff(log(x), x, 3)
```

Out[6]:  $\frac{2}{x^3}$

Рисунок 2 – Пример выполнения программы

```
In [7]: diff(log(x), x, x, x)
```

Out[7]:  $\frac{2}{x^3}$

Рисунок 3 – Пример выполнения программы

```
In [8]: y = log(x**3,10)**3
diff(y,x,2).subs(x,10)
```

Out[8]:  $-2 \sin(10) - 10 \cos(10)$

Рисунок 4 – Пример выполнения программы

```
In [9]: y = log(x**3,10)**3  
diff(y,x,2).subs(x,10).simplify()
```

Out[9]:  $-2 \sin(10) - 10 \cos(10)$

Рисунок 5 – Пример выполнения программы

```
In [10]: y = (x**2 + x - 6) / (x**2 - 10*x + 25)  
z = diff(y, x)  
solve(z, x)
```

Out[10]:  $[7/11]$

Рисунок 6 – Пример выполнения программы

```
In [11]: y = symbols('y')  
f = x**2 + y**2 - 4  
idiff(f, y, x)
```

Out[11]:  $-\frac{x}{y}$

Рисунок 7 – Пример выполнения программы

```
In [12]: y = symbols('y')  
f = x**2 + y**2 - 4  
idiff(f, y, x, 2)
```

Out[12]:  $-\frac{\frac{x^2}{y^2} + 1}{y}$

Рисунок 8 – Пример выполнения программы

```
In [13]: idiff(f, y, x, 2).simplify()
```

Out[13]:  $\frac{-x^2 - y^2}{y^3}$

Рисунок 9 – Пример выполнения программы

```
In [14]: t = symbols('t')
x = t - sin(t)
y = 1 - cos(t)
y_diff = diff(y, t) / diff(x, t)
y_diff
```

Out[14]: 
$$\frac{\sin(t)}{1 - \cos(t)}$$

Рисунок 10 – Пример выполнения программы

```
In [15]: y_2diff = diff(y_diff, t) / diff(x, t)
y_2diff.simplify()
```

Out[15]: 
$$-\frac{1}{(\cos(t) - 1)^2}$$

Рисунок 11 – Пример выполнения программы

```
In [111]: import numpy as np
import matplotlib.pyplot as plt

%matplotlib inline
x = np.linspace(-2, 2, 500)
x[(x > -0.01) & (x < 0.01)] = np.nan
y = np.arctan(1 / x)
plt.plot(x, y)
plt.vlines(0, -1.6, 1.6, color = 'g', linestyle = 'dashed')
plt.show()
```

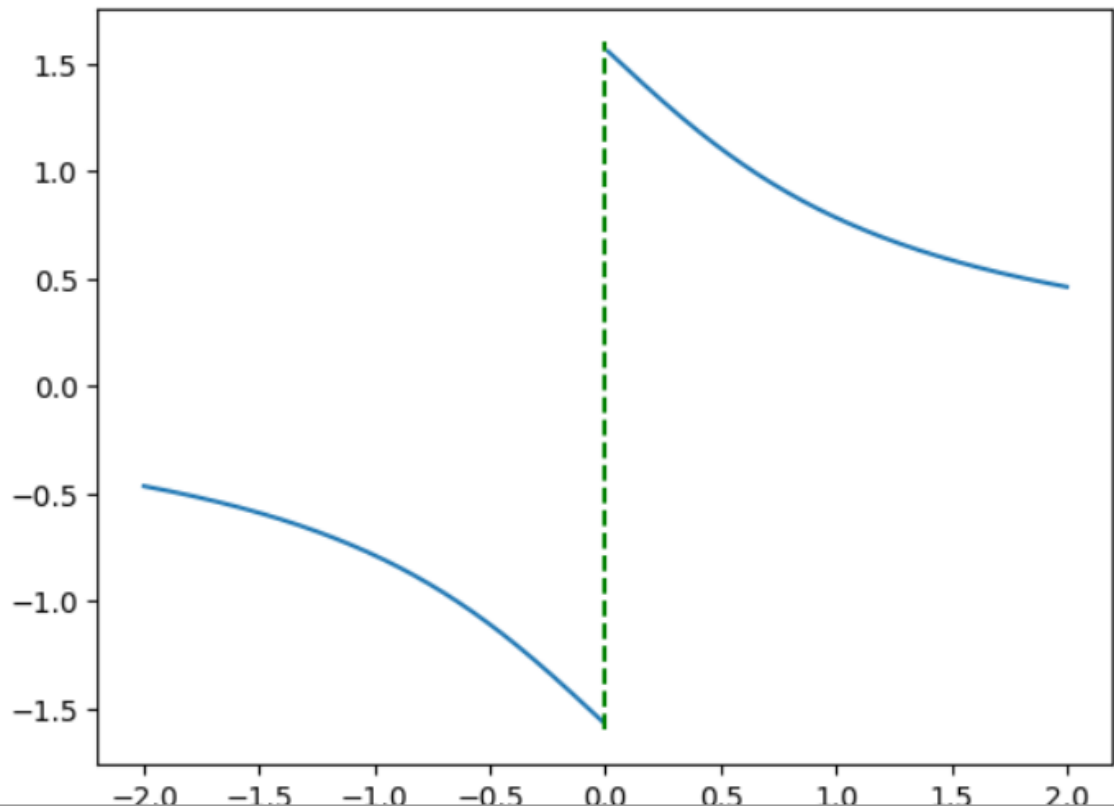


Рисунок 12 – Пример выполнения программы

