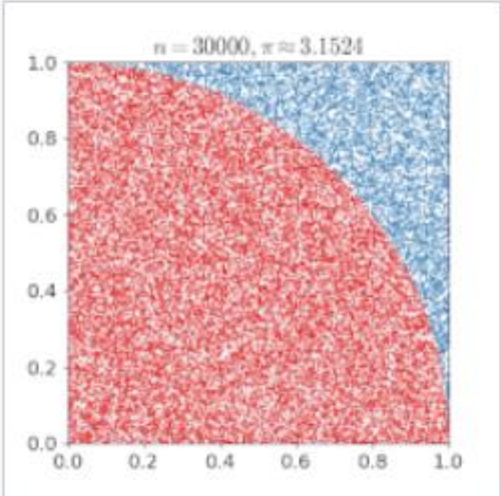
MonteCarlo Simulation

"**Monte Carlo methods**, or **Monte Carlo experiments**, are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. The underlying concept is to use randomness to solve problems that might be deterministic in principle. They are often used in physical and mathematical problems and are most useful when it is difficult or impossible to use other approaches."

https://en.wikipedia.org/wiki/Monte_Carlo_method



Monte Carlo method applied to approximating the value of π.

Figure1. Throwing random darts on the square and counting the number in the red area

The basic idea is to create random variables to the inputs of your experiment, in order to simulate your experiment, and get the variations respect to the theoretical values. The most common way to define random variables is to use the normal distribution.

```
close all
clear all
clc

tiledlayout(2,2) % Requires R2019b or later

nexttile;
x = randn(100000,1); % Centered and sigma = 1
nbins = 300;
h = histogram(x,nbins);

% ======================
nexttile;
x = 3 * randn(100000,1); % centered and sigma = 3
nbins = 300;
h = histogram(x,nbins);

% ======================
nexttile;
x = 10 + randn(100000,1); % sigma = 1 displaced by 10
nbins = 300;
h = histogram(x,nbins);

% ======================
```

```
nexttile;
x = 10 + 3 * randn(100000,1); % sigma = 3 displaced by 10
nbins = 300;
h = histogram(x,nbins);
```

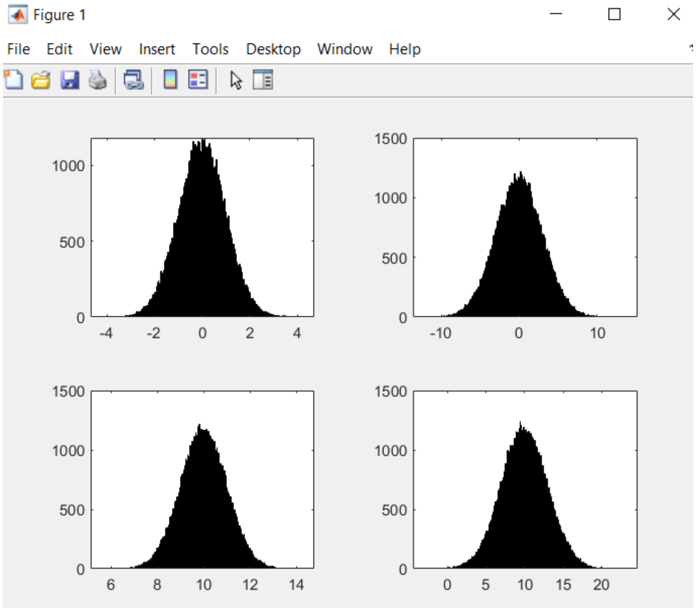


Figure 2- Normal Probabilities with different mean and sigma

Application to the analysis of Laser camera Triangulation

We like to analyze the errors in reconstruction in function of the errors in the subpixel precision of the laser extraction, and the error in the physical points of the cone. We assume that the H matrix (homography matrix) obtained from the calibration is correct.

