

Toán học tính toán: Python buổi 3/10

4 Nội suy

```
1 X = [-1, 0, 1, 2]
2 Y = [4, 3, 2, 7]
```

Mã 16: Nội suy: khai báo

4.1 Nội suy: phương pháp Lagrange

```
1 def L(i, x):
2     prod = 1
3     for j in range(4):
4         if j != i:
5             prod *= (x - X[j]) / (X[i] - X[j])
6     return prod
7
8 from sympy import *
9 x = symbols('x')
10 L(0, x)
11 L(0, x).expand()
12
13 P = 0
14 for i in range(4):
15     P += Y[i] * L(i, x)
16
17 P
18 P.expand()
```

Mã 17: Nội suy: phương pháp Lagrange

4.2 Nội suy: phương pháp Newton tiến, và lùi

```
1 d = lambda k, i: Y[i] if k == 0 else d(k-1, i+1) - d(k-1, i)
2 [ [d(k, i) for i in range(4-k)] for k in range(4) ]
```

Mã 18: Phương pháp Newton: sai phân

```
1 from sympy import *
2 x, t = symbols('x t')
3
4 P = Y[0] # Y[3]
5 for k in range(1, 4):
6     prod = d(k, 0) / factorial(k) # d(k, 3-k)
7     for i in range(k):
8         prod *= t - i # t + i
9     P += prod
10
11 P
12 P.subs(t, (x - X[0]) / 1) # X[3]
```

13 `_ . expand ()`

Mã 19: Nội suy: phương pháp Newton tiến / lùi