```
In [17]: limit(z, x, 0, dir = "+")
```

```
Out[17]: 
$$\frac{(10 - 2x)(x^2 + x - 6)}{(x^2 - 10x + 25)^2} + \frac{2x + 1}{x^2 - 10x + 25}$$

```

Рисунок 13 – Пример выполнения программы

```
In [18]: x = np.linspace(-3, 4, 50)
y1 = x**2
plt.plot(x, y1, lw = 2, c = 'b')

x = np.linspace(-1, 4, 50)
y2 = 4*x - 4
plt.plot(x, y2, c = 'r')

plt.xlabel('x')
plt.ylabel('y')
plt.grid(True, linestyle = '-', color = '0.4')

plt.show()
```

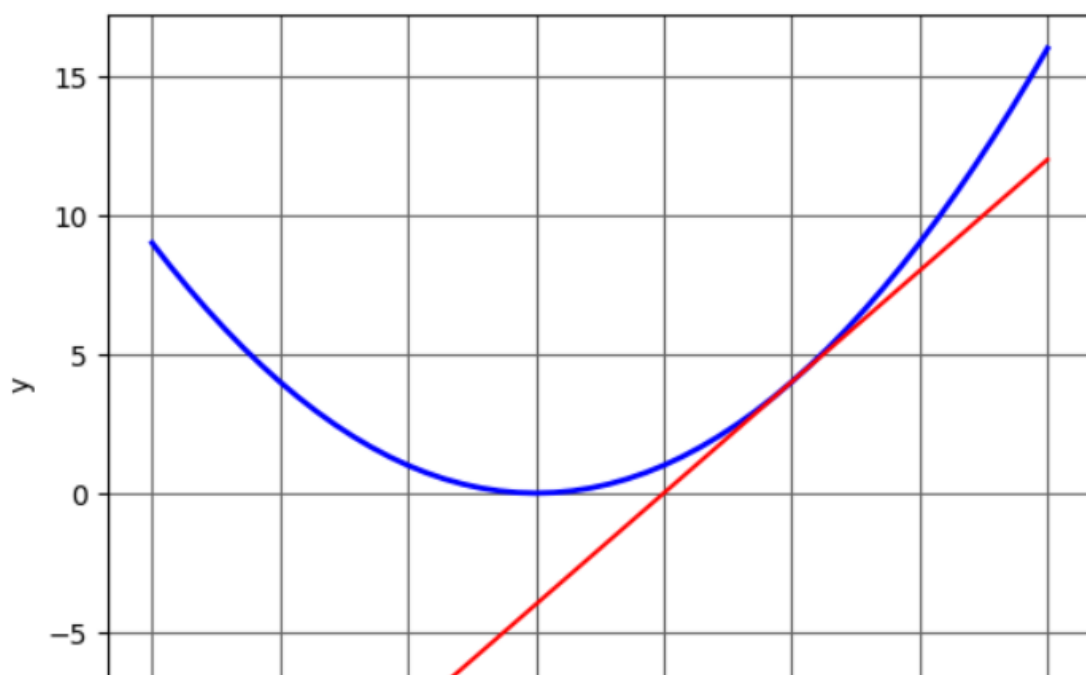


Рисунок 14 – Пример выполнения программы

