Synaptic Maturation Project

J McAllister and C O'Donnell

1 Intro

The model follows the following framework, with 3 possible states and the transitions between them:

$$\text{Potential Synapse Pool} \underset{\text{elimination}}{\overset{\text{creation}}{\rightleftharpoons}} \text{Immature Population} \underset{\text{dematuration}}{\overset{\text{maturation}}{\rightleftharpoons}} \text{Mature Population}$$

For synapses in the Mature Population, we also model "synaptic size". When a synapse becomes mature, we initiate a size of 0, and then simulate the stochastic dynamics of the synaptic weight using a Kesten process

$$x_{t+1} = \epsilon_t x_t + \eta_t$$

where x_t is the synaptic size at time t and ϵ_t and η_t are random variables drawn from some distribution (we use Gaussian, with the mean of ϵ near 1 — or just below — and the mean of η near 0, to ensure a stationary distribution).

2 Model 1: Random Walks

We first model this setup using random walks (with Gillespie's algorithm) and predefined transition probabilities c, e, m, i, for creation, elimination, maturation and immaturation (dematuration) respectively. We set parameters such as potential synapse pool size, time, and Kesten process time step. When a synapse "dematures", we adopt a simplified approach by removing those synapses of size 0. This random walk model has the advantage that its stochastic nature captures the randomness inherent in biological processes, providing a realistic simulation of individual synapse behaviour. However, it is more computationally expensive and less analytically insightful (being reliant on simulations).

Below is an example simulation of this model, with parameters

```
total_time = 100.0
total_pool_size = 1000
c, m, e, i = 0.2,0.2,0.01,0.05
epsilon, eta = 1.0, 0.0
sigma_epsilon, sigma_eta = .5, .5
kesten_timestep = 0.01
```

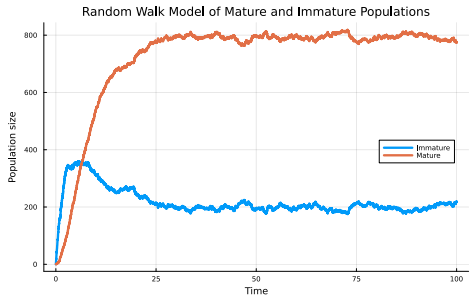


Figure 1: Single simulation of populations using Gillespie's algorithm

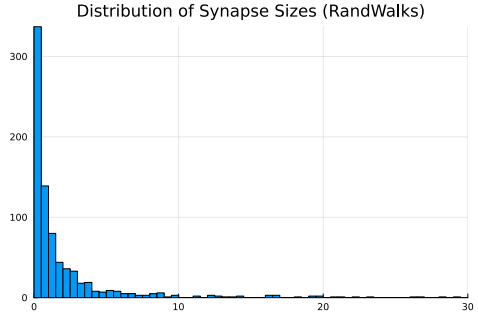


Figure 2: Final distribution of synaptic sizes

We can also keep track of the synapse sizes over time:

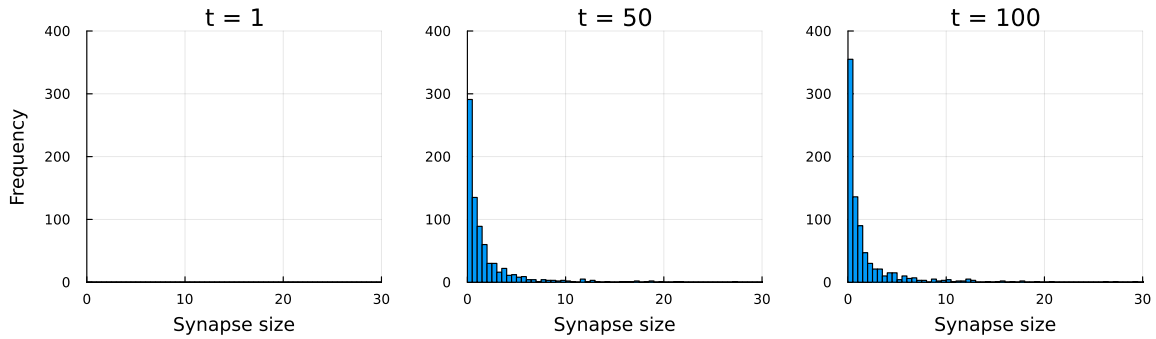


Figure 3: Evolution of the distribution of synaptic sizes over time.

3 Model 2: Differential Equations

3.1 Setup

We then defined a set of differential equations to model the setup in another way. The equations are as follows:

$$\frac{dN_I}{dt} = cN_P - (e+m)N_I + iN_M$$
$$\frac{dN_M}{dt} = mN_I - iN_M$$
$$\frac{dN_P}{dt} = eN_I - cN_P$$

where c, e, m, i are as before, and N_I, N_M, N_P denote the number of Immature, Mature, and Potential (pool) synapses. Again, when a synapse "dematures", we remove those synapses of size 0. This differential equations model preserves the stochasticity of synapse sizes (with the Kesten process), but loses the stochastic nature of the state transitions and models average population behaviour. However, it is more computationally efficient and lends itself more easily to analysis.

