Neural Computation Assignment
Synaptic Depression
Module code: #[REDACTED]

Student ID: [REDACTED]

Reece Hill | April 2022

www.github.com/reecehill

Question 1 (5 points) *Before we use periodic signals, we stimulate the neuron with a constant external current. Simulate a single integrate-and-fire neuron for about a second. Plot the relation between the firing rate in Hz and the external input current.*

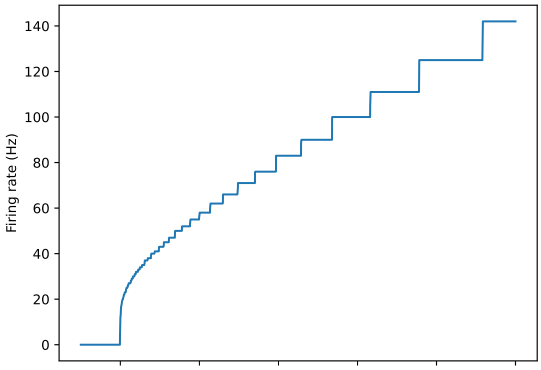

Figure 1: A constant, external current, I_{ext}, is applied to a neurone. The graph shows the relationship between the current applied and the neurone's response.

A single neurone is instantiated using an integrate and fire model written in Python. The neurone will "fire" when the membrane's voltage, V_m, is greater than or equal to the specified threshold, V_{thr} (10mV). At which point, V_m is increased for one timestep to replicate an action potential. The voltage of an action potential, V_{AP}, was not provided in the question; therefore, 40mV was chosen for all simulations to reflect physiology (Chen & Lui, 2022). Other characteristics of the neurone were set according to the question and are as follows. For the timestep that follows an action potential, the voltage is set to 0mV – the reset potential, V_{reset}. Resistance of the neurone's membrane, R_m: 1MΩ. The membrane voltage of the neurone at rest, V_{rest}: 0mV. The membrane's time constant, τ: 10ms. The capacitance of the neurone's membrane, C_m: $\frac{\tau}{R_m}$. Each simulation was run for 1 second, in 1ms timesteps.

To produce Figure 1, we injected a constant current, I_{ext}, into the neurone for the entire duration of the simulation. This simulation was repeated for 1,000 values of I_{ext}, linearly spaced between 0nA and 20nA. The stepped line is an artefact of how firing rate is determined. Briefly, action potentials (or "spikes") per second are counted for each simulation. Therefore, despite an increase in I_{ext} from one simulation to the next, the distance between spikes may reduce but not enough such that an additional spike occurs. For completeness, we demonstrate this by plotting (in Figure 2, below) two time courses of V_m: one for $I_{ext} = 18nA$, and another for $I_{ext} = 19nA$.

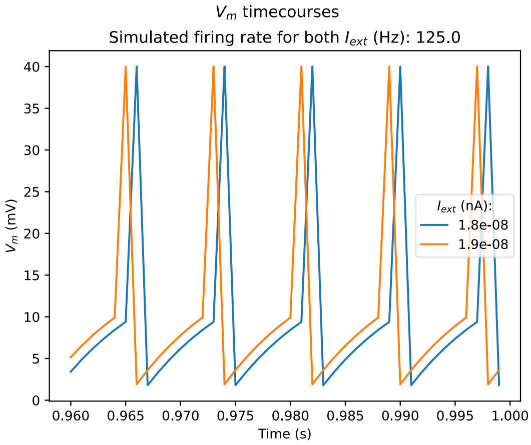

Figure 2: Note that the x-axis does not begin at zero. By increasing I_{ext} (blue to orange trace) the final action potential occurs earlier in the timeseries; the datapoints are transformed in the x-direction. This would yield a higher firing rate if a continuous counting method were used. However, Figure 1 counts (discretely) the number of spikes per second to determine frequency per second.

Figure 1 shows there to be no response in firing activity for $I_{ext} < 10$nA. Intuitively, and with inspection of V_m time courses, we know that the neurone's voltage membrane is less than V_{thr}. Upon meeting this threshold, we observe a steep increase in firing rate for each incremental increase of I_{ext}. This behaviour is momentary, and the steepness reduces after ~10.02nA. Hereafter, the relationship between I_{ext} and firing rate appears linear. A smoothed line could be acquired by the interpolation of fewer datapoints, or by deriving firing rate in a way that is continuous.

Question 2 (5 points) *Simulation for 200 milliseconds. Bin the spike times averaged over the population in 2ms bins. Plot this population rate over time. Make sure to use the correct units and normalization. The mean firing rate should be around 40Hz.*

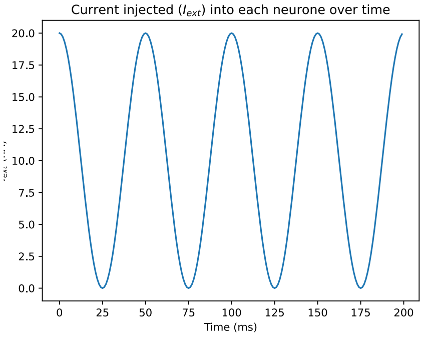

Figure 3: The current injected into each neurone, I_{ext}, oscillates.

Here, we simulate 100 identical neurones (i.e., a population). All neurones are independent of each other; however, the population is homogeneous. The parameters for each simulation/neurone are the same as that used in **Question 1**, with some alterations. The total time to be simulated is reduced to 200ms. Notably, I_{ext} is no longer constant, but oscillates (Figure 3) and takes the form:

$$I_{ext}(t) = \frac{1}{2} I_0 [1 + \cos 2\pi f(t)]$$

$$Where: I_0 = 20nA \text{ } and \text{ } f = 20Hz$$

Given that all neurones in this simulation share the same parameters, we yield a system that behaves identically to a single neurone, just scaled by the size of the neuronal population. Thus, we begin with Figure 4, that shows the average number of spikes per neurone over time. The data is collected into 2ms bins, represented by bar widths of equivalent size. From this graph, we could derive the population's firing activity by multiplying the mean activity per neurone (i.e., 1.0) by the population size (i.e., 100) – provided the population remains homogeneous.

So it can later be referenced in **Question 3**, we also present the raster plot of the population (Figure 5). It also serves as evidence that individual neurones were simulated (as opposed to just scaling by population size, as previously discussed).

Finally, we conclude with the population rate over time (Figure 6), as requested. The same method to bin the data was used here as in Figure 4.

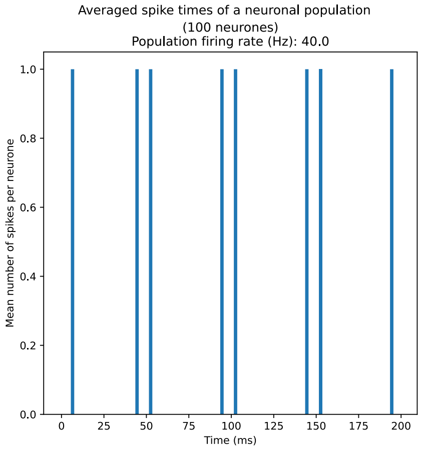

Figure 4: Time course of mean neuronal activity. (See text for parameters)

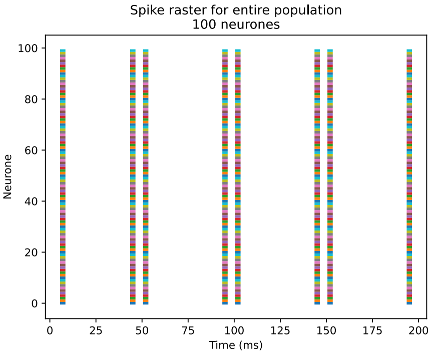

Figure 5: Spike raster/event plot of 100 neurones. (See text for parameters)

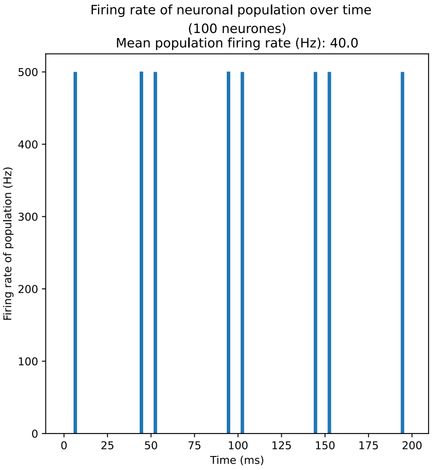

Figure 6: Time course of mean population activity. (See text for parameters)

The data shows all neurones to be firing in unison in response to the oscillating I_{ext}. Each period of excitation leads to two action potentials; this could be interpreted as modelling bursting behaviour – types of neuronal firing patterns – of the neurones. However, it is more an effect of the oscillating current, best shown in Figure 7.

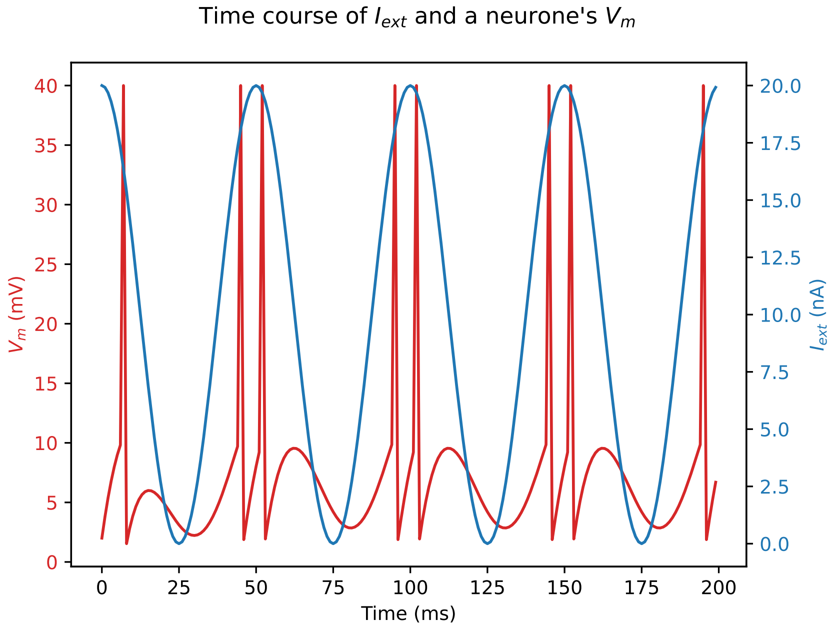

Figure 7: Time course of V_m and I_{ext}. (See question for parameters)

The population response does not exactly reflect the input current. This can be undesirable, when the neurons, say, drive motor actions. It can be improved by, for instance, introducing heterogeneity to the population. For instance, the input resistances can be drawn from a Gaussian distribution around 1MΩ. (Although you might want to ensure that the value is always positive). The spread of the distribution can be easiest characterized by the Coefficient of Variation (CV) defined as standard deviation/mean. Implement heterogeneity in your population. Be careful with the difference between '' and '.*' in Matlab/Octave!*

Question 3 (5 points) *Fit the time course of the firing rate to $g(t) = A + B\cos(2\pi f(t) + \varphi)$, with parameters A, B and φ. (E.g. using octave's nonlin_curvefit). Plot the mean square error in the fit as you vary the heterogeneity, i.e. versus the CV of the input resistance.*

Further inspecting Figure 7 above, we see that the activity of V_m does not align perfectly with an incoming stimulus. That is, for every increase in I_{ext}, there is also an increase in V_m, but it is either delayed or not proportional.

As prompted, we now vary the properties of the neurones within the population to improve this. By introducing heterogeneity, we now observe a system of neurones that do not fire in unison. To show this, and for comparison against Figure 5, we present a raster plot, Figure 8, of a population whose neurones' R_m are first sampled from a Gaussian distribution (mean: 1MΩ, standard deviation: 5MΩ), and then any negative values are changed to zero. The neurones are sorted by their R_m in ascending order for presentation.

