Sebastian Straszak, s1728659

Modelling and Visualization, Checkpoint 2

Contents

1 The Game of Life

1.1 Q1.1

To run the simulation code, simply execute

```
python3 twod_gol.py
```

in terminal, and it should run fine- this is the dressed-down version: any and all the file-based operations won't work on Unix since they're made for Windows (though my uncle taught me about os.join so I'll try to put that in, in the future.)

If you want to run the simulation with FuncAnimate, the functionality is coded up- it just will only work on Unix: it works up to native framerates when it does, though.

Some of the patterns included are larger than 50x50- there is no exception catcher for this, so I recommend lattices around 1000x1000 if you're running anything that you don't recognize.

1.2 Q1.2

See Figure 1 for the histogram relevant to this.

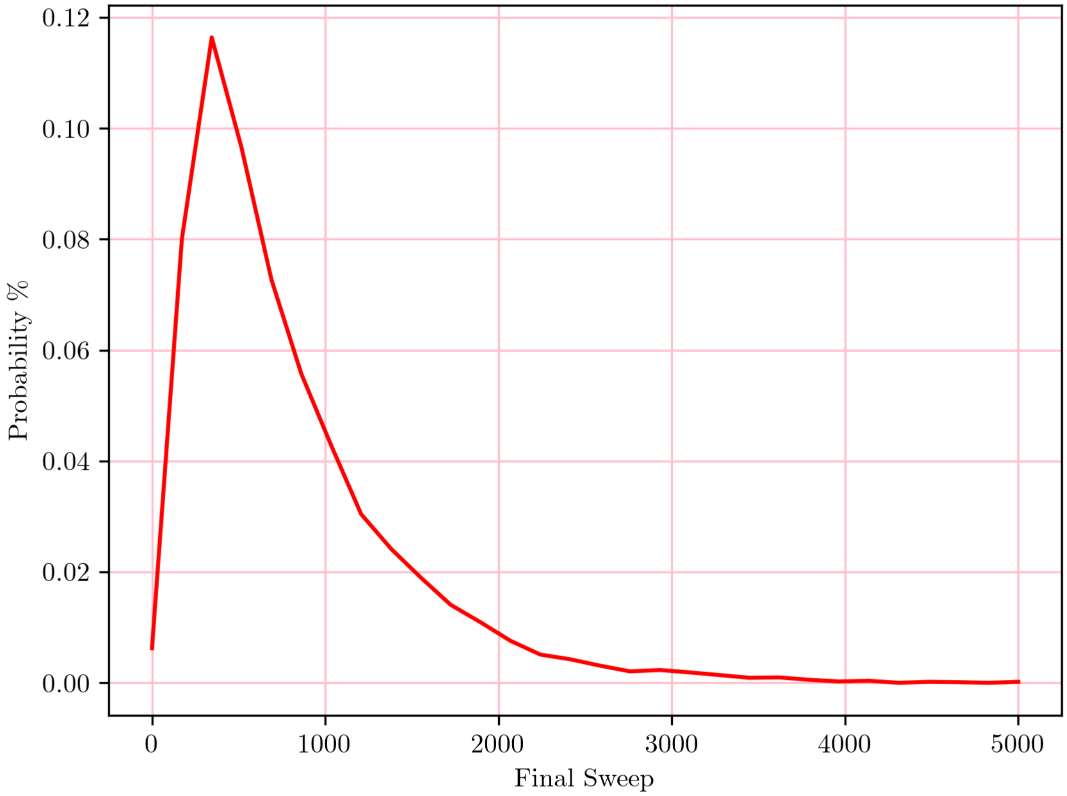

Figure 1: Distribution of times needed to equilibrate, with a few hundred simulations used to get the histogram.

1.3 Q1.3

See Figure 2 to see the demonstration of glider speed calculation functionality.

The code for this will correctly calculate the speeds (speed according to the Life Wiki- they have an odd definition of it that uses x/y displacement rather than absolute distances) of any glider of length/width at most half the size of the lattice, and accounts for periodic boundary conditions.[1]

Centre of mass is calculated by rolling the glider such that, rather than crossing boundaries, it lies entirely within the lattice: the roll is then accounted for in calculating the centre of mass coordinate.

