

實驗物理學（二）
實驗日誌

Fundamental Python
Curve Fit

Group 2

洪 瑜 B125090009

黃巧涵 B122030003

洪懌平 B102030019

2025/04/22

1 實驗條件紀錄

使用講義之參數、條件：

- 設定隨機種子為42
- 產生50個0-10之間的等間距之 x 值
- 定義用來擬合的模型，透過`curve_fit`找到最適合的 A 和 k
- 建立模型： $y = 2.5e^{-1.3x}$ ($A = 2.5$, $k = 1.3$)，並在每一個 y 值上加入一高斯隨機誤差（平均為0，標準差為1）
- 每個點的誤差設為 ± 0.1

程式碼詳情見Fig.1、Fig.2之註解。

```
1  import numpy as np
2  from scipy.optimize import curve_fit
3  import matplotlib.pyplot as plt
4
5  #隨機種子&資料
6  np.random.seed(42)
7  x = np.linspace(0, 10, 50)
8  #y誤差相同，固定為0.1
9  y = 2.5 * np.exp(-1.3*x)+np.random.normal(0, 0.1, x.size)
10 sigma = 0.1 * np.ones_like(y)
11
12 #model
13 def model(x, A, k):
14     return A * np.exp(-k*x)
15
16 #曲線的兩種設定
17 popt_true,pcov_true = curve_fit(model, x, y, sigma=sigma, absolute_sigma=True)
18 popt_false,pcov_false = curve_fit(model, x, y, sigma=sigma, absolute_sigma=False)
19
20 # 殘差計算
21 residuals_true = y - model(x, *popt_true)
22 residuals_false = y - model(x, *popt_false)
23
24 # 擬合曲線
25 x_fit = np.linspace(0, 10, 200)
26 y_fit_true = model(x_fit, *popt_true)
27 y_fit_false = model(x_fit, *popt_false)
28
```

Figure 1: 程式碼(1)

```
30 # 繪圖：資料與擬合結果
31 plt.figure(figsize=(10, 8))
32
33 plt.subplot(2, 1, 1)
34 plt.errorbar(x, y, yerr=sigma, fmt='o', capsize=3, label='Data with error bars')
35 plt.plot(x_fit, y_fit_true, 'r-', label='Fit (absolute_sigma=True)')
36 plt.plot(x_fit, y_fit_false, 'g--', label='Fit (absolute_sigma=False)')
37 plt.title('Curve Fitting with Fixed Sigma = 0.1')
38 plt.xlabel('x')
39 plt.ylabel('y')
40 plt.legend()
41
42 # 繪圖：殘差比較
43 plt.subplot(2, 1, 2)
44 plt.plot(x, residuals_true, 'ro-', label='Residuals (True)')
45 plt.plot(x, residuals_false, 'go--', label='Residuals (False)')
46 plt.axhline(0, color='gray', linestyle='--')
47 plt.xlabel('x')
48 plt.ylabel('Residuals')
49 plt.legend()
50
51 plt.tight_layout()
52 plt.show()
53
54 # 列印擬合參數與不確定度
55 perr_true = np.sqrt(np.diag(pcov_true))
56 perr_false = np.sqrt(np.diag(pcov_false))
57
58 print("=== absolute_sigma=True ===")
59 print(f"A = {popt_true[0]:.4f} ± {perr_true[0]:.4f}")
60 print(f"k = {popt_true[1]:.4f} ± {perr_true[1]:.4f}")
61
62 print("\n=== absolute_sigma=False ===")
63 print(f"A = {popt_false[0]:.4f} ± {perr_false[0]:.4f}")
64 print(f"k = {popt_false[1]:.4f} ± {perr_false[1]:.4f}")
```

Figure 2: 程式碼(2)