Using the same parameter values as in Section 2, this gives:

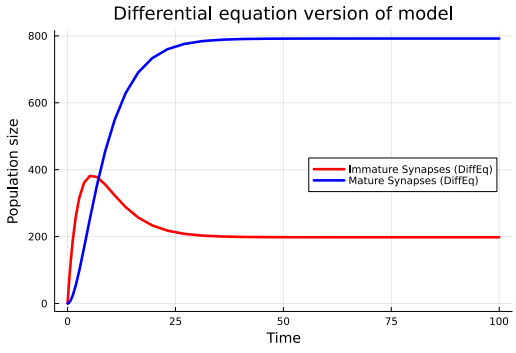


Figure 4: Populations solution using differential equations

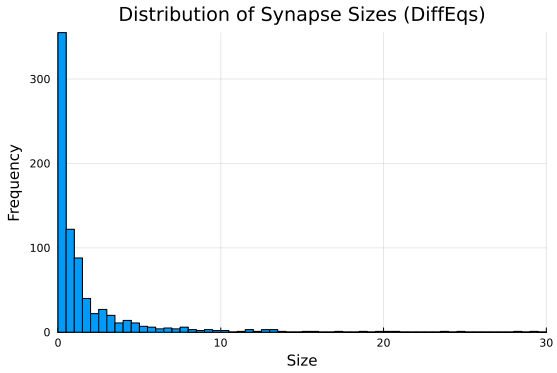


Figure 5: Final distribution of synaptic sizes

This closely matches the random walk version. To check, let us run multiple trials.

3.2 Multiple Trials

We run the above models over N trials (e.g. N=10 below) and this produces:

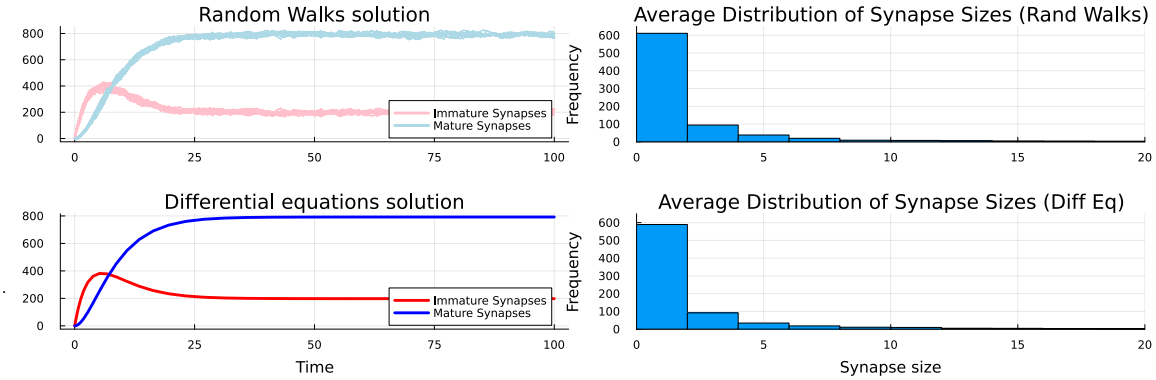


Figure 6: Multiple trials of Random Walks (top) and Differential Equations (bottom).

3.3 Steady State Solutions to the Differential Equations

Taking the differential equations

$$\frac{dN_I}{dt} = cN_P - (e+m)N_I + iN_M$$
$$\frac{dN_M}{dt} = mN_I - iN_M$$
$$\frac{dN_P}{dt} = eN_I - cN_P$$

we can find the final steady state values of N_I, N_m, N_P by letting the derivatives equal 0 and solving.

This gives

$$0 = cN_P - (e+m)N_I + iN_M \Rightarrow cN_P = (e+m)N_I - iN_M$$
$$0 = mN_I - iN_M \Rightarrow mN_I = iN_M$$
$$0 = eN_I - cN_P \Rightarrow eN_I = cN_P$$

We also have that $N_I + N_M + N_P = \text{total}$. This gives us:

$$N_I + \frac{m}{i}N_I + \frac{e}{c}N_I = \text{total}$$
$$\frac{i}{m}N_M + N_M + \frac{ei}{cm}N_M = \text{total}$$
$$\frac{c}{e}N_P + \frac{cm}{ei}N_P + N_P = \text{total}$$

which yields the solutions:

$$N_I = \frac{\text{total}}{(1 + \frac{m}{i} + \frac{e}{c})}$$
$$N_M = \frac{\text{total}}{(1 + \frac{i}{m} + \frac{ei}{cm})}$$
$$N_P = \frac{\text{total}}{(1 + \frac{c}{e} + \frac{cm}{ei})}$$

