DA01 Exercise 3

October 23, 2025

[1]:
```
import pandas as pd
x = [3, 6, 9, 12, 15, 18]
y = [4.2, 22.1, 51.6, 91.8, 141.6, 200.6]
df = pd.DataFrame({"time of flight (ms)":x, "displacement (pixels)":y})
```

[2]:
```
df.head()
```

[2]:

	time of flight (ms)	displacement (pixels)
0	3	4.2
1	6	22.1
2	9	51.6
3	12	91.8
4	15	141.6

[3]:
```
df["time of flight (s)"] = df["time of flight (ms)"]*1e-3
```

[4]:
```
df.head()
```

[4]:

	time of flight (ms)	displacement (pixels)	time of flight (s)
0	3	4.2	0.003
1	6	22.1	0.006
2	9	51.6	0.009
3	12	91.8	0.012
4	15	141.6	0.015

[5]:
```
import numpy as np
from scipy.optimize import curve_fit
import matplotlib.pyplot as plt
```

[6]:
```
X = df["time of flight (s)"]
Y = df["displacement (pixels)"]
def quad(x,a,b,c):
    return a*(x**2)+b*x+c
```

[7]:
```
parameters, covariant_matrix = curve_fit(quad, X, Y,␣
↪absolute_sigma=True,p0=(10,0,0))
A = parameters[0] #extract slope, it is parameter #0
B = parameters[1] #extract intercept, it is parameter #1
```

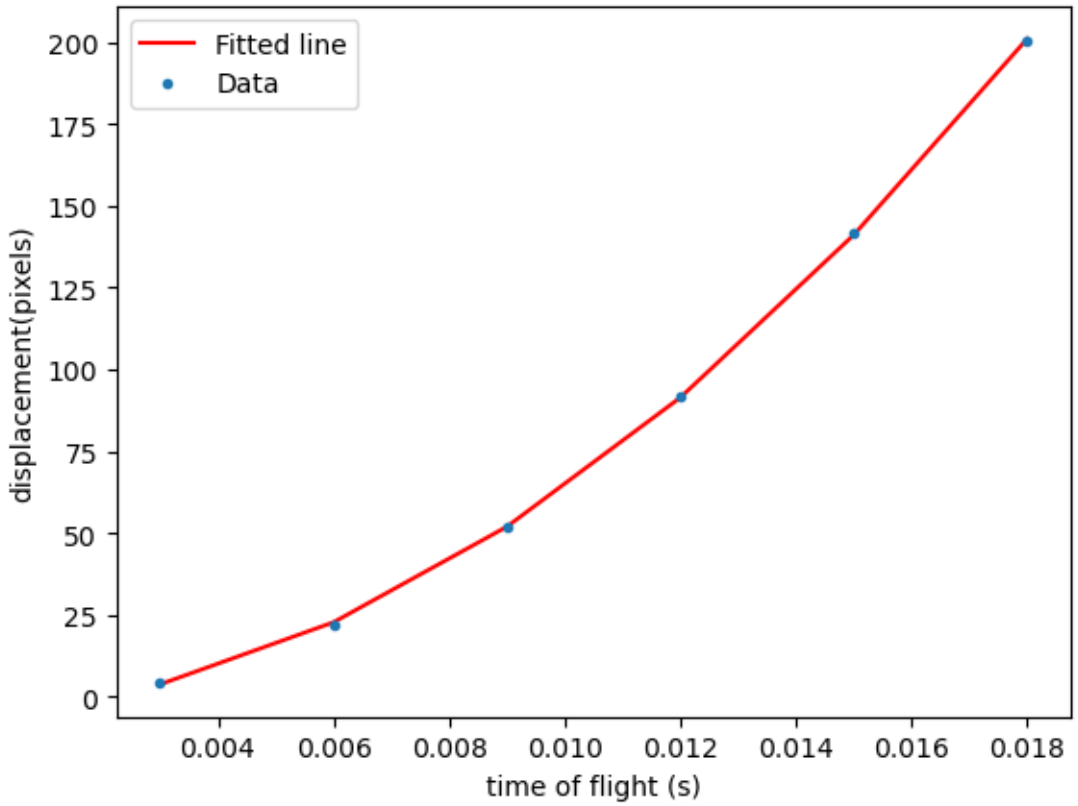
```
C = parameters[2] #extract intercept, it is parameter #2
```

[8]:
```
parameter_uncertainties = np.sqrt(covariant_matrix.diagonal())
a_error = parameter_uncertainties[0]
b_error = parameter_uncertainties[1]
c_error = parameter_uncertainties[2]

print('A: {:.2f} Error on A: {:.2f}'.format(A, a_error))
print('B: {:.2f} Error on B: {:.2f}'.format(B, b_error))
print('C: {:.2f} Error on C: {:.2f}'.format(C, c_error))
```

```
A: 568849.21 Error on A: 18184.83
B: 1203.69 Error on B: 390.11
C: -4.97 Error on C: 1.79
```

[9]:
```
plt.errorbar(X, Y, label='Data', fmt='.', capsize=5)
plt.plot(X, quad(X,A,B,C), 'r', label='Fitted line') # note: we can write any␣
 ↪formula into this plotting function
plt.xlabel('time of flight (s)')
plt.ylabel('displacement(pixels)')
plt.legend()
plt.show()
```



[ ]:

[ ]: