Submission for Statistical Methods for Experimental Physics (Part-I)Assessment 1 - Mihir Koka

1.1 Neutron Time of Flight

The energy of the neutrons has a small Gaussian spread (from doppler broadening), thus the arrivals times will also be Gaussian-distributed. But they will be distorted the by the detector's own Gaussian timing response.

So the plan is: Energy distribution -> time of flight distribution -> measured signal -> our param estimation.

(i) The saddlepoint approximation finds a Gaussian approximation of a strongly peaked function of the form exp [−f (x)] by expanding about the exponent function about its maxima:

$g(x) \approx \exp[-f(x_0)] \exp\left[-\frac{1}{2} f''(x_0)(x-x_0)^2\right]$

(a) Use this method to find an expression for the TOF distribution of deuterium fusion neutrons at the detector,
$g = \frac{dN}{dt}$ using $f = -ln\frac{dN}{dE}$

so we have the intrinsic neutron energy distribution as

$$\frac{dN}{dE} = \phi(E; E_0, \sigma_E)$$

and by subbing in velocity term into the energy equation we get this for the energy as a function of time:

$$E(t) = \frac{1}{2} m_n \left(\frac{d}{t}\right)^2 = \frac{m_n d^2}{2t^2}$$

So we want to use the saddlepoint approximation with $f = -ln\frac{dN}{dE}$ to get a Gaussian approximation for the true TOF distribution

So lets start with the definition of a Gaussian pdf

$$\phi(X; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(X-\mu)^2}{2\sigma^2}\right]$$

So using what we were given we can do this (just subbing in terms):
$\frac{dN}{dE}(E) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(E-E_0)^2}{2\sigma_E^2}\right]$

Now if we evaluate this at $E = E(t) = \frac{m_n d^2}{2t^2}$

=> $\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{(E(t)-E_0)^2}{2\sigma_E^2}\right]$

We define the function (f(t)) according to the saddlepoint prescription:

$f(t) = -\ln\left(\frac{dN}{dE}\Big|_t\right)$ Taking the negative logarithm:

$= -\ln\left(\frac{1}{\sqrt{2\pi}\,\sigma_E}\right) + \frac{(E(t)-E_0)^2}{2\sigma_E^2} = \frac{(E(t)-E_0)^2}{2\sigma_E^2} + \text{const.}$

— Find the peak (t_0) from (f'(t_0) = 0)

Differentiate:

$f'(t) = \frac{E(t)-E_0}{\sigma_E^2}\,E'(t)$

Setting $(f'(t_0) = 0)$, and because $(E'(t_0)! = 0)$ we get

$E(t_0) = E_0$

Substitute the relation $(E(t) = \frac{m_n d^2}{2t^2})$:

$E_0 = \frac{m_n d^2}{2t_0^2} \quad \Rightarrow \quad \boxed{t_0 = d\sqrt{\frac{m_n}{2E_0}}}$

— Compute the curvature (f''(t_0)) and getting the variance from it

Differentiate again:

$f''(t) = \frac{[E'(t)]^2}{\sigma_E^2} + \frac{E(t)-E_0}{\sigma_E^2}E''(t)$

At the peak $(E(t_0) = E_0)$, the second term vanishes, so

$f''(t_0) = \frac{[E'(t_0)]^2}{\sigma_E^2}$

Now, using $(E(t) = \frac{m_n d^2}{2}t^{-2})$:

$E'(t) = -\frac{m_n d^2}{t^3}, \qquad E''(t) = \frac{3m_n d^2}{t^4}$

Therefore,

$[E'(t_0)]^2 = \frac{m_n^2 d^4}{t_0^6} \quad \Rightarrow \quad f''(t_0) = \frac{m_n^2 d^4}{\sigma_E^2\, t_0^6}$

— Variance of the TOF distribution

By the saddlepoint formula, the variance of the Gaussian approximation is the reciprocal of the curvature:

$$\sigma^2_{t,\mathrm{true}} = \frac{1}{f''(t_0)} = \frac{\sigma_E^2\, t_0^6}{m_n^2 d^4}$$

