

exercise3 (Score: 2.0 / 9.0)

1. [Test cell](#) (Score: 2.0 / 2.0)
2. [Test cell](#) (Score: 0.0 / 1.0)
3. [Test cell](#) (Score: 0.0 / 2.0)
4. [Test cell](#) (Score: 0.0 / 2.0)
5. [Test cell](#) (Score: 0.0 / 2.0)

## Lab 3

1. 提交作業之前，建議可以先點選上方工具列的**Kernel**，再選擇**Restart & Run All**，檢查一下是否程式跑起來都沒有問題，最後記得儲存。
2. 請先填上下方的姓名(name)及學號(student\_id)再開始作答，例如：

```
name = "我的名字"  
student_id= "B06201000"
```

3. 演算法的實作可以參考[lab-3 \(https://yuanyuyuan.github.io/itcm/lab-3.html\)](https://yuanyuyuan.github.io/itcm/lab-3.html), 有任何問題歡迎找助教詢問。
4. **Deadline: 10/30(Wed.)**

In [1]:

```
name = "陳彥宇"  
student_id = "B05303134"
```

## Exercise 3

The price (in euros) of a magazine has changed as follows:

Nov. 87	Dec. 88	Nov. 90	Jan. 93	Jan. 95	Jan. 96	Nov. 96	Nov. 00
4.5	5.0	6.0	6.5	7.0	7.5	8.0	8.0

1. Use the interpolating polynomial of **\*degree 7\*** to estimate the price in February 1989, in April 1998 and in November 2002.

Part 0. Import libraries.

In [2]:

```
import matplotlib.pyplot as plt  
import numpy as np
```

Part 1. Define the polynomial interpolation function.

Please refer part of polynomial interpolation function in " *lagrange.ipynb* ".

In [3]:

(Top)

```
def lagrange(points):  
  
    # 請參考 hands-on 的 Lagrange polynomial interpolation  
    # ===== 請實做程式 =====  
    def polynomial(x):  
        total_sum = 0  
        n = len(points)  
        for i in range(n):  
            x_i, y_i = points[i]  
            def g(i, n):  
                product = 1  
                for j in range(n):  
                    if i == j:  
                        continue  
                    x_j, y_j = points[j]  
                    product *= (x-x_j)/float(x_i-x_j)  
                return product  
            total_sum += y_i*g(i,n)  
        return total_sum  
    return polynomial  
    # =====
```

In [4]:

interpolation\_function

(Top)

```
# Test  
P = lagrange((  
    (0, 0),  
    (1, 1),  
    (-1, 1)  
))  
  
print('P(2) =', P(2))  
  
### BEGIN HIDDEN TESTS  
P = lagrange((  
    (0, 0),  
    (1, 1),  
    (-1, 1)  
))  
  
assert P(0) == 0, 'P(0) is wrong!'  
assert P(1) == 1, 'P(1) is wrong!'  
assert P(-2) == 4, 'P(-2) is wrong!'  
assert P(3) == 9, 'P(3) is wrong!'  
### END HIDDEN TESTS
```

P(2) = 4.0

## Part 2. Transfer data to input points (x: dates, y: prices).

In [5]:

(Top)

```
# Hint: points = ((*, *), ..., (*, *))  
# ===== 請實做程式 =====  
points = ((1998+11/12,4.5), (1999,5), (2001+11/12,5.5), (2004+1/12,6), (2006+1/12,6.5),  
          (2007+1/12,7), (2007+11/12,7.5), (2011+11/12,8))  
# =====
```

In [6]:

points\_date

(Top)

```
print('points:', points)

### BEGIN HIDDEN TESTS
data = np.ndarray.flatten(np.array(points))
prices = [4.5, 5., 6., 6.5, 7., 7.5, 8.]

assert len(data) == 16, 'points is wrong!'
assert np.sum(np.isin(data, prices)) == 8, 'Wrong prices in points!'
### END HIDDEN TESTS
```

points: ((1998.9166666666667, 4.5), (1999, 5), (2001.9166666666667, 5.5), (2004.0833333333333, 6), (2006.0833333333333, 6.5), (2007.0833333333333, 7), (2007.9166666666667, 7.5), (2011.9166666666667, 8))

```
-----
AssertionError                                Traceback (most recent call last)
<ipython-input-6-87f3e213e86b> in <module>
      6
      7 assert len(data) == 16, 'points is wrong!'
----> 8 assert np.sum(np.isin(data, prices)) == 8, 'Wrong prices in points!'
      9 ### END HIDDEN TESTS
```

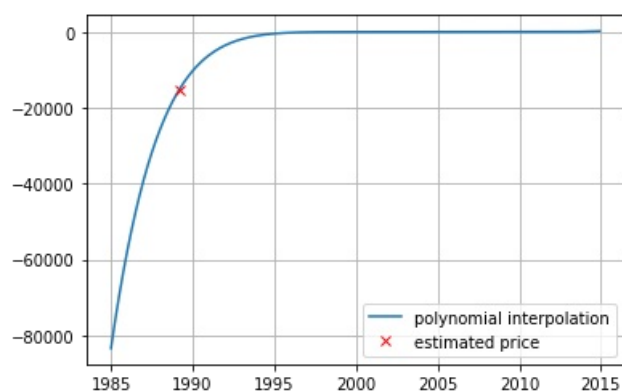
AssertionError: Wrong prices in points!