```
In [29]: def tangent(y, x0):
y0 = y.subs(x, x0)
x1 = x0 + 1
k = diff(y, x).subs(x, x0)
y1 = y0 + k
return Line((x0, y0), (x1, y1))
```

Рисунок 15 – Пример выполнения программы

```
In [31]: x = symbols('x')
y = 6*x**(1/3) + 2*sqrt(x)
x0 = 64
y0 = y.subs(x, x0)
l = tangent(y, 64)
l.equation()
```

Out[31]:  $-\frac{x}{4} + y - 24$

Рисунок 16 – Пример выполнения программы

```
In [55]: p = Point(x0,y0)
l.perpendicular_line(p).equation()
```

Out[55]:  $-x - \frac{y}{4} + 74$

Рисунок 17 – Пример выполнения программы

```
In [58]: x = np.linspace(0, 120, 50)
y1 = 6*x**(1/3) + 2*x**(1/2)
plt.plot(x, y1, lw=2, c='r')

x = np.linspace(10, 120, 50)
y2 = x/4 + 24
plt.plot(x, y2, '--', lw=2, c='b')

x = np.linspace(60, 70, 50)
y3 = 296 - 4*x
plt.plot(x, y3, '-.', lw=2, c='g')

plt.xlabel('x')
plt.ylabel('y')
plt.grid(True, linestyle='-', color='0.4')
plt.axis('equal')
plt.show()
```

Рисунок 18 – Пример выполнения программы

