1. Monte Carlo Method: Definition and Purpose

The **Monte Carlo method** is a computational technique that uses **random sampling** to estimate numerical results for problems that may be deterministic in nature but are too complex for analytical solutions. It is particularly useful in systems involving uncertainty, nonlinear dynamics, or a high number of degrees of freedom.

In a nutshell, Monte Carlo is a straightforward method that can evaluate the sensitivity of any system without having to perform an infinite number of simulations. By assessing the system through randomness, this approach helps reduce the computational burden. However, a valid question arises:

"How many random samples/runs are required inorder to assess all the possible sensitive cases in a system?"

The answer depends on the acceptable tolerance between approximate results and the actual scenario. A valid number of samples/ runs will have to be chosen such that the approximate result converges to the real solution. Or in other words **how the global mean converges to the sample mean as the number of runs/ samples are increased (Theory of large numbers)**

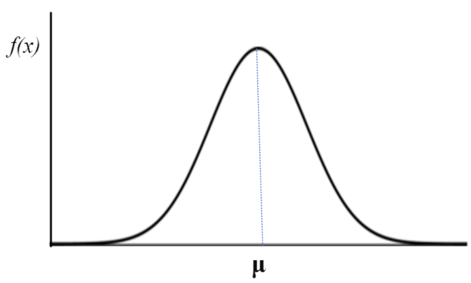
In this specific context, the Monte Carlo approach is applied to simulate **projectile motion** using randomly generated **<u>initial velocities and launch angles,</u>** allowing the study of the resulting **downrange variability** and success probabilities.

2. Types of Variable Distributions

The variables into the system are usually modelled in a way where they follow a certain distribution for their randomness, below there is a general introduction to each of the distribution and for the specific variables in the code, the type of distribution has been listed

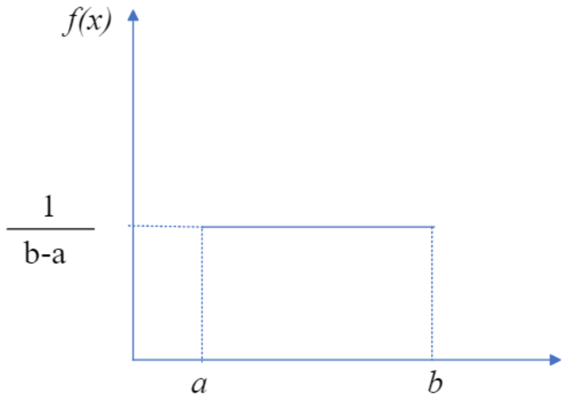
2.1 Normal (Gaussian) Distribution

The normal distribution, also known as the Gaussian distribution, is characterized by its distinctive bell-shaped curve. This distribution is commonly used when the system is expected to yield values that are more likely to occur around the mean and become progressively less probable as they move further away from the mean.



2.2 Uniform Distribution

The uniform distribution is characterised by its definition over a specific interval. Within this interval, every possible value is assigned an equal probability, and no values beyond this interval are permitted. This distribution is often described as conservative because it treats values near the edges of the interval as equally probable as those in the central part of the interval.



2.3 Exponential Distribution

The exponential distribution is used to model the time between events in a Poisson process, such as failure rates or arrival times. It is right-skewed and defined by a rate parameter (λ). Smaller values are more likely, and the probability decreases exponentially for larger values.

2.4 Log-normal Distribution

The log-normal distribution models variables whose logarithms are normally distributed. It is right-skewed and only defined for positive values. This distribution is often used for data influenced by multiplicative effects, like stock prices or biological growth.

2.5 Beta Distribution

The beta distribution is defined on the interval [0, 1] and shaped by two parameters (α and β). It is ideal for modeling probabilities, proportions, and bounded variables. Its shape is flexible, making it useful in Bayesian analysis and uncertainty modeling.

