Q2

(a) In Cholesky's algorithm, we require the input matrix to be symmetric. The output is an upper-triangular matrix. To compute any A[i][j], we need to multiply A[i][0:i] and A[0:i][j] (this is same as using the 2 columns - A[0:i][i] and A[0:i][j]).

- Thus for computing each element we would need to columns : one column corresponding to the element itself, and another column from the left that has already been computed.
- This means we can decompose the input into tasks based on columns. Each task can correspond to computing the result for a given column.
- Once this result is computed it needs to be broadcasted to other processes so that they can use it to compute the values in their allotted columns.

We can allot these column strips in either in blocks, or in a round-robin manner to each task. The round-robin manner (cyclic column strip allocation) would allow for better load balancing compared to column strip block allocation.

- This is because in the resulting upper triangular matrix, we observe that columns on the left have very few elements while columns on the right have more elements. Hence to equally distribute the load, we would need a scheme were each process gets a mix of columns whose average size is somewhat same across all processes. Cyclic column strip allocation allows for this.

(d), (e) The speedup, efficiency and costs were calculated by varying the size of input while keeping n=4 processes as fixed. Below are the graphs:

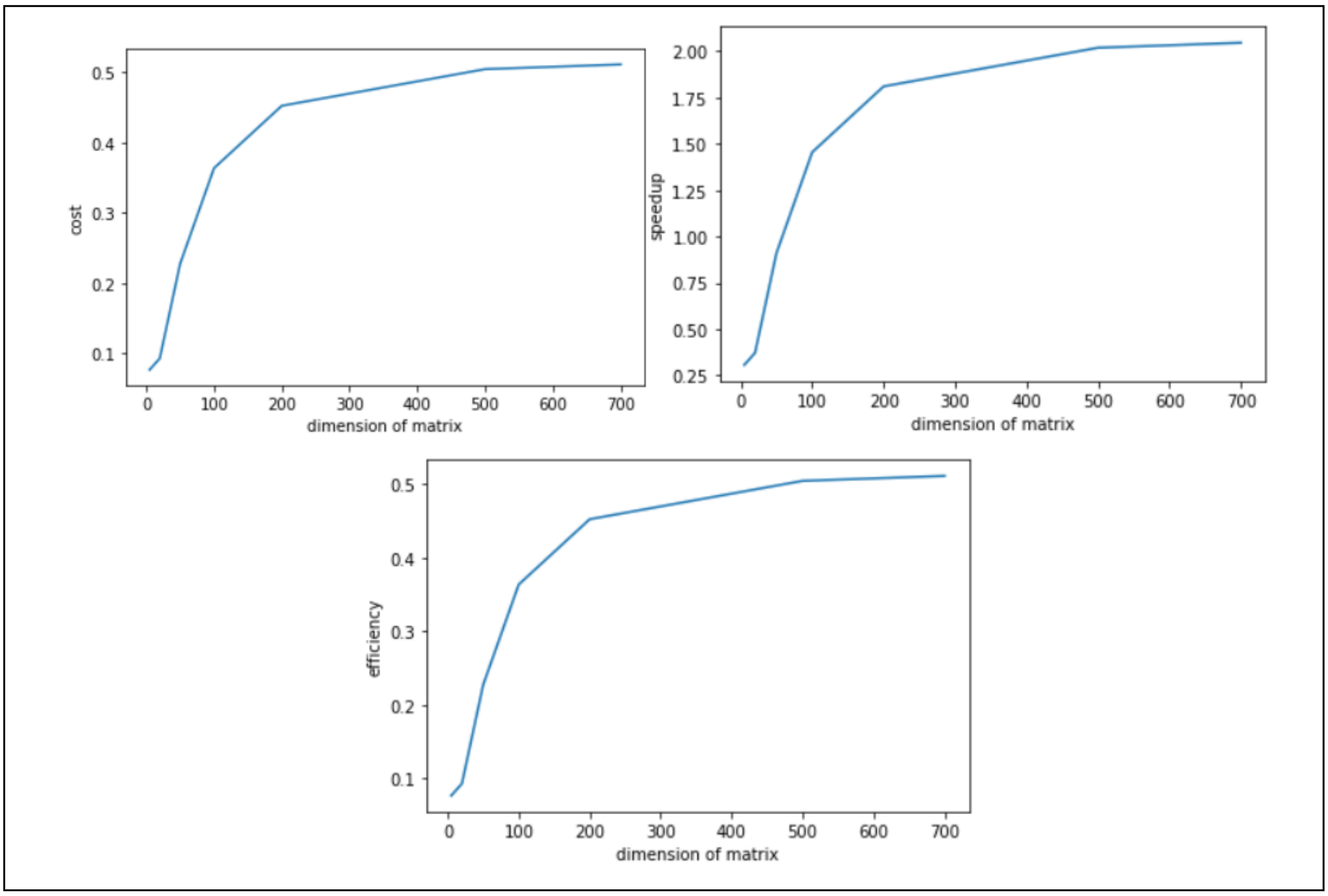

Figure 1 : Speedup, Efficiency and Cost vs size of matrix