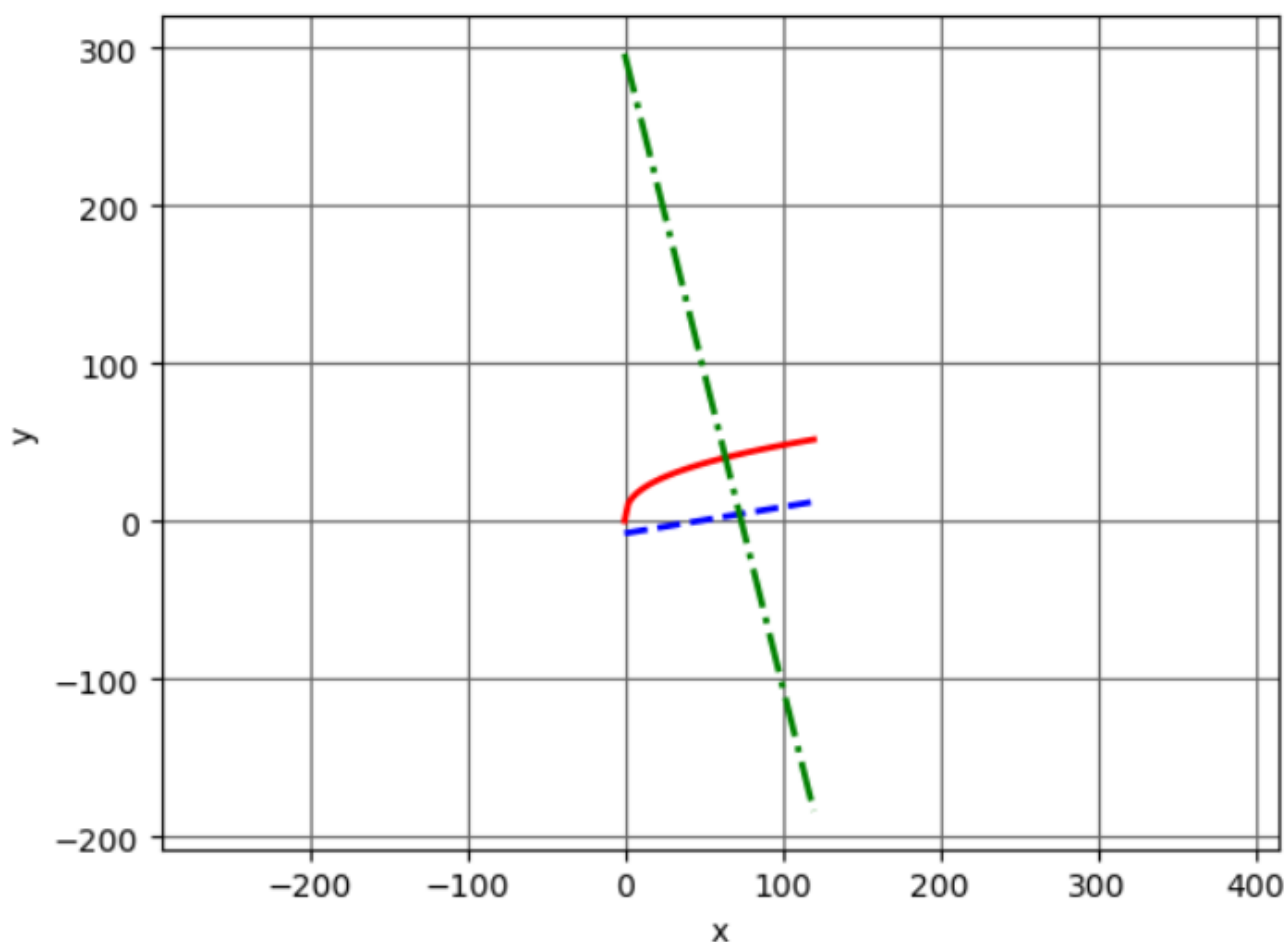


Рисунок 19 – Пример выполнения программы

```
In [69]: def tangent_from_point(y, x1, y1):
x, x0, y0 = symbols('x x0 y0')
y_diff = diff(y, x).subs(x, x0)
y_tang = y_diff*(x - x0) + y0
first_eq = y.subs(x, x0) - y0
second_eq = y_tang.subs(x, x1) - y1
res = solve([first_eq, second_eq], \
            [x0, y0], dict = True)
if len(res) == 1:
    x01 = res[0][x0]
    y01 = res[0][y0]
    return Line((x01, y01), (x1, y1))
else:
    x021 = res[0][x0]; y021 = res[0][y0]
    x022 = res[1][x0]; y022 = res[1][y0]
    return Line((x021, y021), (x1, y1)), \
           Line((x022, y022), (x1, y1))
```

Рисунок 20 – Пример выполнения программы



```
In [78]: x = np.linspace(0, 7, 50)
y1 = np.sqrt(x)
plt.plot(x, y1, lw = 2, c = 'b')

x = np.linspace(-4, 7, 50)
y2 = x/4 + 1
plt.plot(x, y2, c = 'r')

plt.xlabel('x')
plt.ylabel('y')
plt.grid(True, linestyle = '-', color = '0.4')
plt.show()
```



Рисунок 21 – Пример выполнения программы

