## HW8 Game of Life

Initially I have used 2 matrices. One is a “temp” which is a temporary matrix only to store values of elements whose state has been changed. And the other one is the main matrix named “matrix”.

Both temp and matrix have the same size.

In the temp matrix, all elements have the same value. It can be anything except 0s and 1s. I have chosen the value 3.

The matrices:

I have increased the size of the matrix by 2 (both for row and column). I have done that to give an offset (border) to the entire matrix, the extra row and column all has value 0 in them. I did that for checking the values of neighbours, so that it doesn’t take any garbage values for the elements on edge or corner.

For example, a 3 x 3 matrix will look like this:

0 0 0 0 0

0 x x x 0

0 x x x 0

0 x x x 0

0 0 0 0 0

Here, x=random numbers (0s or 1s)

And the temp matrix will look like this

3 3 3 3 3

3 3 3 3 3

3 3 3 3 3

3 3 3 3 3

3 3 3 3 3

MPI execution:

In my code, When I use number of processes=number of rows, It gives me the best result.

Because each process will work on a single row,and will send their values to the master(rank 0) process .

The master takes part in the execution. And it receives all the values from the slave processes.

The master prints the result after every iteration.

Correct execution:

My parallel code doesn’t work for larger matrix. For size 8,9 it gives me segmentation fault.

And for some size, the code only works for one iteration. For further iterations, the state is not changing.