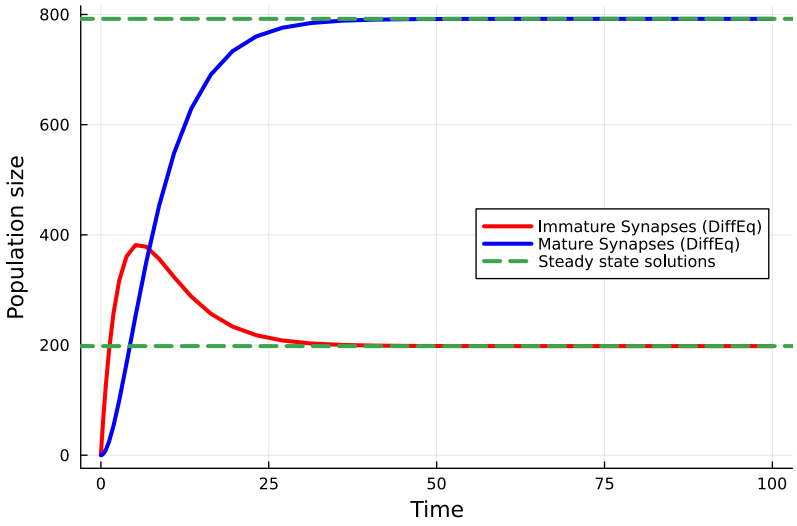


Figure 7: Differential equations model with steady state solutions

3.4 Phase Plane (Vector Field)

Let us plot the **phase plane** of the system. This is a graphical representation of a dynamical system with two state variables. Each point on the plane represents a specific state of the system, and trajectories show how these states evolve over time. A vector field on the phase plane consists of arrows at each point indicating the direction and speed of the system's evolution at that state.

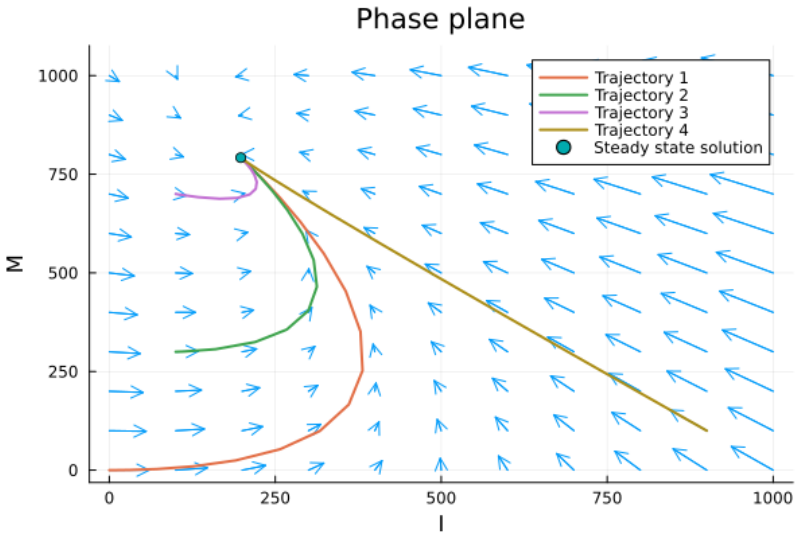


Figure 8: Phase plane of immature (x axis) vs mature (y axis) populations over time. The coloured lines show the different trajectories of populations with different initial values.

We can also plot the nullclines, i.e. the lines where the derivatives are zero. So for the first, we have that

$$cN_P - (e+m)N_I + iN_M = 0 \Rightarrow N_M = \frac{\text{total } c - cN_I - (e+m)N_I}{c-i}$$

and for the second

$$mN_I - iN_M = 0 \Rightarrow N_M = \frac{m}{i}N_I$$

.

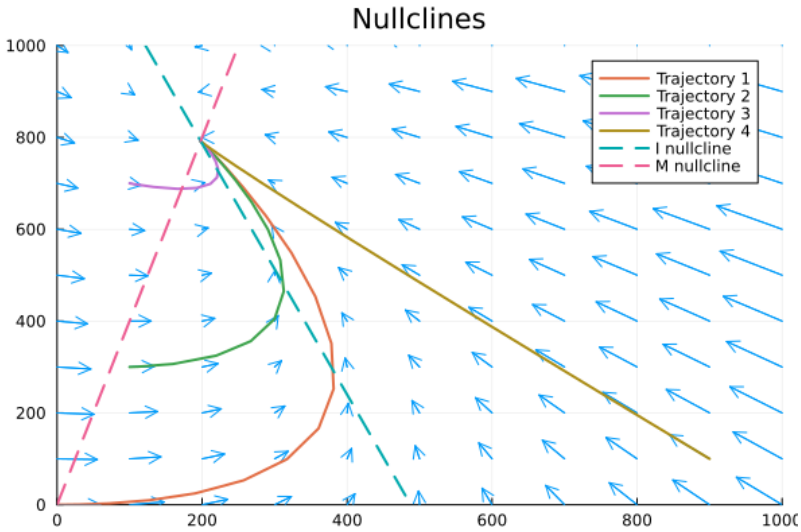


Figure 9: Here we plot the nullclines of the system. Where they intersect is the steady state solution.