```

In [88]: t = np.linspace(-1.5, 5, 100)
f = t**3 - 5*t**2 + 5
fd = 3*t**2 - 10*t
fdd = 6*t - 10
plt.plot(t, f, lw = 2, color = 'red', label = "$y(x)$")
plt.plot(t, fd, '--', lw = 2, color = 'b', label = "$y^{\prime}(x)$")
plt.plot(t, fdd, '-.', color = 'g', label = "$y^{\prime\prime}(x)$")
plt.plot([0], [0], 'o', color = 'y')
plt.plot([0], [5], 'o', color = 'y')
plt.plot([3.3], [0], 'o', color = 'y')
plt.plot([3.3], [-13.4], 'o', color = 'y')
plt.plot([1.65], [0], 'o', color = 'b')
plt.plot([1.65], [-4], 'o', color = 'b')
plt.grid(True, linestyle = '--', color = '0.4')
plt.legend()
plt.show()

```

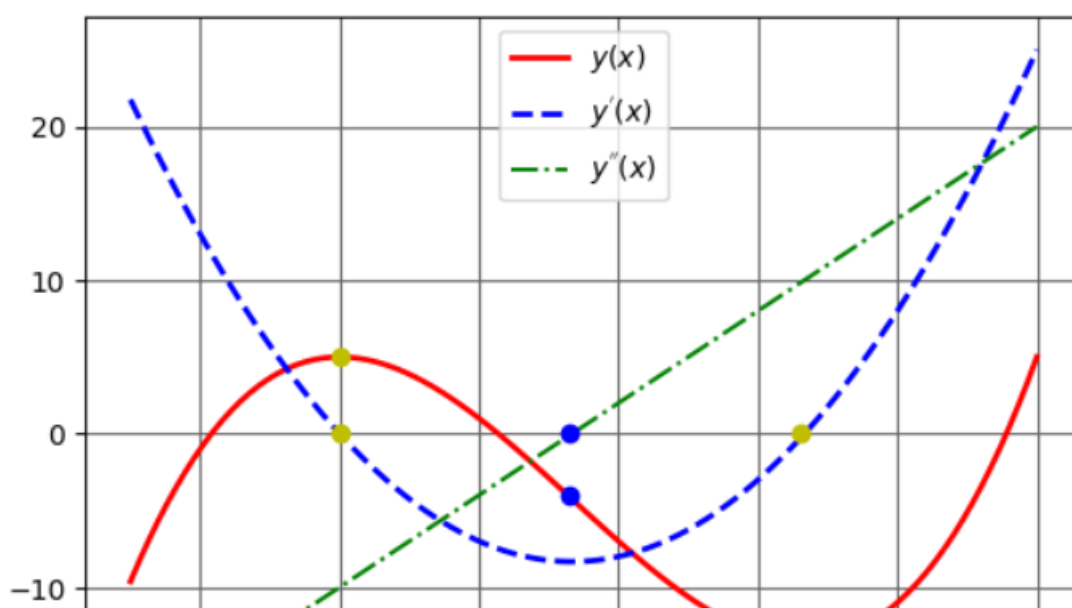


Рисунок 22 – Пример выполнения программы

```

In [89]: from scipy.optimize import minimize

```

```

In [90]: x, y = symbols('x, y')
         solve(x**2 < 3)

```

```

Out[90]:  $-\sqrt{3} < x \wedge x < \sqrt{3}$ 

```

Рисунок 23 – Пример выполнения программы

```
In [92]: solve(x**2 - y**2, x)
```

```
Out[92]: [-y, y]
```

Рисунок 24 – Пример выполнения программы

```
In [93]: f = lambda x: np.exp(-x) - np.exp(-2*x)
```

Рисунок 25 – Пример выполнения программы

