

ЛАБОРАТОРНА РОБОТА №9

Тема: «Наближення функцій поліномом Тейлора»

Трубчанінов Андрій Сергійович

ФІТ 2-8

В-29

Код:

```
import sympy as sp
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import approximate_taylor_polynomial

# Задаємо символічну змінну x та функцію f(x)
x = sp.symbols('x')
f = sp.exp(x) + x**2

# Знаходимо перші три похідні
f1 = sp.diff(f, x)
f2 = sp.diff(f1, x)
f3 = sp.diff(f2, x)

# Знаходимо значення функції та її похідних в точці x=0
x0 = 0
f_x0 = f.subs(x, x0).evalf()
f1_x0 = f1.subs(x, x0).evalf()
f2_x0 = f2.subs(x, x0).evalf()
f3_x0 = f3.subs(x, x0).evalf()

# Обчислюємо значення многочлена Тейлора в точці x=0
T = f_x0 + f1_x0*(x-x0) + (f2_x0/2)*(x-x0)**2 + (f3_x0/6)*(x-x0)**3

# Виводимо значення функції та її наближення за багаточленом Тейлора в точці x=0
print("f(0) =", f_x0)
print("T(x) =", T.evalf())

# Будуємо графіки
x_vals = np.linspace(-2, 2, 1000)

# Виводимо значення функції та її наближення за багаточленом Тейлора в точці x=0
print("f(0) =", f_x0)
print("T(x) =", T.evalf())

# Будуємо графіки
x_vals = np.linspace(-2, 2, 1000)
f_vals = np.array([f.subs(x, xi).evalf() for xi in x_vals])
T_vals = np.array([T.subs(x, xi).evalf() for xi in x_vals])

fig, ax = plt.subplots()
ax.plot(x_vals, f_vals, label="f(x)")
ax.plot(x_vals, T_vals, label="T(x)")
ax.legend()
ax.set_xlabel("x")
ax.set_ylabel("y")
ax.set_title("Графік функції та наближення многочленом Тейлора")
plt.grid(True)
plt.show()
```

