實驗物理學（二）
實驗報告

Fundamental Python
Chi-square fitting - 1

Group 2

洪　瑜 B125090009
黃巧涵 B122030003
洪懌平 B102030019

2025/04/29

摘要

這次實驗探討了least-square fitting與chi-square fitting在數據具有非均勻不確定度時的差異，以及討論自由度對χ^2分布趨勢的影響。其中Practice 1調整了資料集內不同區段的誤差大小，模擬非均勻不確定度的情況。根據擬合結果可以看出，當數據具有不均勻的誤差， chi-square fitting能夠透過調整權重產生更準確的結果，並且chi-square fitting的reduced χ^2（≈0.061）比least square fitting的reduced χ^2（≈0.056）更接近理論值1，代表chi-square fitting能更準確地描述資料分布之特性。透過Practice 2的隨機數據分析可以看出，自由度增加會使卡方分布趨於對稱，並且平均值會接近自由度數值，與理論相符合。本次實驗能對數據擬合方法選擇有更深刻的認識，也進一步了解χ^2值與自由度之間的關聯，有助於日後處理、分析實驗數據。

1 前言

- Least-square fitting and Chi-square fitting:
  Least-square fitting爲最小化殘差平方和S

$$S = \sum_{i=1}^{N} (y_i - F(x_i)) \tag{1}$$

  where y_i is the data of the experiments, $F(x_i)$ is the model with given parameters x_i, and N is the data size.
  而Chi-square fitting引入個別數據點的不確定度，爲最小化加權平方和

$$\chi^2 = \sum_{i}^{N} \left(\frac{y_i - F(x_i)}{\sigma_i} \right)^2 \tag{2}$$

  where y_i is the data of the experiments, $F(x_i)$ is the model with given parameters x_i, σ_i is the uncertainty of the experiment, and N is the data size.

- Reduced χ^2 (χ^2_ν):
  主要用於評估擬合的合理性

$$\chi^2_\nu = \frac{\chi^2}{\nu} \tag{3}$$

  其中$\nu = N - p$，N爲資料點數，p爲擬合參數數量

- Chi-square distribution and probability density function (PDF):
  Chi-square distribution能用來描述獨立標準常態分布變數平方和的分布情形。在資料擬合中，Chi-Square值用來衡量模型與觀測數據的吻合程度 Probability Density Function (PDF) 是描述連續隨機變數在某個數值附近出現的機率密度有多大的函數。透過比較實做出chi-square 值之直方圖與理論的PDF值，我們可以確認實際數值是否符合預期。
  For a chi-square distribution with k degrees of freedom, the PDF is

$$f(x; k) = \frac{1}{2^k \Gamma(k/2)} x^{k/2-1} e^{-x/2}, \; x > 0 \tag{4}$$

  where Γ is the gamma function. When $k \to \infty$, the PDF approaches a normal distribution.

2 實驗步驟

2.1 Practice 1

1. Generate 20 sets of data points defined by a function $y = ax + b +$ noise with `x = np.linspace(1, 10, 10)`.

```python
import numpy as np
from scipy.optimize import curve_fit
import matplotlib.pyplot as plt

np.random.seed(42)
```

✓ 0.6s Python

Figure 1: Import packages and fix the random seed to 42

```python
n_sample    = 10
n_set       = 20

x = np.linspace(1, 10, n_sample, endpoint=True)
x_set = np.tile(x[np.newaxis, :], (n_set, 1))
```

✓ 0.0s Python

Figure 2: Initialize x

2. The noise is sampled from a normal distribution with mean = 0 and standard deviation $= \sigma_0 = 3$.

```python
def linear(x, a, b):
    return a * x + b
```

✓ 0.0s Python

```python
m = 3
k = 1
mean    = 0
sigma_0 = 3 * np.ones((n_set, n_sample))

noise       = np.random.normal(mean, sigma_0, x_set.shape)
y_true      = linear(x_set, m, k)
y_original  = y_true + noise
```