The simulation utilizes Gaussian (Normal) distributions to represent uncertainties in input parameters:

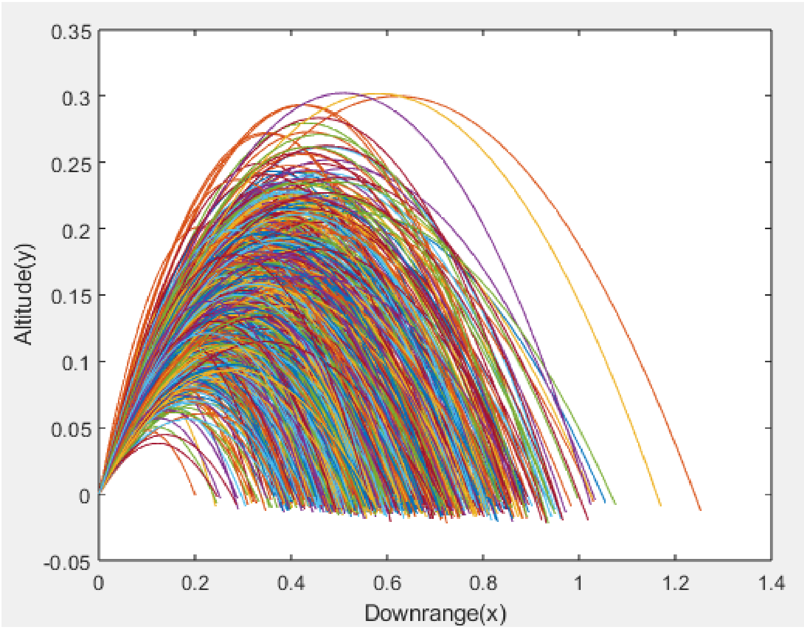
- **Initial Velocity (v)**: Sampled from a **normal distribution with a mean of 2.5 m/s and a standard deviation of 0.3 m/s.**

- **Launch Angle (theta)**: Sampled from a **normal distribution with a mean of 45 degrees and a standard deviation of 5 degrees.**

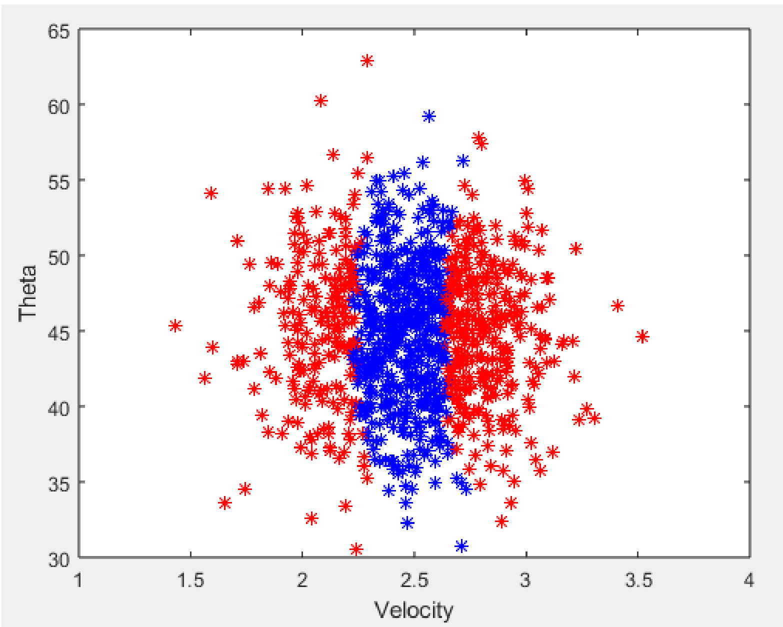
3. Scatter Plots for Analysis

To understand how velocity and angle variations affect the downrange distance, several scatter plots are generated, where some could provide useful inference on the influential level of parameters in the system [3]

- **Trajectory Plot**: Displays the path of all projectile samples. It visually represents the variation in the downrange distance.