3.5 Analytical solutions to differential equations

The differential equations

$$\frac{dN_I}{dt} = cN_P - (e+m)N_I + iN_M$$
$$\frac{dN_M}{dt} = mN_I - iN_M$$
$$\frac{dN_P}{dt} = eN_I - cN_P$$

can be rewritten as

$$\frac{dN_I}{dt} = cT - (e+m+c)N_I + (i-c)N_M$$
$$\frac{dN_M}{dt} = mN_I - iN_M$$

where T is the total possible synapses, and $T = N_P + N_I + N_M$.
This system of ODEs can be written as

$$\vec{s'}(t) = \frac{d}{dt}\begin{bmatrix} N_I \\ N_M \end{bmatrix} = \begin{bmatrix} -(e+m+c) & (i-c) \\ m & -i \end{bmatrix}\begin{bmatrix} N_I \\ N_M \end{bmatrix} + \begin{bmatrix} cT \\ 0 \end{bmatrix}$$

An analytical solution of the form $\vec{s}(t) = \vec{s_h}(t) + \vec{s_p}(t)$ can be found, where $\vec{s_h}$ and $\vec{s_p}$ denote the homogeneous and particular solutions, respectively. The solution to the homogeneous case is in the form

$$\vec{s_h}(t) = c_1\vec{v_1}e^{\lambda_1 t} + c_2\vec{v_2}e^{\lambda_2 t}$$

where λ_1, λ_2 are the eigenvalues of the matrix, $\vec{v_1}, \vec{v_2}$ the eigenvectors, and c_1, c_2 constants. The eigenvalues are

$$\lambda_i = \frac{-(e+m+c+i) \pm \sqrt{(e+m+c+i)^2 - 4(ei+ic+mc)}}{2}$$

and the eigenvectors

$$\vec{v_i} = \begin{bmatrix} 1 \\ \frac{m}{i+\lambda_i} \end{bmatrix}$$

The constants c_1, c_2 can then be determined from the initial values, i.e. $N_I(0) = 0, N_M(0) = 0$. We get that

$$c_1 = \frac{-\lambda_2 cT(i+\lambda_1)}{(\lambda_1-\lambda_2)(ei+ci+cm)}, c_2 = \frac{-\lambda_1 cT(i+\lambda_2)}{(\lambda_2-\lambda_1)(ei+ci+cm)}$$

For the particular solution (the inhomogeneous case), we use the method of undetermined coefficients. Given that the inhomogeneous term is $\begin{bmatrix} cT \\ 0 \end{bmatrix}$, we let $\vec{s_p}(t) = \vec{k} = \begin{bmatrix} k_1 \\ k_2 \end{bmatrix}$, which gives $\vec{s_p}' = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ Substituting this into our differential equation for $\vec{s'}$ gives

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -(e+m+c) & (i-c) \\ m & -i \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} + \begin{bmatrix} cT \\ 0 \end{bmatrix}$$

This allows us to solve for k_1, k_2, yielding

$$k_1 = \frac{icT}{ei+ci+cm}, k_2 = \frac{mcT}{ei+ci+cm}$$

We now have a full expression for the solution:

$$\vec{s}(t) = c_1\vec{v_1}e^{\lambda_1 t} + c_2\vec{v_2}e^{\lambda_2 t} + \vec{k}$$

3.6 Disproving the bump

We want to ask whether or not it is possible for a "bump" to occur in the joint $N_I + N_M$ populations, or if this combined population necessarily increases monotonically. In order to analyse the combined population, we take the analytical solution $\vec{s}(t) = c_1\vec{v_1}e^{\lambda_1 t} + c_2\vec{v_2}e^{\lambda_2 t} + \vec{k}$, and sum the two rows of the system of equations, i.e.

$$\begin{aligned} \text{Combined population (C.P)} &= c_1 v_{11} e^{\lambda_1 t} + c_2 v_{21} e^{\lambda_2 t} + k_1 + c_1 v_{12} e^{\lambda_1 t} + c_2 v_{22} e^{\lambda_2 t} + k_2 \\ &= c_1 e^{\lambda_1 t}(v_{11} + v_{12}) + c_2 e^{\lambda_2 t}(v_{21} + v_{22}) + k_1 + k_2 \end{aligned}$$

