QUESTION # 1

DEVELOPMENT OF ALGORITHM

The piece of code is executable in ROOT session at UNIX workstation. We included necessary C++ libraries and ROOT library "Trandom" and make use of "using namespace std" method to facilitate use of functions included in libraries with much more freedom.

QUESTION1() is main function of our code and we avoided inclusion of user-defined functions keeping in view the simple and natural flow of calculations of given problem. In QUESTION1(), we declared two histograms, one is supposed to contain data for uncorrelated data generation and other contains data for correlated data after rotation.

GENERATING 1000 UNCORRELATED (X,Y) PAIRS USING GAUSSIAN DISTRIBUTION:

We declared two arrays, x[1000] and y[1000] and using the built-in Gaus() function provided by ROOT along with random number generator TRandom3, we generated two Gaussian distributions, one along X-axis with $\mu=1.0$ and $\sigma=2.0$ and second along Y-axis with $\mu=2.0$ and $\sigma=0.5$ in a for-loop. We filled the histogram Gaussian_Uncorrelated with the corresponding data in same for-loop which is displayed in Fig1.1. Using GetMean() method provided in ROOT, we calculate the mean values of generated data along both axis.

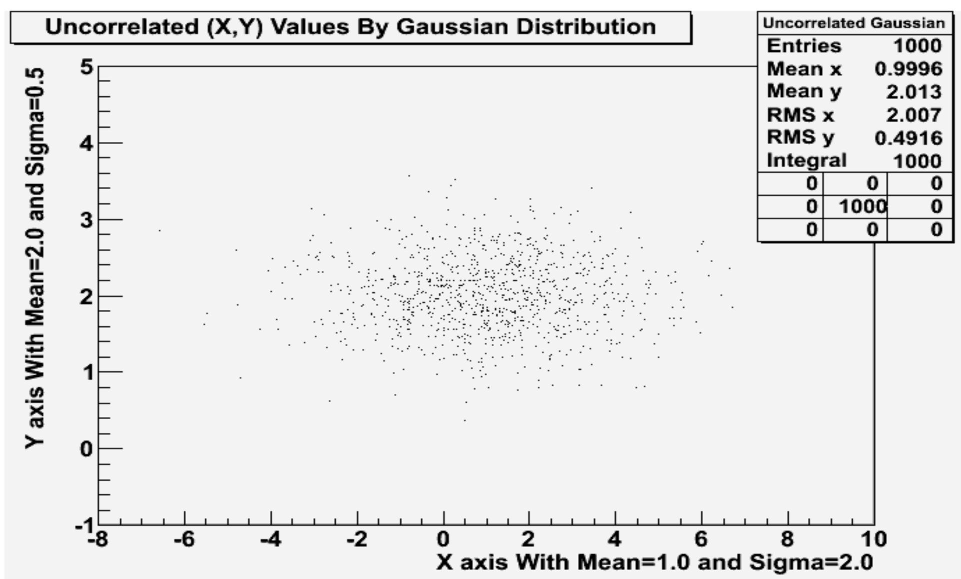

Figure 1.1: Uncorrelated (X,Y) Pair Sampling Using Gaussian Distribution. $\mu_X=1.0$, $\sigma_X=2.0$, $\mu_Y=2.0$ and $\sigma_Y=0.5$

CALCULATION OF ERROR MATRIX, ρ AND Φ:

We calculated elements of Error Matrix V in a for-loop making use of following three equations:

$$V_{XX} = \frac{1}{1000}\sum_{i=1}^{1000}\left[\left(X_i - \mu_X\right)^2\right],$$
$$V_{YY} = \frac{1}{1000}\sum_{i=1}^{1000}\left[\left(Y_i - \mu_Y\right)^2\right],$$

And

$$V_{XY} = \frac{1}{1000}\sum_{i=1}^{1000}\left[\left(X_i - \mu_X\right)\left(Y_i - \mu_Y\right)\right].$$

Calculation of ρ and Φ involved evaluation of following expressions for uncorrelated (X,Y) ordered pairs.