[1] I have tested this and gotten speeds that agree with the Life wiki for a few gliders, including the default glider- hence why I've dumped their definition in since it's easier to verify the code works that way.

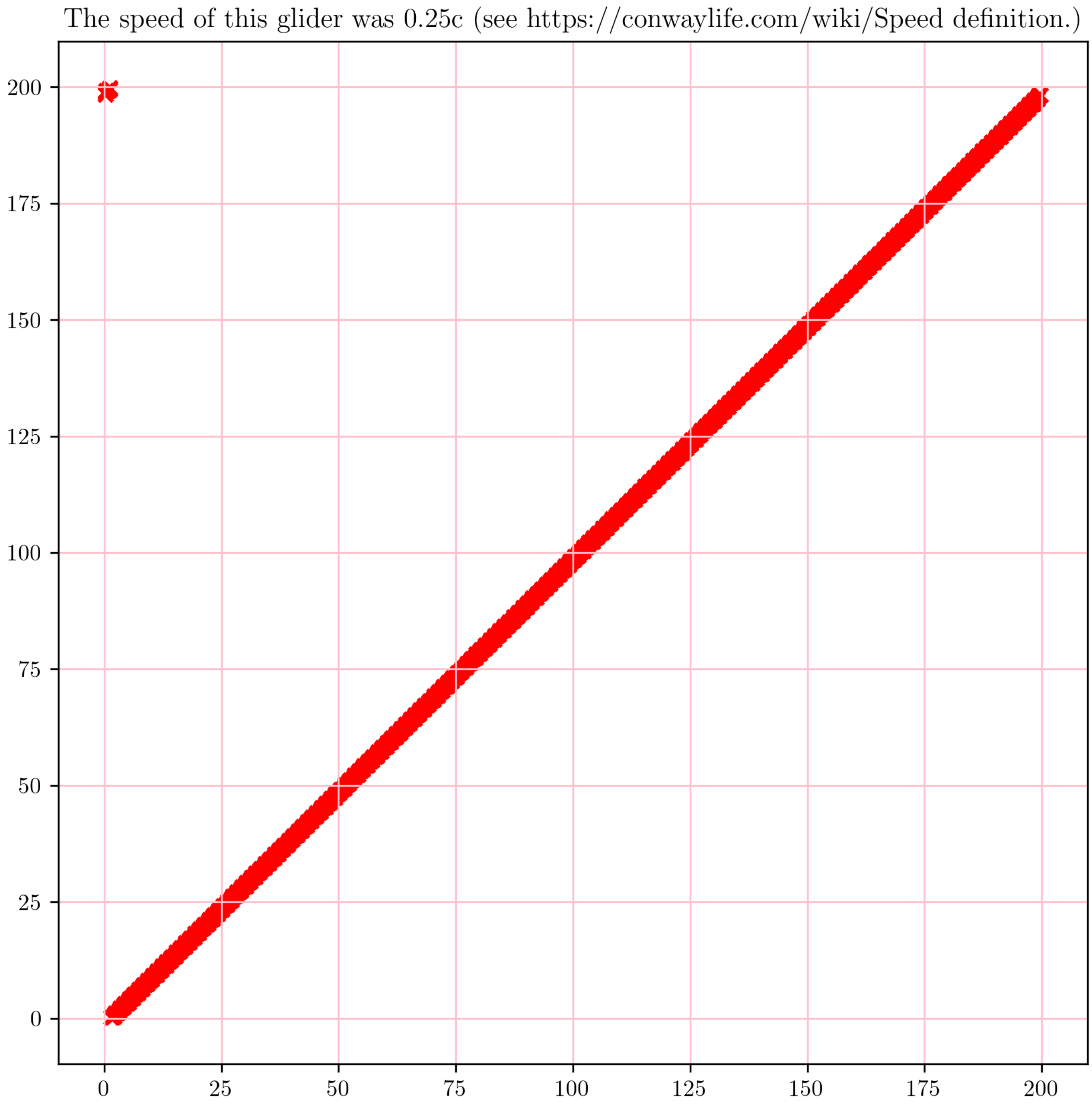

Figure 2: Figure demonstrating functionality to calculate the speed of gliders. Speed is quoted in “c” which is a measure of cells per sweep. This was done for the regular glider on a 200^2 lattice.