```
In [94]: x = np.linspace(0.1, 4, 50)  
plt.plot(x, f(x), 'r')  
plt.xlabel('x')  
plt.ylabel('y')  
plt.grid(True, linestyle = '-', color = '0.4')  
plt.show()
```

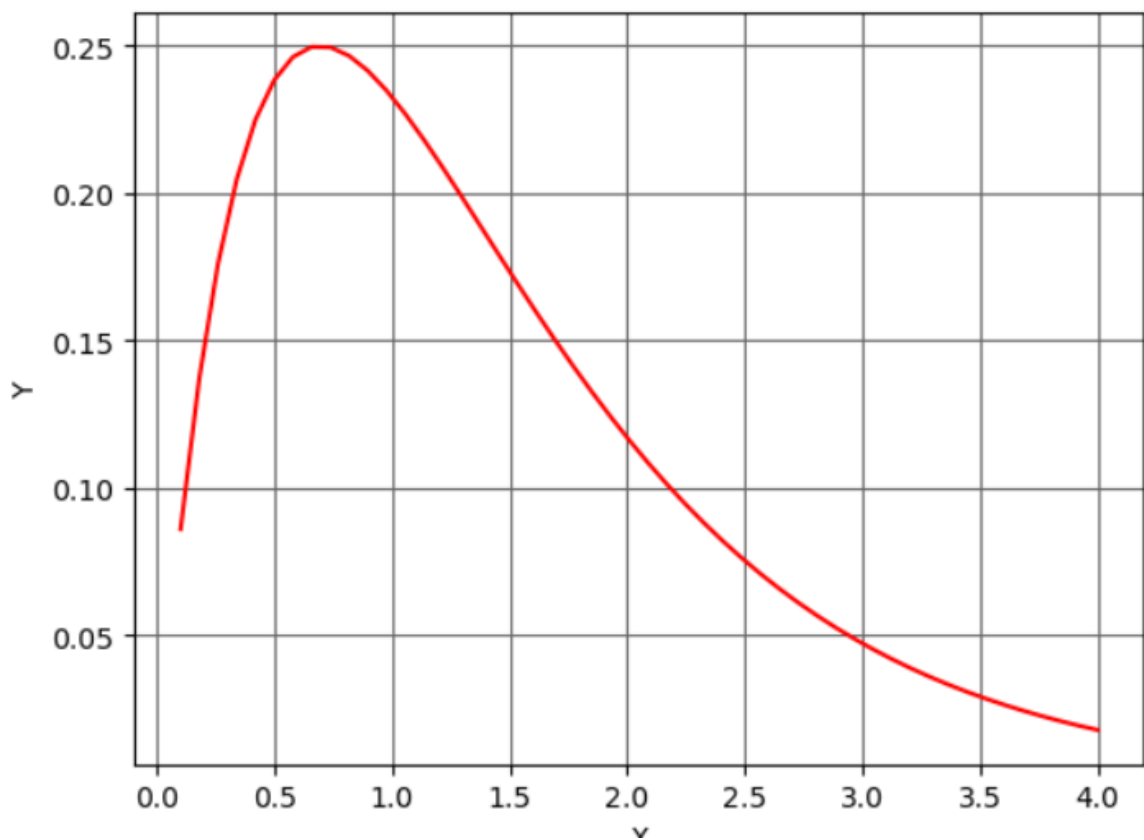


Рисунок 26 – Пример выполнения программы

```
In [95]: f_max = lambda x: -(np.exp(-x) - np.exp(-2*x))
res = minimize(f_max, -2)
print('x_max: %.3f f_max: %.3f' % (res.x, f(res.x)))

x_max: 0.693 f_max: 0.250
```

```
In [96]: res.success
```

```
Out[96]: True
```

```
In [97]: res
```

```
Out[97]: message: Optimization terminated successfully.
success: True
status: 0
      fun: -0.2499999999999441
         x: [ 6.931e-01]
        nit: 12
         jac: [-7.413e-07]
hess_inv: [[ 1.986e+00]]
        nfev: 26
        njev: 13
```

Рисунок 27 – Пример выполнения программы