$$\rho = \frac{V_{XY}}{\sigma_X \sigma_Y}$$

where σ_X and σ_Y are calculated making use of GetRMS() method in ROOT. For Φ, we used following relation:

$$\Phi = \frac{1}{2}\tan^{-1}\left(\frac{2\rho\sigma_X\sigma_Y}{\sigma_X^2 - \sigma_Y^2}\right) \Rightarrow \Phi = \frac{1}{2}\tan^{-1}\left(\frac{2V_{XY}}{\sigma_X^2 - \sigma_Y^2}\right).$$

VARIANCE OF GIVEN FUNCTION USING UNCORRELATED (X,Y) PAIRS:

The function given to us is $f(X,Y) = 5X + 8Y$. We used the following equation to calculate the variance of this function:

$$\sigma_f^2 = a^2 V_{XX} + b^2 V_{YY} + 2abV_{XY} \text{ where } a = 5 \text{ and } b = 8.$$

ROTATION OF UNCORRELATED DATA BY 30° ABOUT CENTER OF DISTRIBUTION:

We observed that uncorrelated (X,Y) events are distributed on XY plane about the center $(\mu_{X\_DATA}, \mu_{Y\_DATA})$ and not about (1,2). We translated our coordinate system using following set of equations:

$$X_{Translated} = X - \mu_{X\_DATA}$$
$$Y_{Translated} = Y - \mu_{Y\_DATA}$$

This is done by using for-loop and defining x_translate[1000] and y_translate[1000] arrays. Thus, our distribution is centered about origin. We make use of rotation matrix to rotate this translated distribution about origin of the distribution (which happens to be origin of coordinate system as we have translated our system). For rotation, we defined x_rotated[1000] and y_rotated[1000] and made use of following matrix equation:

$$\begin{pmatrix} X_{Rotated} \\ Y_{Rotated} \end{pmatrix} = \begin{pmatrix} \cos 30^\circ & -\sin 30^\circ \\ \sin 30^\circ & \cos 30^\circ \end{pmatrix} \begin{pmatrix} X_{Translated} \\ Y_{Translated} \end{pmatrix}.$$

We used inverse-translation equations to shift back our rotated distribution to it centre $(\mu_{X\_DATA}, \mu_{Y\_DATA})$. The rotated data is filled in histogram and is shown in Fig1.2.

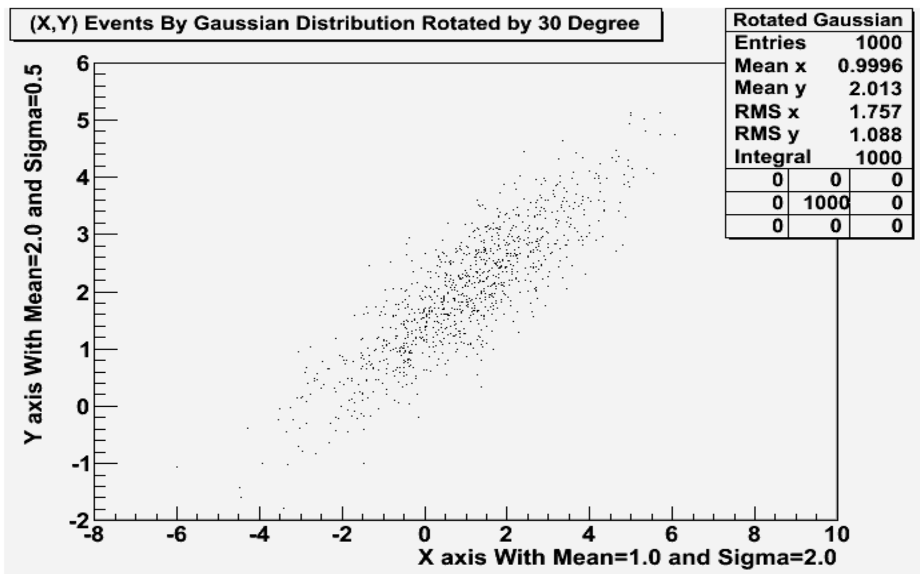

Figure 1.2: 30° Rotation Of (X,Y) Ordered Pairs Sampling About Its Center. μ_X=1.0, σ_X=2.0, μ_Y=2.0 and σ_Y=0.5

