A Mathematical Model of Neurodegeneration in a Cortical Column

Om Biyani, Cole McGuire, Byron Selvage, Kyle Sperber

Colorado School of Mines

MATH 472: Mathematical & Computational Neuroscience

Dr. Cecilia Diniz Behn

October 1, 2025

1 Abstract

Cortical columns are groups of neurons within the cerebral cortex of the brain that play an important role in regular brain functions such as information processing. These columns are composed of four to six distinct horizontal layers, each containing unique neuronal populations. In this project, we investigated the effects of neurodegeneration on cortical column dynamics in a simulated cortical column composed of L2/L3, L4, L5, and L6 neuronal layers. Using a system of mass-spring neuronal models mediated by a connectivity matrix representing inter-neuron interaction, we modeled neurodegeneration by scaling the interactions between neuron populations. We investigated the effect on system rhythm and column functionality. Our results show that neurodegeneration significantly disrupts neural signaling and impairs brain communication pathways. Moreover, different types of neurodegeneration stemming from neurodegenerative diseases like Alzheimer's disease and Frontotemporal Dementia alter brain functions in unique and nonuniform ways.

2 Biological Background

First identified by Vernon Mountcastle in 1957, cortical columns are basic units of the mammalian cerebral cortex. Cortical columns underpin multiple regular brain functions such as information processing and motor commands. Six distinct layers can be recognized within cortical columns:

- **Layer 1 (L1)**: The molecular layer comprised of few scattered neurons. Responsible for collecting and processing widespread information, controlling information flow throughout the neocortex.
- **Layer 2/3 (L2/3)**: The external granular and pyramidal layers contain primarily small to medium pyramidal neurons.
- **Layer 4 (L4)**: The internal granular layer is comprised of stellate and pyramidal neurons.
- **Layer 5 (L5)**: The internal pyramidal layer containing pyramidal neurons and connections to subcortical structures like the basal ganglia.
- **Layer 6 (L6)**: The multiform layer contains pyramidal and multiform neurons.

The exact function and behavior of each layer is still unclear. Within each layer, there are multiple types of excitatory and inhibitory neurons, including pyramidal, stellate, horizontal, and Martinotti cells. [7] The exact neuronal population and behavior vary between layers.

Multiple common neurodegenerative diseases, such as Alzheimer's disease, frontotemporal dementia, and cortical multiple sclerosis damage neurons in these cortical layers, disrupting the behavior and functionality of the cortical column. Alzheimer's disease primarily disrupts the behavior of neurons in layers 2 and 3 (L2/3) of the cerebral cortex. [1] [4] Frontotemporal dementia targets primarily RS and IB Neuron Populations layers 2 and 5. [4] Cortical multiple sclerosis impacts RS and IB Neuron Populations in layers 3 and 5. [6] Accurately modeling neurodegeneration in these layers allows us a clearer understanding of the behavior and impact of these diseases.

3 Mathematical Model

The neural mass model presented by Roberto C. Sotero creates a mathematical framework for modeling the dynamics of cortical columns. This model considers four cortical layers and four neuron populations: Regular Spiking (RS), Intrinsically Bursting (IB), Low-Threshold Spiking (LTS), and Fast-Spiking (FS). Since not every layer contains all types of neurons, this results in a system of 14 interconnected neuron populations. [5] This model is defined:

$$\frac{d^2x_m(t)}{dt^2} = -2k_m b_m \frac{dx_m(t)}{dt} - k_m^2 x_m(t) + G_m k_m (p_m + \sum_{n=1}^{14} \Gamma_{nm} S(x_n(t))) \tag{1}$$

where:

Γ is the connectivity matrix

k is the reciprocal of the time constant, $\left(\frac{1}{\tau}\right)$

b is the damping coefficient

p is the external input

G is the gain coefficient that represents the amplitude sensitivity

The function $S(x)$ is the average firing rate defined as a sigmoid function:

$$S(x) = \frac{e_0}{e^{r(v_0 - x)}} \tag{2}$$

where:

e_0 is the maximum firing rate

v_0 is the membrane potential at which the firing rate is halfway to max

r is the speed at which the firing rate increases.