Substitute $\left(t_0 = d\sqrt{\frac{m_n}{2E_0}}\right)$:

$$t_0^6 = d^6\left(\frac{m_n}{2E_0}\right)^3$$

Hence,

$$\boxed{\sigma^2_{t,\mathrm{true}} = \frac{\sigma_E^2\, d^2\, m_n}{8\, E_0^3}}$$

— Gaussian approximation for the true TOF distribution

Finally, the time-of-flight distribution $\left(g(t) = \frac{dN}{dt}\right)$ is approximately Gaussian with mean (t_0) and variance $(\sigma^2_{t,\mathrm{true}})$:

$$\boxed{g(t) \approx \phi\left(t;\ t_0 = d\sqrt{\frac{m_n}{2E_0}},\ \sigma_{t,\mathrm{true}} = \sqrt{\frac{m_n d^2}{8E_0^3}}\,\sigma_E\right)}$$

(b) The measured TOF data is a convolution of the true TOF distribution with the detector response function, $R_{detector}$. Using the result of part (a), find the Gaussian approximation to the measured TOF signal as afunction of time. Hint: The convolution of two Gaussians is another Gaussian with a variance equal to thesum of the two variances

Ok so we want to take our result from the previous part and convolve it with the detector's Gaussian response which is:

$$R_{\mathrm{detector}}(t) = \phi(t; 0, \sigma_t)$$

Using the hint as know that the convolution of this will also be a Gaussian with variance equal to the sums of the variances of the individual Gaussians.

Part (i)(b) — Measured TOF as a convolution of Gaussians

We have the TOF distribution from (i)(a):

$$g(t) \approx \phi\left(t;\ t_0 = d\sqrt{\frac{m_n}{2E_0}},\ \sigma_{t,\mathrm{true}} = \sqrt{\frac{m_n d^2}{8E_0^3}}\,\sigma_E\right).$$

The detector response is

$$R_{\text{detector}}(t) = \phi(t; 0, \sigma_t).$$

The measured signal is the convolution

$$g_{\text{meas}}(t) \; = \; (g_{\text{true}} * R_{\text{detector}})(t).$$

Since the convolution of two Gaussians is Gaussian, the result has

- mean: $t_0 + 0 = t_0$, (looked into this, not mentioned in the hint but also applies to mean, though trivial here)
- variance: $\sigma_{t,\text{true}}^2 + \sigma_t^2$.

Therefore,

$$\boxed{g_{\text{meas}}(t) \; = \; \phi\Big(t;\; t_0 = d\sqrt{\frac{m_n}{2E_0}},\; \sigma_{t,\text{meas}} = \sqrt{\sigma_{t,\text{true}}^2 + \sigma_t^2}\Big)}$$

where

$$\sigma_{t,\text{true}}^2 = \frac{\sigma_E^2\, d^2\, m_n}{8\, E_0^3} \quad \Rightarrow \quad \sigma_{t,\text{meas}}^2 = \frac{\sigma_E^2\, d^2\, m_n}{8\, E_0^3} + \sigma_t^2.$$

Part (ii)

We model the measured TOF with a Gaussian as seen above

Given data points (t_i, y_i), the residual sum of squares (RSS) is

$$\text{RSS}(\sigma_E;\, \sigma_t) \; = \; \sum_i \Big[y_i - \phi\big(t_i;\; t_0,\; \sigma_{t,\text{meas}}(\sigma_E, \sigma_t)\big)\Big]^2.$$

Normally you'd have some Amplitude, A parameter that is free to fit alongside the others, but in this case we see that the signal is normalized (noted in part iii) so it's going to integrate to 1 anyway. Therefore we can just use the RSS as seen above

parameter degeneracy

so as mentioned previously the measured TOF signal width comes from two effects.

1. The intrinsic energy spread
2. the detector timing

and in part (b) we combined them to get our distribution. in that equation we only have a sum of squares appear, not the individual terms, just their combined effect. So different combinations can give the same total and therefore the same fit with the RSS. These are our degenerate parameters.