CALCULATION OF ERROR MATRIX, ρ AND Φ USING ROTATED (X,Y) PAIRS AND VARIANCE OF GIVEN FUNCTION:

The calculation of error matrix, ρ and Φ are essentially the same of rotated data as they were for uncorrelated (X,Y) values. Same formulae are used to calculate Error Matrix, ρ and Φ. The variance of given function is calculated on the same pattern except that this time, we used rotated data. Table 1.1 shows the calculated error matrix, ρ and Φ for both uncorrelated and rotated data along with variance of function in both cases.

	V_{XX}	V_{YY}	V_{XY}	ρ	Φ (in Degree)	$\sigma^2_{function}$
UNCORRELATED DATA	4.0289	0.2416	-0.0044	-0.0045	-0.0667	115.8356
CORRELATED DATA ROTATED BY 30°	3.0858	1.1846	1.6377	0.8566	29.9340	283.9764

Table 1.1: Values of Error Matrix, ρ, Φ and variance of given function for uncorrelated and correlated cases.

QUESTION # 2

DEVELOPMENT OF ALGORITHM:

The code presents an experimental setup to measure the lifetime of unstable nucleus. It is executable in ROOT session at UNIX workstation. QUESTION2() is the main function of the code besides Solve_For_t(), pdf_fit() and LOG_Likelihood_Function() are added to facilitate development of the algorithm in more versatile fashion besides making it easy for the user to understand working of code in bits and parts.

GENERATION OF 200 EVENTS USING INVERSION TECHNIQUE AND GAUSSIAN SMEARING:

The expected pdf is a convolution of exponential decay and Gaussian resolution. The evaluated convolution integral was provided to us.

$$f(t;\tau,\sigma_t) = \frac{1}{2\tau}\exp\left(\frac{\sigma_t^2}{2\tau^2} - \frac{t}{\tau}\right) erfc\left(\frac{\sigma_t}{\sqrt{2}\tau} - \frac{t}{\sqrt{2}\sigma_t}\right)$$

We note that this function cannot be analytically integrated and therefore, we adopt sample importance method to generate 200 events. We declare an approximate function as followed:

$$f^{approximate} = \frac{1}{2\tau}\exp\left(\frac{\sigma_t^2}{2\tau^2} - \frac{t}{\tau}\right) \Rightarrow f = w \times f^{approximate}$$

And we can now adopt Inversion Technique on approximate function. We note that it is a decaying exponential function and therefore select [0.0sec, 100sec] as our domain, realizing that approximate function falls to zero beyond 100sec. Mathematics of Inversion Technique described in following few steps.

$$\lambda = \frac{\int_{-5}^{t} \frac{1}{2\tau}\exp\left(\frac{\sigma_t^2}{2\tau^2} - \frac{t}{\tau}\right) dt}{\int_{-5}^{100} \frac{1}{2\tau}\exp\left(\frac{\sigma_t^2}{2\tau^2} - \frac{t}{\tau}\right) dt}$$

Where λ is a random number generated using TRandom3 RNG. Solving for *t* yields:

$$t = -\tau \ln\left[\lambda\left\{\exp\left(\frac{-100}{\tau}\right) - \exp\left(\frac{5}{\tau}\right)\right\} + \exp\left(\frac{5}{\tau}\right)\right]$$

A user-defined function Solve_For_t() included in the program solves above equation when this function is called in main program within do-while() loop. We realize that weight *w* is the error function provided in pdf. The *erfc()* is provided in TMath library.

$$w = erfc\left(\frac{\sigma_t}{\sqrt{2}\tau} - \frac{t}{\sqrt{2}\sigma_t}\right)$$

And hence, we select a particular t if w evaluated at that t is greater than product of a new random number λ_2 and maximum value of w.

