Homework: 7

Jesse Harper

Introduction 1.1:

This problem examines the behavior of discrete organisms' population growth behavior through iterative simulation. A set of parameters which govern the life and death of an array of cells is set and a 300 generation simulation is then allowed to determine the behavior of this particular random population.

Models and Methods 1.2:

The script runs as a function, with 2 supporting local helper functions. The two helper functions, “advance_cell” and “test_cell”, each run the finer details of the simulations. First the driver function initializes an array of living and dead cells call “cell”. This array will be iteratively updated by the helper functions in a 300 loop script. The script runs a 300 generation loop, which call advance_cell each generation to update the cell. Likewise, “advance_cell” goes through each element of the cell array and calls “test_cell” to determine if that element will contain a live or dead cell .

Calculations and Results 1.3:

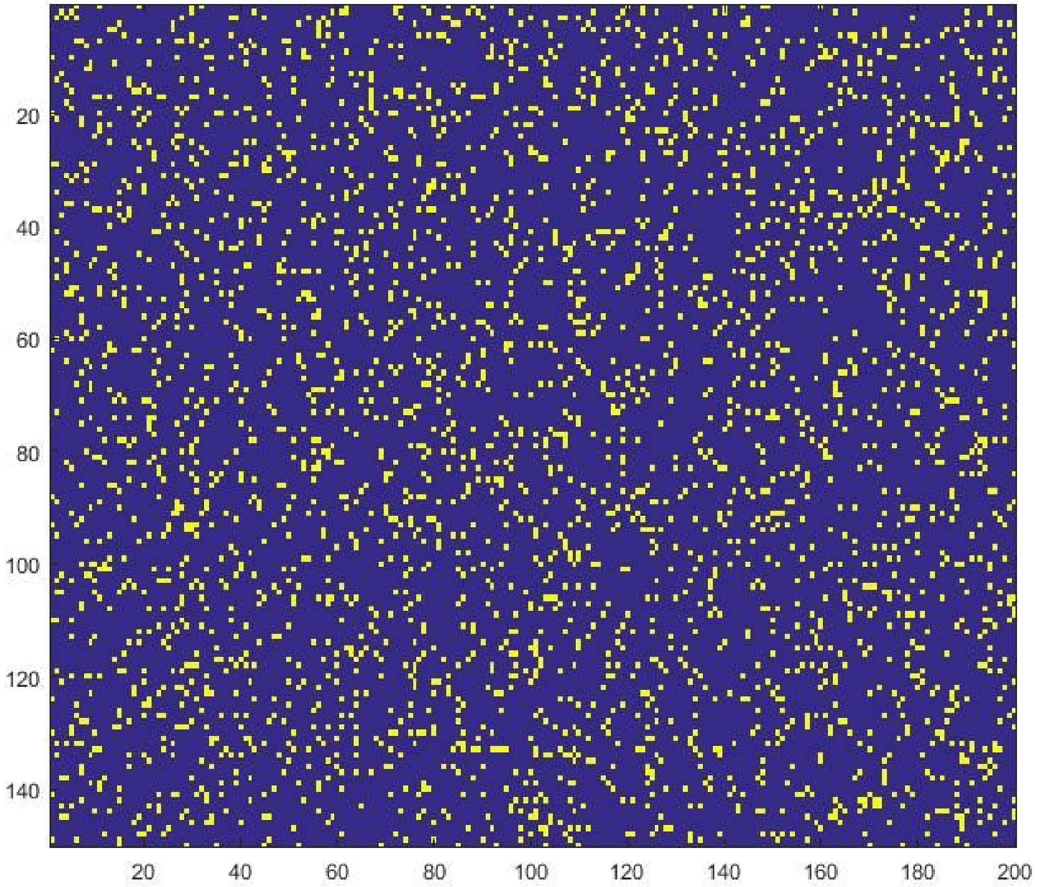

Figure 1.3.1: initial distribution of living cells across the 150 by 200 point grid