```
In [133]: def tangent_plane(F, M):
F_diff_x = diff(F, x).subs({x:M.x, y:M.y, z:M.z})
F_diff_y = diff(F, y).subs({x:M.x, y:M.y, z:M.z})
F_diff_z = diff(F, z).subs({x:M.x, y:M.y, z:M.z})
n = Point(F_diff_x, F_diff_y, F_diff_z)
p = Plane(M, normal_vector = n).equation()
K = Point(M.x+n.x, M.y+n.y, M.z+n.z)
l_n = Line(M, K).arbitrary_point()

return p, l_n
```

Рисунок 28 – Пример выполнения программы

```
In [134]: x, y, z = symbols('x y z')
M = Point(1, 1, 1)
F = x**2 + y**2 + z**2 - 9
p, l_n = tangent_plane(F, M)
```

```
In [122]: p
```

```
Out[122]: 2x + 2y + 2z - 6
```

```
In [123]: l_n
```

```
Out[123]: Point3D(2t + 1, 2t + 1, 2t + 1)
```

Рисунок 29 – Пример выполнения программы

```
In [125]: z = lambda w: (w[0] - 1)**2 + (w[1] - 3)**4
res = minimize(z, (0, 0))
```

```
In [126]: res.x
```

```
Out[126]: array([0.99999999, 2.98725136])
```

```
In [127]: res = minimize(z, (9.999, 3.001))
res.x
```

```
Out[127]: array([1.00000001, 3.001    ])
```

```
In [129]: z((1, 3)) < z((0.999, 3.001))
```

```
Out[129]: True
```

```
In [130]: z((1, 3))
```

```
Out[130]: 0
```

Рисунок 30 – Пример выполнения программы

```
In [131]: z = lambda w: w[0]**4 + w[1]**4 - 2*w[0]**2 + \
          4*w[0]*w[1] - 2*w[1]**2
```

```
In [143]: z = 4.5*x**(0.33) * y**(0.66)
z_x = diff(z, x)
z_y = diff(z, y)

E_x = (x/z) * z_x
E_y = (y/z) * z_y
print('E_x: %.2f E_y: %.2f' % (E_x, E_y))

E_x: 0.33 E_y: 0.66
```

Рисунок 31 – Пример выполнения программы

```
In [144]: K, V0 = symbols('K, V0')
V = V0*log(5+K**2)
Vprim2 = diff(V, K, 2)
Vprim3 = diff(V, K, 3)
s = solve(Vprim2, K)
s
```

```
Out[144]: [-sqrt(5), sqrt(5)]
```

```
In [145]: Vprim3.subs(K, s[1])
```

```
Out[145]: 
$$-\frac{\sqrt{5}V_0}{25}$$

```

Рисунок 32 – Пример выполнения программы

# Индивидуальное задание, вариант 10

Задание №1. Найти вторую производную функции  $y = xe^{x^2}$