We normalize the distribution function obtained by Inversion Technique adopting standard procedure and draw expected pdf curve on top of normalized distribution function using user-defined pdf_fit() function.

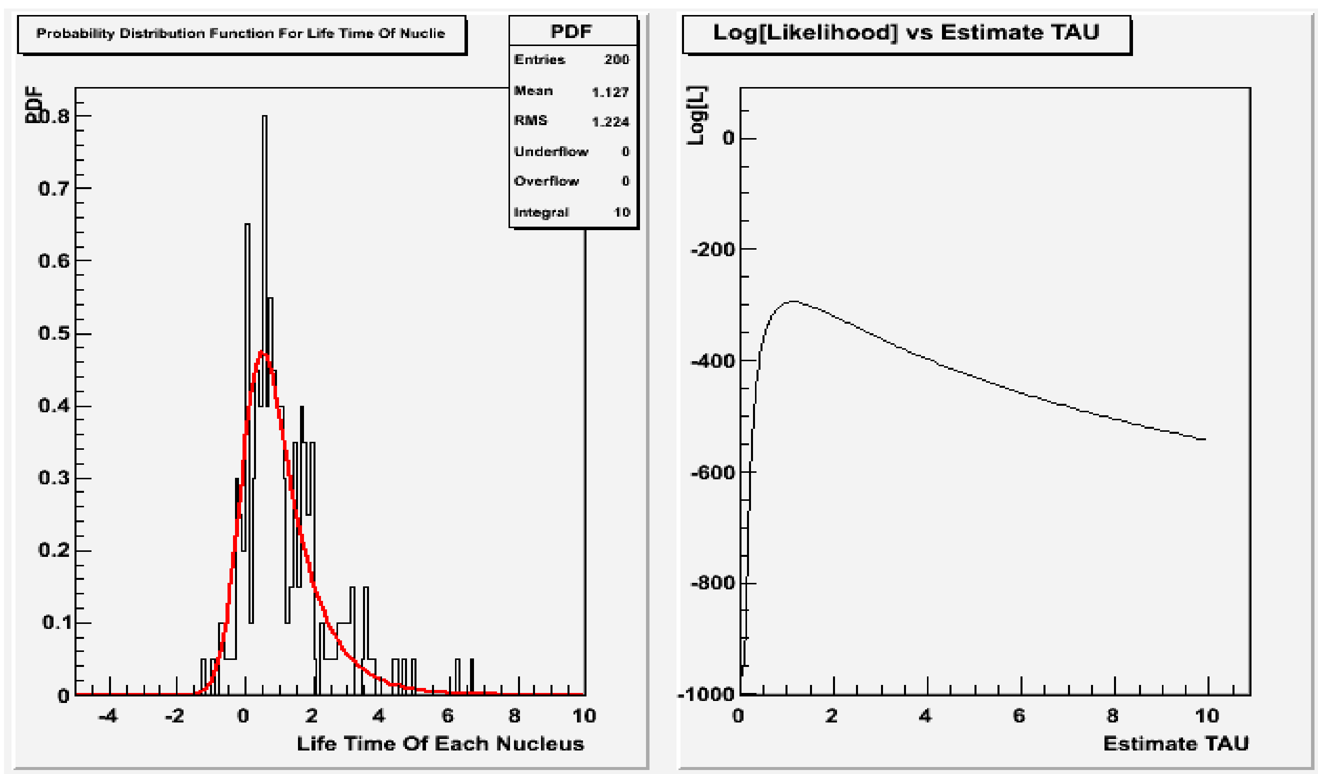

Figure 2.1: On left hand, expected pdf after Inversion Technique/Gaussian Smearing is plotted. Red curve is the given pdf plot. On right hand, Likelihood function is plotted against Estimate τ.