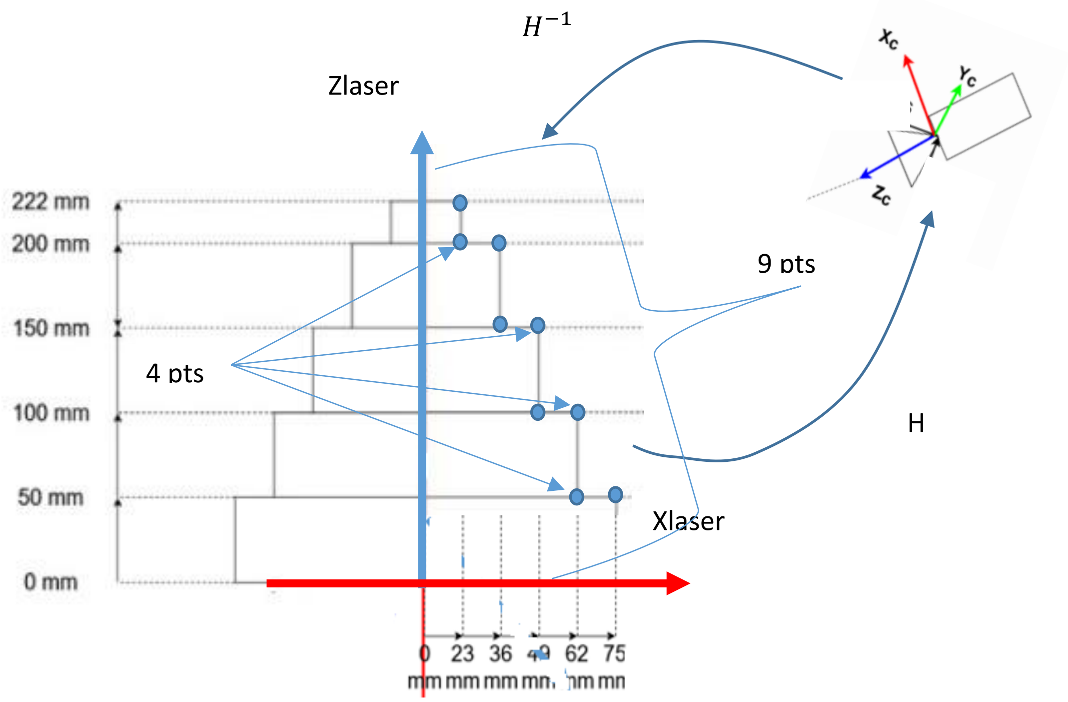


Figure 3. Point correspondence for MonteCarlo analysis

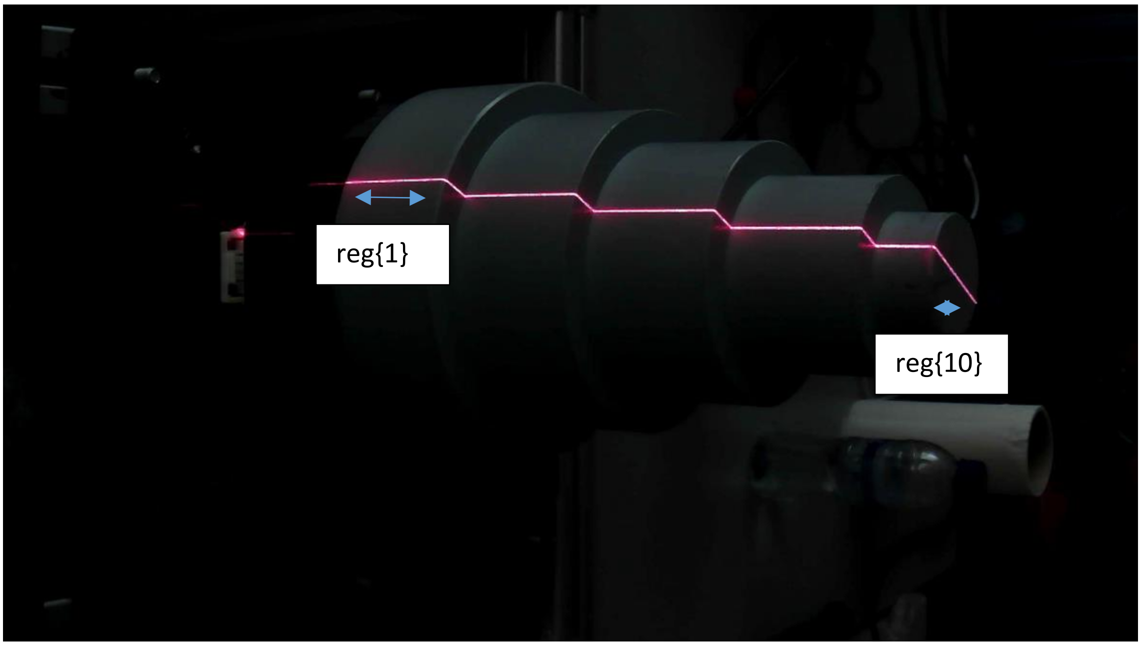
In order to do the Monte-Carlo Analysis, we project the set points in the cone (theoretical points plus a random noise) to the image. To the points in the image, we add a random noise. We find straight lines and intersections, as previously done, and re-project the image points to the laser plane. We calculate the errors between the theoretical points in the cone and the re-projected ones.

Hint:

Define the same set of regions as in the previous example:

```
reg{1} = [600, 741];
reg{2} = [750, 780];
reg{3} = [788, 959];
reg{4} = [970, 995];
reg{5} = [1009, 1200];
reg{6} = [1205, 1232];
reg{7} = [1235, 1445];
reg{8} = [1456, 1473];
reg{9} = [1478, 1568];
reg{10} = [1581, 1627];
```

```
PtsCono = [75 62 62 49 49 36 36 23 23; ...
           50 50 100 100 150 150 200 200 222];
```



We calculate for each region the theoretical points in the image vertical line, calculating which theoretical point of the cone is the corresponding one. Add noise to the theoretical point in the cone, calculated the corresponding image, and add point to this image. Calculate the straight lines and intersection of the lines in order to calculate the modified homography. Re-project the points in all segments with the new homography and calculate the errors with the theoretical points in the cone. Calculate the statistics repeating the calculation n = 100 times. Calculate the statistics for the following sigma values (laser and images).

```
% Variances of added normal noises
sigmaPw = [0, 0.001, 0.005, 0.01, 0.05, 0.1];
sigmaPixel = [0, 0.01, 0.05, 0.1];
```