

CAAM 615 Graduate Project

Javeria Ahmed (ja71@rice.edu)

January 12, 2023

1. Abstract

We consider a neuron with a large dendritic tree such as the Lobula Giant Movement Detector (LGMD) neuron and attempt to build its Hines matrix to ultimately obtain eigenvalue decomposition. Once obtained, we use the eigenvalues and eigenvectors to further inspect the relationship between the mother and the daughter dendrites. Previously, we have seen a simple forked dendrite and constructed its Hines matrix from the current balance equation and cell circuitry. The resulting almost-symmetric Hines matrix is acted upon by suitable transformations to bring it to symmetry. We build up the LGMD model from the simple forked model by adding upon and tweaking the already existing cell circuit until it resembles the LGMD model. We then use a similar strategy to convert the resulting asymmetric matrix to a symmetric one and then exploit its symmetry to obtain its eigenvalues and eigenvectors.

2. Introduction & Background

The Lobula Giant Movement Detector (LGMD) is a wide-field visual neuron that is found in the Lobula layer of the Locust central nervous system. The LGMD is known to increase its firing rate in response to the velocity and proximity of an approaching object. It has also been found that it can respond to looming stimuli very quickly and trigger avoidance reactions¹.

The Lobula Giant Movement Detector (LGMD) is the neuron responsible for detecting collision in the locust (Peron et al. (2009))². One proposed structure of the LGMD neuron was characterized by Peron et al. (2009)

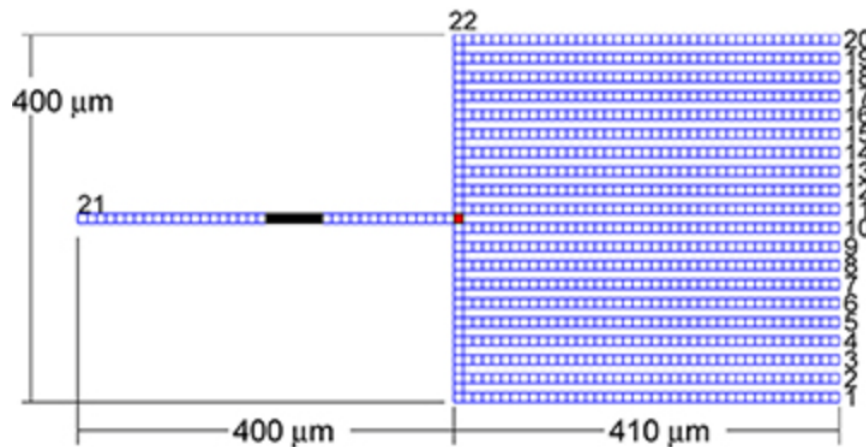


Figure 1

¹Meng, Hongying Appiah, Kofi Yue, Shigang Hunter, Andrew Hobden, Mervyn Priestley, Nigel Hobden, Peter Pettit, Cy. (2010). A modified model for the Lobula Giant Movement Detector and its FPGA implementation. Computer Vision and Image Understanding, 114. 1238-1247. 10.1016/j.cviu.2010.03.017.

²Peron, S. P., Jones, P. W., and Gabbiani, F. (2009). Precise subcellular input retinotopy and its computational consequences in an identified visual interneuron. *Neuron* 63, 830–842. doi: 10.1016/j.neuron.2009.09.010

In this particular template the LGMD neuron has 20 daughter dendrites, a junction dendrite (that connects each daughter to the mother and the other dendrites), the mother and finally the soma and has a distinctive rake like structure as seen above.

In the past, we have constructed smaller but similar templates of neurons divided into compartments as seen in the diagram below. This compartmentalization has allowed modeling these neurons discretely using matrices.

One important matrix in modeling that we came across the branched cell with soma is the Hines matrix. It emerges in the simulations of mathematical models describing the initiation and propagation of action potentials in a neuron. A Hines matrix is a symmetric matrix where every row of the matrix has only one nonzero element with a column index bigger than the row index ³. The almost symmetric nature of the Hines matrix and its similarity to a tri-diagonal matrix makes it very useful. In fact, when calculating the eigenvectors of a slightly transformed version of \mathbf{H} , we find that they are orthonormal in the weighed sense ⁴.

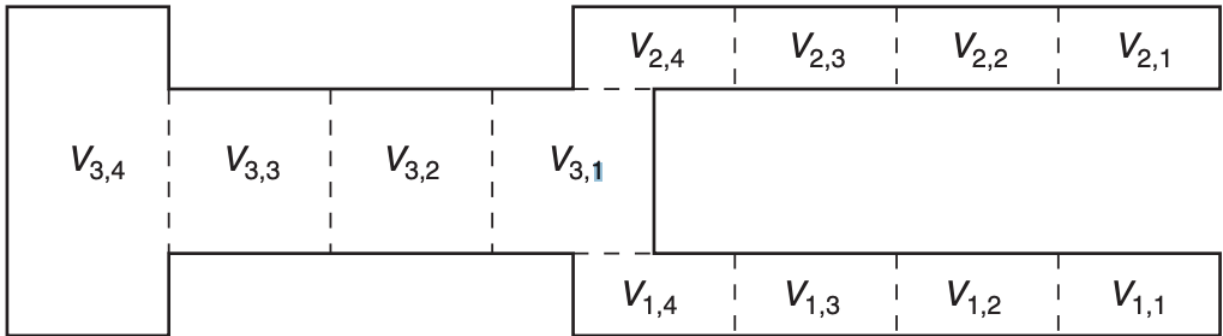
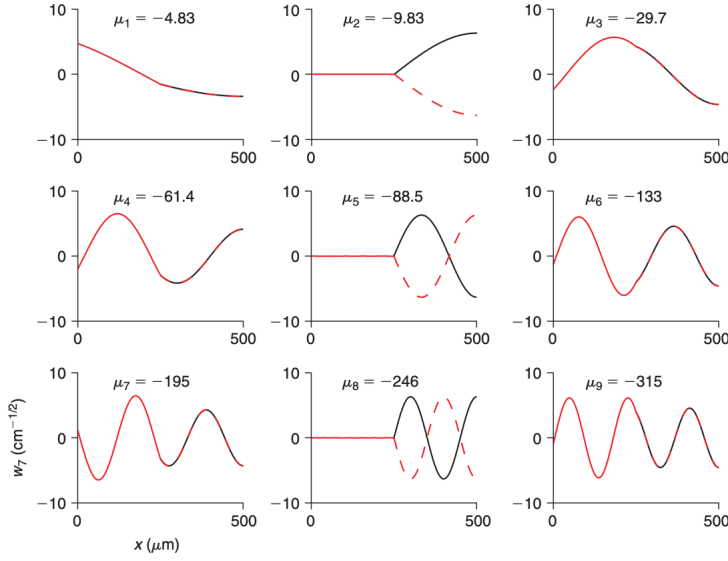