2 實驗數據整理

輸出：

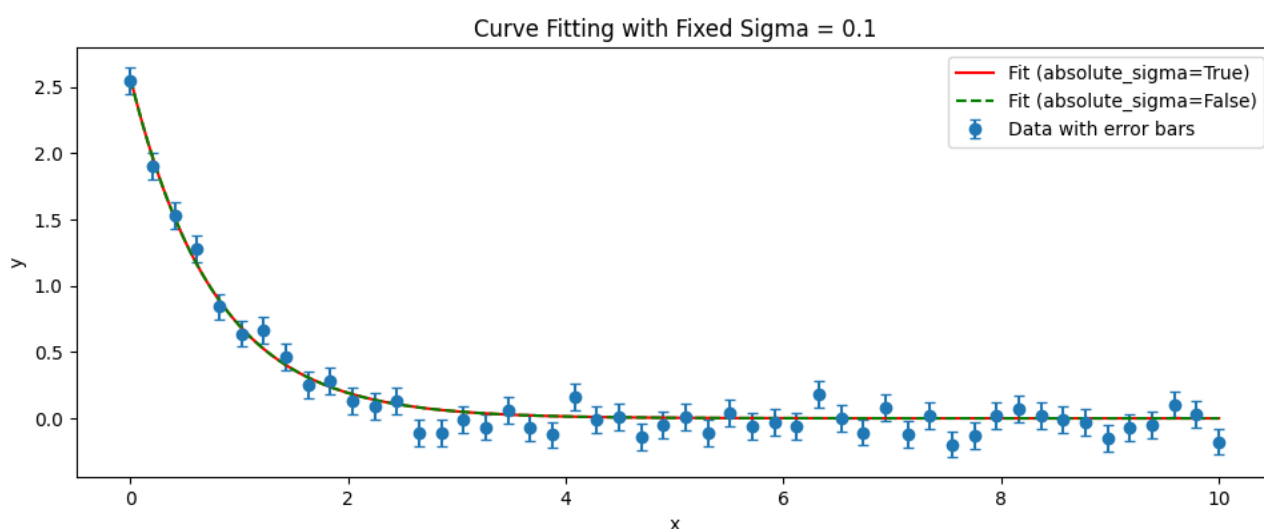


Figure 3: original data with error bars

透過Fig.3可以確認擬合出的曲線是否貼近資料點。

同時也可看出，`absolute_sigma=True`與`absolute_sigma=False`的曲線高度重疊，這是因為 `absolute_sigma` 並不會影響擬合曲線，而只會影響參數的誤差估計。