So, σ_E and σ_t are degenerate parameters in this model because they both control the overall broadening, and their individual effects can't be seen individually using the TOF data alone.

part (iii)

Quick write up: We're estimating the neutron energy spread σ_E using TOF data at a fixed detector width of 10ns. This is done by fitting the instrinsic spread by minimizing the RSS we got in the previous step.

what the code does ?

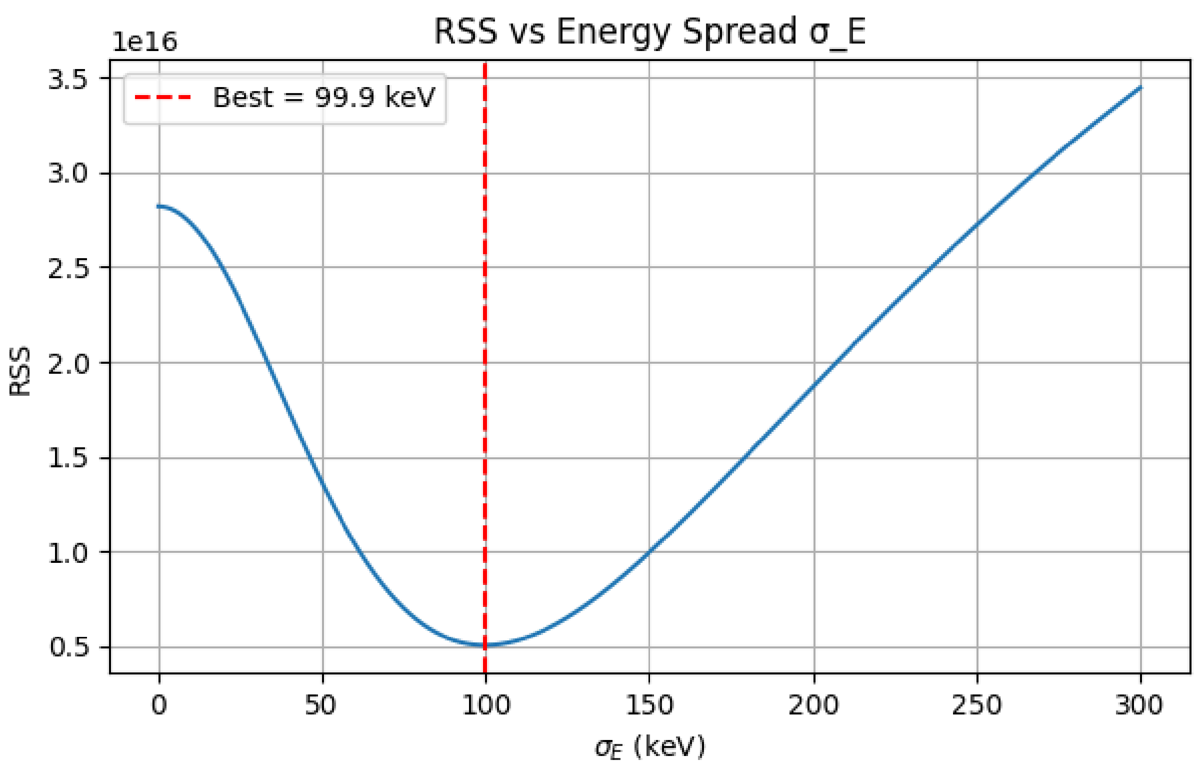
- we load the measured TOF data
- define the constants we're using
- calculate the mean TOF for 2.45Mev neutron at 15m (the dataset is for 15m)
- Loop over the possible values for σ_E, computes the Gaussian model and calculates the RSS
- Find the value for σ_E that gives us the smallest RSS
- Plots: The RSS vs σ_E to show the best fit minimum The measured data vs the Gaussian model at that best fit width we found.