4 Contributions

To model neurodegeneration, we scale the affected neuron population by a linear percentage.

$$p_{m\%} \cdot \frac{\partial^2 x_m(t)}{\partial t^2}$$

In this way, decreasing the entry's value represents the loss of neurons within that population, allowing us to implement neurodegeneration before simulating the model dynamics.

Using this scheme, we investigated single neuron population degeneration by only scaling one population at a time. We then targeted multiple neuron populations to simulate the effects of Alzheimer's disease and Frontotemporal Dementia.

5 Results

5.1 Single Neuron Degeneration

We investigated single neuron degeneration by applying the prescribed linear percentage scaling shown above. Applying this, we expected the neuron signal to decay as populations began to die off. However, what we found was chaos in the neurons as the signal became wildly unstable. Some neurons would become cut off from the network while others would attempt to stabilize. Meanwhile, secondary neurons in the connection would become resilient. One of the populations is shown below:

Figure 1: A Single Neuron Population as it Decays

Note that this figure shows a perfectly healthy neuron population on the right and a dead population on the left. This shows how unstable the neuron population becomes as the neurons begin to decay. Patterns in spiking become erratic in frequency, becoming chaotic. Simulating neurodegeneration in other single neuron populations creates similar results. We therefore choose not to include more graphs here as they are not enlightening.

These results, while interesting, are not significantly biologically relevant as, to our knowledge, no neurodegenerative disease targets only one specific neuron population. Therefore, we take these results as validation of our methodology and move to modeling more complex neurodegeneration, like the kinds you would see in neurodegenerative disorders.

5.2 Alzheimer's Disease

Alzheimer's disease disrupts the behavior of all neuron populations within the second and third cortical layers (L2/3) of the cerebral cortex. To model this behavior, we scaled the L2/3 populations by a linear percentage and simulated the system dynamics. This resulted in the following behavior.

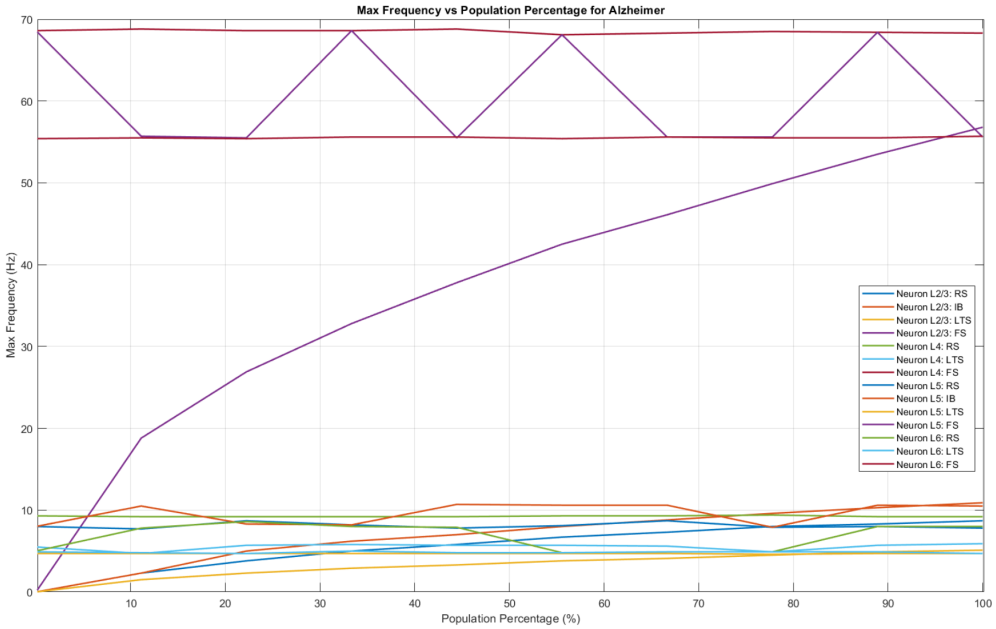

Figure 2: Neuronal Firing Frequency Under Neurodegeneration from Alzheimer's

This graph shows logarithmic decay in all chosen neuron populations, decay in the spiking frequency of L4 RS neurons, instability in the L5 FS neurons, and stability in the L6 RS neuron population.