LIKELIHOOD METHOD, EVALUATION OF $\hat{\tau}$ AND $\sigma_{\hat{\tau}}$:

The LogLikelihood method is adopted to evaluate estimate $\hat{\tau}$ and $\sigma_{\hat{\tau}}$. The governing equations are reported below.

$$\sum_{i=1}^{200} \ln f(t_i;\tau,\sigma_t) = \sum_{i=1}^{200}\left[\ln\left(\frac{1}{2\tau}\right) + \left(\frac{\sigma_t^2}{2\tau^2} - \frac{t_i}{\tau}\right) + \ln\left(erfc\left(\frac{\sigma_t}{\sqrt{2}\tau} - \frac{t_i}{\sqrt{2}\sigma_t}\right)\right)\right]$$

We have written a user-defined function LOG_Likelihood_Function() to evaluate above summation for each value of $\hat{\tau}$ ranging from 0 to 10.0 and the plot is given in Figure2.1 on right side. The maximum of this plot occurs at estimate value $\hat{\tau}$ and $\sigma_{\hat{\tau}}$ is calculated by noting abscissa when LogLikelihood function is reduced by 0.5 as mentioned in lectures. It is done numerically in our code. Estimate value of $\hat{\tau}$ and $\sigma_{\hat{\tau}}$ for three such run, randomly chosen are reported in Table 2.1 as followed.

Experiment Number	$\hat{\tau}$	$\sigma_{\hat{\tau}}$
1	0.941	0.0735
2	0.958	0.0750
3	1.019	0.0790

Table 2.1: Values of $\hat{\tau}$ and $\sigma_{\hat{\tau}}$ for randomly chosen three simulated experiments.

AUTOMATION OF CODE FOR ACCOMULATING STATISTICS OF 100 EXPERIMENTS AND FITTING GAUSSIAN CURVE:

We introduced a for-loop, converted the variable for estimator and uncertainty to array of size 100 and accumulated 100 values for $\frac{\hat{\tau}-\tau}{\sigma_{\hat{\tau}}}$ in a histogram. We observed a Gaussian behaviour and fitted a Gaussian curve to the histogram using TF1 method. The plot is shown in Figure2.2.

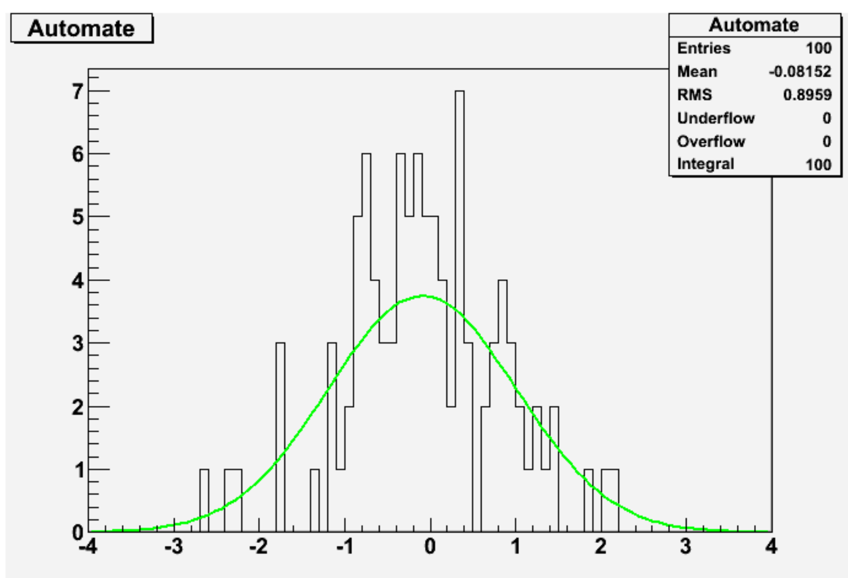

Figure 2.2: Automation of Code to perform 100 experiments and obtain distribution of $\frac{\hat{\tau}-\tau}{\sigma_{\hat{\tau}}}$.

The Gaussian curve in green is the best fit obtained by ROOT. The fitting parameters for Gaussian curve to best represent the statistics of experimental data accumulated in histogram are presented in Table 2.2.

	Constant	**Mean**	**Sigma**
Gaussian Parameters	3.74857	-0.0911636	1.10001

Table 2.2: Gaussian curve best fit parameters.

It shows that the histogrammed data represents a unit Gaussian trend.