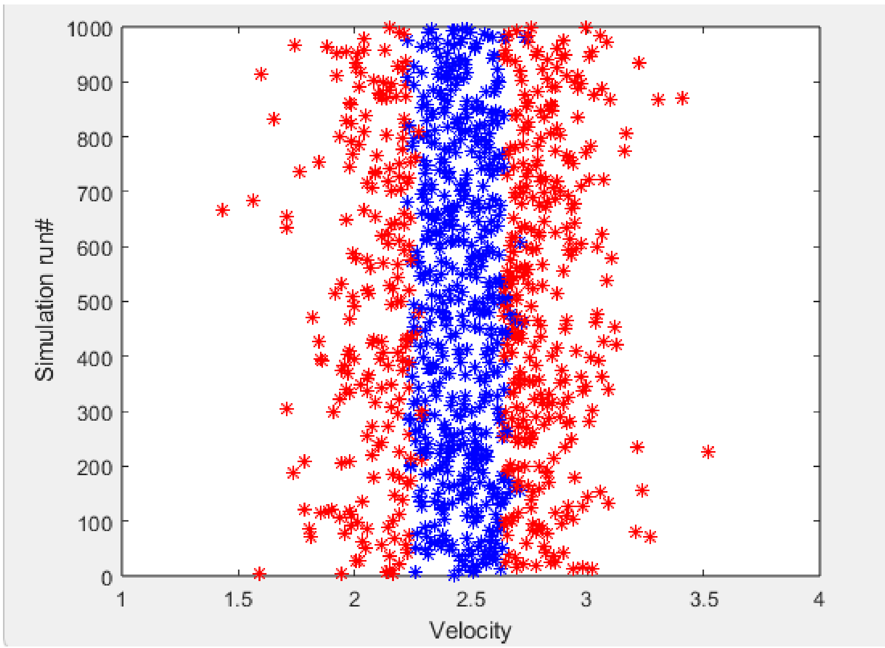


- **Velocity vs Angle Scatter**: Highlights the successful and unsuccessful input combinations. Blue stars indicate success; red stars indicate failure.



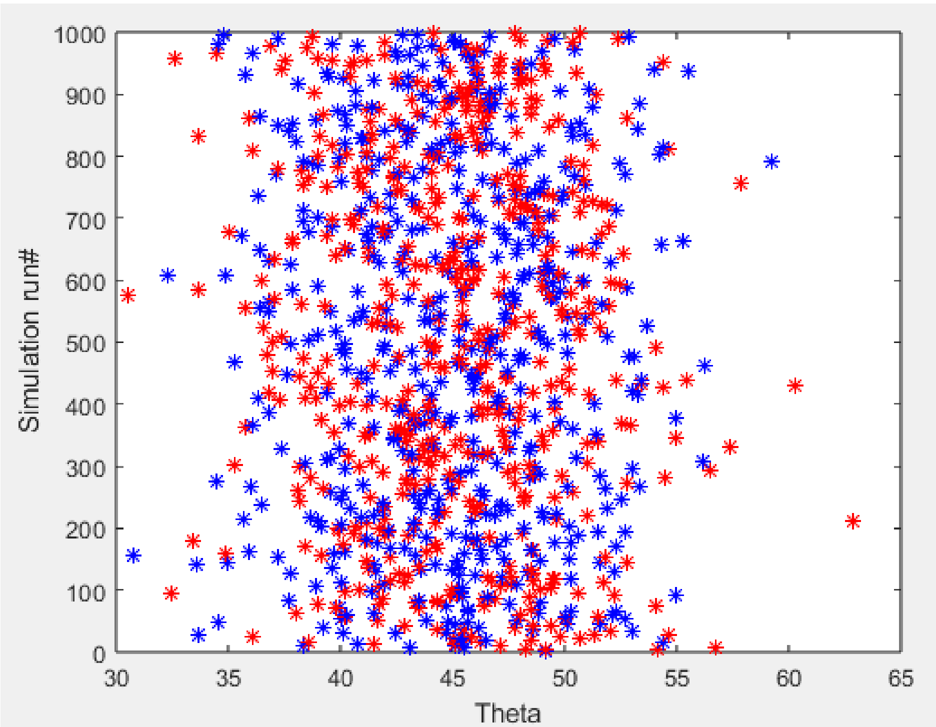
Inference: A clear pattern of success and failure is observed with velocity, where there is no such inference with theta

- **Velocity vs Simulation Run#**: Helps visualize trends or clustering of successful velocity values across simulation iterations.