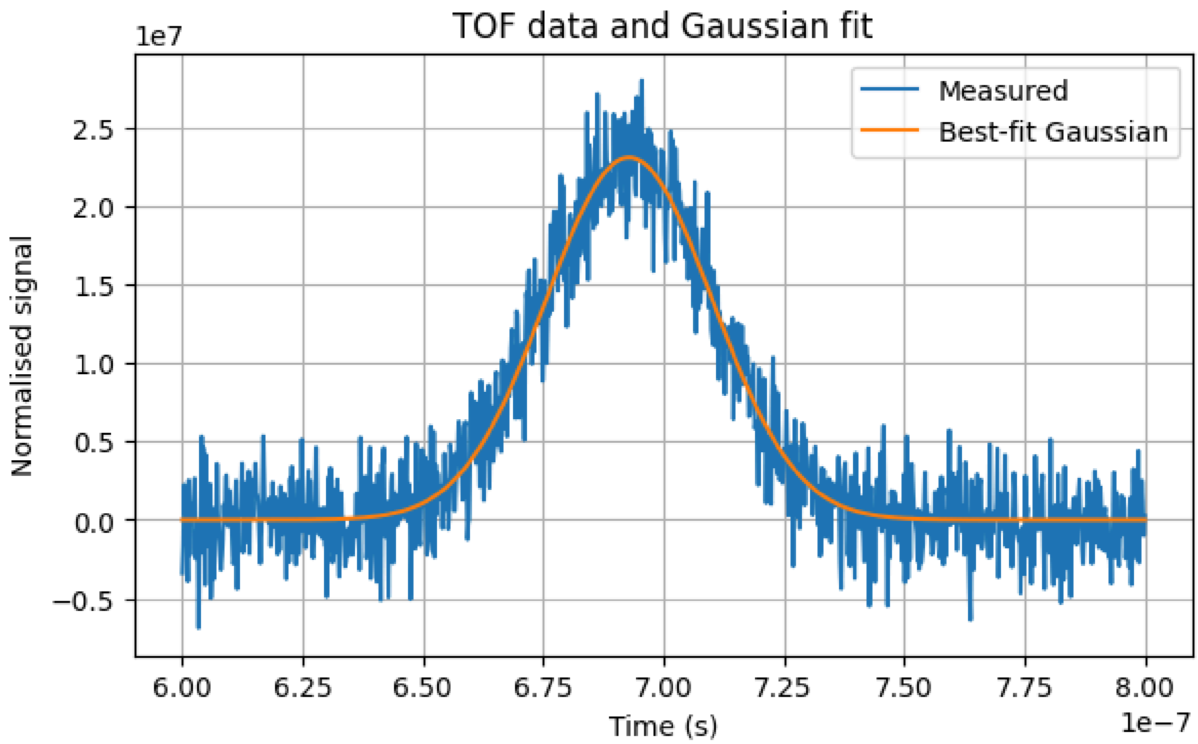
In [4]:

```
import numpy as np, pandas as pd, matplotlib.pyplot as plt

# Load the CSV (in my case the notebook and csv are in the same directory, if this
df = pd.read_csv("nTOF_data_15m_detector.csv")
t = df['# t (s)'].to_numpy()
y = df[' normalised signal'].to_numpy()

# Physical constants we're using
e  = 1.602176634e-19     # J per eV
mn = 1.67492749804e-27   # neutron mass (kg)
E0 = 2.45e6 * e           # 2.45 MeV → J
d  = 15.0                 # detector distance (m)
sigma_t = 10e-9           # 10 ns detector width

# Mean flight time
t0 = d * np.sqrt(mn / (2*E0))
print(f"t0 = {t0:.3e} s")

# Grid search for sigma_E (doing this in keV)
sigmaE_vals = np.linspace(0.1, 300, 2000)
rss = []

for sigmaE_keV in sigmaE_vals:
    sigmaE_J = sigmaE_keV * 1e3 * e
    sigma_t_true = np.sqrt((mn*d**2/(8*E0**3)) * sigmaE_J**2)
    sigma_t_meas = np.sqrt(sigma_t_true**2 + sigma_t**2)
    f = (1/(np.sqrt(2*np.pi)*sigma_t_meas)) * np.exp(-0.5*((t-t0)/sigma_t_meas)**2)
    rss.append(np.sum((y - f)**2))

rss = np.array(rss)
```

```python
best_sigmaE = sigmaE_vals[np.argmin(rss)]
print(f"Best σ_E = {best_sigmaE:.2f} keV")

# --- Plot RSS vs σ_E
plt.figure(figsize=(7,4))
plt.plot(sigmaE_vals, rss)
plt.axvline(best_sigmaE, color='r', ls='--', label=f"Best = {best_sigmaE:.1f} keV")
plt.xlabel(r"$\sigma_E$ (keV)")
plt.ylabel("RSS")
plt.title("RSS vs Energy Spread σ_E")
plt.legend(); plt.grid(); plt.show()

# --- Plot data vs best-fit Gaussian
sigmaE_J = best_sigmaE * 1e3 * e
sigma_t_true = np.sqrt((mn*d**2/(8*E0**3)) * sigmaE_J**2)
sigma_t_meas = np.sqrt(sigma_t_true**2 + sigma_t**2)
f_best = (1/(np.sqrt(2*np.pi)*sigma_t_meas))*np.exp(-0.5*((t-t0)/sigma_t_meas)**2)

plt.figure(figsize=(7,4))
plt.plot(t, y, label="Measured")
plt.plot(t, f_best, label="Best-fit Gaussian")
plt.xlabel("Time (s)"); plt.ylabel("Normalised signal")
plt.title("TOF data and Gaussian fit")
plt.legend(); plt.grid(); plt.show()
```

```
t0 = 6.928e-07 s
Best σ_E = 99.87 keV
```