Figure 2

The eigenvector and eigenvalues are used to observe the behavior of the mother and the two daughter strands. In the figure below, the red line represents the mother, dashed red the first daughter, and black the second daughter. We can see that sometimes the daughters follow the behavior of the mother completely, and when they don't, their behavior is equal and opposite of each other.

³Vooturi, D. T., Kothapalli, K., Bhalla, U. S. (2017, December). Parallelizing Hines matrix solver in neuron simulations on GPU. In 2017 IEEE 24th International Conference on High Performance Computing (HiPC) (pp. 388-397). IEEE.

⁴Fabrizio Gabbiani, Steven James Cox, Chapter 8 - The Passive Dendritic Tree, Editor(s): Fabrizio Gabbiani, Steven James Cox, Mathematics for Neuroscientists (Second Edition), Academic Press, 2017, Pages 109-123, ISBN 9780128018958, doi: 10.1016/B978-0-12-801895-8.00008-7

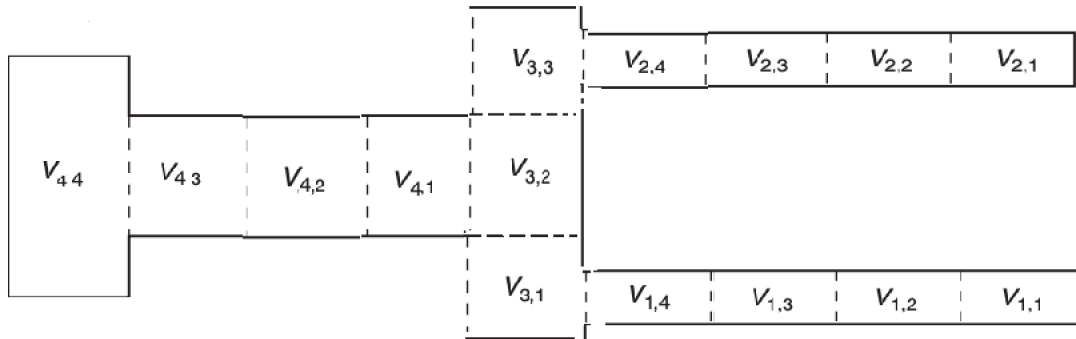
**Figure 3**

We build a 879 compartment model of the LGMD neuron. After it's current balance equations have been set up, we construct the Hines matrix from the ground up starting with a simple branched cell compartmentalization **Fig. 2**. We find the eigenvalues and eigenvectors of the Hines matrix corresponding to the LGMD neuron and compare with those of the simple branched cell (Fig. 3).

3. Method

Structuring the Hines Matrix

For the purpose of constructing the Hines matrix and its corresponding eigenvalues for the LGMD neuron, we start by building up from our existing model i.e. (**Fig. 2**). Step by step, we can bring this simple branched cell closer and closer to the LGMD neuron cell. The first integral change we make is that we make a distinct 'junction dendrite' that is separate from the mother

**Figure 4 : Sub – structure 1**

Next, we inspect how this addition of a junction dendrite to the original structure cascades to the Hines matrix. Using the current balance equations we obtain the following system