```
In [1]: #!/usr/bin/env python
# -*- coding:utf-8 -*-

from sympy import *
x = symbols('x')
y = x*exp(x**2)
diff(y, x, 2)
```

Out[1]:  $2x(2x^2 + 3)e^{x^2}$

Рисунок 33 – Пример выполнения индивидуального задания

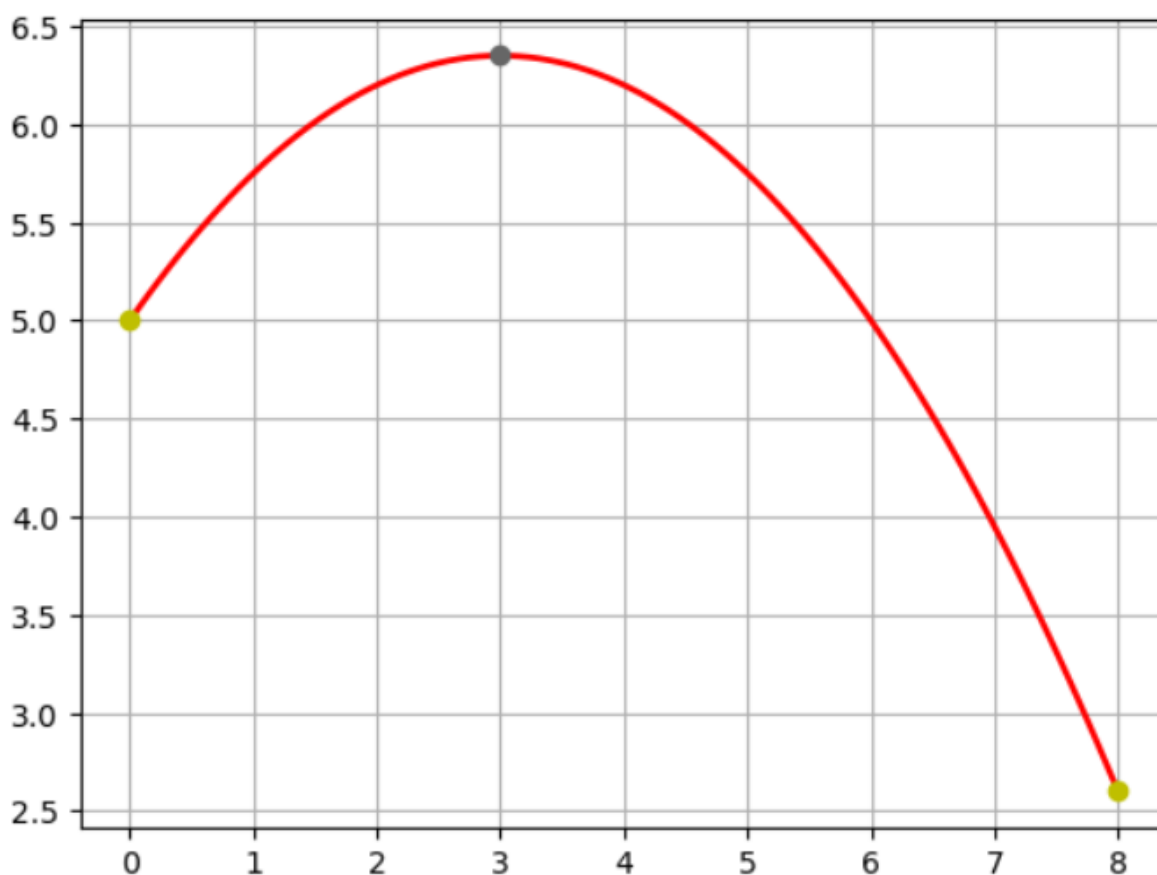
```
In [22]: #!/usr/bin/env python
# -*- coding:utf-8 -*-

from sympy import *
import numpy as np
import matplotlib.pyplot as plt
t = symbols('t')
q = -0.05*t**3 + 0.45*t**2 + 5*t
y = diff(q, t)
print(y)

-0.15*t**2 + 0.9*t + 5
```

```
In [55]: t = np.linspace(0, 8, 100)
z = -0.15*t**2 + 0.9*t + 5
plt.plot(t, z, '-', lw = 2, color = 'r')
plt.grid(True)
plt.plot([0], [5], 'o', color = 'y')
plt.plot([3], [6.35], 'o', color = '0.4')
plt.plot([8], [2.6], 'o', color = 'y')
plt.show()
```

Рисунок 34 – Пример выполнения индивидуального задания



Графиком этой зависимости  $y$  от  $t$  является парабола с ветвями, направленными вниз (рис.1). Приведенный график показывает, что производительность труда в первые три часа работы возрастает (с 5 до 6,35 единиц продукции в час), а потом монотонно падает, и к концу рабочего дня составляет лишь 2,6 единиц продукции в час.

Рисунок 35 – Пример выполнения индивидуального задания

Ссылка на репозиторий: [https://github.com/tamaranesterenko/PI\\_LR\\_3](https://github.com/tamaranesterenko/PI_LR_3)