✓ 0.0s Python

Figure 3: Initialize original y

3. Change the σ_0 of y at $x = 4, 5, 6, 7$ to $\sigma_0 = 10$.

```
sigma_0_adjust = sigma_0.copy()
sigma_0_adjust[:, 3:7] = 10

y_adjust = linear(x_set, m, k) + np.random.normal(mean, sigma_0_adjust, x_set.shape)
```
✓ 0.0s Python

Figure 4: Adjust the σ_0 of y at $x = 4, 5, 6, 7$

4. You might see that the deviation of the data points at these x values is larger than the others.

5. Get the *mean* and *standard deviation* of the data points (mean value of y might be close to $a * x + b$).

```
mean_y_adjust = np.mean(y_adjust, axis=0)
sigma_y_adjust = np.std(y_adjust, axis=0)
```

Figure 5: Get the *mean* and *standard deviation* of the data points

6. Fit these mean values of y with the function $y = a * x + b$ using `curve_fit`.

7. Compare the difference between the least-square fitting method and the chi-square fitting method by calculating the χ^2 value.

```
parms_chisq, cov_chisq = curve_fit(linear, x_set.flatten(), y_adjust.flatten(),
                                   sigma=sigma_0_adjust.flatten(), absolute_sigma=True)
parms_uncertainty_chisq = np.sqrt(np.diag(cov_chisq))
parms_ls, cov_ls = curve_fit(linear, x_set.flatten(), y_adjust.flatten())
parms_uncertainty_ls = np.sqrt(np.diag(cov_ls))
```

Figure 6: Fit these mean values of y with the linear function and

2.2 Practice 2

1. Generate a 1-d array with 500 elements, each element is sampled from a normal distribution with mean = 0 and standard deviation = 1.

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import chi2

# 固定隨機數種子
np.random.seed(0)
```

Figure 7: Import packages and fix the random seed to 42

2. Calculate the chi-square value of each data point: chisq = $\frac{(\text{data\_point}-\mu)^2}{\sigma}$ You'll find out $\sigma \approx 1, \mu \approx 0$. (Note that `len(chisqr)` = 500 in this case. You might feel confused about the definition of chisqr here, but just accept it for now. You can think about what it probably means in the fitting process you've done before.)

```python
data = np.random.normal(0, 1, 500)

#計算每個點的卡方值(自由度=1)
chisqr = (data - 0)**2 / 1  # μ=0, σ=1
```

Figure 8: Initialize 500 elements and calculate the chi-square value.

3. Plot the histogram of these chi-square values.
4. Compare the histogram with the chi-square distribution with degree of freedom = 1.

```python
#畫直方圖
plt.figure(figsize=(10, 6))
plt.hist(chisqr, bins=50, density=True, alpha=0.6, label='Simulated Data')

#畫理論自由度1的卡方分布
x_vals = np.linspace(0, 10, 500)
plt.plot(x_vals, chi2.pdf(x_vals, df=1), label='Chi-square Distribution (dof=1)', color='red')

plt.xlim(left=0)
plt.xlabel('Chi-square value')
plt.ylabel('Probability Density')
plt.title('Chi-square Distribution (dof=1)')
plt.legend()
plt.savefig('./fig/chi_square_dof1.pdf', transparent=True)
plt.close('all')
```

Figure 9: Compare the histogram with the chi-square distribution with degree of freedom = 1.

5. Create several (2 to 10) 1-d arrays similar to the ones above, each array containing 500 elements sampled from a normal distribution with mean = 0 and standard deviation = 1. Then, calculate the chi-square value of each data point in each array and sum them up to get a new 1-d array that also contains 500 elements representing the sum of chi-square values. (e.g., array1 = $[1, 2, 3, \ldots, (500^{\text{th}} \text{ value})]$, array2 = $[4, 5, 6, \ldots, (500^{\text{th}} \text{ value})]$, sum_array = $\left(\frac{\text{array1}}{\sigma_1}\right)^2 + \left(\frac{\text{array2}}{\sigma_2}\right)^2$, `len(sum_array) = 500`)

6. Plot the histogram of these chi-square values (`sum_array`).

7. Compare the histogram with the chi-square distribution with degree of freedom equal to the number of 1-d arrays (2∼10) you generated.