$$\begin{aligned}
\tau v'_{1,1} + v_{1,1} - \lambda_1^2(v_{1,2} - v_{1,1})/dx^2 &= 0 \\
\tau v'_{1,2} + v_{1,2} - \lambda_1^2(v_{1,3} - 2v_{1,2} + v_{1,1})/dx^2 &= 0 \\
\tau v'_{1,3} + v_{1,3} - \lambda_1^2(v_{1,4} - 2v_{1,3} + v_{1,2})/dx^2 &= 0 \\
\tau v'_{1,4} + v_{1,4} - \lambda_1^2(v_{3,1} - 2v_{1,4} + v_{1,3})/dx^2 &= 0 \\
\tau v'_{2,1} + v_{2,1} - \lambda_2^2(v_{2,2} - v_{2,1})/dx^2 &= 0 \\
\tau v'_{2,2} + v_{2,2} - \lambda_2^2(v_{2,3} - 2v_{2,2} + v_{2,1})/dx^2 &= 0 \\
\tau v'_{2,3} + v_{2,3} - \lambda_2^2(v_{2,4} - 2v_{2,3} + v_{2,2})/dx^2 &= 0 \\
\tau v'_{2,4} + v_{2,4} - \lambda_2^2(v_{3,3} - 2v_{2,4} + v_{2,3})/dx^2 &= 0 \\
\tau v'_{3,1} + v_{3,1} + \frac{a_1 \lambda_1^2}{a_3 dx^2}(v_{3,1} - v_{1,4}) - \lambda_3^2(v_{3,2} - v_{3,1})/dx^2 &= 0 \\
\tau v'_{3,2} + v_{3,2} + \frac{a_4 \lambda_4^2}{a_3 dx^2}(v_{4,1} - v_{3,2}) - \lambda_3^2(v_{3,1} - 2v_{3,2} + v_{3,3})/dx^2 &= 0 \\
\tau v'_{3,3} + v_{3,3} + \frac{a_2 \lambda_2^2}{a_3 dx^2}(v_{3,3} - v_{2,4}) - \lambda_3^2(v_{3,2} - v_{3,3})/dx^2 &= 0 \\
\tau v'_{4,1} + v_{4,1} - \lambda_4^2(v_{3,2} - 2v_{4,1} + v_{4,2})/dx^2 &= 0 \\
\tau v'_{4,2} + v_{4,2} - \lambda_4^2(v_{4,1} - 2v_{4,2} + v_{4,3})/dx^2 &= 0 \\
\tau v'_{4,3} + v_{4,3} - \lambda_4^2(v_{4,2} - 2v_{4,3} + v_{4,4})/dx^2 &= 0 \\
\tau v'_{4,4} + v_{4,4} - \frac{A_4}{A_S} \lambda_4^2(v_{4,3} - v_{4,4})/dx^2 - \frac{I_{stim}}{g_{Cl} A_S} &= 0
\end{aligned}$$

and similarly we are able to write the above system as

$$\begin{aligned}
\mathbf{v}'(t) &= \mathbf{B}\mathbf{v}(t) + \mathbf{f}(t) \\
\mathbf{B} &= (\mathbf{H} - \mathbf{I})/\tau \\
\mathbf{f}(t) &= \frac{I_{stim}}{g_{Cl} A_S} \mathbf{e}_{15}
\end{aligned}$$

where \mathbf{H} is the Hines matrix and is as follows

$$-\frac{1}{dx^2} \begin{bmatrix}
-\lambda_1^2 & \lambda_1^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\lambda_1^2 & -2\lambda_1^2 & 1\lambda_1^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \lambda_1^2 & -2\lambda_1^2 & \lambda_1^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \lambda_1^2 & -2\lambda_1^2 & 0 & 0 & 0 & 0 & \lambda_1^2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -\lambda_2^2 & \lambda_2^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \lambda_2^2 & -2\lambda_2^2 & \lambda_2^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \lambda_2^2 & -2\lambda_2^2 & \lambda_2^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \lambda_2^2 & -2\lambda_2^2 & 0 & 0 & \lambda_2^2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & r_1 \lambda_1^2 & 0 & 0 & 0 & 0 & -\lambda_3^2 - r_1 \lambda_1^2 & \lambda_3^2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_3^2 & -2\lambda_3^2 + r_4 \lambda_4^2 & \lambda_3^2 & -r_4 \lambda_4^2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & r_2 \lambda_2^2 & 0 & \lambda_3^2 & -\lambda_3^2 - r_2 \lambda_2^2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_4^2 & 0 & -2\lambda_4^2 & \lambda_4^2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_4^2 & -2\lambda_4^2 & \lambda_4^2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_4^2 & -2\lambda_4^2 & \lambda_4^2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \rho \lambda_4^2 & -\rho \lambda_4^2
\end{bmatrix}$$

where $\rho = \frac{A_4}{A_S}$ and $r_i = \frac{a_i}{a_3}$

Immediately, we can see the matrix differs (as expected) only at the junction dendrite sub-matrix and one unit ahead (or behind) i.e. $v_{2,1}$, $v_{2,4}$ and $v_{4,1}$. We call this area, where the matrix differs, as the one-compartment radius of the junction dendrite.

We explore one more sub-structure before we reach the generalized LGMD Hines matrix