2 The SIRS Model

2.1 Definitions

I am going to be using ψ as the "infection fraction" for the lattice- 1 being entirely infected, 0 not-infected. Given the parallels to the Ising model in her variance, I am going to use χ to represent the variance in ψ (it just makes sense I hope...)

As previously, see

```
    datafile.hdf5
```

for the datafiles for each individual part of this checkpoint with the SIRS model (i.e. part 3,4,5.)

2.2 Q2.1

To run the simulation code, simply execute

```
    python3 twod_sirs.py
```

in terminal, and it should run fine- this is the dressed-down version: any and all the file-based operations won't work on Unix since they're made for Windows (though my uncle taught me about os.join so I'll try to put that in, in the future.)

2.3 Q2.2

See Figures 3,4,5 for wavelike, near-equilibrium and absorbing state behaviours. Probabilities that led to them are displayed in the captions (though the titles of the graphs take the form

```
lx_p1_p2_p3...
```

and could be used to infer this, more specifically.)

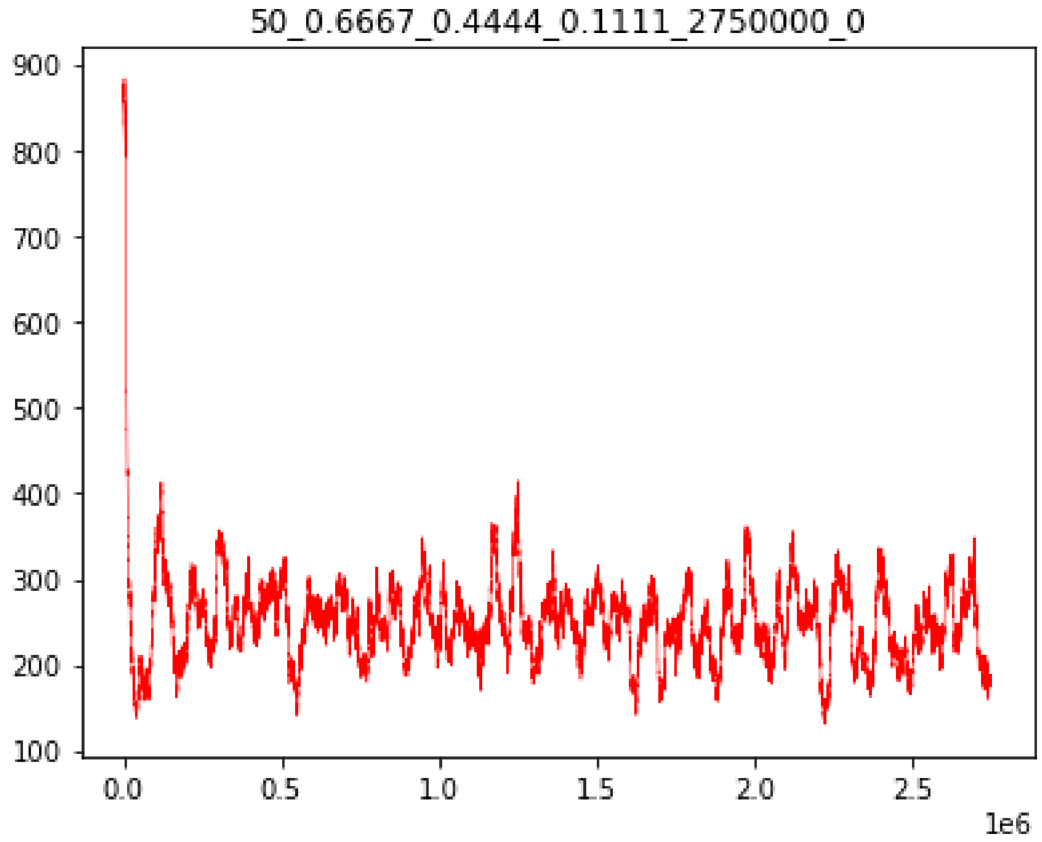

Figure 3: Wavelike behaviour in the SIRS Model for $p_1, p_2, p_3 = 0.67, 0.44, 0.11$. I've manually searched a sleuth of parameter sets and this one seems somewhat wavelike/reasonable. The y-axis is the total number infected (not an average/etc- the total infected count) while the x-axis is the flip (not sweep) since the inception of the simulation.