```python
# 設定自由度範圍
dofs = range(2, 11)
# 建立subplots 3*3
fig, axes = plt.subplots(3, 3, figsize=(10, 9), sharex=True, sharey=True)
# 調整子圖間距
plt.subplots_adjust(left=0.03, right=0.97, top=0.95, bottom=0.05, wspace=0.0, hspace=0.0)
x_vals = np.linspace(0, 30, 1000)
# 遍歷每個自由度與子圖
for idx, dof in enumerate(dofs):
    row = idx // 3
    col = idx % 3
    ax = axes[row, col]

    # 模擬卡方資料
    sum_chisqr = np.zeros(500)
    for _ in range(dof):
        data = np.random.normal(0, 1, 500)
        sum_chisqr += data**2

    # 畫直方圖
    ax.hist(sum_chisqr, bins=50, density=True, alpha=0.6, label='Simulated Data')
    # 畫理論曲線
    ax.plot(x_vals, chi2.pdf(x_vals, df=dof), color='red', label='Chi-square PDF')

    ax.set_xlim(0, 30)
    ax.set_title(f'dof = {dof}')
    if dof == 4:
        ax.legend()
    if dof == 8:
        ax.set_xlabel('Chi-square value')
        ax.set_ylabel('Density')

plt.suptitle('Chi-square Distributions (dof = 2 to 10)', fontsize=16)
plt.tight_layout(rect=[0, 0, 1, 0.96])  # 調整 layout 以容納 suptitle
# plt.show()
plt.savefig('./fig/output_2_1.pdf', transparent=True)
```

Figure 10: Compare the histogram with the chi-square distribution with different degrees of freedom.

3 實驗數據與分析

3.1 Practice 1

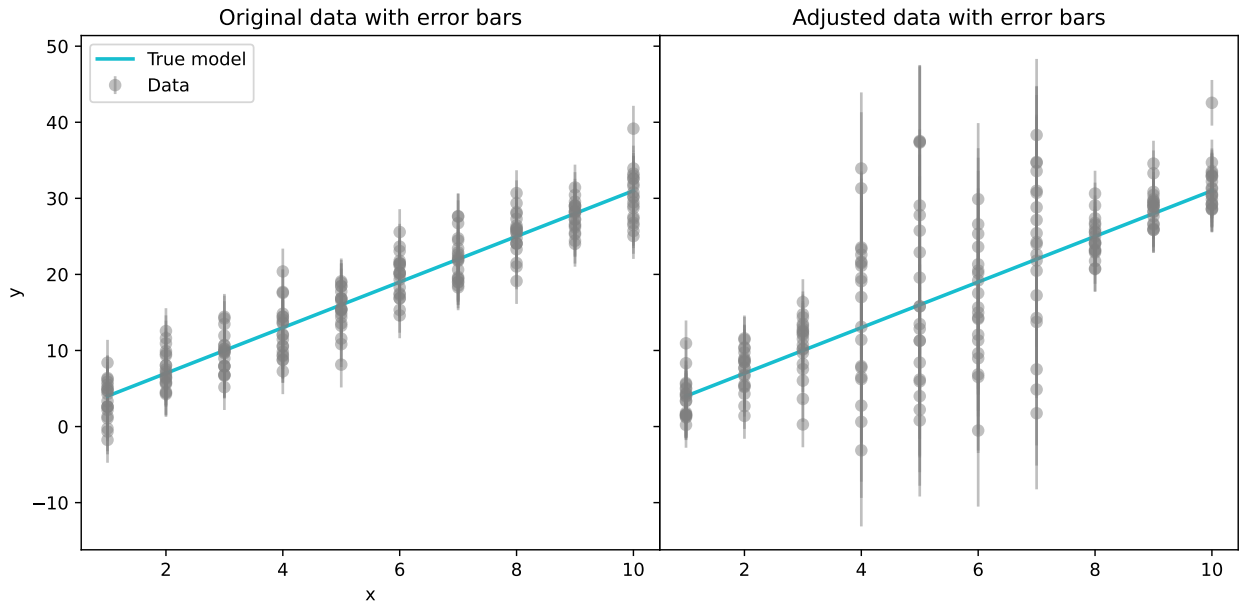

Figure 11: Comparison between the original and σ_0-adjusted y data points.(*left*: Original data with $\sigma_0 = 3$; *right*: The same with the left panel except for $x = 4-7$, where the σ_0 are changed to 10.)