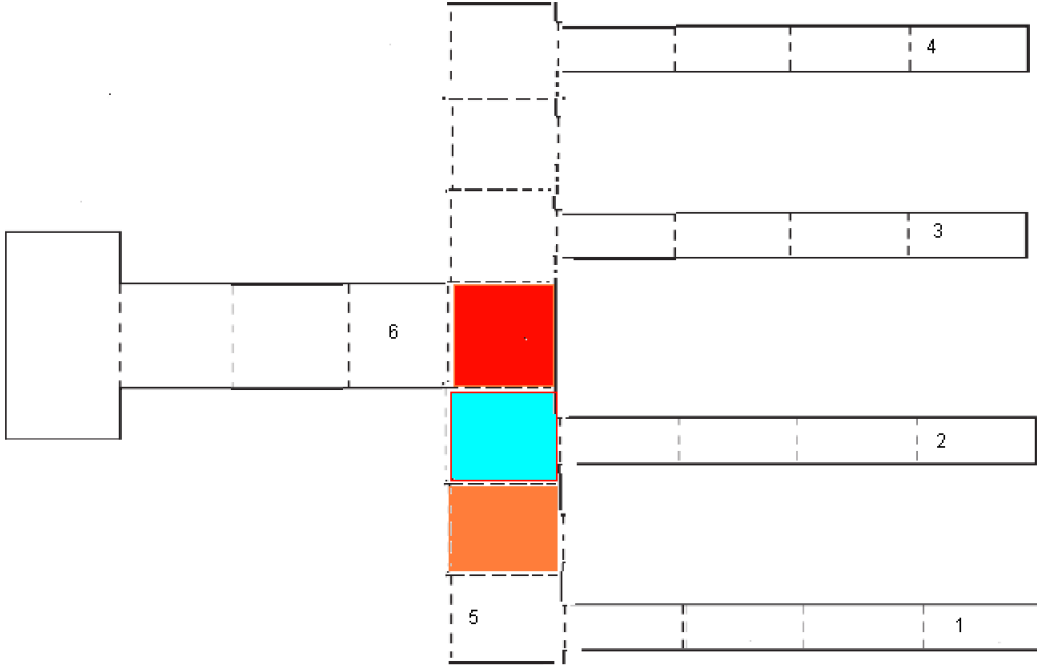


Figure 5 : Sub – structure 2

We have already observed that the truly dynamic nature of the Hines matrix comes from the variation in the junction dendrite and its one-compartment radius. However, in the previous case because the junction dendrite consisted only of 3 compartments - the compartments were either the **bridge** (blue compartment in **Fig. 5**) to a daughter dendrite or the **central junction** (red compartment in **Fig. 5**). In a case, where the neuron is larger we observe a third kind of compartment present in the junction dendrite. We call this compartment a **connector** (orange compartment in **Fig. 5**) and it links the other two kinds of compartments together. A *connector* is either the connection between two *bridges* or a *bridge* and the *central junction*. This distinction will help us construct and encode a general Hines matrix for the LGMD neuron.

Using the current balance equations we can write the system for the junction-dendrite with a single compartment radius

$$\begin{aligned}
\tau v'_{1,4} + v_{1,4} - \lambda_1^2(v_{5,1} - 2v_{1,4} + v_{1,3})/dx^2 &= 0 \\
\tau v'_{2,4} + v_{2,4} - \lambda_2^2(v_{5,3} - 2v_{2,4} + v_{2,3})/dx^2 &= 0 \\
\tau v'_{3,4} + v_{3,4} - \lambda_3^2(v_{5,5} - 2v_{3,4} + v_{3,3})/dx^2 &= 0 \\
\tau v'_{4,4} + v_{4,4} - \lambda_4^2(v_{5,7} - 2v_{4,4} + v_{4,3})/dx^2 &= 0
\end{aligned}$$

$$\begin{aligned}
\tau v'_{5,1} + v_{5,1} + \frac{a_1 \lambda_1^2}{a_5 dx^2}(v_{5,1} - v_{1,4}) - \lambda_5^2(v_{5,2} - v_{5,1})/dx^2 &= 0 \\
\tau v'_{5,2} + v_{5,2} - \lambda_5^2(v_{5,3} - 2v_{5,2} + v_{5,1})/dx^2 &= 0 \\
\tau v'_{5,3} + v_{5,3} + \frac{a_2 \lambda_2^2}{a_5 dx^2}(v_{5,3} - v_{2,4}) - \lambda_5^2(v_{5,4} + 2v_{5,3} + v_{5,2})/dx^2 &= 0 \\
\tau v'_{5,4} + v_{5,4} + \frac{a_6 \lambda_6^2}{a_5 dx^2}(v_{6,1} - v_{5,4}) - \lambda_5^2(v_{5,5} - 2v_{5,4} + v_{5,3})/dx^2 &= 0 \\
\tau v'_{5,5} + v_{5,5} + \frac{a_3 \lambda_3^2}{a_5 dx^2}(v_{5,5} - v_{2,4}) - \lambda_5^2(v_{5,4} + 2v_{5,5} + v_{5,6})/dx^2 &= 0 \\
\tau v'_{5,6} + v_{5,6} - \lambda_5^2(v_{5,7} - 2v_{5,6} + v_{5,5})/dx^2 &= 0 \\
\tau v'_{5,7} + v_{5,7} + \frac{a_4 \lambda_4^2}{a_5 dx^2}(v_{5,7} - v_{4,4}) - \lambda_5^2(v_{5,6} - v_{5,7})/dx^2 &= 0 \\
\tau v'_{6,1} + v_{6,1} - \lambda_6^2(v_{5,4} - 2v_{6,1} + v_{6,2})/dx^2 &= 0
\end{aligned}$$

At this point we can make certain assumptions about the specific structure of Hines matrix for the LGMD neuron.

Let n be the number of daughter dendrites

1. the number of compartments in the junction dendrite are $2n - 1$
2. the *central junction* is present at the n -th compartment
3. there are n *bridge* compartments and $n - 2$ *connector* compartments
4. the first and last compartment of a junction dendrite will always be a *bridge*
5. the *bridges* occur at every off number between 1 and $2n - 1$
6. if a compartment isn't a *bridge* or *central junction*, it is a *connector*

This information allows us to efficiently write code for the one-compartment radius of the Hines matrix.

Finally, note that by observing **Fig. 1** we can see that the length of each compartment is $dx = 10\mu m$, the length of each daughter and the mother is $400\mu m$ and the width of each compartment in the junction is $10\mu m$.

$$Compartments = 20 \cdot \frac{400}{10} + \frac{390}{10} + (2 \cdot 20 - 1) \cdot \frac{10}{10} + 1 = 879$$

There are 20 compartments in each daughter, 39 compartments in the mother, 39 compartments in the junction dendrite and 1 compartment for the soma.

Coding the Hines Matrix

The following constants are used

$$\begin{aligned}
 C_m &= 1 \mu F/cm^2 \\
 g_{Cl} &= \frac{1}{15} mS/cm^2 \\
 A_s &= 400\pi \mu m^2 \\
 R_a &= 0.3 K\Omega \\
 a_i &= 1 \mu m \\
 dx &= 1 \mu m
 \end{aligned}$$

We begin with

```

N = [250 7 249 1];
Ntemplate = [4 1 1 1];

```

N represents the number of compartments in each daughter strand, junction, mother and soma in that order. **Ntemplate** represents the number of daughters, junctions, mothers and somae.

There are four sections of the code that constructs the Hines matrix; daughters, junction, mother and soma.

Daughters

The daughters follow a clear cut mostly tri-diagonal pattern encoded in *HinesSub*

```

for i=1:branch_count
    r2 = r2+ branch_len;%N(1,i);
    H_sub = HinesSub(branch_len);%
    H(r1:r2,r1:r2) = (lambda(i)^2)* H_sub;%
    r1 = r2 +1;
end

```

Junction

Within the code for the Junction there are three further cases; bridges, central junction and connectors.

```

junction_len = 2*Ntemplate(1)-1;
junction_sub = zeros(junction_len,junction_len);

```

Bridge

Within the case of the bridge, there are three cases; if the bridge is the first port, the last one or one in the middle;

```

kk = branch_count+1;
if mod(i,2)==1 % bridge
    if i==1 %start
        junction_sub(i,1) = -lambda(kk)^2;
        junction_sub(i,2) = lambda(kk)^2;
    elseif i==junction_len %end

```

```

        junction_sub(i,junction_len) = -1*lambda(kk)^2;
        junction_sub(i,junction_len-1) = 1*lambda(kk)^2;
    else %middle
        junction_sub(i,i+1) = 1*lambda(kk)^2;
        junction_sub(i,i-1) = 1*lambda(kk)^2;
        junction_sub(i,i) = -2*lambda(kk)^2;
    end
    j = (i+1)/2;
    junction_sub(i,i) = junction_sub(i,i) -r(j)*lambda(j)^2;

```

Central Junction

```

elseif i==Ntemplate(1)
    junction_sub(i,i+1) = 1*lambda(kk)^2;
    junction_sub(i,i-1) = 1*lambda(kk)^2;
    junction_sub(i,i) = -2*lambda(kk)^2;
    junction_sub(i,i) = junction_sub(i,i)-r(end)*lambda(end)^2;
    H(junction_index+i,m)= -r(end)*lambda(end)^2;

```

Connectors

```

else %connector
    junction_sub(i,i+1) = 1*lambda(kk)^2;
    junction_sub(i,i-1) = 1*lambda(kk)^2;
    junction_sub(i,i) = -2*lambda(kk)^2;

```

Then re-initialized into the main **H** matrix

```

    rj_start = junction_index+1;
    rj_end= junction_index+1+junction_len -1;
    H(rj_start:rj_end,rj_start:rj_end) = junction_sub;

```

Mother

The mother also follows a tri-diagonal structure with a few alterations.

```

    mother_len = N(end-1);
    M_sub = HinesSub(mother_len);

    rm_start = m;
    rm_end = m+mother_len-1;

    H(rm_start:rm_end,rm_start:rm_end) = (lambda(end-1)^2)*M_sub;

```

Soma


```
H(end,end-1)=rho*lambda(end)^2;
H(end,end)=-rho*lambda(end)^2;
```

We find the Hines matrix using

```
H = lgmdHines(N,Ntemplate,lambda,a, rho);
```

Finding the Eigenvalues & Eigenvectors

Once we find the raw Hines matrix we scale it

```
S = -(1/(dx^2))*H;
```

and apply the method outlined in discussed in Mathematics of Neuroscience (Gabbiani (2017)) to obtain a symmetric version of the S matrix

```
Sv = sD*S*sDi;
[Q,Z] = eig(Sv);

z = diag(Z);
[sz,si] = sort(z);

Qs = Q(:,si);
q = sDi*Qs(:,k+1);
```

The relevant sub-arrays corresponding to the first daughter, the last daughter and the mother are consequently plotted.

4. Results

The red line represents the mother, the dashed black line represents the first daughter and the dashed-red line represents the last daughter.