To test for the existence of a "bump", we find the derivative of this expression:

$$\frac{d}{dt}\text{C.P.} = \lambda_1 c_1 e^{\lambda_1 t}(v_{11} + v_{12}) + \lambda_2 c_2 e^{\lambda_2 t}(v_{21} + v_{22})$$

In order for there to be a bump, there must be a t such that the derivative is equal to 0. Setting it equal to 0 gives

$$\begin{aligned} \lambda_1 c_1 e^{\lambda_1 t}(v_{11} + v_{12}) &= -\lambda_2 c_2 e^{\lambda_2 t}(v_{21} + v_{22}) \\ \Rightarrow \frac{e^{\lambda_1 t}}{e^{\lambda_2 t}} = e^{(\lambda_1 - \lambda_2)t} &= \frac{-\lambda_2 c_2 (v_{21} + v_{22})}{\lambda_1 c_1 (v_{11} + v_{12})} \end{aligned}$$

We solve for t:

$$(\lambda_1 - \lambda_2)t = \ln(\kappa) \Rightarrow t = \frac{\ln(\kappa)}{\lambda_1 - \lambda_2}$$

where

$$\kappa = \frac{-\lambda_2 c_2 (v_{21} + v_{22})}{\lambda_1 c_1 (v_{11} + v_{12})}$$

Therefore, t exists when $\kappa > 0$. In order to disprove the existence of a bump, we need to show that $\kappa < 0$ always. Let us substitute in the expressions of c_1, c_2 presented above.

$$\Rightarrow \kappa = \frac{-\lambda_2 (\frac{-\lambda_1 cT(i+\lambda_2}{(\lambda_2-\lambda_1)(ei+ci+cm)})(v_{21} + v_{22})}{\lambda_1 (\frac{-\lambda_2 cT(i+\lambda_1}{(\lambda_1-\lambda_2)(ei+ci+cm)})(v_{11} + v_{12})} = \frac{-(i+\lambda_2)(\lambda_1 - \lambda_2)(v_{21} + v_{22})}{(i+\lambda_1)(\lambda_2 - \lambda_1)(v_{11} + v_{12})}$$

Now let us write in the terms for $v_{11}, v_{12}, v_{21}, v_{22}$:

$$\kappa = \frac{-(i+\lambda_2)(\lambda_1 - \lambda_2)(1 + \frac{m}{i+\lambda_2})}{(i+\lambda_1)(\lambda_2 - \lambda_1)(1 + \frac{m}{i+\lambda_1})} = \frac{-(\lambda_1 - \lambda_2)(i+m+\lambda_2)}{(\lambda_2 - \lambda_1)(i+m+\lambda_1)}$$

The part of the expression $\frac{(\lambda_1 - \lambda_2)}{(\lambda_2 - \lambda_1)}$ is -1, which means

$$\kappa = \frac{(i+m+\lambda_2)}{(i+m+\lambda_1)}$$

Substituting in the formulae we have for the eigenvalues yields

$$\kappa = \frac{(i+m+\frac{-(e+m+c+i)-\sqrt{(e+m+c+i)^2-4(ei+ic+mc)}}{2})}{(i+m+\frac{-(e+m+c+i)+\sqrt{(e+m+c+i)^2-4(ei+ic+mc)}}{2})}$$

$$= \frac{i+m-e-c-\sqrt{(e+m+c+i)^2-4(ei+ic+mc)}}{i+m-e-c+\sqrt{(e+m+c+i)^2-4(ei+ic+mc)}}$$

Multiplying top and bottom by the denominator's conjugate yields:

$$\kappa = \frac{\left(i+m-e-c-\sqrt{(e+m+c+i)^2-4(ei+ic+mc)}\right)^2}{(i+m-e-c)^2-(e+m+c+i)^2+4(ei+ci+cm)}$$

The numerator is always positive, therefore we need only check the denominator. Simplifying it gives $-4em$, and since e, m are positive, $-4em$ is negative, making $\kappa < 0$. Therefore, there is no t such that the combined population's derivative equals 0.