The 20 sets of data points defined by a function $y = mx+k+noise$ with `x = linespace(1, 10, 10)` are plotted altogether in the Fig.11. As Fig. 11 shows, the error bars of $x = 4-7$ on the *right panel* are significantly enhanced due to the manually changed noise σ_0 from 3 to 10. This result looks ideal. Additionally, Table. 1 lists the mean and standard deviation of the synthetic datasets, denoted as μ_{data} and σ_{data}, respectively, compared to the actual model $y = mx+k$ and the given gaussian noise σ_0. The discrepancy between the mean and the model, Δ, and the error percentage between σ_0 and σ_{data} are also combined in the Table. 1.

In the Table. 1, the discrepancy Δ between the model $y = mx+k$ and the μ_{data} are generally larger if $x = 4 \sim 7$ (bold text in the table), where the σ_0 is set to be 10. This may be due to the small size of the data set, recalling the law of large numbers (LNN). The sample size of each x is merely 20, in which some of the largely biased data points can still influence the μ_{data}. This reason can also be applied to the cause of the arbitrary and random large error between σ_0 and σ_{data} at some x (e.g., $x = 3, 6, 9$).

Table 1: The μ and σ of the data compared to the ideal y and given σ_0

x	$y = mx + k$	μ_{data}	Δ	σ_0	σ_{data}	error(%)
1	4	3.731	−0.269	3	2.572	14.27
2	7	7.510	+0.510	3	2.706	9.80
3	10	10.622	+0.622	3	3.810	27.00
4	**13**	**15.388**	**+2.388**	**10**	**9.850**	**1.50**
5	**16**	**17.271**	**+1.271**	**10**	**11.601**	**16.01**
6	**19**	**15.907**	**−3.093**	**10**	**7.449**	**25.51**
7	**22**	**22.852**	**+0.852**	**10**	**10.067**	**0.67**
8	25	24.728	−0.272	3	2.488	17.07
9	28	28.933	+0.933	3	2.232	25.60
10	31	31.625	+0.625	3	3.110	3.67

