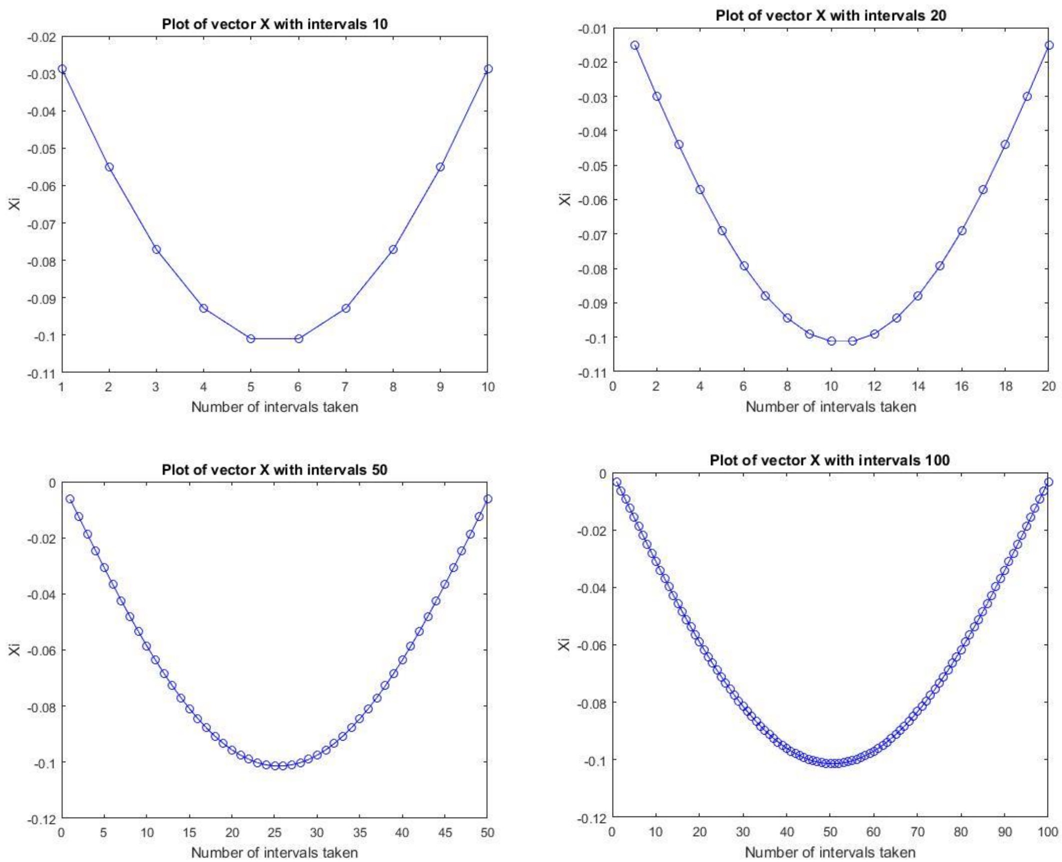
The purpose of this assignment is using Crout factorization to solve a linear system Ax=b. Crout factorization is a specific type of LU decomposition of a matrix and it has the property that the upper triangular matrix is normalized (i.e. having all 1s on the diagonal). This contrasts with Doolittle factorization which has a normalized lower triangular matrix. In the case of a tridiagonal matrix, Crout factorization can compute the result of a linear system much more quickly compared to other standard methods. In addition, Crout factorization is not limited to tridiagonal or symmetric matrix; any invertible square matrix has a valid LU Crout decomposition, as sources on Wikipedia suggest[1].

After initializing the matrix and the vector using the given parameters, the Crout algorithm returns one lower triangular matrix and one normalized upper triangular matrix. Using these 2 matrices, we can solve the system in 2 consecutive steps using backward substitution: first obtain the intermediate solution y using L and vector b, then obtain the solution x using U and y. Plotting the vector x gives the following graph:



As number of interval n increases, the graph increasingly resembles a lower portion of a sine curve, and as note 2 suggests, it indeed forms a solution to the differential equation Uxx =sin(πx) with boundary conditions at 0 and 1. Intuitively, the function y = -k*sin(πx) should give back sin(πx) after differentiating twice, which suggests that the vector x should be a trigonometric function.

1. Source: Wikipedia on Crout factorization (along with sample code implementation) https://en.wikipedia.org/wiki/Crout_matrix_decomposition