4 Weight-Dependent Random Walk Model

Above we simplified the approach by just removing the mature synapses with size 0, but we next model the mature-to-immature transition in a weight-dependent manner. We do this by defining a probability function over synapse size. Realistically, this must capture the fact that smaller synapses are more likely to demature (and large synapses are very unlikely to demature). This can be modelled simply with an exponential function

$$p(s \text{ dematuring }) = Ae^{-\frac{s}{\lambda}}$$

where s is the synapse size, and A, λ are parameters determining the exponential function. e.g.,

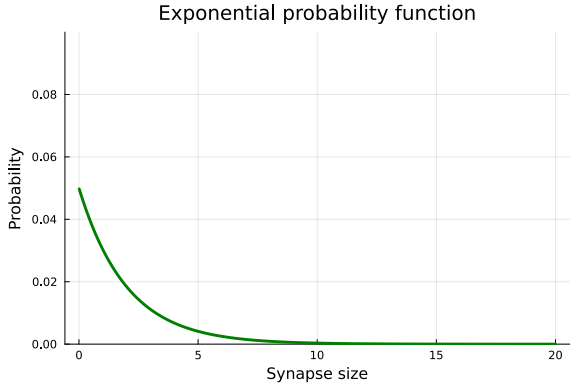


Figure 10: Exponential function for $A = 0.05, \lambda = 2$. Here we have $A = 0.05$ as this was the value of i used in previous simulations.

Adopting this in a constant timestep stochastic state transition model, we get the following:

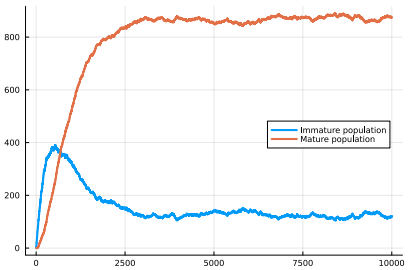


Figure 11: Single simulation of populations using weight dependent dematuring probabilities

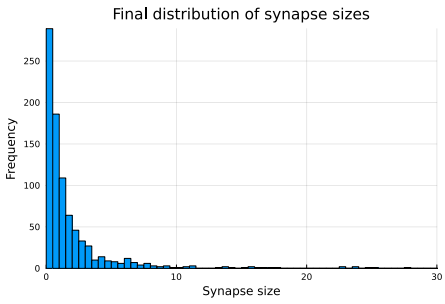


Figure 12: Final distribution of synaptic sizes.

5 Weight-Dependent Differential Equation Model

We similarly use this weight dependent idea in a differential equations setup, where we determine the rate of mature-to-immature by the exponential pdf. This gives:

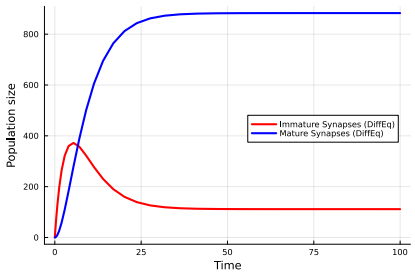


Figure 13: Differential Equation solution of populations using weight dependent dematuring probabilities

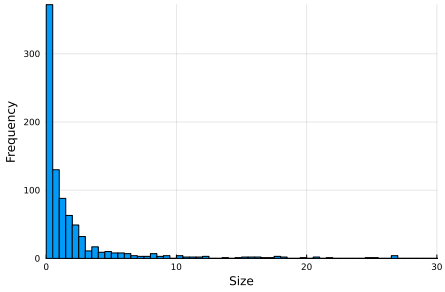
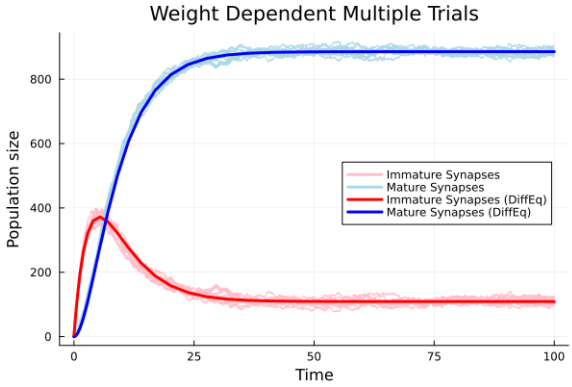


Figure 14: Final distribution of synaptic sizes.

Running this over some trials and comparing the probabilistic version with differential equations:



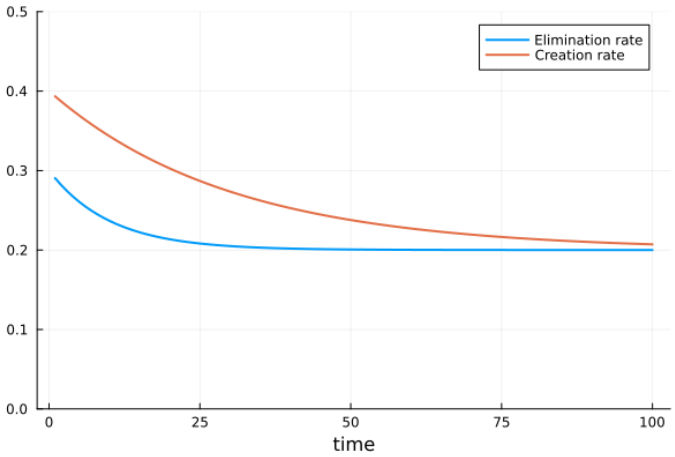
6 Variable rates of creation and elimination