In [7]:
```
# could also use scipy to minimize like this intstead, gives the same value
import numpy as np
from scipy.optimize import minimize

def rss_sigmaE(params):
    sigmaE_keV = params[0]
    if sigmaE_keV <= 0:
        return 1e30  # prevent negative vals
    sigmaE_J = sigmaE_keV * 1e3 * e
    sigma_t_true = np.sqrt((mn * d**2 / (8 * E0_J**3)) * sigmaE_J**2)
    sigma_t_meas = np.sqrt(sigma_t_true**2 + sigma_t**2)
    f_model = (1 / (np.sqrt(2*np.pi)*sigma_t_meas)) * np.exp(-0.5*((t - t0)/sigma_t
    return np.sum((y - f_model)**2)

# Run the minimization
result = minimize(rss_sigmaE, x0=[100], bounds=[(0.1, 300)])
best_sigmaE_keV = result.x[0]

print(f"Best-fit σ_E = {best_sigmaE_keV:.2f} keV")
```

```
Best-fit σ_E = 99.79 keV
```

part(iii) results summary

- The best fit is around ~100 keV or to get it in joules we get $1.6 \cdot 10^{-14} J$

so if we plug σ_E as 100keV into our equations from before we get:

$\sigma_{true} \approx 14.1ns$ and $\sigma_{measured} \approx 17.3ns$

as the instrinsic TOF width from neutron energy spread and the total measured width when including the detector respectively