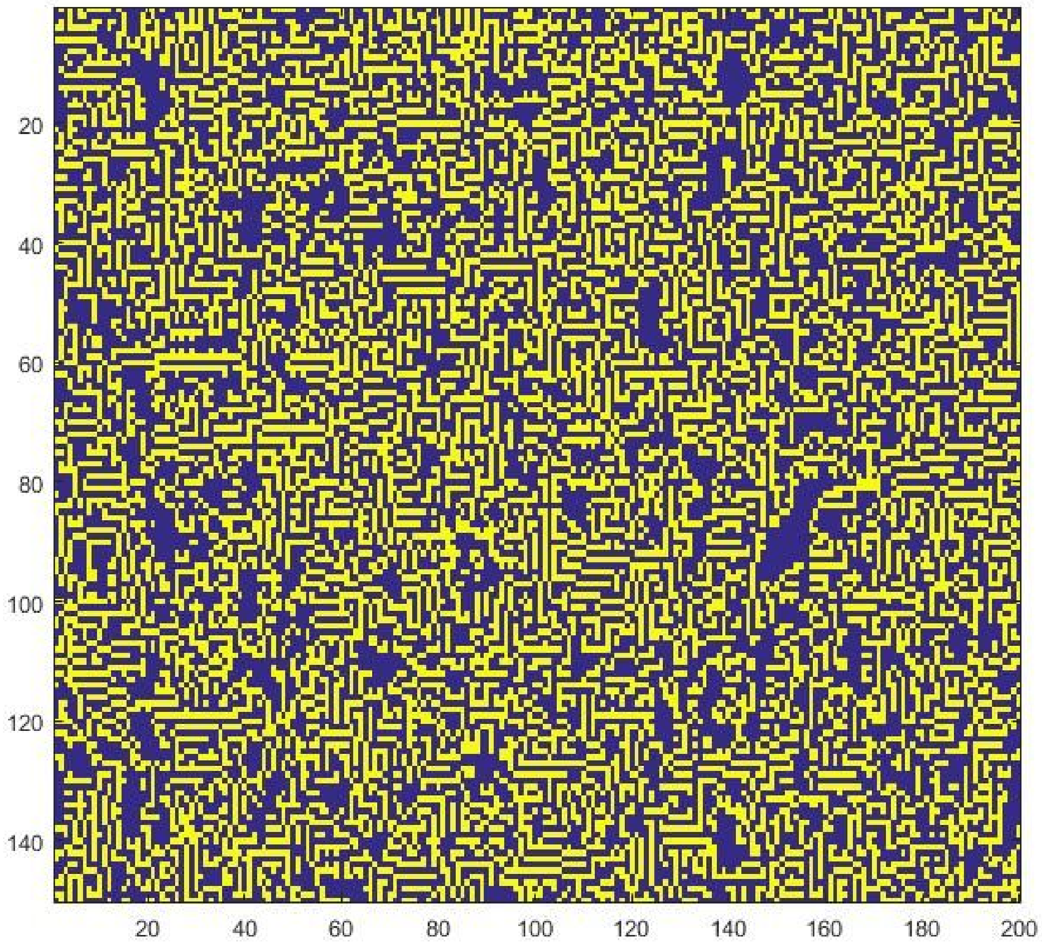

Figure 1.3.2: final distribution of living cells on grid

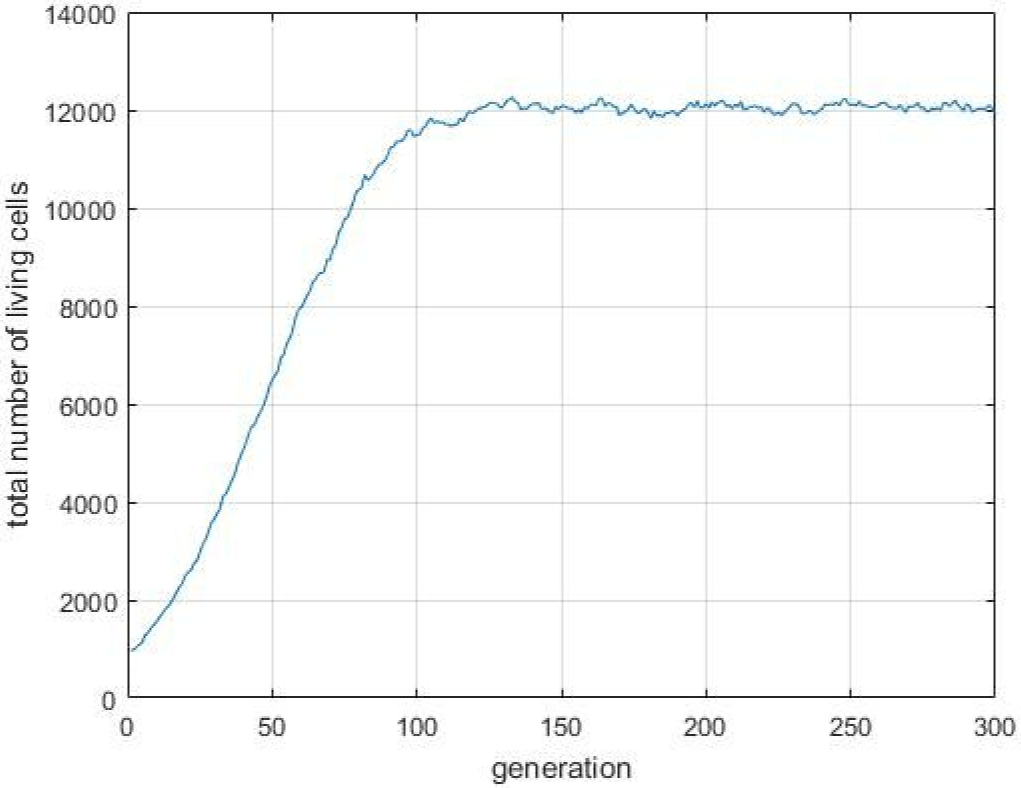

Figure 1.3.3: a graph of the number of living cells with each generation

Discussion 1.4:

After a few trials a clear asymptotic pattern can be seen from the graphs of the total number of living cells versus generation. The total number reaches a stable number. This number is roughly 26% of the total array area for the initial value given. As long as the ratio of initial value living cells is kept below 50% this trend seems to persist, however, the higher the initial ratio of living cells is, the more erratic the simulation becomes. Under the normal initial parameters, some other patterns can be seen in the final results. The organization of the living cells appear in alleys and road like structures. The final results look like a maze. This is a fitting structure which allows more cells to propagate without clustering and killing each other off.

Introduction 2.1:

This problem uses a numerical method called to solve the euler beam bending problem.

Models and Methods 2.2:

Calculations and Results 2.3:

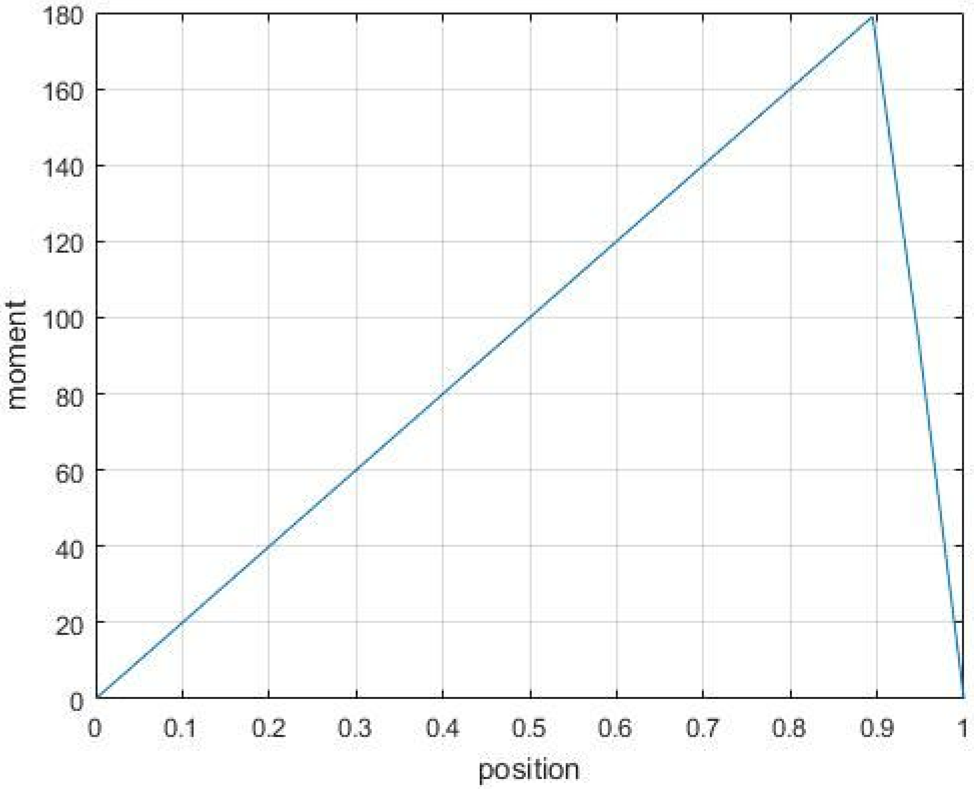

Figure 2.3.1: moment versus position on beam

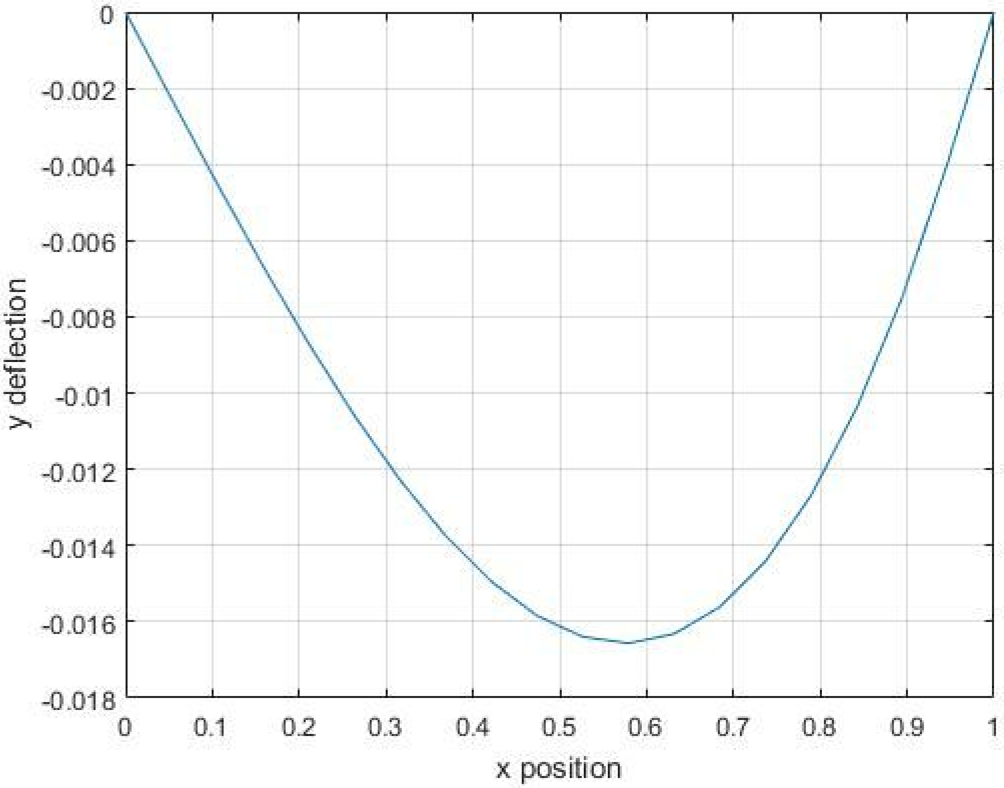

Figure 2.3.2: deflection versus position on beam

Discussion 2.4:

It works..