Fig. 12 shows the least-square and chi-square fitting results using `curve_fit`, including the best-fit curve (dashed line) and the uncertainties (shaded area) of each fitting method. The detailed values of the best-fit parameters are listed in the Table. 1. (chi-square and least-square fitting are called χ^2 and LS fitting, hereafter)

As shown in the Table. 1, the best-fit parameters of the two fitting methods, χ^2 and LS fitting, m and k, are near the ones of the true model ($m = 1$ and $k = 1$). The best-fit curves also share the same results in Fig. 12 as the χ^2 fitting (purple dashed line) and LS fitting (orange dashed line) overlap the true model (cyan solid line). The χ^2 values and the reduced χ^2 in the Table. 1 demonstrates that there is not much deviation between the two fitting methods regarding the values of parameters.

However, there is a significant dissimilarity in the uncertainties of the two parameters Δm and Δk between the two fitting methods, χ^2 and LS fitting. The uncertainties of the χ^2 fitting are roughly 2 times smaller than those of the LS fitting. This is rooted in the fundamental difference between the two methods: the χ^2 fitting weights the sum of the least-squares with the uncertainties of each experiment. In our case, the enhanced uncertainties at $x = 4 \sim 7$ decrease those terms' significance and lessen the covariance matrix's diagonal terms when performing the χ^2 method. At the same time, the LS fitting doesn't consider the experimental uncertainties.

Table 2: Best-fit parameters of the chi-square and least-square fitting

Best-fit	χ^2 fitting	LS fitting	True model
m	3.037	2.985	3
Δm	0.076	0.167	–
k	1.156	1.438	1
Δk	0.495	1.037	–
χ^2	0.488	0.447	–
χ^2_ν	0.061	0.056	–

Despite the uncertainty difference, the shaded area (uncertainties) of the two fitting methods in Fig. 12 both involve the true model, reassuring that in our data set, there is not a notable contrast between the χ^2 and LS fitting.

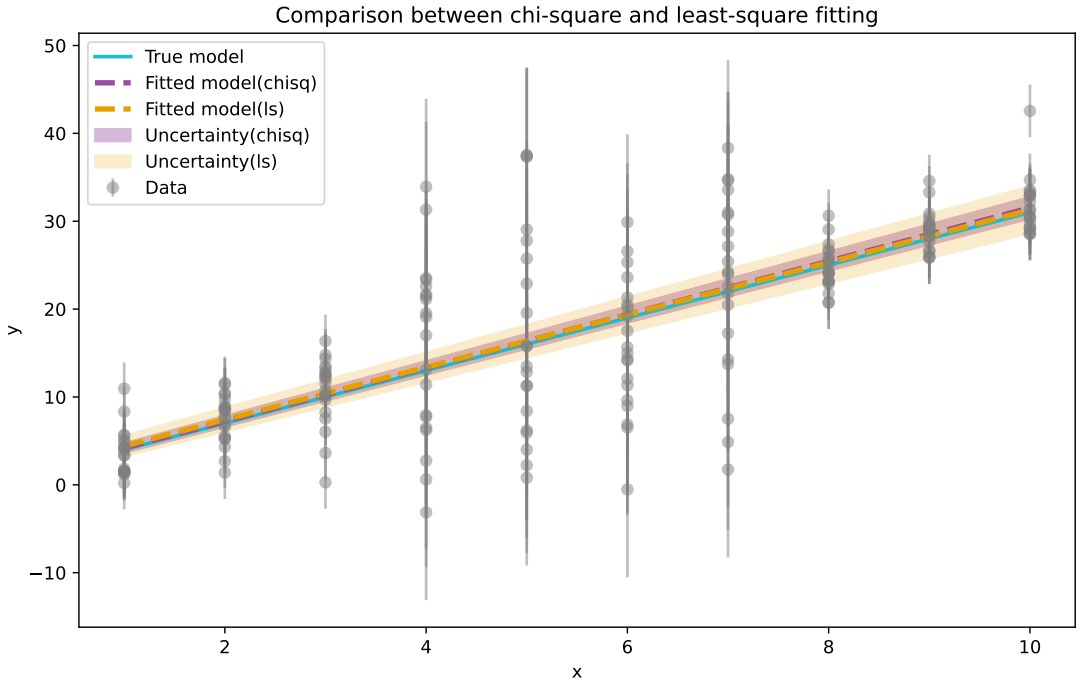

Figure 12: Comparison between least-square fitting and chi-square fitting.

3.2 Practice 2

The chi-square distribution (χ^2-distribution) is a special case of the gamma distribution, widely used in statistics, especially in hypothesis testing and confidence interval estimation for variance. It describes the distribution of the sum of the squares of k independent standard normal random variables.

There are some properties about the probability density function (PDF) of the chi-square distribution (Eq. 4):

- The mean μ is equal to the degrees of freedom (dof; k)
- For small k, it is highly right-skewed.
- As $k \to \infty$, it approaches a normal distribution via the central limit theorem.

In our practice, we generate a 1-d array with 500 elements, which are sampled from a normal distribution with mean = 0 and standard deviation = 1, calculate the chi-square value of each data point, and plot this χ^2 value to histograms. Moreover, we overlap the ideal PDF of the chi-square distribution onto those histograms to compare the two.

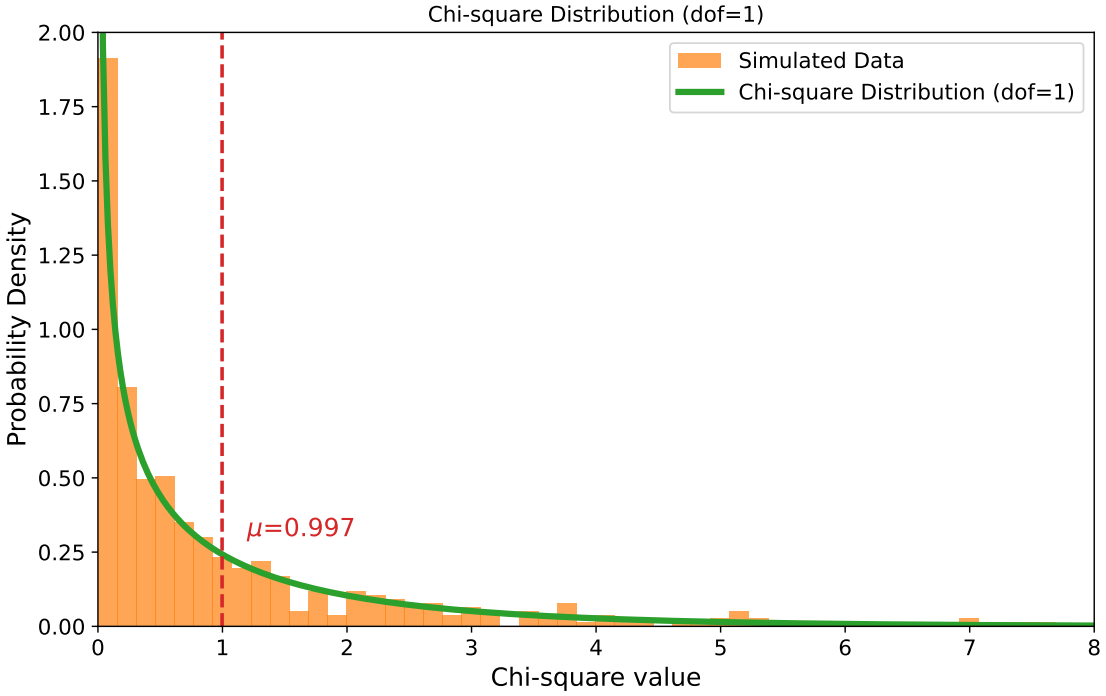

Figure 13: Histogram of the calculated chi-square values of the 500 elements with comparison of chi-square distribution with the degrees of freedom = 1