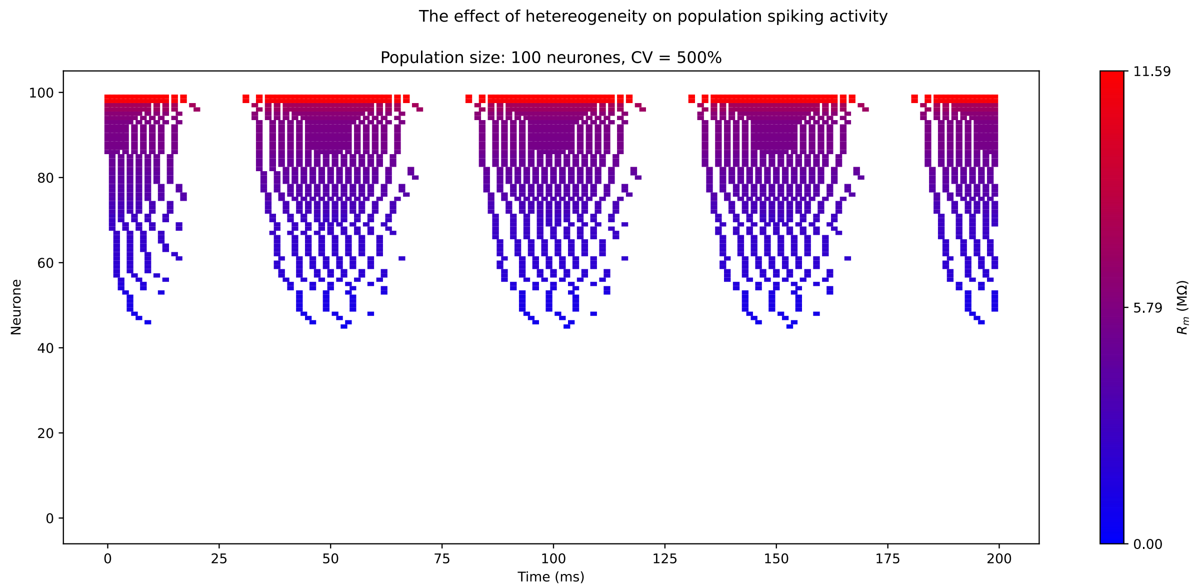

Figure 8: A raster plot showing the effect of heterogeneity (varying R_m of each neurone within a population) on firing activity. Each box represents spiking activity from that neurone. Parameters of the neurones, such as τ, are as defined in Question 2. R_m is varied (see text). The mean neural activity from 5 seeds is used. Coefficient of Variation = CV.

Moving down the y-axis of Figure 8, from the 100th neurone onwards, we note that some neurones fire rapidly, some less so, and the rest not at all (approximately 45% of this population). The proportions of activity are not fixed and are affected by the chosen Co-efficient of Variation (CV). Given that we have fixed the Gaussian distribution's mean, smaller CVs will yield a smaller standard deviation. Thus, the number of neurones assigned an R_m incompatible with spiking behaviour in response to the current regime (i.e., I_{ext}) is reduced.

With heterogeneity evidenced, we move to fitting our data to $g(t) = A + B\,cos(2\pi f t + \varphi)$.

Put briefly, A will translate the oscillating curve in the y-direction, φ in the x-direction, and B is a stretch along the y-axis. Or, A is the mean firing rate of the population, B is the amplitude, and φ is a time offset. Intuitively, we begin with approximated initial values for A, B and φ that are dependant on the simulated population firing rate.

As an aside, we also experimented with curve fitting that started with initial parameters that were fixed for all percentages of CVs. Interestingly, and shown in Figure 9, fixing the initial parameters for all CVs had a negligible effect on the loss function – mean-squared error (MSE). Regardless, in our discussion we vary our initial parameters according to the data we are trying to fit – as is standard.

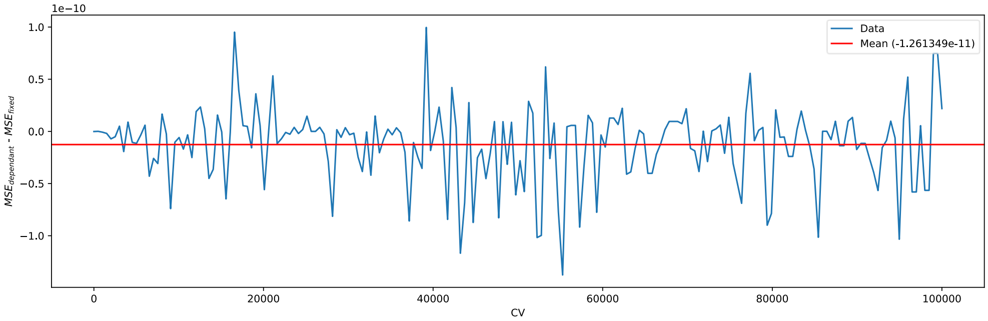

Figure 9: A graph showing the difference in mean-squared error (MSE) achieved by fitting $g(t)$ to the population's firing rate over time for initial parameters that are fixed (MSE_{fixed}) and simulation-dependant ($MSE_{dependant}$). Note the magnitude of the y-axis. CV: Co-efficient of Variance. Parameters unchanged from Figure 8, but CV is sampled 200 times from 0 to 1e5. Fixed parameters: $\{A: 65/16, B: 95/128,\ \varphi: 3/16\}$.

From basic mathematics, to fit the time course of firing activity to $g(t)$ we assign the initial parameters as per Table 1 (see also, Figure 10). The time offset, φ, was a challenge to derive, but from graphing it was clear it could be small, and many possible solutions could exist.

$A_{initial} = \dfrac{Population\ Firing\ Rate\ (Hz)}{Population\ Size}$
$B_{initial} = Maximum\ Population\ Firing\ Rate\ (Hz) - \dfrac{Population\ Firing\ Rate\ (Hz)}{Population\ Size}$
$\varphi_{initial} = Size\ of\ each\ timestep\ in\ simulation\ (s)$

Table 1: The initial parameters used for fitting the curve changed for each population. Table shows how A, B and φ were derived.

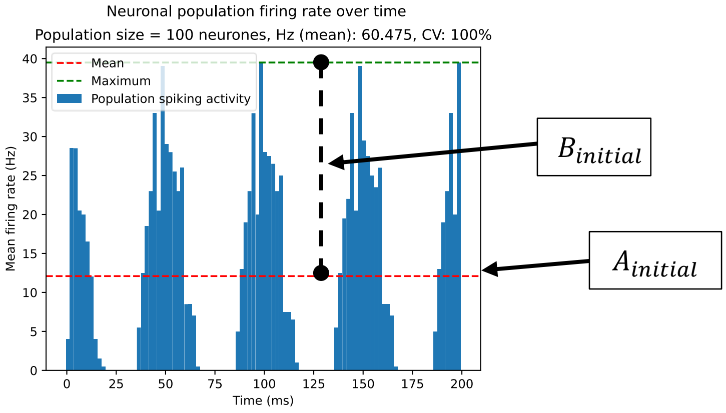

Figure 10: Timeseries of population firing activity. Overlaid are annotations that show how $A_{initial}$ and $B_{initial}$ are derived.

From the initial parameters for $g(t)$, we yield a curve (Figure 11, solid purple) that approximates the population's firing rate over time. By using the *curve_fit()* function in the Python library, SciPy, we are able to use this curve as a starting point from which, through non-linear least squares fitting, we can derive another curve that is a supposed better fit of our data (Figure 11, dashed green).

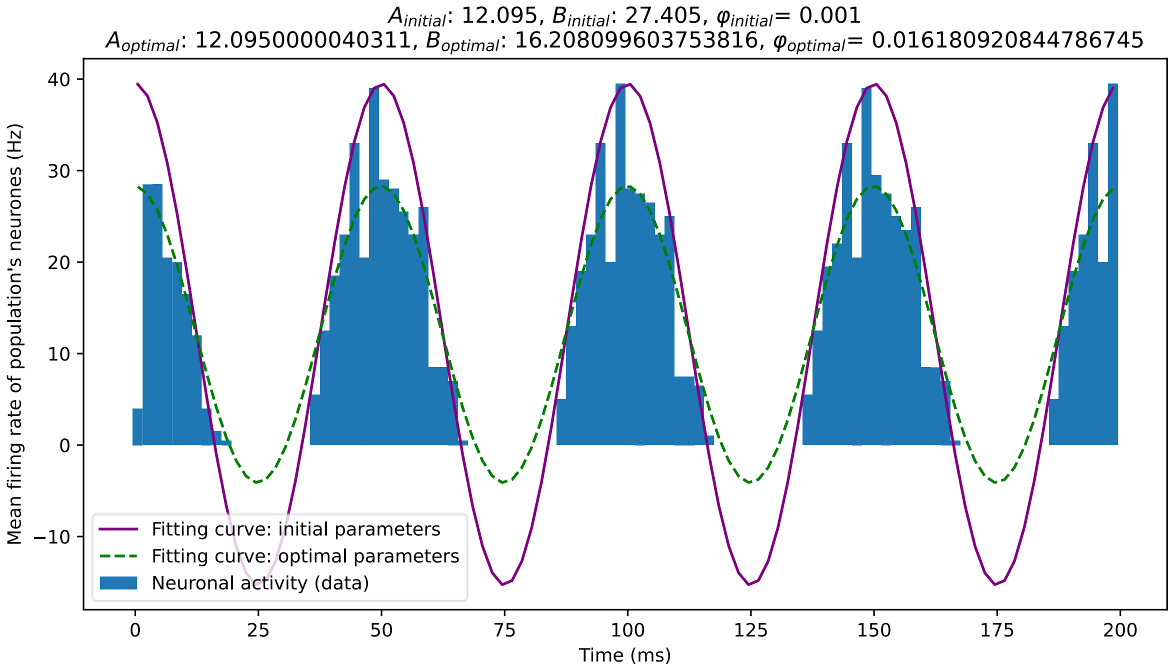

Figure 11: The time course of the neuronal population's firing activity is plotted, and overlaid with g(t) using initial (purple) and optimal (green) parameters.

The MSE of the optimal curve is an indication of how well the curve fits the neuronal activity data. Hence, and to conclude this question's response, we vary CV (of the population's R_m) and plot the MSE as we do so (Figure 12).

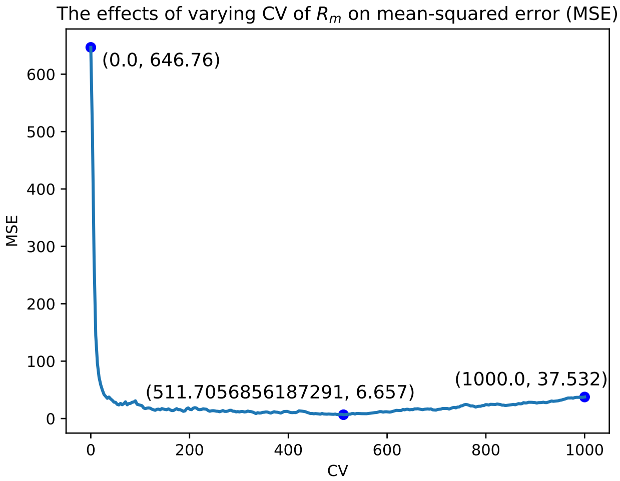

Figure 12: Graph showing MSE to decrease for small perturbations of CV (R_m) from zero. Then, it appears to steadily rise again as CV is further increased. Three annotations are included: the first and last datapoints, and the value of minimum MSE. Parameters are unchanged from Figure 8, with some exceptions. CV: 300 values between 0 and 1e3 are plotted.

Figure 12 shows a rapid decline in MSE as CV moves away from zero, followed by a positive gradient in MSE thereafter. This range of CV was chosen to emphasise the effect of CV upon MSE when it is small. It should be noted that the position of MSE_{min} is only approximate – an increase in granularity (i.e., by sampling more CVs) will yield a different value. Here, we show a value of 6.657. However, by varying CV further, and exploring at larger magnitudes, we present the landscape of CV, showing an eventual plateau of MSE at 97 (Figure 13).

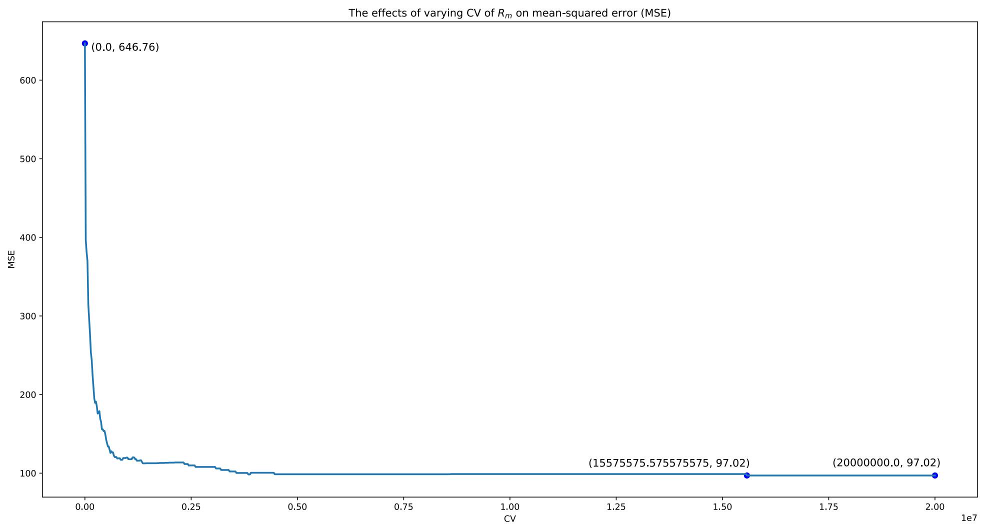

Figure 13: Graph showing MSE against CV. Three annotations are included: the first and last datapoints, and the value of minimum MSE identified by this simulation. Parameters are unchanged from Figure 12, with one exception: CV: 1,000 values between 0 and 2e7 are plotted.

It is possible that our identified MSE_{min} is one that is only local. However, to determine this without mathematical analysis would demand even greater granularity and, thus, computing power.

Question 4 (5 points) *Examine also the heterogeneity in membrane capacitance and threshold voltage. Compare your findings with the previous question.*

Next, we vary the CV for other properties of the population's neurones and plot the resulting MSE.