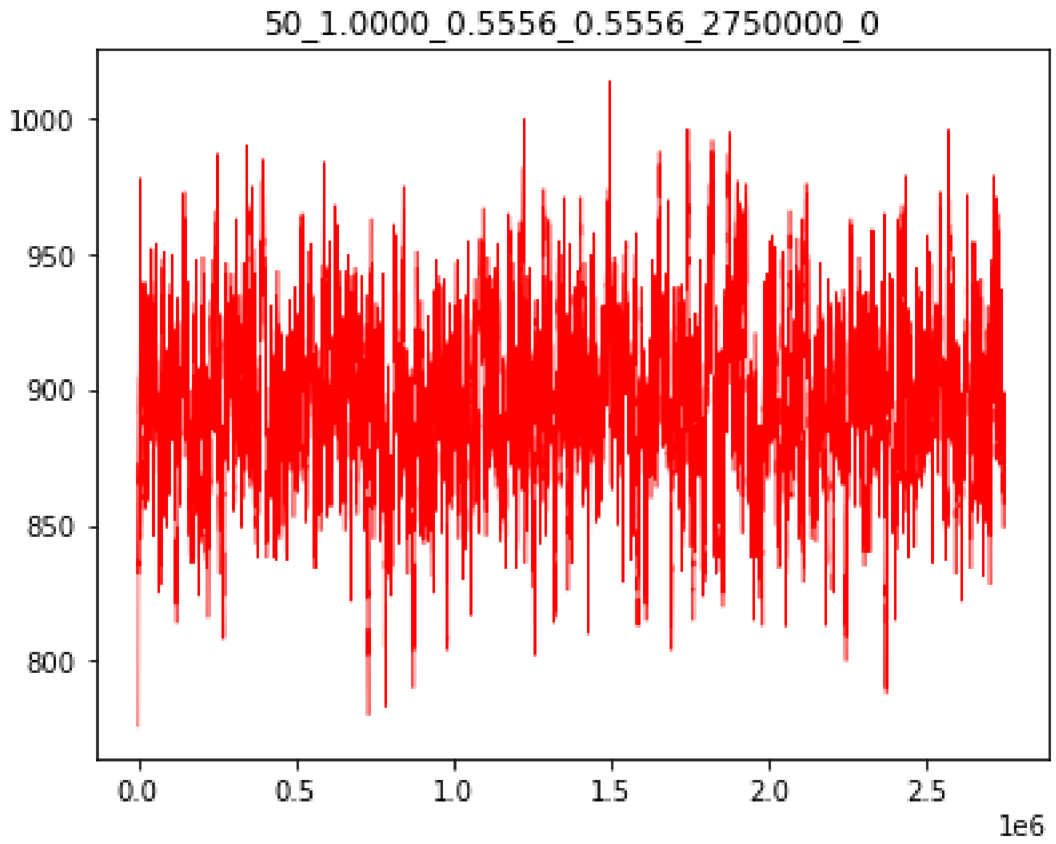

Figure 4: A state nearing dynamic equilibrium with $p_1, p_2, p_3 = 1, 0.56, 0.56$. The y-axis is the total number infected (not an average/etc- the total infected count) while the x-axis is the flip (not sweep) since the inception of the simulation.