We now model the rates of synapse formation ($c(t)$) and elimination ($e(t)$) in a time-dependent manner. We try this using exponential functions such that the rates start off higher and then decrease to a baseline k.

$$c(t) := A_1 e^{\frac{-t}{\lambda_1}} + k_1$$

$$e(t) := A_2 e^{\frac{-t}{\lambda_2}} + k_2$$

We use $A_1, A_2, \lambda_1, \lambda_2, k_1, k_2 = 0.2, 0.1, 30, 10, 0.2, 0.2$, which makes these functions look like follows:



Implementing this (along with the weight-dependent dematuring approach as above), gives us something like the plot below.

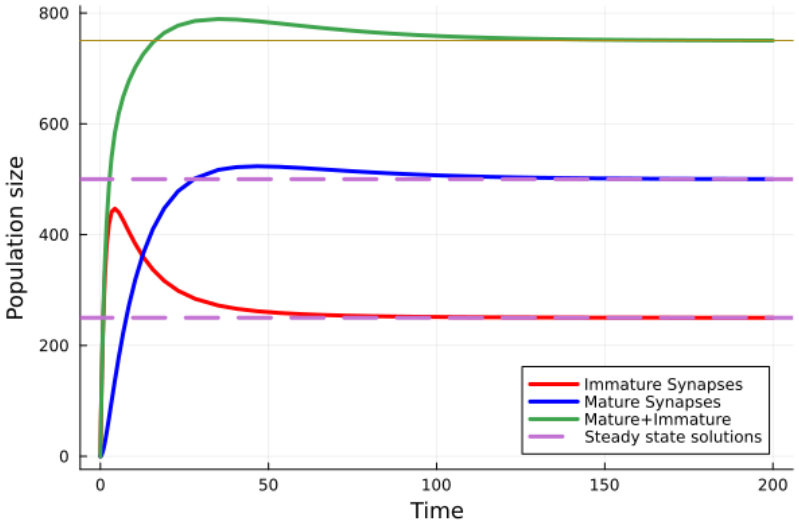


Figure 15: Immature, mature, combined populations over time modelling with time-dependent rates.

7 Tracking times in each state

We would like to be able to examine in the model how long each synapse lasts in the three states (resource pool, immature state, and mature state). Here is an example of the states of a single synapse across time, using constant rates

```
c, m, e, i = 0.2, 0.2, 0.01, 0.05
```

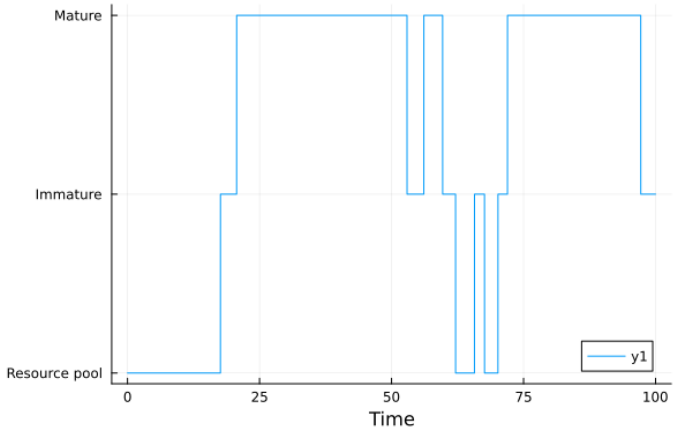


Figure 16: Example keeping track of single synapse: times spent in each possible state.

We can illustrate the distributions of durations in the three states across all synapses:

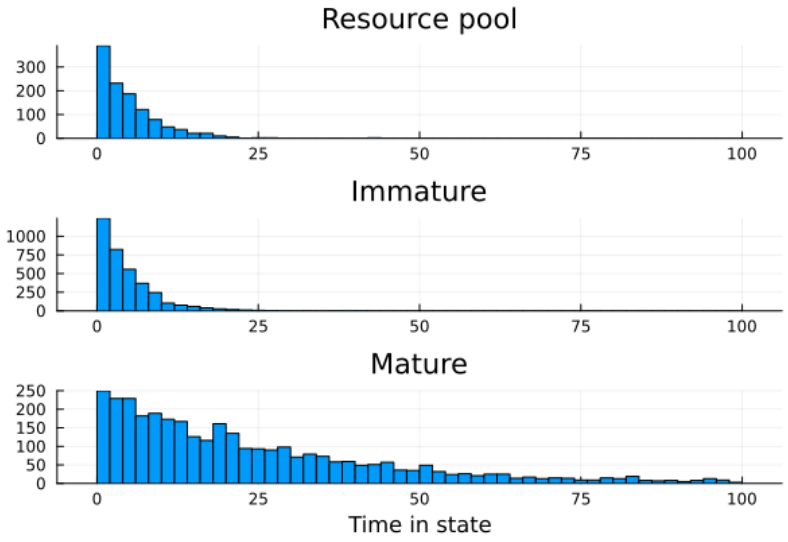


Figure 17: Distributions of time spent in the three states. Mean times 4.9, 4.6, and 23.6.