### Part 3-1. Estimate the price in February 1989.

In [7]:

(Top)

```
# ===== 請實做程式 =====
estimated_price = lagrange(points)(1989+2/12)
x_range = np.arange(1985,2015,1/12)
plt.plot(x_range,lagrange(points)(x_range), label = 'polynomial interpolation')
plt.plot(1989+2/12, estimated_price, 'rx', label = 'estimated price')
plt.grid(True)
plt.legend()
plt.show()
# =====
```



In [8]:

Feb\_1989

(Top)

```
print("My estimated price in February 1989 is", estimated_price)

### BEGIN HIDDEN TESTS
assert abs(estimated_price - 5.09) < 5e-2, 'Estimated price is wrong!'
### END HIDDEN TESTS
```

My estimated price in February 1989 is -15293.991483208742

```
-----
AssertionError                                Traceback (most recent call last)
<ipython-input-8-e546e2d84dac> in <module>
      2
      3 ### BEGIN HIDDEN TESTS
----> 4 assert abs(estimated_price - 5.09) < 5e-2, 'Estimated price is wrong!'
      5 ### END HIDDEN TESTS
```

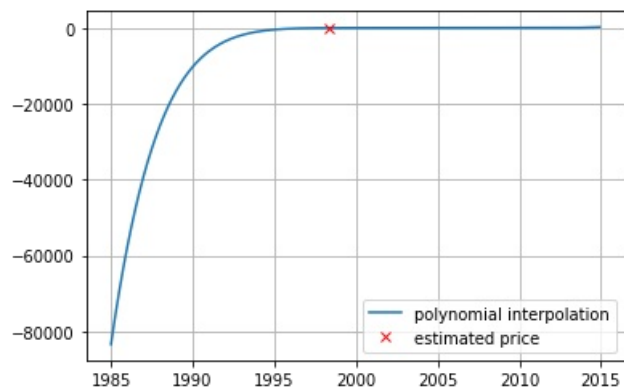
AssertionError: Estimated price is wrong!

## Part 3-2. Estimate the price in April 1998.

In [9]:

(Top)

```
# ===== 請實做程式 =====
estimated_price = lagrange(points)(1998+4/12)
x_range = np.arange(1985,2015,1/12)
plt.plot(x_range,lagrange(points)(x_range), label = 'polynomial interpolation')
plt.plot(1998+4/12, estimated_price, 'rx', label = 'estimated price')
plt.grid(True)
plt.legend()
plt.show()
# =====
```



In [10]:

April\_1998

(Top)

```
print("My estimated price in April 1998 is", estimated_price)

### BEGIN HIDDEN TESTS
assert abs(estimated_price - 8.67) < 5e-2, 'Estimated price is wrong!'
### END HIDDEN TESTS
```

My estimated price in April 1998 is -1.9226976671218634

```
-----
AssertionError                                Traceback (most recent call last)
<ipython-input-10-ff44be72b30c> in <module>
      2
      3 ### BEGIN HIDDEN TESTS
----> 4 assert abs(estimated_price - 8.67) < 5e-2, 'Estimated price is wrong!'
      5 ### END HIDDEN TESTS
```

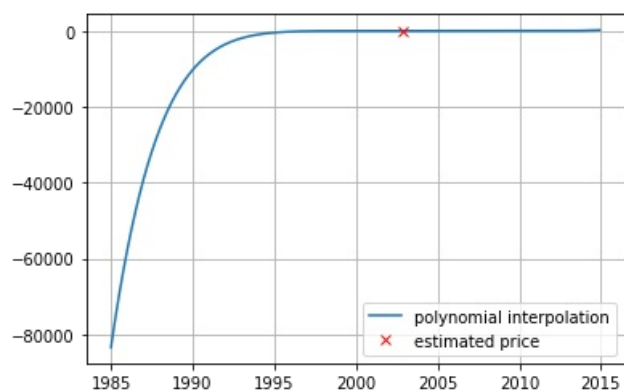
AssertionError: Estimated price is wrong!

### Part 3-3. Estimate the price in November 2002.

In [11]:

(Top)

```
# ===== 請實做程式 =====
estimated_price = lagrange(points)(2002+11/12)
x_range = np.arange(1985,2015,1/12)
plt.plot(x_range,lagrange(points)(x_range), label = 'polynomial interpolation')
plt.plot(2002+11/12, estimated_price, 'rx', label = 'estimated price')
plt.legend()
plt.grid(True)
plt.show()
# =====
```



In [12]:

Nov\_2002

(Top)

```
print("My estimated price in November 2002 is", estimated_price)

### BEGIN HIDDEN TESTS
assert abs(estimated_price - 11.24) < 5e-2, 'Estimated price is wrong!'
### END HIDDEN TESTS
```

My estimated price in November 2002 is 5.624590311057412

```
-----
AssertionError                                Traceback (most recent call last)
<ipython-input-12-429d3b403597> in <module>
      2
      3 ### BEGIN HIDDEN TESTS
----> 4 assert abs(estimated_price - 11.24) < 5e-2, 'Estimated price is wrong!'
      5 ### END HIDDEN TESTS
```

AssertionError: Estimated price is wrong!