This suggests that the neuron populations of layers 2/3 are the first neurons affected by Alzheimer's disease, as expected. Additionally, we expect that after 30% decay of the L2/3 neuron populations, physical damage will cause decay and death throughout the entire cortical column. [1] However, while our model sees that we would have a loss of system rhythm, it does not accurately capture system collapse after this 30% threshold.

5.3 Frontotemporal Dementia

Frontotemporal dementia is characterized by neuronal loss in Layers 2–3 and damaged structures in Layer 5. Unlike Alzheimer's disease, which targets all neuron populations, frontotemporal dementia targets primarily the Regular Spiking (RS) and Intrinsically Bursting (IB) neuron populations. Like with Alzheimer's disease, we used a scaling of the neuron population to simulate neurodegeneration in these neuron populations. This created the following plot.

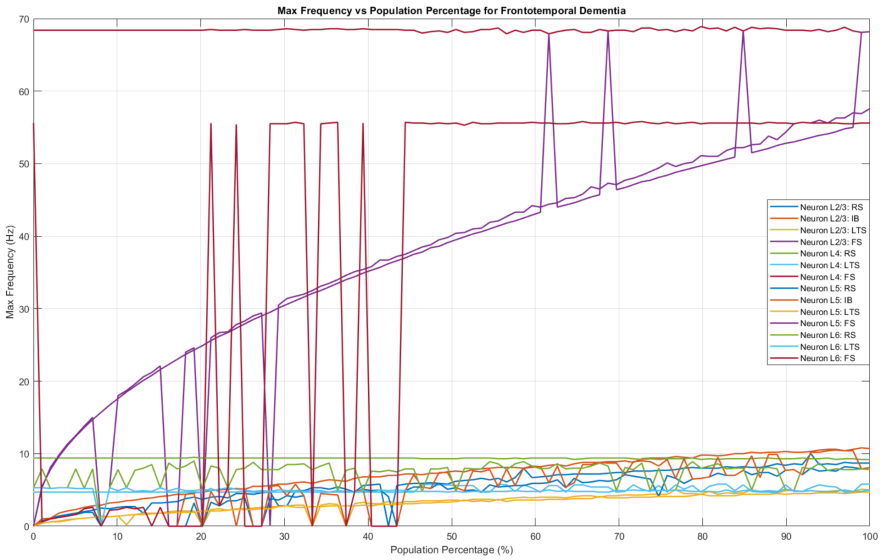

Figure 3: Neuronal Firing Frequency Under Neurodegeneration from Frontotemporal Dementia

As expected, we saw logarithmic decay in the affected neuron populations (RS and IB). We also see extreme instability of L4 FS neurons after 50% neurodegeneration, instability of the L5 FS neurons after 30%, instability of the L2/3 IB neurons after 90% with complete ceasing of activity at 16%. In contrast to these neurons, the L4 RS and LTS neurons and L6 FS neurons were stable and unaffected by the neurodegeneration.

Research suggests that 40%- 70% neurodegeneration will cause decay through the entire cortical column. [4] Our model results are consistent with this as we see total system collapse at around the 50% neurodegeneration level. This is likely because more interconnected groups were affected, causing the damage to cascade through the system.

6 Conclusion

In this study, we developed and implemented a mathematical model of neurodegeneration within a cortical column by simulating neuronal interactions across four cortical layers and 14 neuron populations. Using a system of second-order differential equations derived from a mass-spring model, we represented neurodegeneration through linear scaling of affected neuronal populations. This framework allowed us to investigate both isolated and disease-specific patterns of neurodegeneration, including those observed in Alzheimer's disease and frontotemporal dementia.

Our findings suggest that neurodegeneration disrupts cortical function in complex and often nonlinear ways. While localized neuron loss led to instability and chaotic activity in certain populations, interconnected networks often exhibited unexpected resilience or vulnerability depending on their connectivity. Simulations of Alzheimer's and frontotemporal dementia revealed progressive breakdowns in system rhythm and structure, aligning with clinical observations of disease progression.

Although our model captures important qualitative features of neurodegeneration, it does not fully account for the cascading effects observed beyond certain thresholds of damage. Future work could improve fidelity by incorporating more detailed biological features such as synaptic plasticity, neurotransmitter dynamics, and glial cell involvement. Overall, our model provides a foundational approach for understanding how cellular-level neurodegeneration can impair large-scale brain function.

Model Parameters Initialization

```
%% Model Parameters Initialization - RUN THIS ALWAYS
clear;
close all;

gamma = [
    19.23, 12.53, 34.17, 14.07, 1.61, 0, 0, 3.82, 1.61, 0, 0, 0, 0, 0;
    12.53, 12.53, 27.47, 14.07, 1.61, 0, 0, 3.82, 1.61, 0, 0, 0, 0, 0;
    -23.45, -23.45, -52.93, -6.70, 0, 0, 0, -23.45, -33.50, 0, 0, -16.75, -16.75, -11.39;
    -3.35, -5.36, -6.03, -20.1, 0, 0, 0, -3.35, -6.70, 0, -2.01, -3.35, 0, -2.01;
    9.72, 0, 0, 0, 22.98, 34.17, 58.96, 7.77, 8.17, 0, 2.14, 0, 0, 0;
    0, 0, 0, 0, -23.45, -52.93, -8.71, 0, 0, 0, 0, 0, 0, 0;
    0, 0, 0, 0, -6.03, -6.03, -61.64, 0, 0, 0, 0, 0, 0, 0;
    1.47, 0, 0, 0, 0.47, 0, 0, 32.89, 5.36, 20.77, 8.71, 2.14, 0, 0;
    1.21, 0, 0, 0, 1.14, 0, 0, 46.90, 20.77, 8.71, 4.69, 0, 0, 0;
    -23.45, 0, 0, 0, -23.45, -23.45, -52.93, -2.01, -16.75, 0, -5.36, 0, 0, 0;
    0, 0, 0, 0, -2.68, -2.68, -61.64, 0, 0, 0, 0, 0, 0, 0;
    0, 0, 0, 0, 0, 0, 0, 0.40, 1.88, 0, 0, 48.78, 34.17, 15.41;
    -23.45, 0, 0, 0, 0, 0, 0, -16.75, -16.75, -5.36, -23.45, -66.33, -8.71, 0;
    0, 0, 0, 0, 0, 0, 0, -9.38, -29.48, -28.14, 0, 0, 0, 0;
];

population = diag(ones(14,1));
tspan = [0, 10];
u0 = zeros(28, 1);
NueronLabels = ["L2/3: RS", "L2/3: IB", "L2/3: LTS", "L2/3: FS", ...
                "L4: RS", "L4: LTS", "L4: FS", ...
                "L5: RS", "L5: IB", "L5: LTS", "L5: FS", ...
                "L6: RS", "L6: LTS", "L6: FS"];
```