Vary: V_{thr}
Fixed: R_m, C_m, τ

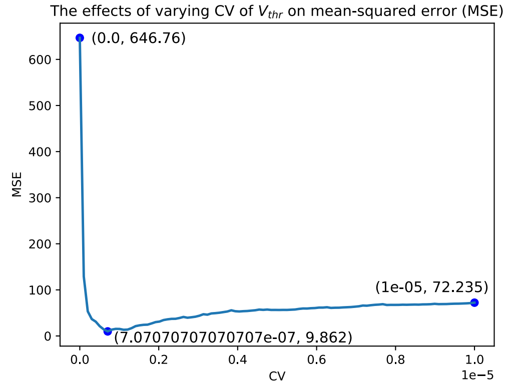

Figure 14: Graph showing MSE to decrease for small perturbations of CV (V_{thr}) from zero. Then, it appears to steadily rise again as CV is further increased. Three annotations are included: the first and last datapoints, and the value of minimum MSE. Parameters are unchanged from Figure 8, with some exceptions. CV: 300 values between 0 and 1e-5 are plotted.

Vary: C_m
***Fixed: R_m, V_{thr}, τ**

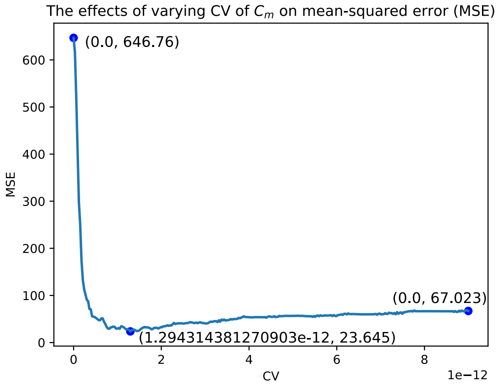

Figure 15: Graph showing MSE to decrease for small perturbations of CV (C_m) from zero. Then, it appears to steadily rise again as CV is further increased. Three annotations are included: the first and last datapoints, and the value of minimum MSE. Parameters are unchanged from Figure 8, with some exceptions. CV: 300 values between 0 and 1e-12 are plotted.

**For this simulation, we ensure $C_m = \frac{\tau}{R_m}$ is maintained. We set R_m =1MΩ and $\tau = C_m R_m$.*

From Figures 14 and 15, we can see that heterogeneity, even when in V_{thr} and C_m, leads to similar landscapes of MSE; a rapid decrease in MSE as CV moves from zero, then the MSE_{min}, followed by a gradual increase that leads to a plateau (although scaling is different). This shape was subtle when we varied the CV of R_m; but, it becomes clearer when the MSE_{min} of Figure 12 (6.657) is considered in the context of Figure 13.

For the alternative sources of heterogeneity presented here (Figures 14-15), the MSE_{min} appears for much lower values of CV; close to zero. Yet, they achieve an MSE of similar magnitude. This comparison is made with reservations, as due to the aforementioned issue of computing power, we are unable to conclude which type of heterogeneity yields the lowest MSE_{min} with certainty.

A question that follows in this discussion is the effect upon MSE if we were to introduce more than one source of heterogeneity into a neuronal population. Given that MSE is, as explained previously, a measure of how well $g(t)$ can fit the neuronal firing activity, we can conclude that if low MSE values are unobtainable then $g(t)$ is not an accurate model of said activity. Put simply, our neuronal population would not behave in a way that closely reflects its input, I_{ext}.

Theoretically, to find the lowest possible MSE value, we must implement heterogeneity into the neuronal population in a way that is optimal. This could include different CVs for R_m versus C_m, for example. As a final activity, we take the optimal CV values, where MSE_{min} were identified, for three neuronal properties. We set the CV values as follows: CV_{R_m}: 511.70…91%, $CV_{V_{thr}}$: 7.07…07e-7%, and CV_{C_m}: 1.29…03e-12 (refer to Figure 14 and 15 for untruncated values).

For illustrative purposes, the following figures represent an “optimised” neuronal population. Figure 16 shows the mean firing rate of the population in response to oscillating I_{ext}. We now see a population firing rate that closely follows $g(t)$.

