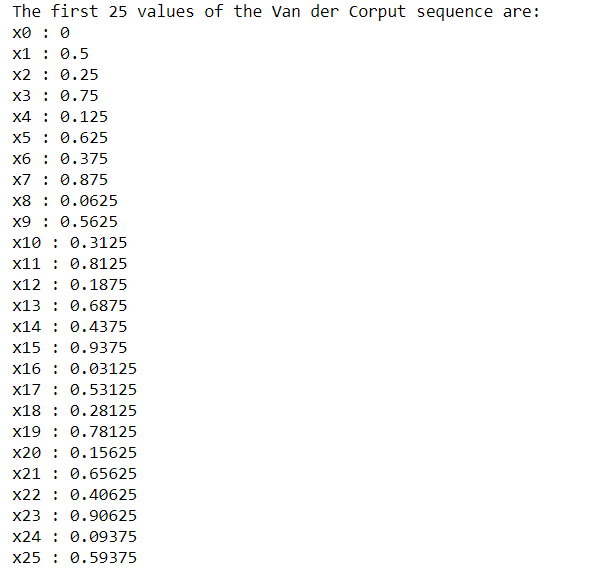
MA 323 (2020) Monte Carlo Simulation Lab 12

**Name:** Udandarao Sai Sandeep

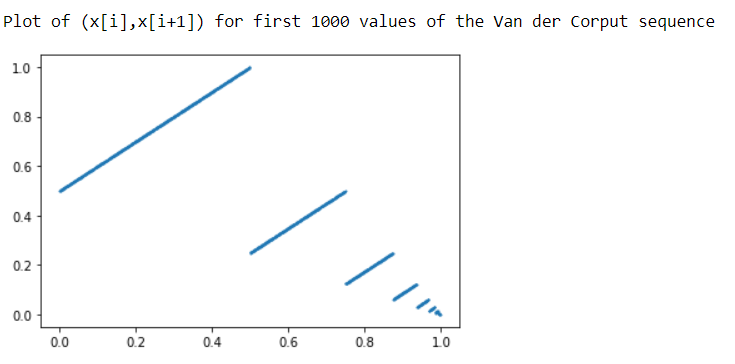
**Roll Number:** 180123063

**Dept.:** Mathematics and Computing

**Q1.**

The left screenshot shows the initial 26 values (including **0**) of the Van Der Corupt sequence using the radical inverse function **xi: = φ2(i)**. The generation of the Van Der Corupt sequence has been implemented in the **gen\_vander\_corupt\_seq()** function in the code.

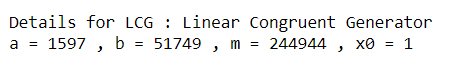
The below screenshot shows the pattern of **(x[i], x[i+1])** of the Van Der Corupt sequence. The scatter plot displays a series of line segments with **slope 1** and decreasing lengths. The line segments seem to converge to the point **(1,0)**. The plot is not random, and noticeable linear dependency is observed.



x[i+1]

x[i]

Details of LCG used:



Comparison of density plots of **LCG** and **Van Der Corupt** Sequence for n = 100 and n = 100000 is as follows:

|  |  |  |
| --- | --- | --- |
| n | LCG | Van Der Corupt Sequence |
| 100 |  |  |
| 100000 |  |  |

It can be observed that for **n = 100**, the **Van der Corupt Sequence** and the **LCG** sequence somewhat mimic uniformity. But **100** is a very less value to judge the uniformity of a sequence. For **n = 100000**,both the sequence strongly displays strong signs of uniformity. In fact, **Van der corrupt** sequence mimics uniformity better than the **LCG** itself.

**Q2.**

Firstly, first **100** values of the **Halton Sequence xi = (φ2(i), φ3(i))** were generated. A 2D plot was created using python matplotlib module of python. Similarly, first **100000** values of the **Halton Sequence xi = (φ2(i),φ3(i))** was then generated. It was also plotted out. The plots are shown in the next page. For **n = 100**, the plot is very less dense, and the points are scattered out without any specific pattern. The points are spread out haphazardly on the 1x1 square in the 2D plane. For **n = 100000**,points fill the entire 1x1 square on the 2D plane. This shows that for larger values of n, Halton sequence mimics uniformity strongly, making it a great sequence for generating random numbers.