Single Population Test

```
population = diag(ones(14,1));
population_percentage = 0.3;
nueronNum = [1 ,2, 8, 9];
e0 = 5; v0 = 6; r = 0.56; b = 0.001;
parameters = [e0, v0, r, b];

for i = 1:length(nueronNum)
    population(nueronNum(i), nueronNum(i)) = population_percentage;
end

label = "";
for i = 1:length(nueronNum)-1
    label = [label, NueronLabels(nueronNum(i)),", "];
end
label = [label, NueronLabels(length(nueronNum))];

[t_1pop,u_1pop] = ode45(@(t,u)PACM_prime(t,u, population, parameters),tspan,u0);

N = length(t_1pop);
t_uniform = linspace(t_1pop(1), t_1pop(end), N);
dt = t_uniform(2) - t_uniform(1);
Fs = 1/dt;
freq = (0:N-1)*(Fs/N);
half_N = floor(N/2);

num_neurons = 14;
for start_idx = 1:3:num_neurons
    figure;
    tiledlayout(3, 2, 'TileSpacing', 'compact', 'Padding', 'compact');
    for offset = 0:2
        i = start_idx + offset;
```

```
        if i > num_neurons
            break;
        end

        x = u_1pop(:, i);
        x_uniform = interp1(t_1pop, x, t_uniform, 'linear');
        f = fft(x_uniform);

        nexttile;
        plot(t_uniform, x_uniform, 'LineWidth', 1.2);
        xlabel('Time (s)');
        ylabel('Amplitude');
        title(['Neuron ', NueronLabels(i), ' Output']);
        grid on;

        nexttile;
        plot(freq(1:half_N), abs(f(1:half_N)), 'LineWidth', 1.2);
        xlabel('Frequency (Hz)');
        ylabel('Magnitude');
        title(['Neuron ', num2str(i), ' Spectrum']);
        grid on;
        xlim([0 100]);
    end
    sgtitle(['Neuron Outputs and Spectra (Neurons ', num2str(start_idx), '   ', num2str(min(
        start_idx+2,num_neurons)), ')']);
end
```

Varying Population Percentages

```
population = diag(ones(14, 1));
options = odeset('RelTol',1e-6,'AbsTol',1e-8);
nueronNum = [1 ,2, 8, 9];
e0 = 5; v0 = 6; r = 0.56; b = 0.001;
parameters = [e0, v0, r, b];

population_percentages = linspace(0, 1, 100);
max_frequencies_all = [];

for p_idx = 1:length(population_percentages)
    population_percentage = population_percentages(p_idx);
    population = diag(ones(14, 1));
    for i =1:length(nueronNum)
        population(nueronNum(i), nueronNum(i)) = population_percentage;
    end

    [t_1pop, u_1pop] = ode45(@(t, u) PACM_prime(t, u, population, parameters), tspan, u0,
        options);

    N = length(t_1pop);
    t_uniform = linspace(t_1pop(1), t_1pop(end), N);
    dt = t_uniform(2) - t_uniform(1);
    Fs = 1 / dt;
    freq = (0:N-1) * (Fs / N);
    half_N = floor(N / 2);

    num_neurons = 14;
    max_frequencies = zeros(1, num_neurons);

    for i = 1:num_neurons
        x = u_1pop(:, i);
        x_uniform = interp1(t_1pop, x, t_uniform, 'linear');
        f = fft(x_uniform);
        [max_value, max_index] = max(abs(f(2:half_N*2)));
        max_frequency = freq(max_index);
        max_frequencies(i) = max_frequency;
```

```
    end
    max_frequencies_all = [max_frequencies_all; max_frequencies];
end

figure;
for i = 1:num_neurons
    plot(population_percentages * 100, max_frequencies_all(:, i), ...
        'LineWidth', 1.5, 'DisplayName', ['Neuron ', NueronLabels{i}]);
    hold on;
end
xlabel('Population Percentage (%)');
ylabel('Max Frequency (Hz)');
title(['Max Frequency vs Population Percentage for Neuron ', num2str(nueronNum)]);
legend show;
grid on;
```