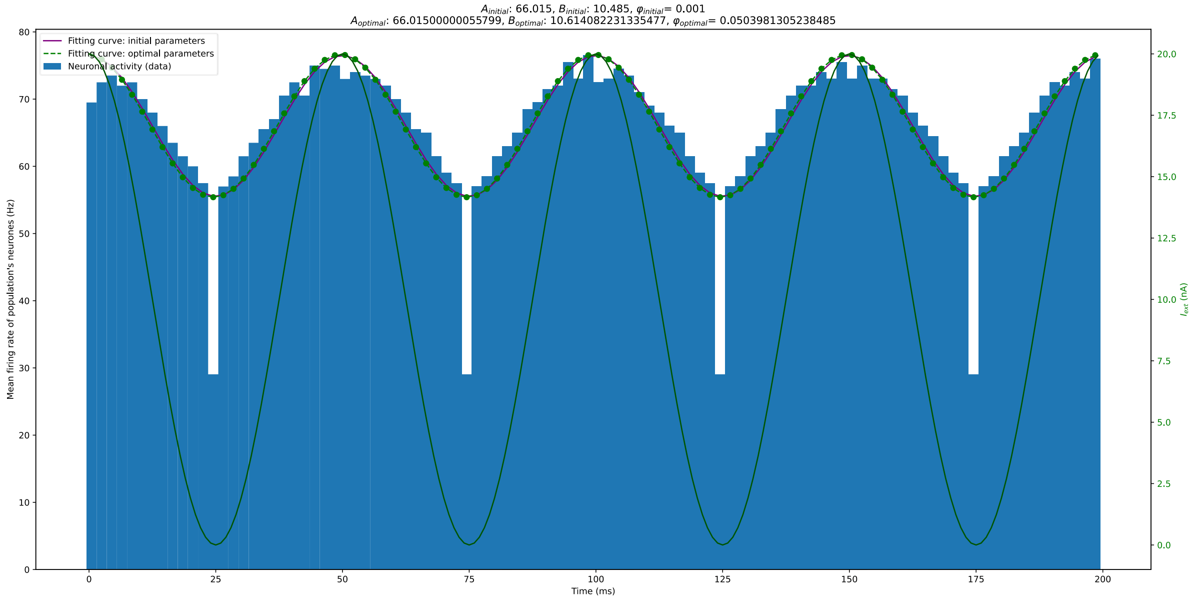

Figure 16: The time course of the firing activity of an optimal neuronal population is plotted, and overlaid with g(t) using initial (purple) and optimal (green) parameters.

Lastly, Figure 17 (below) shows four neurones and their different bursting patterns that have been generated by this “optimal” neuronal population. From this, we can deduce that the neuronal population is incredibly diverse, and now able to support a variety of behaviours such that it has greater manoeuvrability and is no longer limited to the entire population acting in unison (as we saw with the homogenous population).

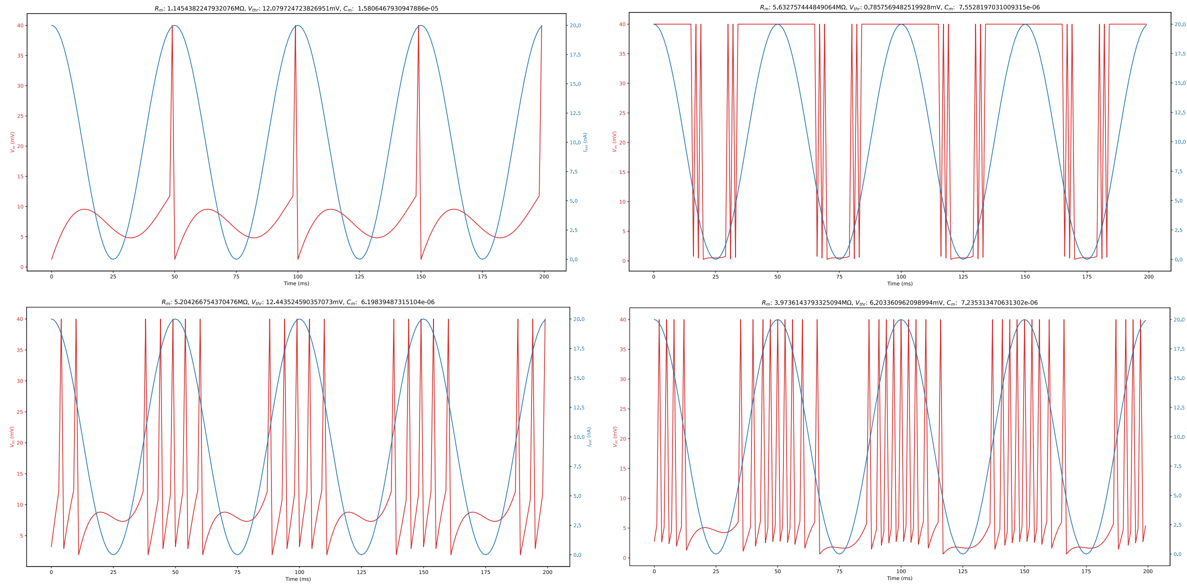

Figure 17: Time courses of V_m and I_{ext} for four different neurones taken from the optimal population.

References

Chen, I., & Lui, F. (2022). Neuroanatomy, Neuron Action Potential. In *StatPearls*. Treasure Island (FL).