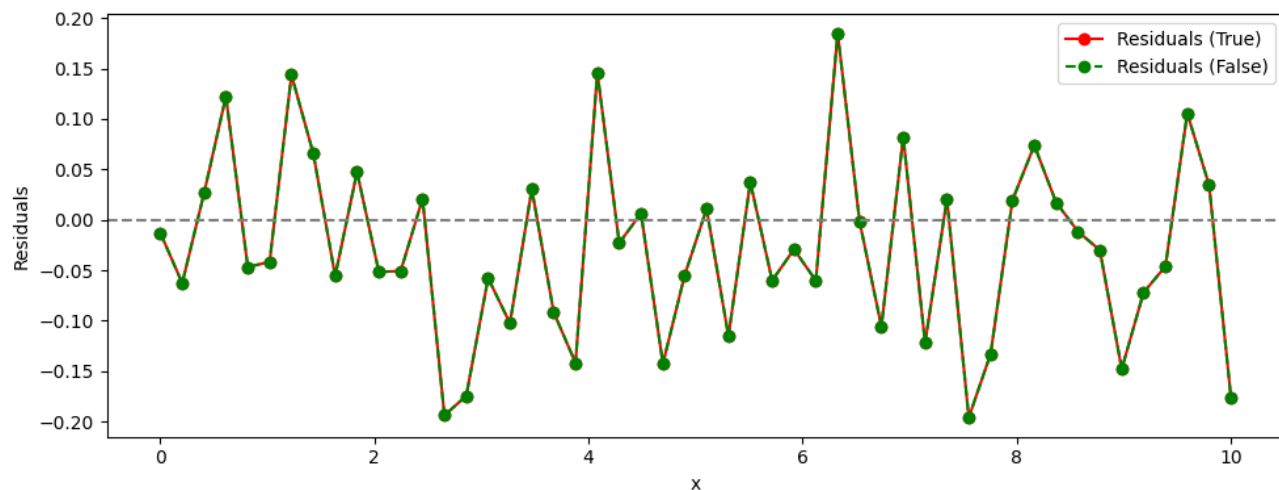


Figure 4: residuals for each of the two cases separately

殘差圖可以比較這兩種擬合之下資料與模型之間的偏差。可以看出二者的殘差分布皆沒有明顯趨勢，且分布於0附近，顯示出模型與資料吻合程度相當高，可以應證`absolute_sigma`只會影響參數的誤差估計而不影響擬合結果與殘差本身。

```
=== absolute_sigma=True ===  
A = 2.5626 ± 0.0808  
k = 1.2977 ± 0.0657  
  
=== absolute_sigma=False ===  
A = 2.5626 ± 0.0776  
k = 1.2977 ± 0.0631
```

Figure 5: 最佳擬合參數

輸出：

absolute_sigma=True

- $A = 2.5626 \pm 0.0808$
- $k = 1.2977 \pm 0.0657$

absolute_sigma=False

- $A = 2.5626 \pm 0.0776$
- $k = 1.2977 \pm 0.0631$

初步分析：

3 實驗誤差分析

本實驗所擬合的資料為理論模型 $y = A * \exp(-k * x)$ 加上隨機誤差後的模擬資料。我們採用高斯分佈誤差（平均為0，標準差為0.1），模擬測量過程中的隨機誤差。由於所有資料點的誤差皆相同（ $\sigma=0.1$ ），因此在擬合過程中，我們能夠合理使用固定誤差來估計參數不確定性。

模型形式與產生資料的理論模型完全相同，表示無模型誤差（model error），因此我們主要關注的是擬合演算法因為「資料點隨機誤差」所產生的擬合誤差（fitting error），這可從參數的協方差矩陣估計出其不確定性。

4 具體說明嘗試除錯的方法

1. 確認隨機誤差是否符合期望分布：

- 使用 `np.random.seed(42)` 確保每次模擬結果一致。
- 使用 `np.random.normal(0, 0.1, x.size)` 人為加入標準差為0.1的誤差。

2. 檢查誤差陣列 `sigma` 是否為常數：

- 使用 `sigma = 0.1 * np.ones_like(y)` 產生與 `y` 同長度的誤差陣列。
- 確保 `curve_fit` 中 `sigma` 的傳入方式為正確格式（非 `None`）。

3. 驗證擬合模型正確性：

- 使用與資料產生相同的數學模型 $A * \exp(-k * x)$ ，避免模型不一致造成系統性偏差。
- 比較 `absolute_sigma=True` 和 `False` 擬合結果，確認兩者擬合曲線相同，但參數不確定度不同。

4. 測試 `reduced chi-square` 的合理性：

- 加入程式碼手動計算 $\chi^2_\nu = \chi^2 / (N - p)$ ，並觀察 `absolute_sigma=False` 時是否將協方差乘上 χ^2_ν 。

5 具體說明實驗遇到的問題，或分析可能的問題

5.1 `absolute_sigma=True` or `False`?

根據擬合的結果，我們發覺 `absolute_sigma=True` 和 `False` 的輸出結果相同，僅差別在不確定度，且差距恰巧為 $\sqrt{\chi^2_\nu}$ ，推測可能的原因：

- 兩種方式的 `covariance matrices` 算法相差了 χ^2_ν ，使得不確定度差了 $\sqrt{\chi^2_\nu}$
- `curve_fit` 的原先設計讓結果 `self-calibrate` `overfitting` 或 `underfitting` 的情況，並給出相較於輸入的不確定度更實際的不確定度。

詳細的分析將會在結報中闡述。