PACM_prime Function

```
function dudt = PACM_prime(t, u, population, parameters)
    G = [3.25, 3.25, 30, 10, 3.25, 30, 10, 3.25, 3.25, 30, 10, 3.25, 30, 10]';
    k = [60, 70, 30, 350, 60, 30, 350, 60, 70, 30, 350, 60, 30, 350]';

    e0 = parameters(1);
    p = zeros(14, 1);
    p(5) = 500;
    p(7) = 150;
    v0 = parameters(2);
    r = parameters(3);
    b = parameters(4);

    gamma = [
        19.23, 12.53, 34.17, 14.07, 1.61, 0, 0, 3.82, 1.61, 0, 0, 0, 0, 0;
        12.53, 12.53, 27.47, 14.07, 1.61, 0, 0, 3.82, 1.61, 0, 0, 0, 0, 0;
        -23.45, -23.45, -52.93, -6.70, 0, 0, 0, -23.45, -33.50, 0, 0, -16.75, -16.75,
            -11.39;
        -3.35, -5.36, -6.03, -20.1, 0, 0, 0, -3.35, -6.70, 0, -2.01, -3.35, 0, -2.01;
        9.72, 0, 0, 0, 22.98, 34.17, 58.96, 7.77, 8.17, 0, 2.14, 0, 0, 0;
        0, 0, 0, 0, -23.45, -52.93, -8.71, 0, 0, 0, 0, 0, 0, 0;
        0, 0, 0, 0, -6.03, -6.03, -61.64, 0, 0, 0, 0, 0, 0, 0;
        1.47, 0, 0, 0, 0.47, 0, 0, 32.89, 5.36, 20.77, 8.71, 2.14, 0, 0;
        1.21, 0, 0, 0, 1.14, 0, 0, 46.90, 20.77, 8.71, 4.69, 0, 0, 0;
        -23.45, 0, 0, 0, -23.45, -23.45, -52.93, -2.01, -16.75, 0, -5.36, 0, 0, 0;
        0, 0, 0, 0, -2.68, -2.68, -61.64, 0, 0, 0, 0, 0, 0, 0;
        0, 0, 0, 0, 0, 0, 0, 0.40, 1.88, 0, 0, 48.78, 34.17, 15.41;
        -23.45, 0, 0, 0, 0, 0, 0, -16.75, -16.75, -5.36, -23.45, -66.33, -8.71, 0;
        0, 0, 0, 0, 0, 0, 0, -9.38, -29.48, -28.14, 0, 0, 0, 0;
    ];

    x = population * u(1:14);
    dx = u(15:28);

    S_x = e0 ./ (1 + exp(r * (v0 - x)));

    ddx = -2 .* k .* b .* dx - k.^2 .* x + G .* k .* (p + gamma * S_x);

    dudt = [dx; ddx]; % necessary to return a column vector
end
```

References

[1] John H. Morrison and Patrick R. Hof. Life and death of neurons in the aging cerebral cortex. In *The Neurobiology of Epilepsy and Aging*, volume 81 of *International Review of Neurobiology*, pages 41–57. Academic Press, 2007.

[2] Yuzhen Qin, Tommaso Menara, Danielle S. Bassett, and Fabio Pasqualetti. Phase-amplitude coupling in neuronal oscillator networks. *Phys. Rev. Res.*, 3:023218, Jun 2021.

[3] Yousef Salimpour, Kelly A. Mills, Brian Y. Hwang, and William S. Anderson. Phase- targeted stimulation modulates phase-amplitude coupling in the motor cortex of the human brain. *Brain Stimulation: Basic, Translational, and Clinical Research in Neuromodulation*, 15(1):152–163, Jan 2022.

[4] William W Seeley. Selective functional, regional, and neuronal vulnerability in frontotemporal dementia. *Curr Opin Neurol*, 21(6):701–707, December 2008.

[5] Roberto C. Sotero. Topology, cross-frequency, and same-frequency band interactions shape the generation of phase-amplitude coupling in a neural mass model of a cortical column. *PLOS Computational Biology*, 12(11):1–29, 11 2016.

[6] Bruce D Trapp, Megan Vignos, Jessica Dudman, Ansi Chang, Elizabeth Fisher, Susan M Staugaitis, Harsha Battapady, Sverre Mork, Daniel Ontaneda, Stephen E Jones, Robert J Fox, Jacqueline Chen, Kunio Nakamura, and Richard A Rudick. Cortical neuronal densities and cerebral white matter demyelination in multiple sclerosis: a retrospective study. *Lancet Neurol*, 17(10):870–884, August 2018.

[7] Jana Vaskovic and Dimitrios Mytilinaios. Cytoarchitecture of cerebral cortex. *Kenhub*, Nov 2023.