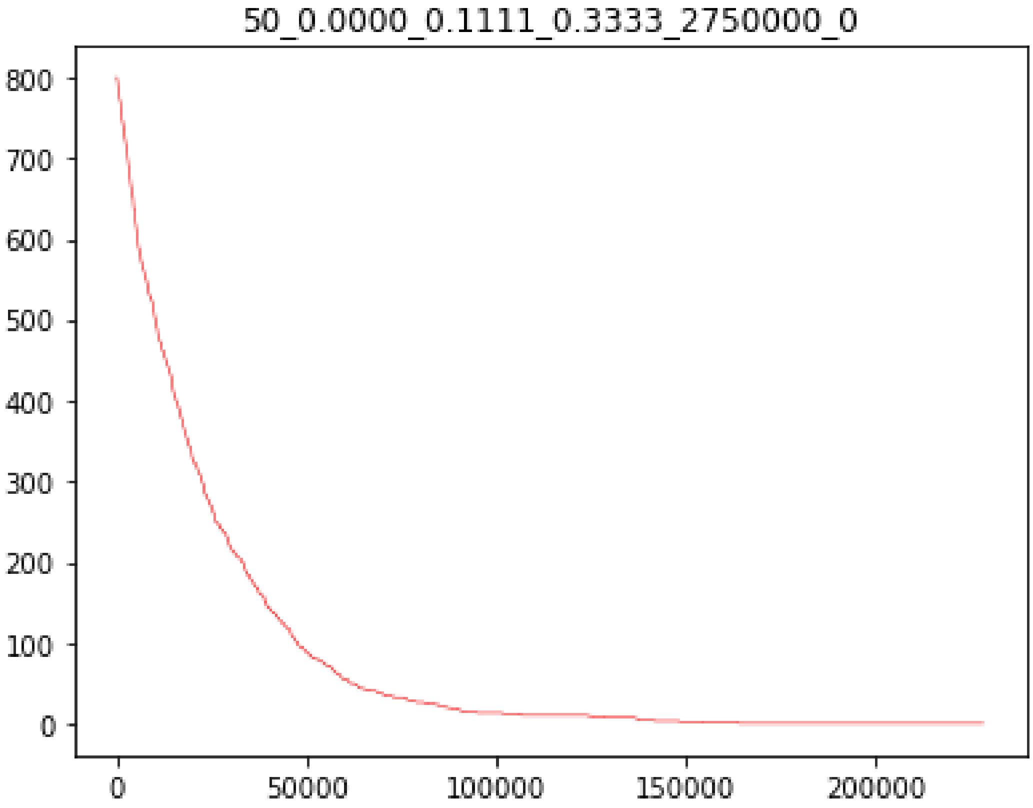

Figure 5: Absorbing behaviour in the SIRS Model for $p_1, p_2, p_3 = 0, 0.11, 0.33$. The y-axis is the total number infected (not an average/etc- the total infected count) while the x-axis is the flip (not sweep) since the inception of the simulation.

2.4 Q2.3

See Figures 6 and 7 for the average infection fraction $\langle\psi\rangle$ and her variance $\langle\chi\rangle$ for the requested probability domains.

In this case, the plotted values of $\langle\psi\rangle$ and $\langle\chi\rangle$ are averaged from the averages obtained from five independent runs, these averages obtained via the exact same bootstrapping method as in the previous checkpoint (with autocorrelation of 10 sweeps.)

The exact method by which these independent runs were combined is as follows, where i denotes the run i...

$$\langle\psi\rangle = \frac{1}{N}\sum_i \langle\psi\rangle_i \tag{1}$$

$$\sigma^2_{\langle\psi\rangle} = \frac{1}{N}\sqrt{\sum_i \sigma^2_{\langle\psi\rangle i}} \tag{2}$$

where the second line (the error^2/the variance) follows via propagation of errors.

Examining Figure 7, the region where waves form is clear as the boundary between the absorbing state ($\langle\psi\rangle = 0$) and the regions of dynamic equilibrium in the upper-right of the diagram, with the boundary having the highest variance in $\langle\psi\rangle$ over the course of the simulation.