In Fig. 13 and 14, the chi-square distributions of the generated arrays with dof from 1 to 10 are showcased. When dof = 1 and 2, the χ^2 distribution and the PDF are highly left-skewed, while as the dof are larger, the distributions become less skewed and closer to a normal distribution, matching the abovementioned properties.

Moreover, the means μ of the chi-square distributions are in red on Fig. 13 and 14. As a result, the means are around the same as the degrees of freedom. This also goes with the characteristic of the PDF of the chi-square distribution.

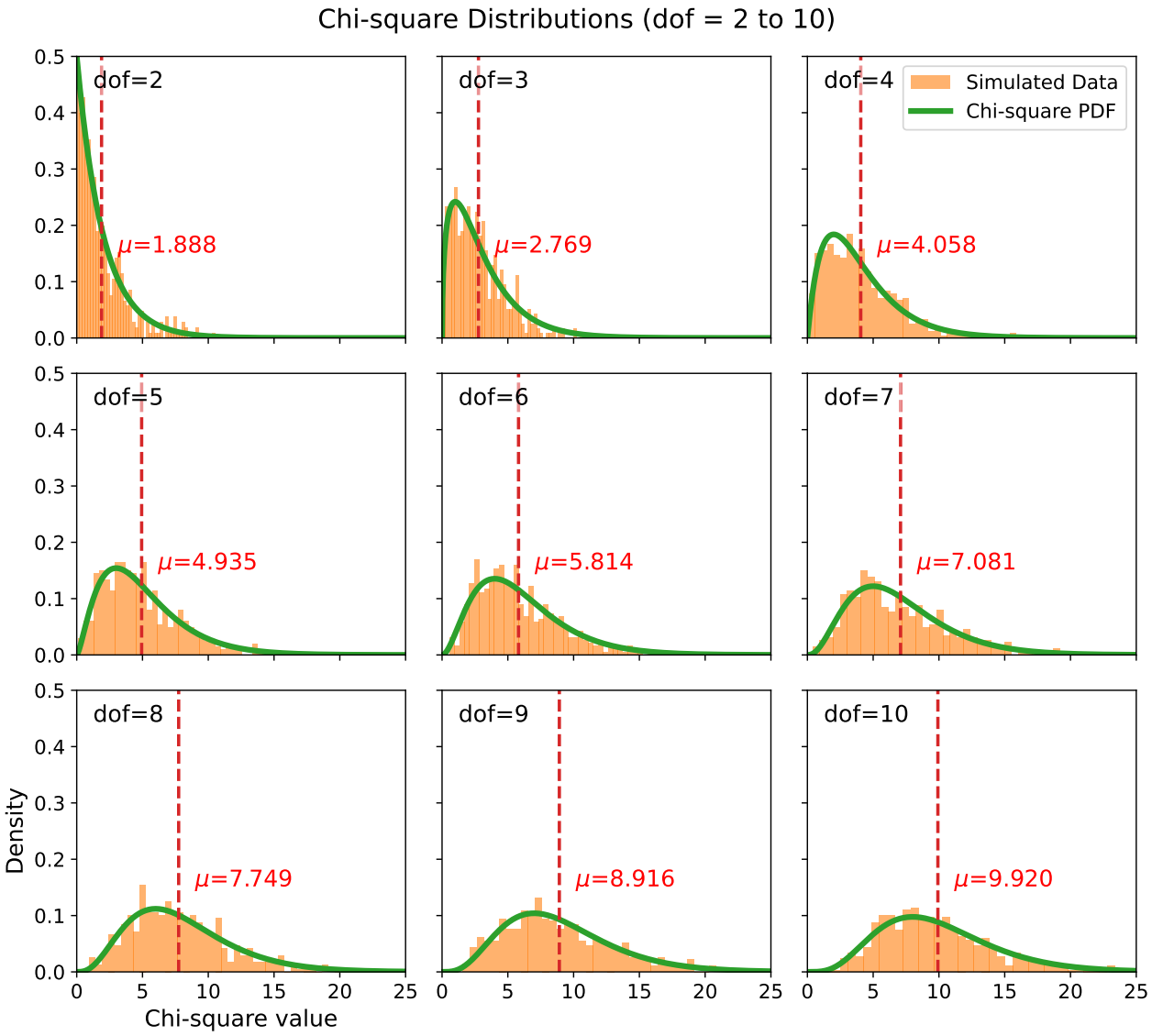

Figure 14: Histograms of the calculated chi-square values of the 500 elements with comparison of chi-square distribution with the degree of freedom from 2 to 10 (from top to bottom and left to right).

4 問題討論

4.1 Practice 1

4.1.1 Which fitting method is better in this case? State the reason.

從先前的練習可以知道透過比較χ^2_ν可判斷擬合結果的好壞，依據如下：

- $\chi^2_\nu \approx 1$：符合預期
- $\chi^2_\nu \ll 1$：過度擬合
- $\chi^2_\nu \gg 1$：欠擬合

Table.2中，可見$\chi^2_{LS} = 0.488$，而$\chi^2_{chi} = 0.447$；又本次實驗共生成10個數據，其自由度爲：

$$\nu = 10 - 2 = 8$$

可得：

- $LS\ fitting$：$\chi^2_\nu = \frac{0.488}{8} = 0.061$
- $chi\ fitting$： $\chi^2_\nu = \frac{0.447}{8} \approx 0.056$

觀察結果可見，$LS\ fitting$之χ^2_ν較接近1，理論上爲擬合效果較好的方式，這並不符合我們的預期結果，我們所期望的應是由有加入權重的$chi\ fitting$擬合程度較佳；造成此結果我們認爲並非$LS\ fitting$眞的較爲精準，而是由於數據、參數等等皆由電腦所生成，不是眞正的實驗數據，導致擬合結果「過度準確」；可由兩者的χ^2_ν皆遠小於1來佐證。

我們亦可由 Fig.12觀察到$LS\ fitting$所估的參數不確定性明顯大於$chi\ fitting$，表示在考慮誤差大小的情況下，$chi\ fitting$能提供更穩定的參數估計，進一步說明其爲更精準的擬合方法。

4.2 Practice 2

4.2.1 What do you find?

