VERKFRÆÐILEGAR BESTUNARAÐFERÐIR

Day 10 - Group 2

T-423-ENOP

Arnar Gylfi Haraldsson
`arnarh23@ru.is`

Hafþór Árni Hermannsson
`hafthorh20@ru.is`

Ragnheiður Gná Gústafsdóttir
`ragnheidurg@ru.is`

May 14, 2024

Exercise 1

Description

Implement procedure for evaluating the uniformity measure of the set of sample points in the design space as described during the lecture. Implement visualization procedure that is able to display projections of the sample points in n-dimensional design space $[l_1, u_1] \times [l_2, u_2] \times \cdots \times [l_n, u_n]$ onto two-dimensional sub-spaces $[l_k, u_k] \times [l_l, u_l]$, where $k, l \in \{1, 2, \ldots, n\}, k \neq l$. Consider $n = 2, 3, 4$ and 5. Test all the procedures using sets of sample points of your choice.

Functionality

The uniformity measure is updated using the equation given in lecture slides:

$$E = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \left(\frac{1}{||x_i - x_j||^2} \right)$$

Solution

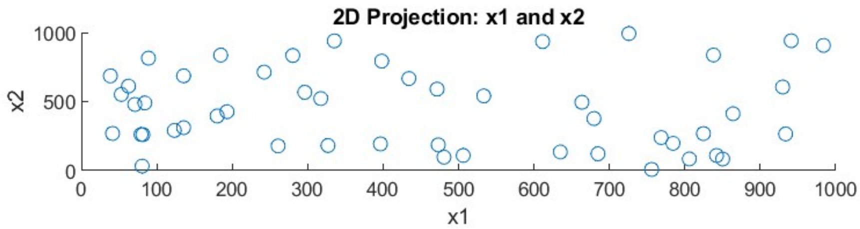


Figure 1: n = 2

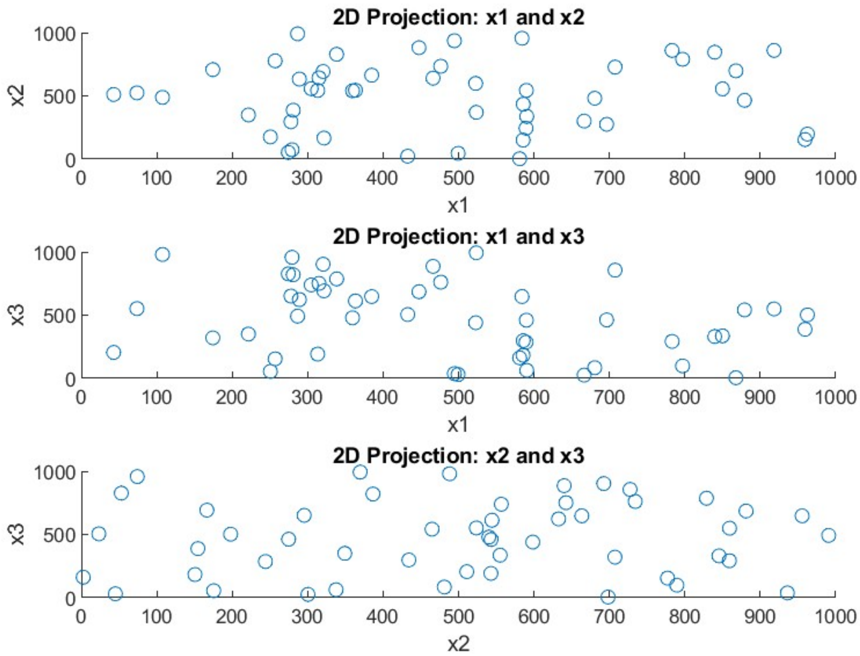


Figure 2: n = 3

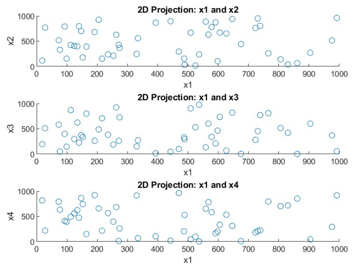


Figure 3: n = 4

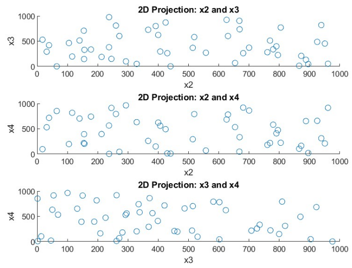


Figure 4: n = 4

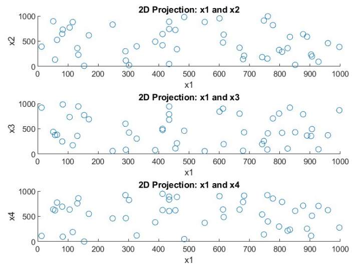


Figure 5: n = 4

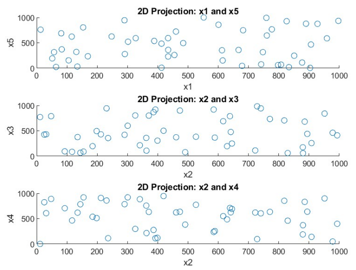


Figure 6: n = 5

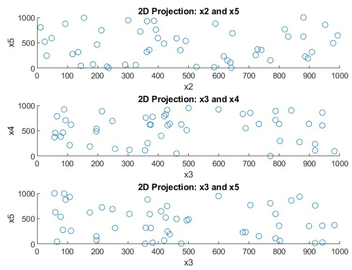