Sub – structure 1 Behavior

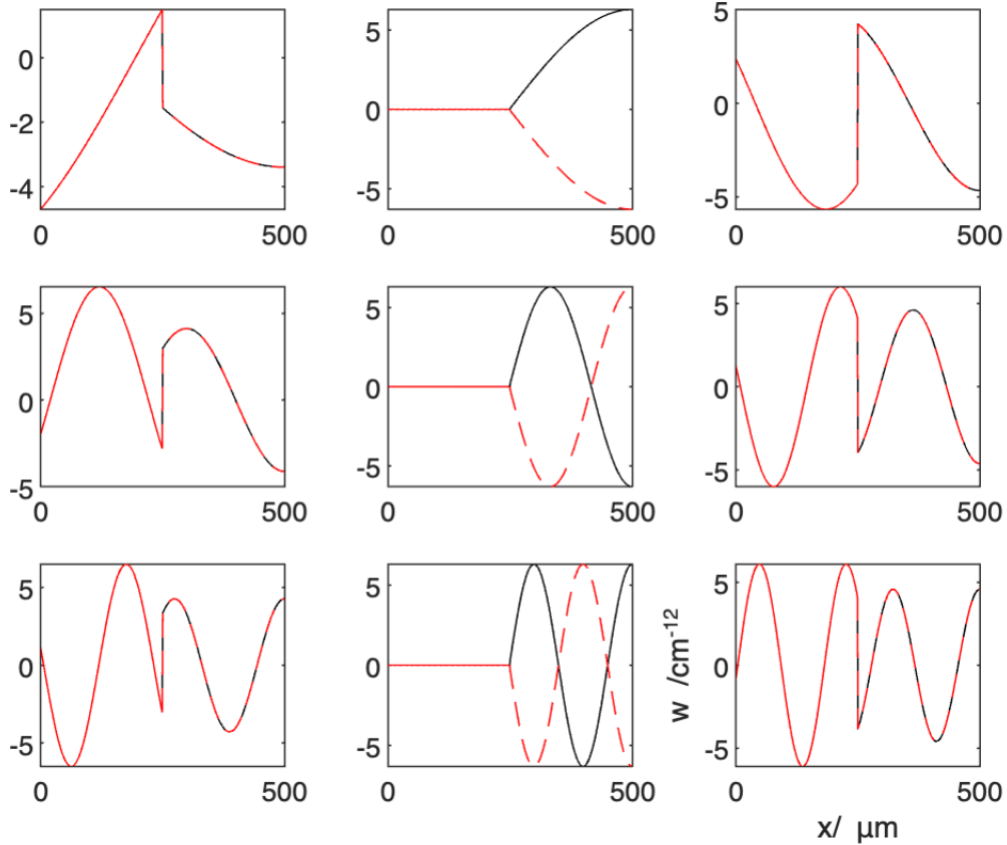


Figure 6

For simulating sub-structure 1 (**Fig. 4**) we used 250 compartments for each of the daughters, $2(2) - 1 = 3$ compartments for the junction dendrite, 249 compartments for the mother and 1 compartment for the soma.

We can see some similarities with **Fig. 1** in that, there are cases where the daughters follow the mother and others where the mother is silent and daughters are equal and opposite of each other. However, we do notice that the behavior of both the mother and the daughter changes and, in fact, while the order of the *equal – opposite* behavior remains the same - the behavior of the cases where the mother and the daughters show the same behavior become less smooth.

Sub – structure 2 Behavior

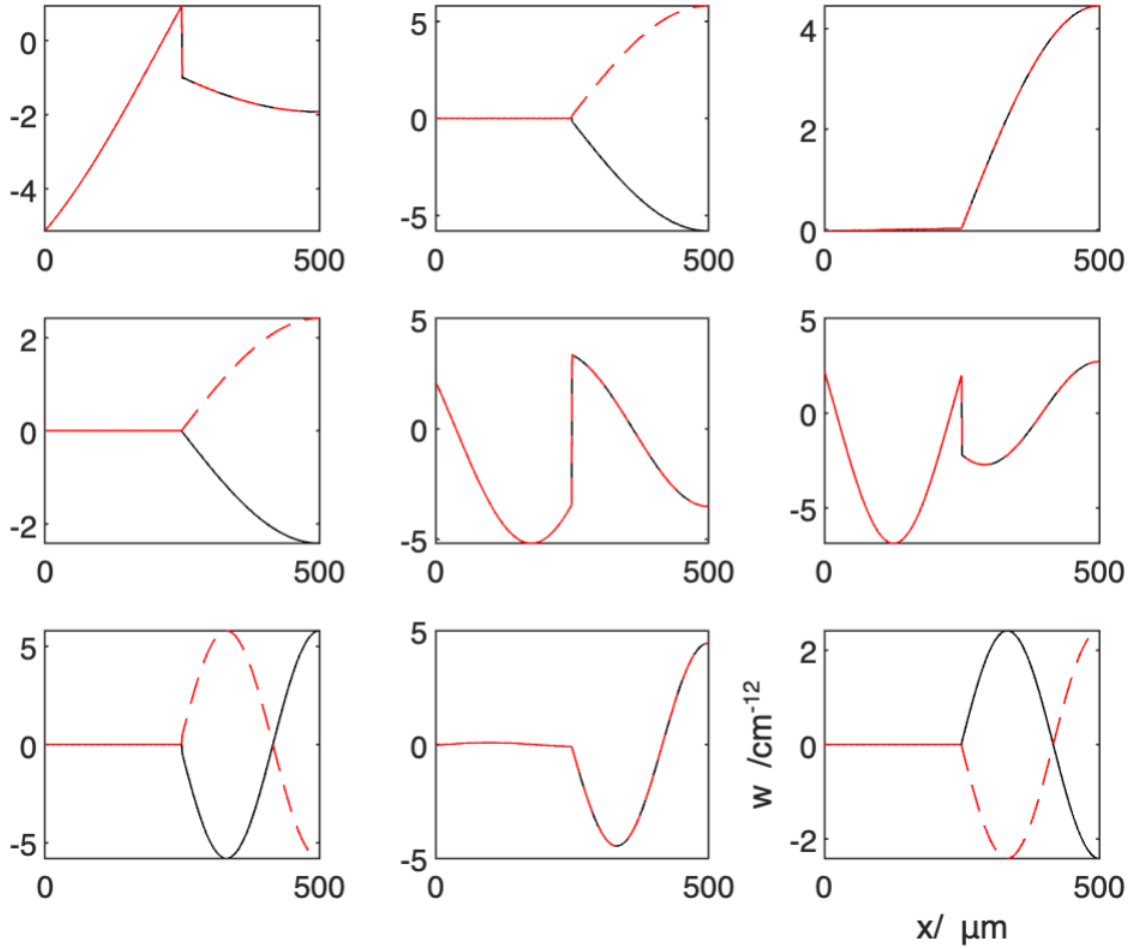


Figure 7

For simulating sub-structure 2 (**Fig. 5**) we used 250 compartments for each of the four daughters, $2(4) - 1 = 7$ compartments for the junction dendrite, 249 compartments for the mother and 1 compartment for the soma.

Just as before, we observe some equal and opposite cases and other cases where the daughters follow the mother. However, unlike the last two cases (**Fig. 1** and **Fig. 6**), these cases no longer occur in the same order of the eigenvalues. We also see some of the sinusoidal behavior being dampened or reduced in frequency.

LGMD Neuron Behavior

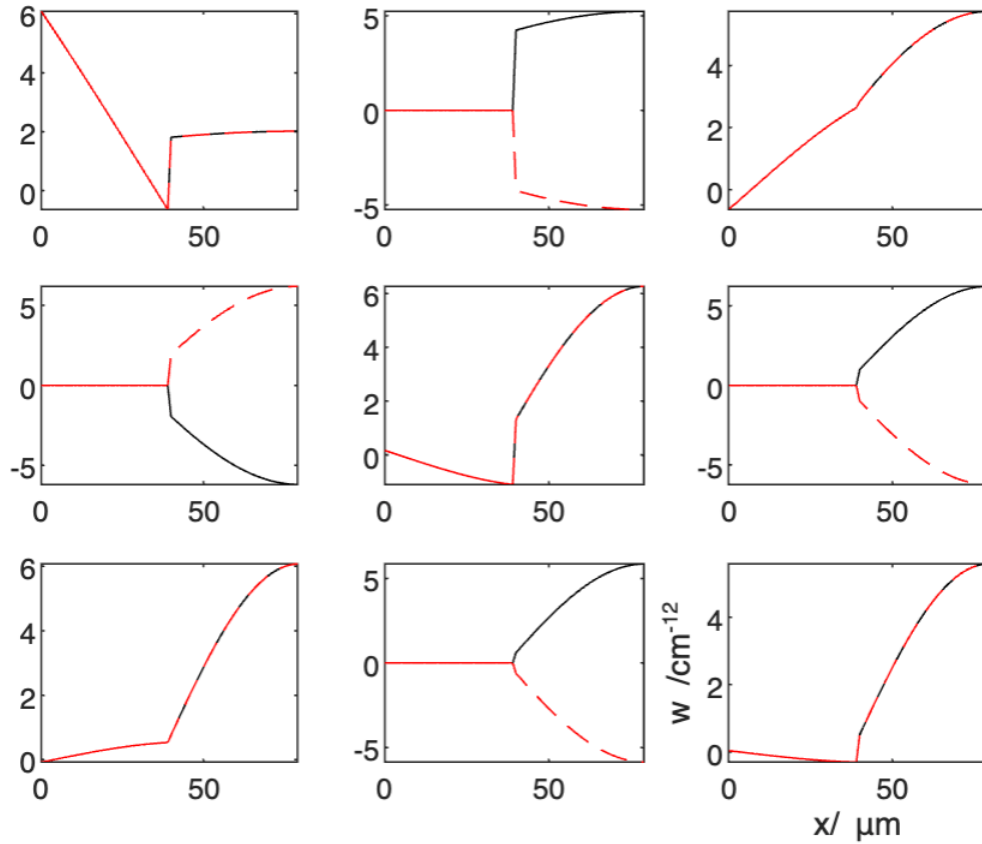


Figure 9

For simulating the LGMD neuron (**Fig. 5**) we used 40 compartments for each of the 20 daughters, $2(20) - 1 = 49$ compartments for the junction dendrite, 39 compartments for the mother and 1 compartment for the soma.

Here the order of equal and opposite behavior has changed orders again. The shift overall is more dampened and there little change in the behavior seen, at least, compared to the sinusoidal behavior in the last few figures.

Overall, we also observe a tuning effect as we increase the number of daughter dendrites. If we begin with **Fig.** we see smooth sinusoidal behavior. This continually decreases as we increase the number of dendrites and when we observe the LGMD neuron - almost any semblance of sinusoidal behavior is gone.

5. Conclusion

In this paper, we propose an LGMD model, consisting of the Hines matrix, to obtain the eigenvalue decomposition of an LGMD neuron. The model requires little additional computational cost as we build it up from the existing simple forked model by using the current balance equation to tweak the cell circuit.

Once encoded, we were able to observe the behavior of the the daughters (first and last) with the mother in all of sub-structures. One of the major similarities we observed despite the increase in the number of

dendrites was that there were always instances of the mother being silent and the daughters being equal and opposite to each other. There were also always cases of the daughter following the path of the mother.

In terms of difference, we noticed that the order in which the 'equal and opposite' cases occurred was not necessarily retained. Also, there was a very visible dampening effect on the initial sinusoidal behavior we saw in the compartmentalization of the branched cell.

In the future, we can examine the change in behavior through the mother and the daughters by slowly changing the structure of the LGMD neuron. This would allow us to see to what degree the unique qualities of the LGMD neuron rely on its very specific rake like structure. Secondly, we can also try to increase the number of daughter dendrites to and observe the cascading effect on the conductance to perhaps understand better what the optimal number of daughter dendrites are.

6. Appendix

```

1 %% CAAM 615 Final Project
2 %% Author: Javeria Ahmed
3
4 % Constants
5 dx = 10^-4;
6 As = 400*pi*10^-8;
7 R_a = 0.3;
8 g_cl = 1/15; C_m = 1;
9
10
11 N = [250 7 249 1];
12 % N(1) number of compartments in each daughter strand
13 % N(2) 2N-1 number of compartments in junction where is the number of
14 %     daughter strands
15 % N(3) number of compartments in mother
16 % N(4) number of compartments in soma, usually one.
17
18 Ntemplate = [4 1 1 1];
19
20 % Ntemp(1) number of daughter strands
21 % Ntemp(2) 1 junction
22 % Ntemp(3) 1 mother
23 % Ntemp(4) 1 Soma
24
25 num_strand = sum(Ntemplate);
26 branch_count = Ntemplate(1);
27
28
29 a = 1e-4*ones(num_strand,1);
30 A3 = 2*pi*a(end-1)*dx; rho = A3/As;
31 lambda = sqrt(a./(2*R_a*g_cl));
32
33 H = lgmdHines(N,Ntemplate,lambda,a, rho);
34 S = -(1/(dx^2))*H;
35
36 % D Transformation Matrix Construction
37 Db = [];
38 for i=1:branch_count
39     Db = [Db a(i)*ones(1,N(1))];
40 end
41
42 kk=0;
43 for i=(branch_count+1):(num_strand-1)
44     Db = [Db a(i)*ones(1,N(2+kk))];
45     kk=kk+1;
46 end