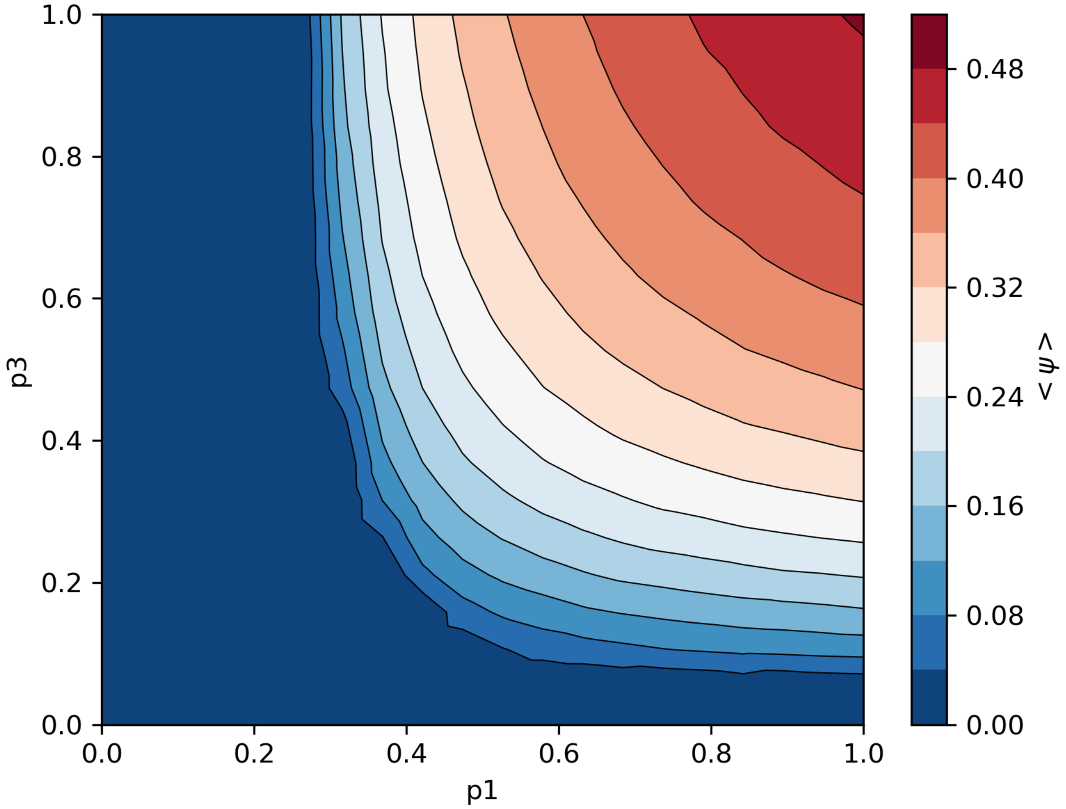

Figure 6: Contour plot/phase diagram for $p_1, p_3 \epsilon [0, 1]$ with $p_2 = 0.5$ at intervals of $\Delta p = 0.05$, for average infection fraction.

