Progress Report

[Two-Component HVC(RA) Neuron Modeling System]

Summary

[I have designed and began implementation of a general c++ system strucutre in order to model systems of ODEs. This system also includes the functionality to generate octave scripts to export data for visualization. Furthermore, I have implemented classes in order to model the two-compartment HVC(RA) neuron model, as well as the Integrate-and-Fire model for testing purposes. Unfortunately, I have run into some issues which verification of my implementation of the 4th-order Runge-Kutta method for solving the represented ODE systems.]

Completed Tasks

- Implemented HVC(RA) neuron model
- Implemnnted Integrate-and-Fire neuron model
- Implemented the data export framework Exporto.

Current Work

- Verification of RK4 method/neuron models.
- Further revisions to Exporto

Upcoming Tasks

- Implement ability to construct neuron networks.
- Replicate HVC(RA) Figures presented in Jin 2009

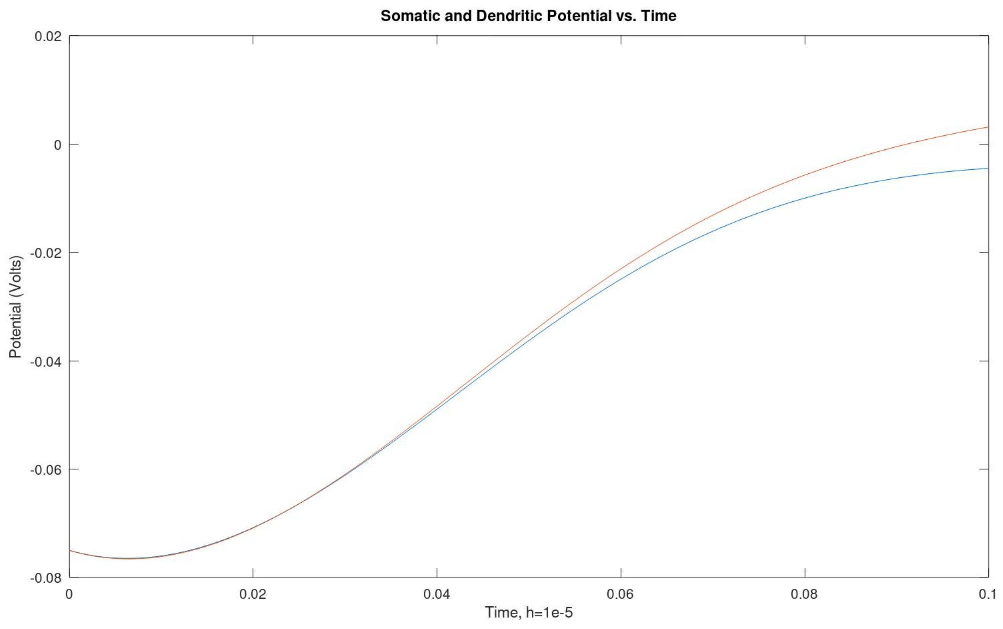
Challenges and Issues

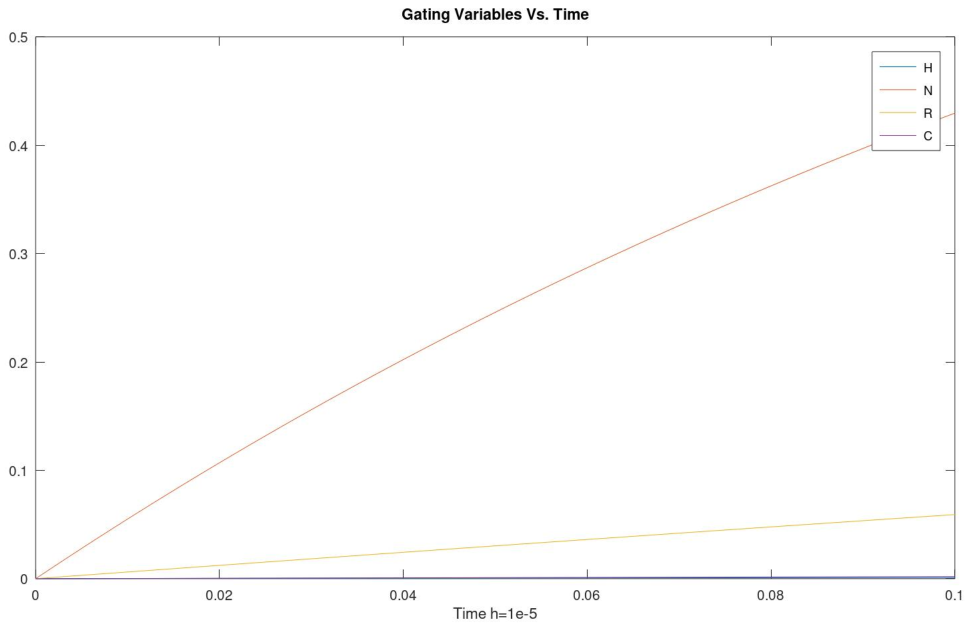
The main challenge I have arrived at is in the verification of my implementation of the 4-th order Runge Kutta method. For my implementation I adapt the code included in Ch.16 of A First Course in Nunerical Methods (Ascher, Uri M., Greif, Chen)

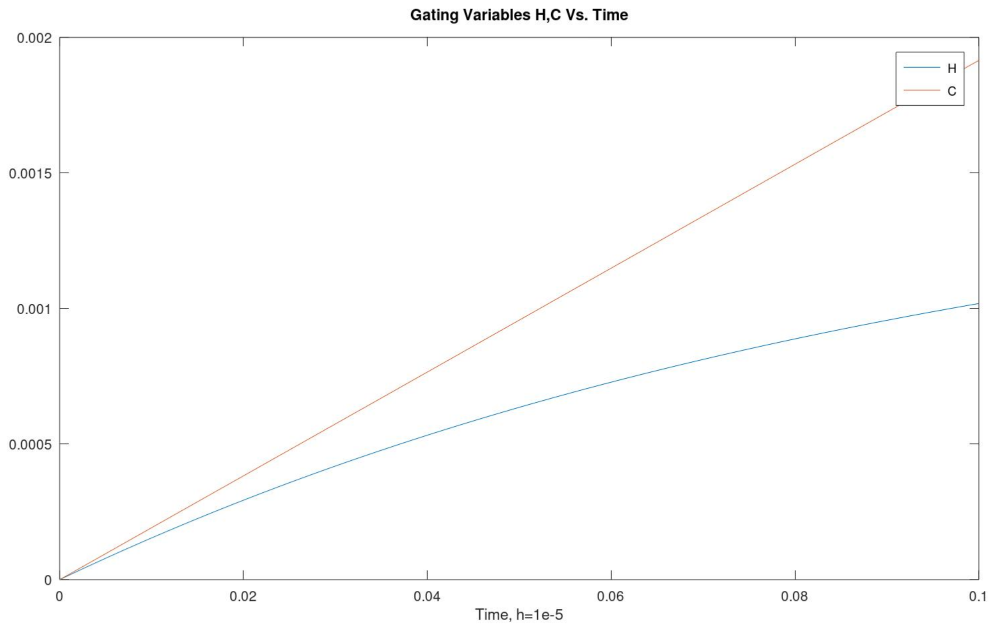
```
% Integrate
for i=1:N
% Calculate the four stages
K1 = feval(f, t(i),y(:,i) );
K2 = feval(f, t(i)+.5*h, y(:,i)+.5*h*K1);
K3 = feval(f, t(i)+.5*h, y(:,i)+.5*h*K2);
K4 = feval(f, t(i)+h, y(:,i)+h*K3 );
% Evaluate approximate solution at next step
y(:,i+1) = y(:,i) + h/6 *(K1+2*K2+2*K3+K4);
end
```

