(1) A processor suffers transient failures as a Poisson process, with rate $\lambda$. When failure strikes, it takes a deterministic time, $t_{rec}$, to recover and reboot the processor. Once it is rebooted, the processor rolls back to the latest available checkpoint and restarts execution from there. The execution time is $t_e$ and the time per checkpoint is $t_c$. You have $n_c$ checkpoints in all, evenly spaced.

In the derivation in class, we used $t_e/n_c + t_c$ as the duration of an intercheckpoint segment: this translates to having a checkpoint at the end of the execution. In this question, assume that this is not true, i.e., there is no checkpoint at the end. The duration of a segment will now become $t_e/(n_c + 1) + t_c$. (In your code, you can for simplicity ignore the fact that the last segment will be shorter by $t_e$ than the others).

In this work, you will need to find the sum of a number of uniformly distributed random variables, representing the computation lost when a transient strikes. The density function of this is a piecewise linear curve, which can be tedious to obtain. Instead, simplify by doing this numerically after digitizing the time axis, using some small constant $\Delta$. Consider that the computation lost is always some multiple of $\Delta$: the density functions will now become mass functions. Clearly, $\Delta$ must be kept very small to reduce quantization error. Use a $\Delta$ small enough that reducing it further makes no significant difference to the result.

Write a computer program to obtain the probability that a deadline is missed. Make sure your program is fully documented, and include your source code in what you hand in. Also provide a separate description, containing the mathematical derivation that underlies your program.

(a) Fix $t_e = 1$, $t_c = 0.1$, $t_{rec} = 0.1$, and $t_{deadline} = 1.5$. Plot the probability of missing the deadline, varying $n_c$ from 0 to 5: draw three separate curves (on the same plot) for the cases $\lambda = 10^{-3}, 10^{-2}, 10^{-1}$. Discuss your result.

(b) Repeat (a) if $t_{deadline} = 2.0$. Discuss any differences you observe between these results and those in part (a).

(c) Now, develop the following first-order approximation to the failure probability. We have:

$$\text{Prob[deadline is missed]} = \sum_{i=0}^{\infty} \text{Prob[deadline is missed|} i \text{ failures occur]} \times \text{Prob[i failures occur]}.$$ 

A lower bound to the failure probability can be obtained by using the fact that

$$\text{Prob[deadline is missed|} i \text{ failures occur]} \geq \text{Prob[deadline is missed|} 1 \text{ failure occurs]}$$ for $i = 1, 2, \cdots$
An upper bound can be obtained by using the following:

\[
\text{Prob}[\text{deadline is missed} | i \text{ failures occur}] \leq 1 \quad i = 2, 3, \ldots.
\]

Write a program to compute these upper and lower bounds. Add an explanatory note showing the derivation for this program.

Plot the upper bound of the deadline missing probability using the parameters in part (a). Separately, plot the difference between the upper and lower bounds.

(2) Consider the problem of false positives and false negatives. You are given that the true value of some output is observed to have distribution

\[
F(x) = \begin{cases} 
0 & \text{if } x < 0 \\
1 - e^{-5x^2} & \text{if } 0 \leq x < 10 \\
1 & \text{if } x \geq 10 
\end{cases}
\]

It is believed that faulty outputs will be uniformly distributed over the range \([0, 20]\). The cost of a false negative is 100; the cost of a false positive is 500.

(a) What should be the lower and upper bounds used by an acceptance test to minimize the expected cost associated with an output?

(b) Repeat (a) if faulty outputs are believed to be uniformly distributed over the range \([0, 100]\).