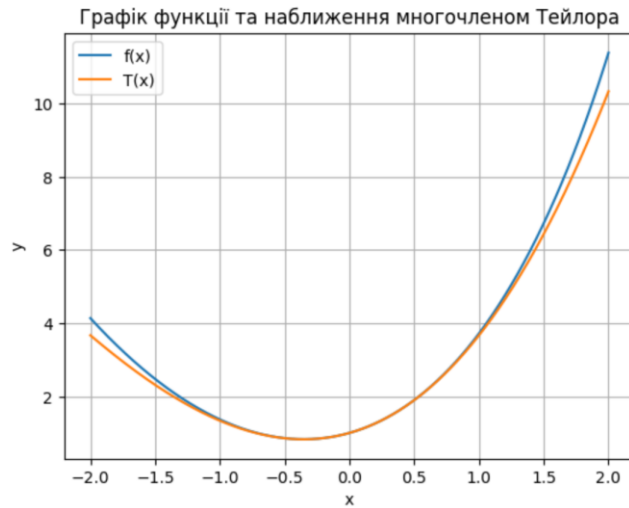
f(0) = 1.00000000000000

T(x) = 0.166666666666667*x**3 + 1.5*x**2 + 1.0*x + 1.0

Графік функції та наближення многочленом Тейлора



$f(0) = 1.0000000000000000$
 $T(x) = 0.1666666666666667 \cdot x^3 + 1.5 \cdot x^2 + 1.0 \cdot x + 1.0$



```

# Побудова багаточлена Тейлора за допомогою approximate_taylor_polynomial
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import approximate_taylor_polynomial

# Задана функція
def f(x):
    return np.exp(x) + x**2

x = np.linspace(-2.0, 2.0, num=400)

plt.figure(figsize=(10, 6))
plt.plot(x, f(x), label="f(x) curve", color='blue')
degree = 3

taylor = approximate_taylor_polynomial(f, 0, degree, 1)
print('taylor=', taylor)

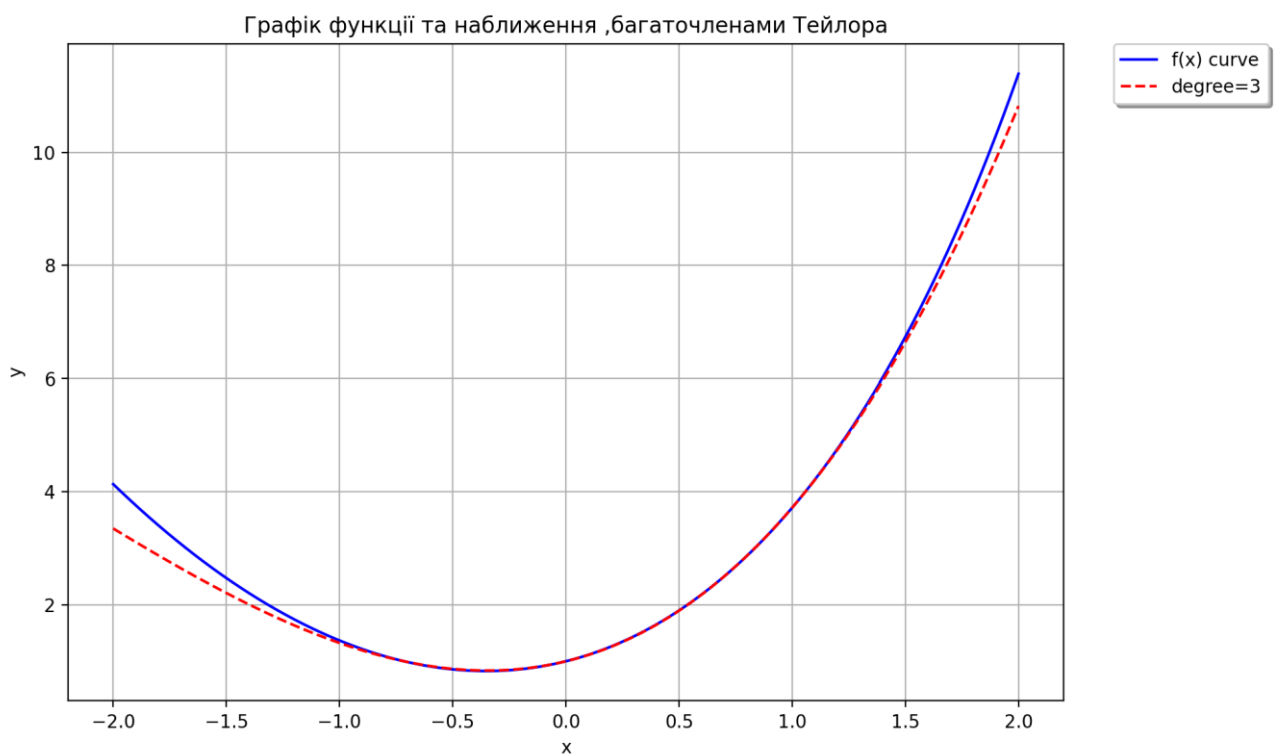
plt.plot(x, taylor(x), label=f"degree={degree}", color='red', linestyle='--' )

plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0.0, shadow=True)
plt.xlabel("x")
plt.ylabel("y")
plt.title("Графік функції та наближення ,багаточленами Тейлора")
plt.tight_layout()
plt.grid()
plt.show()

taylor=
      3      2
0.2233 x + 1.521 x + 0.9738 x + 1

```

Графік функції та наближення ,багаточленами Тейлора



Код зі скрінів:

```
import sympy as sp
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import approximate_taylor_polynomial

# Задаємо символну змінну x та функцію f(x)
x = sp.symbols("x")
f = sp.exp(x) + x**2

# Знаходимо перші три похідні
f1 = sp.diff(f, x)
f2 = sp.diff(f1, x)
f3 = sp.diff(f2, x)

# Знаходимо значення функції та її похідних в точці x=0
x0 = 0
f_x0 = f.subs(x, x0).evalf()
f1_x0 = f1.subs(x, x0).evalf()
f2_x0 = f2.subs(x, x0).evalf()
f3_x0 = f3.subs(x, x0).evalf()

# Обчислюємо значення многочлена Тейлора в точці x=0
```

$$T = f_{x0} + f1_{x0} * (x - x0) + (f2_{x0} / 2) * (x - x0) ** 2 + (f3_{x0} / 6) * (x - x0) ** 3$$

Виводимо значення функції та її наближення за багаточленом Тейлора в точці $x=0$

```
print("f(0) =", f_x0)
```

```
print("T(x) =", T.evalf())
```

Будуємо графіки

```
x_vals = np.linspace(-2, 2, 1000)
```

```
f_vals = np.array([f.subs(x, xi).evalf() for xi in x_vals])
```

```
T_vals = np.array([T.subs(x, xi).evalf() for xi in x_vals])
```

```
fig, ax = plt.subplots()
```

```
ax.plot(x_vals, f_vals, label="f(x)")
```

```
ax.plot(x_vals, T_vals, label="T(x)")
```

```
ax.legend()
```

```
ax.set_xlabel("x")
```

```
ax.set_ylabel("y")
```

```
ax.set_title("Графік функції та наближення многочленом Тейлора")
```

```
plt.grid(True)
```

```
plt.show()
```

```
# Побудова багаточлена Тейлора за допомогою
approximate_taylor_polynomial

import numpy as np

import matplotlib.pyplot as plt

from scipy.interpolate import approximate_taylor_polynomial


# Задана функція
def f(x):

    return np.exp(x) + x**2


x = np.linspace(-2.0, 2.0, num=400)


plt.figure(figsize=(10, 6))
plt.plot(x, f(x), label="f(x) curve", color="blue")
degree = 3


taylor = approximate_taylor_polynomial(f, 0, degree, 1)
print("taylor=", taylor)


plt.plot(x, taylor(x), label=f"degree={degree}", color="red",
linestyle="--")


plt.legend(bbox_to_anchor=(1.05, 1), loc="upper left",
borderaxespad=0.0, shadow=True)
```

```
plt.xlabel("x")
```

```
plt.ylabel("y")
```

```
plt.title("Графік функції та наближення ,багаточленами  
Тейлора")
```

```
plt.tight_layout()
```

```
plt.grid()
```

```
plt.show()
```

GitHub: <https://github.com/Rey-ui/chiselni-metody>