藉由Fig.14可見，在$\nu > 2$後，當自由度增加時，χ^2的機率密度函數之峰值會向右移，且分佈形狀逐漸平滑；而在Fig.13和14觀察出當自由度等於1、2時，χ^2之PDF在$x \to 0$時發散，整體趨勢類似爲exponential decay；所以透過自由度低較不接近常態分佈的觀察可推測出：當自由度較少時，其χ^2_ν數值的變動會比較敏感。
$(Hint: \nu =$ 數據量 $-$ 擬合所使用的參數$)$

4.2.2 Considering your previous work on chi-square fitting, describe the relation between those μ, σ, elements, and the fitting process.

（*Hint* : μ爲平均數、σ爲標準差）

χ^2計算公式如下：

$$\chi^2 = \sum_i^N (\frac{(y_i - F(x_i))}{\sigma_i})^2$$

其中，y_i爲實驗數據(在本實驗由電腦生成)，$F(x_i)$爲模型擬合之數據，σ_i爲每個數據的不確定性。

由於我們的數據點是從常態分佈產生的，則有以下性質：

- μ和σ個別描述資料的中心與其分散程度
- 計算χ^2時，μ可等效爲期望值，σ則表示成數據誤差的權重
- normalization的公式爲：

$$x_{normalized} = \frac{x - \mu}{\sigma}$$

  意義爲衡量該數據偏離常態的程度

接著根據*practice* 1的結果，實際在進行曲線擬合時，可發現若資料點之σ不同，*chi fitting*將進行加權擬合；將誤差大的數據(σ較大)透過計算使其對於χ^2值的貢獻較低，反之則反；而如果我們沒有對數據進行加權擬合的話，其結果等效爲*LS fitting*。

最後，根據*practice* 2的結果，可見χ^2和自由度(ν)存在一定關係；欲判斷擬合結果好壞，將仰賴於其比值：

$$\chi^2_\nu = \frac{\chi^2}{\nu}$$

判斷依據參考4.1.1。

5 總結

實驗中分析了least-square fitting 和 chi-square fitting在具有不同不確定度數據時的差異。Practice 1爲了模擬非均勻不確定度，調整了noise的標準差在數據集的數值，並且分別用least square fitting 和 chi-square fitting進行擬合。從結果比較可以看出，非均勻的不確定度對於擬合參數有影響。與least square fitting相比，chi-square fitting對變化的不確定度有更明顯的反應，因此能得出chi-square fitting更適用於具有不均勻不確定度的數據，此外延續前面幾次的實驗，這次也使用reduced χ^2來判斷擬合程度的好壞。Least-square fitting的reduced χ^2值0.056，Chi-square fitting的reduced χ^2值0.061，擬合結果都遠小於理論值（reduced $\chi^2$1），但卡方擬合的數值更接近1，表示其結果較符合預期。得出在error bar變化較大的時候，透過chi-square fitting能得到更準確的擬合結果。Practice 2中，實驗結果之χ^2 值與理論值接近，可推測出自由度會影響卡方分布的趨勢變化。

6 分工

- 洪瑜： 問題討論
- 黃巧涵： 摘要、前言、總結、分工
- 洪懌平： 實驗步驟、實驗數據與分析

7 Appendix

7.1 Source code

- https://github.com/hyp0515/exp_phy_ii/tree/main/apr29