8 Example Optimisation

Julia's Optimization.jl package is useful for formulating a problem and finding optimal parameters. For instance, I defined a function which runs the model, finds the stable value of the mature population, compares it to a target value (in this case 100), and calculates the squared error. This function gets passed into an optimiser in order to find the rate parameters that will minimise this error (i.e. give a mature population

of 100). I provide it with an initial guess, a lower and upper bound, and then it runs (for several minutes). Code:

```
function_to_optimise()
    ...
    mature_population
    target = 100
    error = mature_population - target
    return error^2
end

# Define initial guesses for parameters c, m, e, i
x0 = [0.5, 0.5, 0.1, 0.1]

lower_bounds = [0.0, 0.0, 0.0, 0.0]
upper_bounds = [2.0, 2.0, 2.0, 2.0]

# Set up the optimization problem with bounds
opt_function = OptimizationFunction(function_to_optimise, Optimization.AutoForwardDiff())
prob = Optimization.OptimizationProblem(opt_function, x0, p, lb=lower_bounds, ub=upper_bounds)

# Solve the optimization problem using LBFGS
result = Optimization.solve(prob, LBFGS())
```

This give the values of 0.6820606226870286, 0.3896891612648679, 1.1943516162074415, 1.2748396318634099 for the parameters, which does indeed give a stable mature population of 100.

9 A Possibility – Work in Progress...

9.1 Calculating the average mature-to-immature fraction

The plan for the next step is to try make this a bit more analytically or computationally tractable. The idea is to work out the stationary limiting distribution p_1 of the synapse sizes (from the Kesten process, subject to certain conditions), and using the exponential weight-dependent probability function p_2, calculate the "average" fraction of synapses that make the mature-to-immature transition with the integral from 0 to ∞ of the product of p_1 and p_2. This may be calculable analytically (I have yet to work it out, seems hard), but if not, it can be done numerically (e.g. Julia's QuadGK package was used below). This "average" value can then be implemented in a probabilistic state transition model, or in the differential equations model.

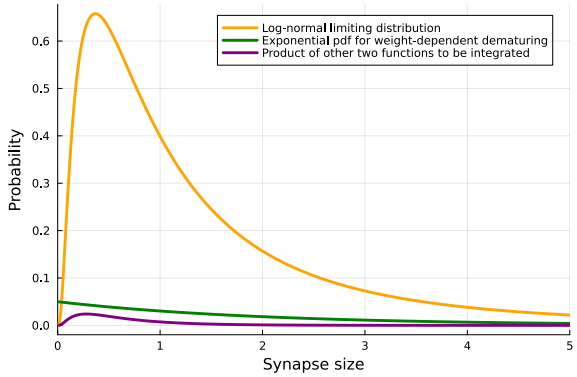


Figure 18: Limiting log-normal synapse size distribution (orange), exponential weight-dependent dematuring pdf (green), and their product (purple)

Using the distributions given above, $\int_0^\infty p_1 p_2 \approx 0.0191$ (QuadGK numerical integration package). Implementing this value in the state transition model from Section 5 gives the below simulation:

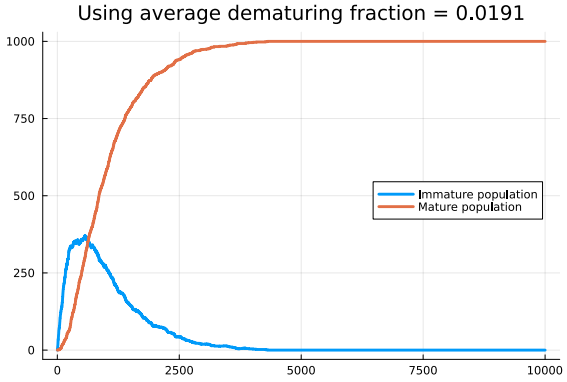


Figure 19: State transition model using constant average dematuring fraction of 0.0191

9.2 Incorporating developmental periods

The above models outlined above could be considered to only deal with a specific age in brain development because the parameters $c, m, e, \lambda, \sigma_\epsilon, \sigma_\eta$ are all fixed. Ultimately we want these to vary across the developmental trajectory. We will do this either by

- Fitting the model parameters separately to the data at each developmental timepoint recorded
- Fitting some parametric function for each parameter to model its smooth change across development (i.e. fitting all the developmental timepoints in one go).