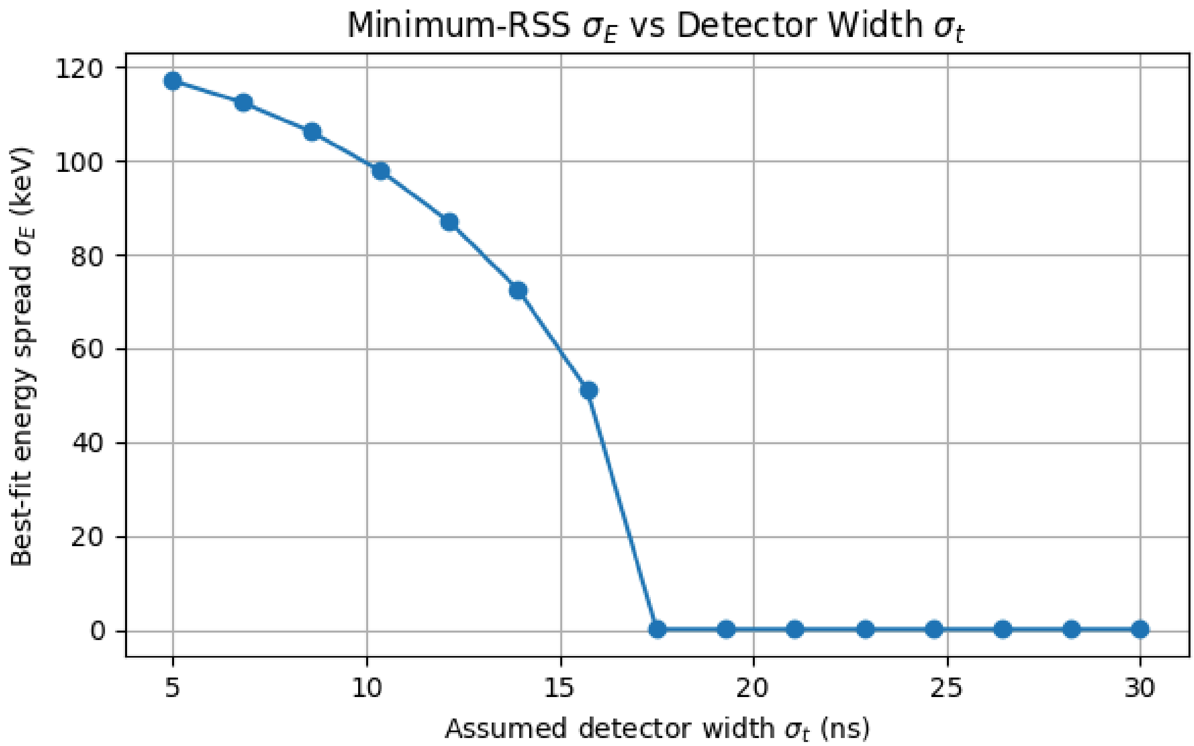
part (iv)

Plot the minimum RSS estimate of the energy spread (σE) as a function of assumed detector responsewidth (σt). What does this reveal about the model?

So what we're doing now is we take the detector width σ_t as a variable instead of fixing it like we did in part(iii).
we then:

1. Minimize the RSS over σ_E like in part (iii)
2. record the best fit value for σ_E again
3. plot the best σ_E against the width

In [9]:
```
import numpy as np
from scipy.optimize import minimize
import matplotlib.pyplot as plt

# Range of detector widths to test
sigma_t_values = np.linspace(5e-9, 30e-9, 15)   # 5 ns → 30 ns is the range we're t
best_sigmaE_list = []

for sigma_t_assumed in sigma_t_values:
    # define RSS for this sigma_t
    def rss_sigmaE(params):
        sigmaE_keV = params[0]
        if sigmaE_keV <= 0:
            return 1e30
        sigmaE_J = sigmaE_keV * 1e3 * e
        sigma_t_true = np.sqrt((mn * d**2 / (8 * E0_J**3)) * sigmaE_J**2)
        sigma_t_meas = np.sqrt(sigma_t_true**2 + sigma_t_assumed**2)
        f_model = (1 / (np.sqrt(2*np.pi)*sigma_t_meas)) * np.exp(-0.5*((t - t0)/sig
        return np.sum((y - f_model)**2)

    result = minimize(rss_sigmaE, x0=[100], bounds=[(0.1, 300)])
    best_sigmaE_list.append(result.x[0])

# Convert sigma_t to ns for plotting
sigma_t_ns = sigma_t_values * 1e9

# Plot σ_E(best) vs σ_t
plt.figure(figsize=(7,4))
plt.plot(sigma_t_ns, best_sigmaE_list, marker='o')
plt.xlabel(r"Assumed detector width $\sigma_t$ (ns)")
plt.ylabel(r"Best-fit energy spread $\sigma_E$ (keV)")
plt.title(r"Minimum-RSS $\sigma_E$ vs Detector Width $\sigma_t$")
plt.grid(True)
plt.show()
```



We can see in the plot that when we're at ~100keV the assumed width is at 10ns, as we should expect from our result in part (iii) \

As the width increases the energy decreases, this curve basically shows us the $\sigma_E - \sigma_t$ degeneracy.

What does this reveal about the model ?

so this shows us a inverse relationship between the assumed width of the detector and the energy spread. Since the model can't differentiate if the broadening of the TOF peak is coming from the intrinsic energy spread of the neutrons or the detector timing response (they're degenerate parameters).