Inference: A clear pattern of success and failure is observed with velocity, where we could see only **a certain value of velocity is resulting in success.**

- **Angle vs Simulation Run#**: Similar to velocity, this shows angle contributions to success or failure,



- **Inference:** shows no distinction in the simulation runs vs theta, so **there is less impact of theta onto the success or failure**

4. Categorical Success Criteria and Kernel Density Estimation (KDE)

The success criterion is defined categorically as:

- **Success**: Downrange lies between 0.5 m and 0.7 m (i.e., mean ± standard deviation of 0.6 ± 0.1 m).

Small introduction to Kernel Density Estimation:

In simple words, one can define density estimation as PDF (Probability density function) but categorically, meaning put them into success or failure. Density estimation can be done in any form, such as gaussian,cubical ,kernels , etc.

Here, we plot the density estimation for the parameters and how much success or failure has been resulting from that, so we can say that **the more distinction between these PDFs of success and failure , the more influential or the more effective the parameter is in creating success/failure !**

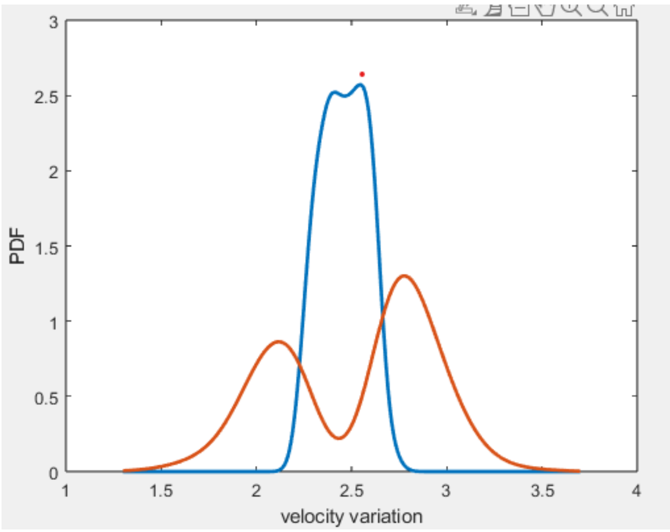
$$p_{success}(x_j) = \frac{1}{N_{success}} \sum_{i=1}^{N_{success}} \frac{1}{(2\pi h_{success}^2)^{1/2}} e^{\frac{||x_j - x_i||^2}{2h_{success}^2}}$$

$$p_{failure}(x_j) = \frac{1}{N_{failure}} \sum_{i=1}^{N_{failure}} \frac{1}{(2\pi h_{failure}^2)^{1/2}} e^{\frac{||x_j - x_i||^2}{2h_{failure}^2}}$$

For a clear theory, on step size ‘h’ in the formula, please refer paper [3]

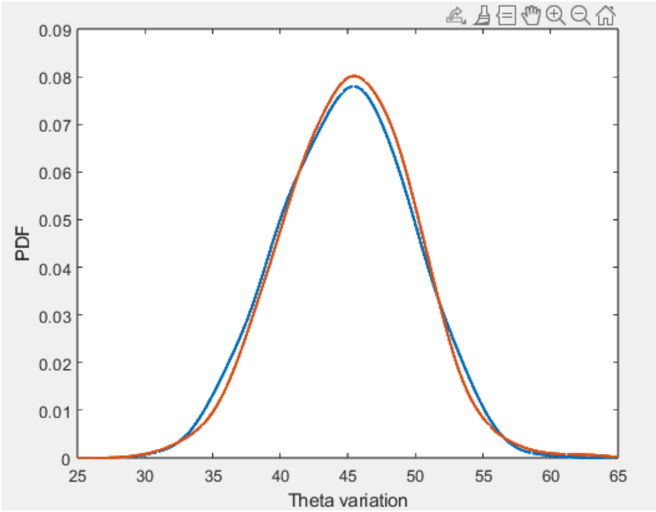
Kernel Density Estimation was performed on the velocity and angle datasets to estimate their probability density functions (PDFs):