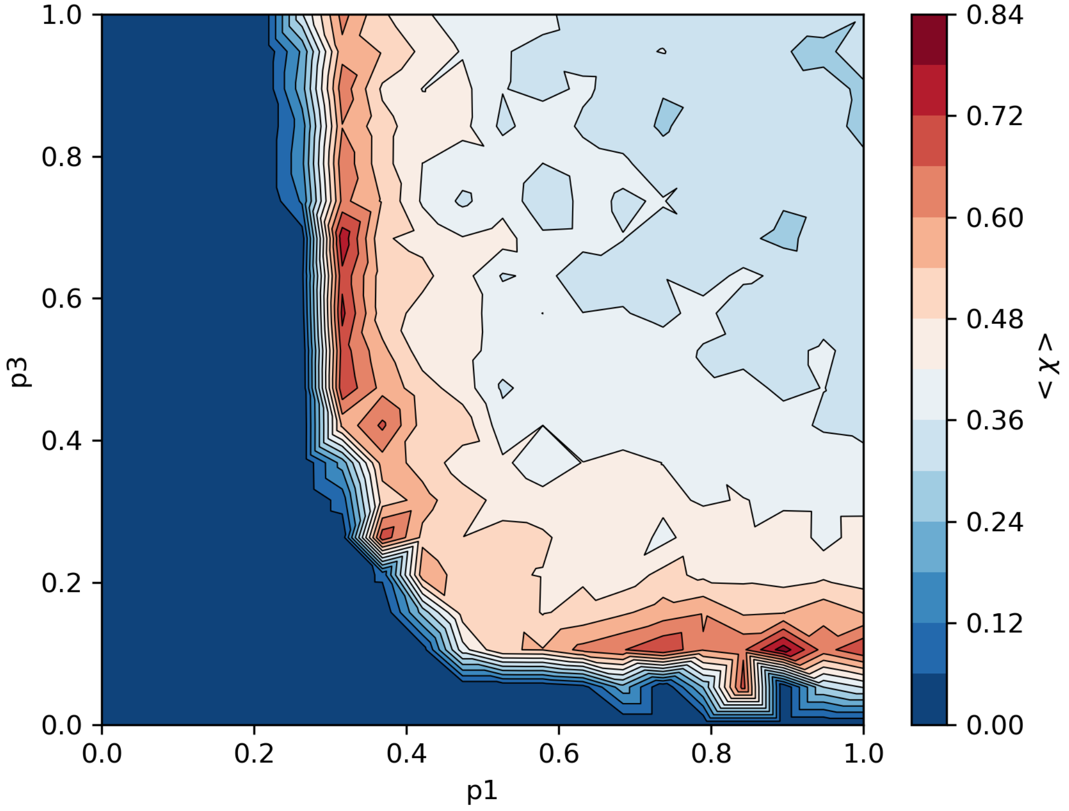

Figure 7: Contour plot/phase diagram for $p_1, p_3 \epsilon [0, 1]$ with $p_2 = 0.5$ at intervals of $\Delta p = 0.05$, for variance in average infection fraction (analogue to susceptibility in the Ising model.)

2.5 Q2.4

See Figure 8 for the plot of χ, the variance in ψ, for the requested probability parameters. The transition from the absorbing state and the state of dynamic equilibrium is evident for this particular slice, being somewhere around 0.308 (just guessing by eye and not coding anything to estimate it.)