Figure 7: n = 5

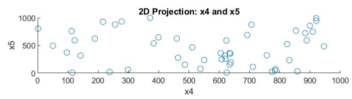


Figure 8: n = 5

The uniformity measure for the points for each dimension was computed:

n	2	3	4	5
E	0.1747	0.0069	0.0034	0.0021

Table 1: Uniformity for each n

Exercise 2

Description

Implement the Latin Hypercube Sampling algorithm that generates N points in n-dimensional design space $[l_1, u_1] \times [l_2, u_2] \times \cdots \times [l_n, u_n]$. Visualize the output of the algorithm for $n = 2$ and 3 for the following values of N: 50, 100, and 200.

Functionality

The algorithm works as follows: It divides each of n intervals into N subintervals, where N is the number of samples (this yields N^n bins in the design space). Next, select N samples so that

(i) each sample is randomly placed inside a bin, and

(ii) for all one-dimensional projections of the N samples and bins, there will be one and only one sample in each bin.

(This information was found in the slides provided by Slawomir Koziel)
Additionally, a plot function was made that can plot points in any multidimensional space in two dimensions, it was coded to include 3 subplots per figure and then make a new figure.

Result

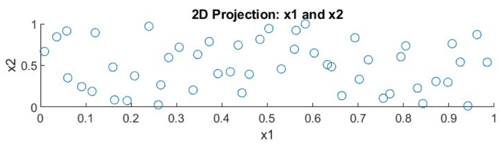


Figure 9: n = 2, N = 50.

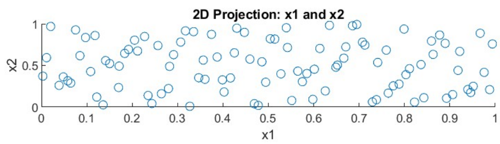


Figure 10: n = 2, N = 100.

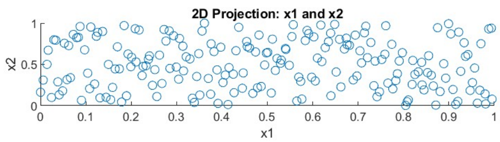


Figure 11: n = 2, N = 200.