- **KDE for Successful vs Failed Velocities**



Inference: here there is clear distinction between success and failure PDFs for a certain bandwidth of velocity

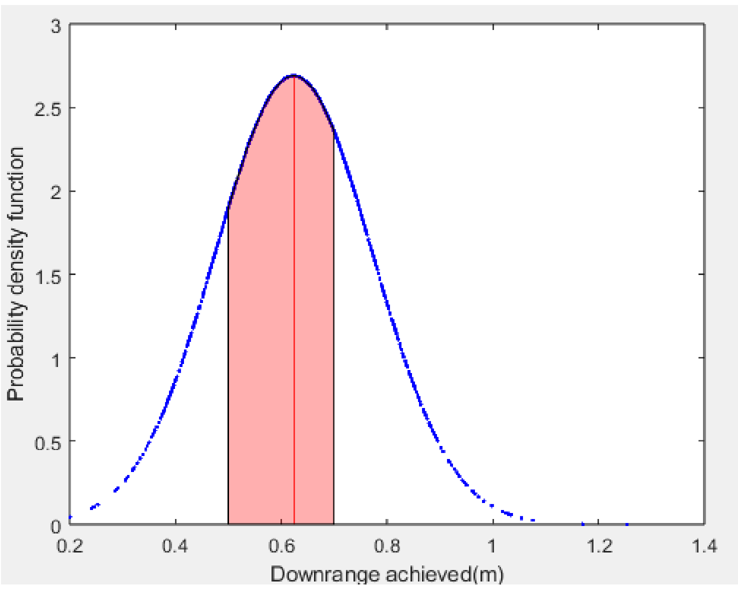
- **KDE for Successful vs Failed Angles**



Inference: here there is not much distinction between success and failure PDFs

These visualizations help identify how the distributions differ between successful and unsuccessful outcomes.

An overlay of the downrange PDF was also plotted with a shaded area representing the region of success.



5. Non-Categorical Correlation and Cost Function Analysis

To quantify the impact of each parameter on the output variability:

- A cost function was calculated as the integral of the absolute difference between PDFs of successful and failed samples for each input (velocity and angle).

$$J = \sum_{j=1}^{M} |p_{success}(x_j) - p_{failure}(x_j)| * D$$

Output displayed in the terminal:

```
>> monte
the mean of the output data:0.63326
the standard deviation of the output data:0.15382
the cost function of the output data due to velocity variation:1.4475
the cost function of the output data due to theta variation:0.1555
the confidence level of the success0.47465
```

The cost functions were:

- **J1 (velocity)**: Higher values indicate stronger influence of velocity on success probability.
- **J2 (theta)**: Similarly measures the effect of launch angle.

6. Confidence Interval Estimation

Based on the normal distribution fit of the downrange values:

- The confidence level that a random trial will result in success (i.e., land in the target zone) was computed using the cumulative distribution function (CDF).

7. Conclusion:

The code attached 'Monte_Carlo' is clearly discussed along with theory in this document.

Even Though this is a simple system used for illustrating this method, this could be implied to many applications that could be used for testing various systems sensitivity and identifying influential parameters in them.

References

- [1] Kroese, D. P., Brereton, T., Taimre, T., & Botev, Z. I. (2014). "Why the Monte Carlo method is so important today." *Wiley Interdisciplinary Reviews: Computational Statistics*, 6(6), 386–392.

- [2] Rubinstein, R. Y., & Kroese, D. P. (2016). *Simulation and the Monte Carlo method*. John Wiley & Sons.

- [3] NASA Technical Reports (18599348.pdf & 20100038453.pdf): Techniques for flight simulation using stochastic methods and uncertainty analysis.