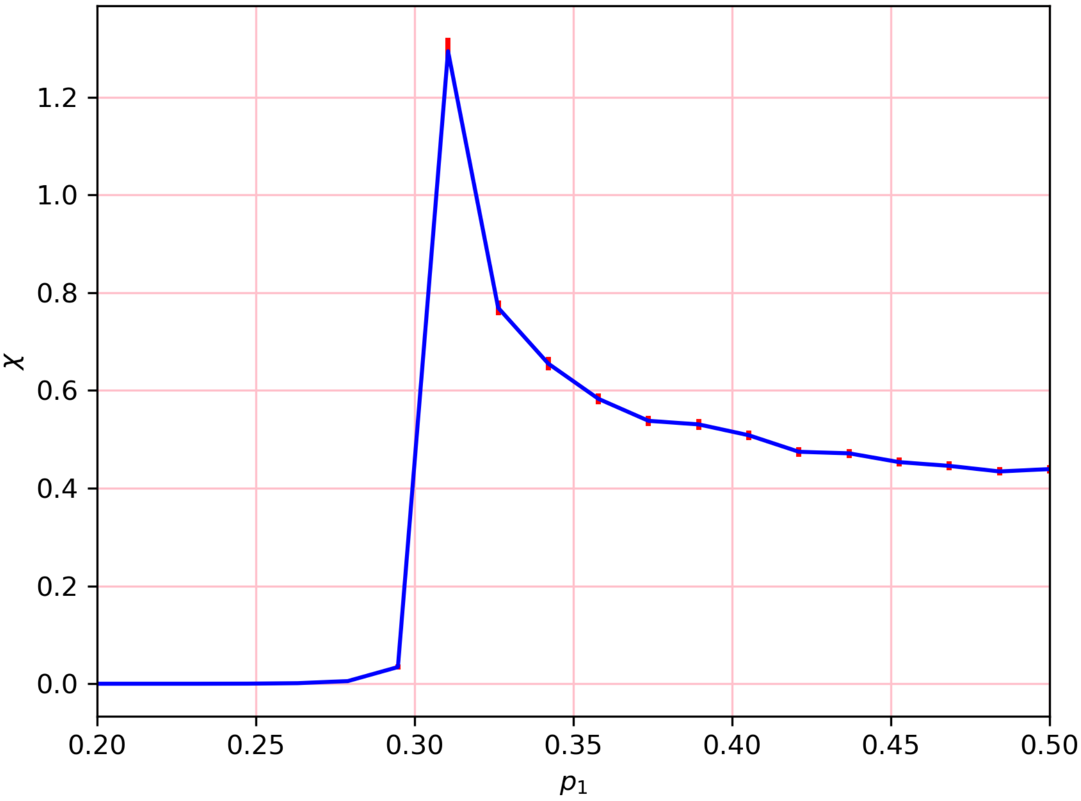

Figure 8: Plot of χ and her error for $p_2 = p_3 = 0.5$ with $p_1 \epsilon [0.2, 0.5]$. The cross-over point between dynamic equilibrium and the absorbing state, where wave-like behaviour is observed, is evidenced by the peak. This was done for 10,000 sweeps (with an equilibration of 100 sweeps) as requested.

2.6 Q2.5

See Figure 9 for the average infection fraction as a function of immunisation fraction (permanently immune.) Somewhere around $R = 0.4$ seems to be a fair requirement for all $p_i = 0.5$ to prevent infection spread. The averages have been done, as previously, over five independent runs.

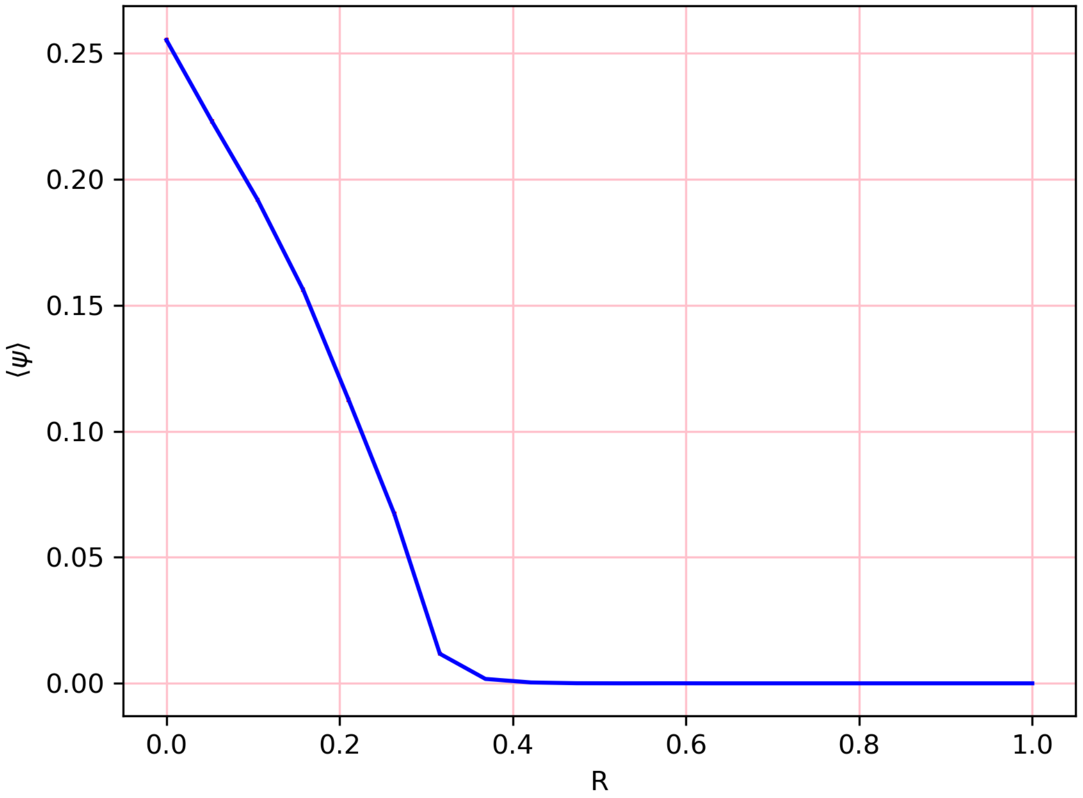

Figure 9: Plot of average infection fraction vs. recovery ratio for all $p_i = 0.5$ for all requested parameters in the checkpoint. Errors are illustrated (but appear negligible for this calculation... though that might be a problem on my part.