```

```

47 Db = [Db a(end-1)/rho];
48
49 D = diag(Db);
50 sD = diag((Db).^(1/2));
51 sDi = diag((Db).^(-1/2));
52
53 % Transformation to Symmetric Matrix
54 Sv = sD*S*sDi;
55
56 % Eigenvalue and Eigenvectors of Sv
57 [Q,Z] = eig(Sv);
58 z = diag(Z);
59 [sz,si] = sort(z);
60 Qs = Q(:,si);
61
62 x1 = N(end-1):(N(end-1)+N(1));
63 x2 = N(end-1):(N(end-1)+N(1));
64 x3 = 0:N(end-1);
65
66 for k=1:9
67     q = sDi*Qs(:,k+1); %eigenvectors of H
68
69     q1 = fliplr(q(1:N(1))'); %first daughter
70     q2 = fliplr(q(((branch_count-1)*N(1)+1):branch_count*N(1))'); %last strand
71     q3 = fliplr(q(branch_count*N(1)+N(2)+1:end)'); %mother
72
73     subplot(3,3,k)
74     plot(x3,q3,'r')
75     hold on
76     plot(x2,[q3(end) q2'],'k')
77     plot(x1,[q3(end) q1'],'r--')
78     hold off
79
80 end
81 xlabel('x/ \mum')
82 ylabel('w /cm^{-12}')

```

```

1 %% CAAM 615 Final Project
2 %% Author: Javeria Ahmed
3
4 function H = lgmdHines(N,Ntemplate,lambda,a, rho)
5
6 total_len = N*Ntemplate';
7 H = zeros(total_len,total_len);
8 branch_count = Ntemplate(1);
9 branch_len = N(1);
10 r = a./a(branch_count+1);
11
12 %% Daughter Branches
13 r1 = 1; r2 = 0;
14 for i=1:branch_count
15     r2 = r2+ branch_len;%N(1,i);
16     H_sub = HinesSub(branch_len);%
17     H(r1:r2,r1:r2) = (lambda(i)^2)* H_sub;%
18     r1 = r2 +1;
19 end
20 junction_index = N(1)*Ntemplate(1);
21 for i=1:branch_count
22     H(i*branch_len,2*i-1+junction_index) = (lambda(i)^2); %%*
23 end
24
25 %% Junction
26
27 junction_len = 2*Ntemplate(1)-1;
28 junction_sub = zeros(junction_len,junction_len);

```

```

29 m = N(1)*Ntemplate(1)+junction_len+1;
30
31 for i=1:junction_len %this iterates over the rows
32     kk = branch_count+1;
33     if mod(i,2)==1 % bridge
34         if i==1 %start
35             junction_sub(i,1) = -lambda(kk)^2;
36             junction_sub(i,2) = lambda(kk)^2;
37         elseif i==junction_len %end
38             junction_sub(i,junction_len) = -1*lambda(kk)^2;
39             junction_sub(i,junction_len-1) = 1*lambda(kk)^2;
40         else %middle
41             junction_sub(i,i+1) = 1*lambda(kk)^2;
42             junction_sub(i,i-1) = 1*lambda(kk)^2;
43             junction_sub(i,i) = -2*lambda(kk)^2;
44         end
45         j = (i+1)/2;
46         junction_sub(i,i) = junction_sub(i,i) -r(j)*lambda(j)^2;
47
48
49     elseif i==Ntemplate(1) % central junction
50
51         junction_sub(i,i+1) = 1*lambda(kk)^2;
52         junction_sub(i,i-1) = 1*lambda(kk)^2;
53         junction_sub(i,i) = -2*lambda(kk)^2;
54         junction_sub(i,i) = junction_sub(i,i)-r(end)*lambda(end)^2;
55         H(junction_index+i,m)= -r(end)*lambda(end)^2;
56
57
58     else %connector
59
60         junction_sub(i,i+1) = 1*lambda(kk)^2;
61         junction_sub(i,i-1) = 1*lambda(kk)^2;
62         junction_sub(i,i) = -2*lambda(kk)^2;
63
64     end
65 end
66
67 for i=1:branch_count
68     H(2*i-1+junction_index,i*branch_len) = r(i)*(lambda(i)^2); %%
69 end
70
71 rj_start = junction_index+1;
72 rj_end= junction_index+1+junction_len -1;
73 H(rj_start:rj_end,rj_start:rj_end) = junction_sub;
74
75 %% Mother
76 %
77 mother_len = N(end-1);
78 M_sub = HinesSub(mother_len);
79 M_sub(1,1) =-2;
80 M_sub(end,end) =-2;
81
82 rm_start = m;
83 rm_end = m+mother_len-1;
84
85 H(rm_start:rm_end,rm_start:rm_end) = (lambda(end-1)^2)*M_sub;
86 H(m,junction_index+Ntemplate(1))= -lambda(end-1)^2;
87 H(end-1,end) = lambda(end-1)^2;
88 %% Soma
89 H(end,end-1)=rho*lambda(end)^2;
90 H(end,end)=-rho*lambda(end)^2;
91
92 end

```

```
1 %% CAAM 615 Final Project
2 %% Author: Javeria Ahmed
3
4 function H_sub = HinesSub(N)
5 H_sub = diag(-2*ones(1,N)) + diag(ones(1,N-1),1) + diag(ones(1,N-1),-1);
6 H_sub(1,1)=-1;
7 end
```