Toán học tính toán: Python buổi 4/10

3 Tính gần đúng đạo hàm tích phân

```
1 from sympy import *
2 f = lambda x: E ** sin(x)
3 import numpy as np
4 X = np.linspace(0, 2, 11)
5 Y = [ f(x) for x in X ]
6 h = (2 - 0) / 10
```

Mã 20: Chia lưới

3.1 Tính gần đúng đạo hàm

```
1 (-Y[2] + 4*Y[1] - 3*Y[0]) / 2 / h
2 [ (Y[i+1] - Y[i-1]) / 2 / h for i in range(1, 10) ]
3 (3*Y[10] - 4*Y[9] + Y[8]) / 2 / h
4 from sympy import *
5 x = symbols('x')
6 f(x).diff()
7 [ f(x).diff().subs(x, a) for a in X ]
```

Mã 21: Đạo hàm cấp một: công thức ba điểm

```
1 (Y[2] - 2*Y[1] + Y[0]) / h**2
2 [ (Y[i+1] - 2*Y[i] + Y[i-1]) / h**2 for i in range(1, 10) ]
3 (Y[10] - 2*Y[9] + Y[8]) / h**2
4 f(x).diff(x, 2)
5 [ f(x).diff(x, 2).subs(x, a) for a in X ]
```

Mã 22: Đạo hàm cấp hai: công thức ba điểm

3.2 Tính gần đúng tích phân

```
sum( [(X[i] - X[i-1]) * (Y[i] + Y[i-1]) / 2 for i in range(1, 11)] ) # công thức tổng quát

M2 = N(2 * E, 6)
M2 * (2-0)**3 / 12 / 10**2

4 sqrt( M2 * (2-0)**3 / 12 / 10**-4 )
```

Mã 23: Phương pháp hình thang

```
1 sum( [(X[2*i] - X[2*i-2]) * (Y[2*i] + 4*Y[2*i-1] + Y[2*i-2]) / 6 for i in range(1, 6)]
    )
2 M4 = N(15 * E)
3 M4 * (2-0)**5 / 180 / 10**4
```

```
4 ( M4 * (2-0)**5 / 180 / 10**-4 ) ** (1/4)
```

Mã 24: Phương pháp Simpson

```
1 from sympy import *
  x, t, h, x0 = symbols('x t h x0')
                                       # n mốc nội suy x_0, x_1, \dots, x_{n-1}, ứng với công thức n-điểm
3 n = 2
4 Y = MatrixSymbol('Y', n, 1)
5 D = [Y[i] for i in range(n)] # sai phân
6 P = D[0]
                                       # đa thức nội suy
7 for k in range(1, n):
    for i in range(n - k):
      D[i] = D[i + 1] - D[i] \qquad \# \Delta^k y_i
9
     N = D[O]
10
   for i in range(k):
   N *= (t - i) / (i + 1)
     P += N
                                       # t(-Y_{0,0} + Y_{1,0}) + Y_{0,0}
14 P
15 P.diff(t) / h
16 P.integrate((t, 0, n-1)) * h
```

Mã 25: Xây dựng công thức tính gần đúng đạo hàm, tích phân