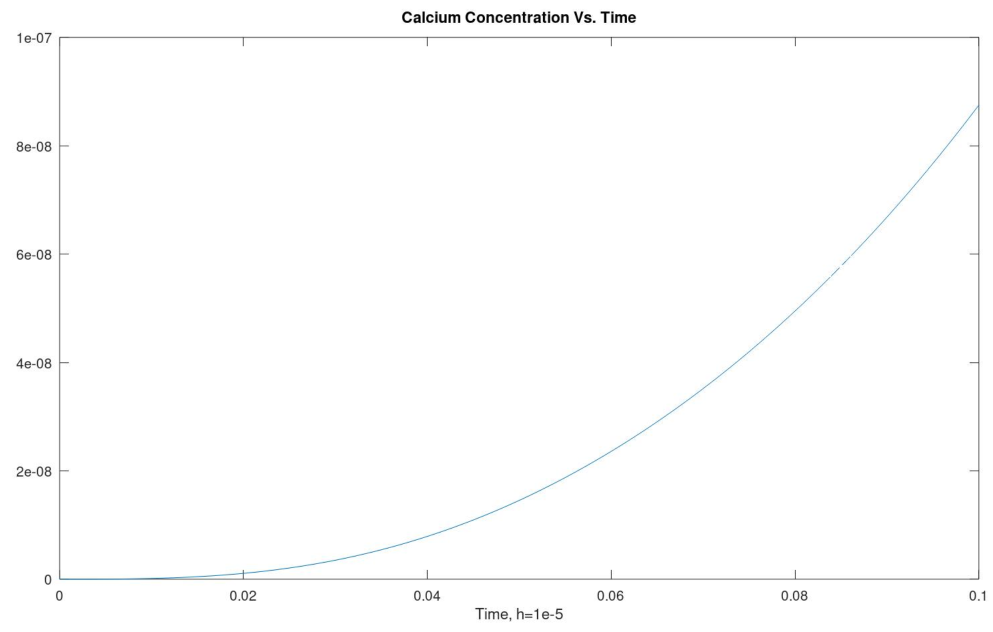
I have used this section of their code to implement the step portion of the RK4 method. After not obtaining proper stabilization results from initial simultations of the HVC(RA) model, I began exploring verification of this method via use of the simpler Integrate-and-Fire method.

Current Results



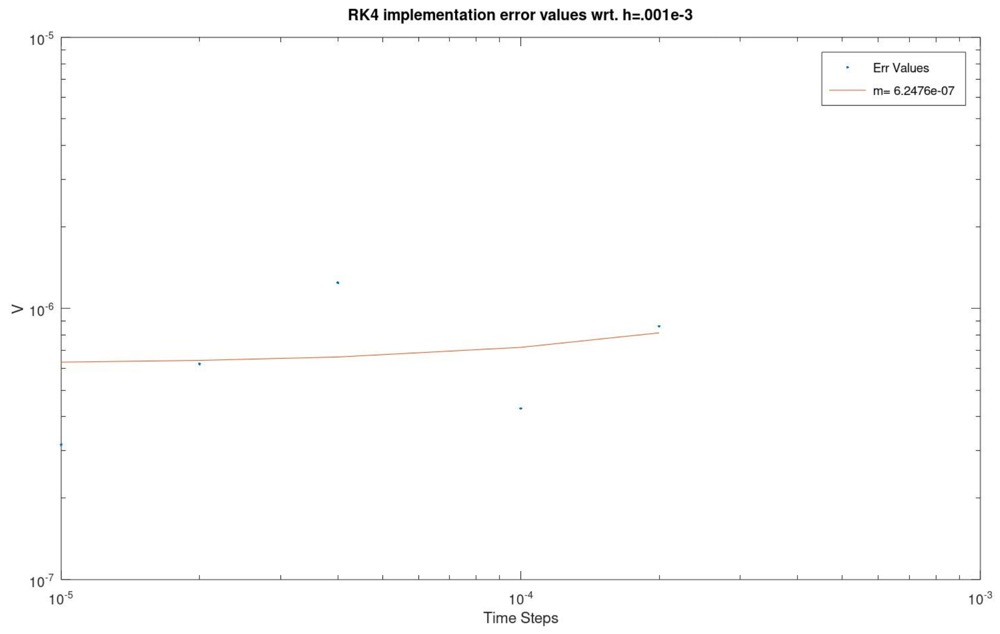






Error Verification Via Integrate-and-Fire Model

In order to attempt to diagnose the problems within this system. I first attempted to verify the error of my method via simulating the much simpler IF neuron model as described in the Training Module Assignment. Model - This consists of one differential equation for membrane potential $\tau\frac{dV}{dt} = L - V + I_0(1 + \sin t)$. $L = -70mV$ being the resting membrane potential, $I_0 = 5mV$ being the injected current, $\tau = 20ms$ being our time constant. - We simulate this model for 40ms with the initial condition $V(0) = L$ for timesteps $.01, .02, .04, .1, .2, .001ms$. - We then plot the error of the final value of V between the larger timesteps and $.001ms$ on a loglog graph and fit a linear function to the graph.



- We should expect our error to follow a linear pattern on a log scale, but that is not what we see in this case

Notes

- Something about the gating variables (specifically R, N) having such large values compared to H, C does not seem right.