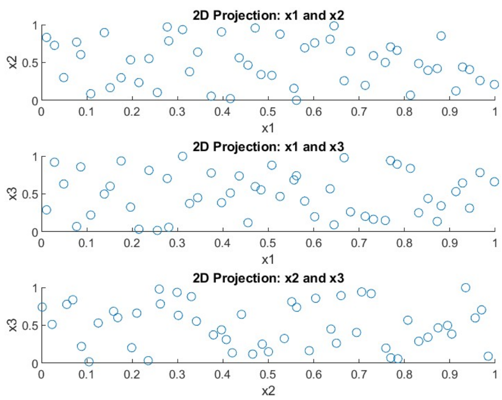


Figure 12: n = 3, N = 50.

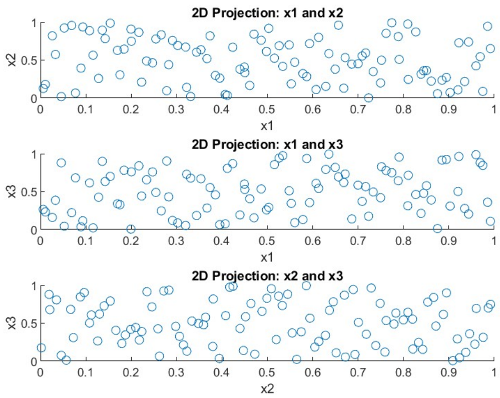


Figure 13: n = 3, N = 100.

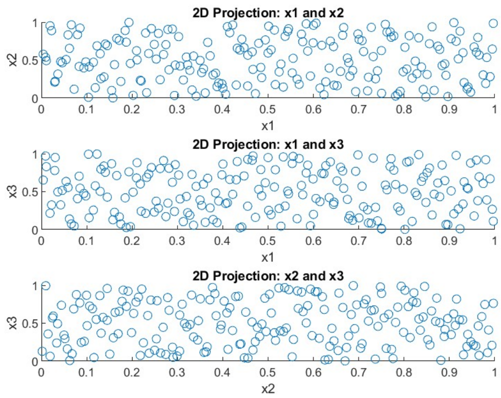


Figure 14: n = 3, N = 200.

Exercise 3

Description

Functionality

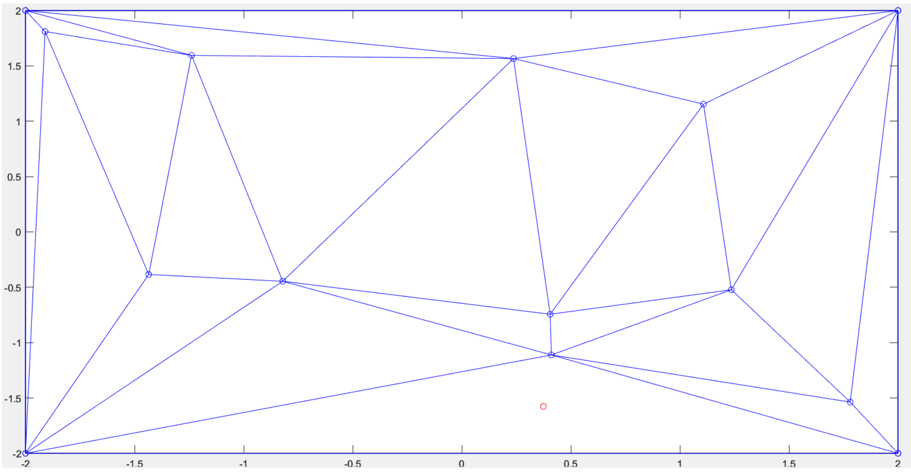
We generate N_0 initial points randomly and change the first four points to be the corners defined by upper and lower bounds. We use the inbuilt `MATLAB` function `delaunayTriangulation` to create triangles between the initial points. The geometric center for each triangle is computed again using an inbuilt `MATLAB` function `incenter`. We compute the largest triangle by iterating through each triangle and computing the area using the determinant method

$$\text{Area} = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$$

then we append the center point of the largest triangle to the sample points and repeat the process.

Results

Using $N_0 = 10$ we have at 5 iterations:



and at 50 iterations: