Diffusion of Acetylcholine in the Synaptic Cleft

October 29, 2010

BENG 221

Thomas Hagan
Matt Janssen
Ryan LaCroix
Nate Vacanti
Kevin Vincent

Table of Contents

Introduction

A Synapse allows for functional connections between two neurons or between a neuron and another cell type such as a muscle or gland; for this reason, they are essential for communication between the nervous system and other parts of the body. This process begins as an action potential propagates along the length of the axon until it reaches the presynaptic membrane; the signal depolarizes the membrane allowing calcium ions to flow into the presynaptic cell. The influx of calcium causes the neurotransmitters, which are bundled inside vesicles, to move toward the synapse. Upon binding with the presynaptic membrane, the vesicles release the neurotransmitters into the synapse where they will hopefully bind to neurotransmitter receptors on the postsynaptic membrane. Alternative neurotransmitter fates include escape from the synaptic cleft or enzymatic degradation. The binding of the neurotransmitter activates the receptor and generates an excitatory or inhibitory postsynaptic potential once the binding threshold has been reached. After activation, the neurotransmitter is released from the receptor and is either degraded in the cleft or reabsorbed by the presynaptic cell.[2] This process is shown in Figure 1.[1]

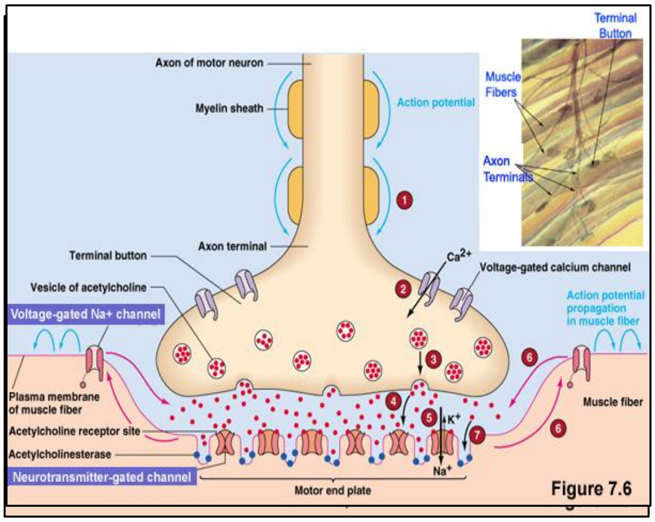


Figure 1: Acetylcholine diffusion across synapse[1]

The neurotransmitter acetylcholine plays a vital role in the peripheral nervous system during muscle movement as well as in the central nervous system where it is believed to affect an individual's learning and memory. As previously mentioned, neurotransmitters can be enzymatically degraded within the synapse; acetylcholinesterase hydrolytically degrades acetylcholine into acetate and choline.[3] Several prescription drugs take advantage of this mechanism to treat diseases such as Alzheimer's and myasthenia gravis. Patients suffering from Alzheimer's disease have damaged acetylcholine receptors, which results in memory loss, mood swings, and language degeneration. Acetylcholinesterase inhibitors are employed to slow the degradation rate and allow for more acetylcholine to bind to the postsynaptic receptors.[4]

Sarin gas also targets acetylcholinesterase inhibitors, but the level of inhibition is much greater. Sarin gas is a highly volatile nerve agent that is both colorless and odorless; its devastating effects on the nervous system make it an extremely potent chemical weapon. The mechanism utilizes competitive inhibition to bind to the active site of acetylcholinesterase and renders it biologically inactive. Since there is no method for acetylcholine to be removed from the synaptic cleft, it continues to bind to receptors and send signals. As a result, the affected individual begins to convulse as the muscles seize up. Eventually, the synaptic membranes depolarize and halt all receptor activation causing the individual to lose control of muscle function. Long-term exposure can lead to paralysis, respiratory failure, and death.[5, 6]

Computational models can provide insight into the complex dynamics of acetylcholine diffusion during normal and pathological states. They also provide an opportunity to test potential pharmaceutical interventions *in-silico*. The goal of this project was to create a simple model of acetylcholine diffusion across a synapse and determine how this diffusion is altered by sarin gas. Three levels of modeling were performed. The first was an analytical solution for acetylcholine diffusing in one dimension. The next level numerically approximated the one dimensional diffusion equation in MATLAB. Lastly, the diffusion of acetylcholine from a vesicle across the synaptic cleft was modeled in a representative two dimensional geometry using Finite Element Analysis in MATLAB.

Set-up

For the analytical solution and one dimensional numerical approximation, the diffusion of acetylcholine across the synaptic cleft was modeled as a slab with the presynaptic neuron located at X=0 and the postsynaptic neuron at X=L. The basic setup for the problem can be seen in Figure 2 and all required constant values are listed in Table 1. The differential equation governing this problem is given as Equation 1, with boundary conditions given as Equation 2 and Equation 3, and the initial condition given as Equation 4.

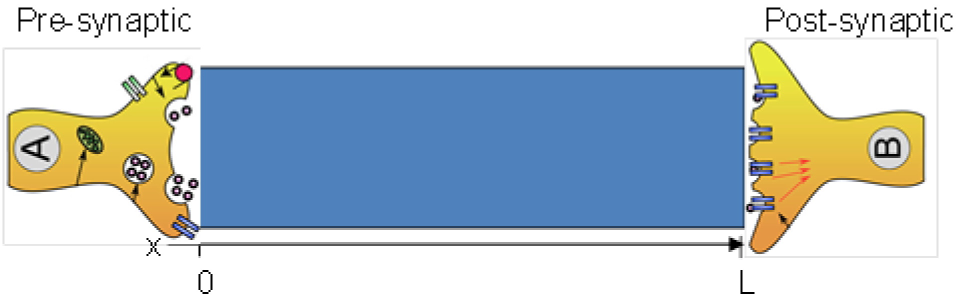


Figure 2: Simplified diagram of synapse illustrating problem setup

$$\frac{\partial C_{(t,x)}}{\partial t} = D\frac{\partial^2 C_{(t,x)}}{\partial x^2} - kC_{(t,x)} \qquad (1)$$

$$\left.\frac{\partial C}{\partial x}\right|_{(t,x=0)} = 0 \qquad (2)$$

$$-D\left.\frac{\partial C}{\partial x}\right|_{(t,x=L)} = \beta C_{(x=L,t)} \qquad (3)$$

$$C_{(t=0,x)} = C_s\delta_{(x)} \qquad (4)$$

Table 1: Values for the constants shown in Figure 2

Constant Description	Variable	Value
Diffusivity of acetylcholine [7]	D	400 µm^2/s
Synaptic cleft distance [8]	L	20nm
Acetylcholinesterase rxn rate [9]	k	2333 s^{-1}
Acetylcholine receptor binding rate [9]	β	4.7 E 7 M^{-1}s^{-1}
Initial [acetylcholine] released [10, 11]	C_i	0.006 mmol/m^2

Synapse shape [11]	-	r = 150nm
		L = 20nm

At the presynaptic neuron, no reabsorption was modeled thus we had a no flux boundary condition (Eq. 2). At the X=L boundary acetylcholine bound to the ligand gated ion channels. When bound, the acetylcholine was no longer free to contribute to the overall concentration of acetylcholine. Thus, at the postsynaptic neuron, a Robin boundary condition (Eq. 3) was utilized. Furthermore, to model the degradation of acetylcholine by acetylcholinesterase, there was a general consumption term that consumed according to a standard first order reaction. To model the release of acetylcholine from the presynaptic neuron, an initial condition with a delta function was used (Eq. 4).

In modeling the synaptic cleft in this manner, some key assumptions were made.

Assumptions:

1) The concentrations of acetylcholine are of a magnitude large enough that diffusion can be looked at in a continuous manner. Unfortunately, due to very low number of neurotransmitter molecules released (~3000 molecules) a continuous model may not be ideal. To make this solvable within the scope of the diffusion equation, this assumption was made.
2) The neurotransmitters cannot diffuse out of the synaptic cleft. Once again, physiologically speaking, this is not ideal since surrounding the synaptic cleft there are typically glial cells which are able to absorb neurotransmitters. However, this was assumed to be negligible compared to the degradation of acetylcholine.
3) Degradation of acetylcholine by acetylcholinesterase is a first order reaction. This assumption ensures that our problem is linear and thus solvable. This assumption is quite accurate as long as inhibitors are not involved. For the pathological study, in which the inhibition of acetylcholinesterase is modeled, the reaction term was eliminated.

Numerical Analysis Set-up: PDEPE

MATLAB's numerical partial differential equation solver is PDEPE. It solves partial differential equations (PDEs) in the form shown below.

$$c(x,t,u,\frac{\partial u}{\partial x})\frac{\partial u}{\partial t}=\frac{\partial}{\partial x}(f(x,t,u,\frac{\partial u}{\partial x}))+s(x,t,u,\frac{\partial u}{\partial x}). \qquad (5)$$

To match the PDE for this problem (Eq. 1) to this form, c=1, $f=D\frac{\partial C}{\partial x}$, and s=$-kC$, where D is the diffusion coefficient of acetylcholine, and k is the rate constant of the degradation of acetylcholine catalyzed by acetylcholinesterase. These coefficient values were used in the sub-function pdex1pe to define the PDE.

The initial condition in this case was a delta function. Although MATLAB has no delta function for numerical purposes, the delta function can be approximated by making the initial condition vector such that C=Ci at x=0 and C=0 at all other *x* data points. This was done using an "if-else statement" in the initial condition sub-function, pdexlic. Since there is a finite number of x points used in the simulation, this method only approximates the delta function initial condition.

PDEPE also requires that the boundary conditions satisfy the form in Equation 6.

$$p(x,t,u)+q(x,t)f(x,t,u,\frac{\partial u}{\partial x})=0, \qquad (6)$$

where $f=D\frac{\partial C}{\partial x}$.

In this problem, the boundary conditions were that there was no flux of acetylcholine at the left synapse boundary (Eq. 2), and that the flux at the right synapse boundary was proportional to the amount of acetylcholine present (Eq. 3). To write these boundary conditions in the form of Equation 6, p=0 and q=1 at the left boundary and $p=-\beta C_R$and q= -1 at the right boundary, where β is the binding rate of the acetylcholine receptor, and C_R is the concentration of acetylcholine at the right boundary. These coefficients were used in the sub-function pdex1bc to define the boundary conditions.

The main purpose of implementing the numerical solution in this problem was to find the flux of acetylcholine out of the right boundary. This corresponds to the amount of acetylcholine bound to receptors on the postsynaptic membrane and this is what ultimately activates the neural response. By estimating the integral of the flux at the right boundary, the approximate number of acetylcholine molecules bound to the receptors per unit area was determined. In order to implement this in MATLAB, a "for loop" was used to approximate the flux at each time point as follows:

$$J \approx -D\frac{C(t, x = L) - C(t, x = L - \Delta x)}{\Delta x}, \qquad (7)$$

where Δx is the distance between *x* points in the simulation (0.5nm in this case).

The integral of the flux at each successive time point was then approximated by multiplying the flux by the time step and adding it to the previous value of the integral:

$$\int_0^t J\,dt \approx \sum_{i=1}^{t/\Delta t} J_i \Delta t\ . \qquad (8)$$

This approximated integral represents the bound acetylcholine molecules per unit area as a function of time, and was plotted as such. Appendix B contains the MATLAB code used to implement PDEPE.

Numerical Analysis Set-up: Finite Element Analysis (FEA)

The MATLAB PDE Toolbox provides a graphical user interface (GUI) to construct and numerically approximate the solution to PDEs in two spatial dimensions using Finite Element Analysis (FEA). FEA constructs a network of triangles, called a mesh, from the defined geometry and approximates the solution to the PDE as a system of ordinary differential equations at each of the nodes of the mesh.

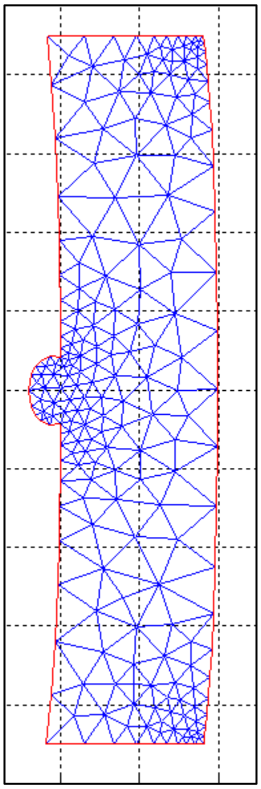


Figure 3: FEA Geometry and Mesh

The PDE Toolbox can be opened by calling pdetool in the command window. The geometry generated for our simulation, shown in Figure 3, contained a vesicle fusing to the presynaptic membrane (left side), the post synaptic membrane (right side), and the edges of the synaptic cleft (top and bottom). As with the previous methods, a no flux boundary condition was given for the presynaptic membrane and a Robin boundary condition was defined for the postsynaptic membrane. The edges of the synaptic cleft were defined as a no flux boundary condition. Figure 3 also contains the mesh generated by the PDE Toolbox. The geometry, boundary conditions, and mesh were exported to the MATLAB workspace for additional analysis.

A MATLAB script was used to add the initial conditions and consumption terms. The initial condition was a high concentration of acetylcholine inside the merging vesicle and zero elsewhere (Figure 4). For the normal synaptic physiology simulation, a consumption term was used to represent the effects of acetylcholinesterase. In the pathological sarin gas simulation, the consumption term was removed. The function parabolic was used to iteratively approximate the PDE solutions for multiple action potentials. For each subsequent action potential, identical initial conditions were added to the solution from the previous action potential.

Appendix C provides a more detailed explanation of the set-up using the PDE Toolbox, and Appendix D contains the MATLAB code for

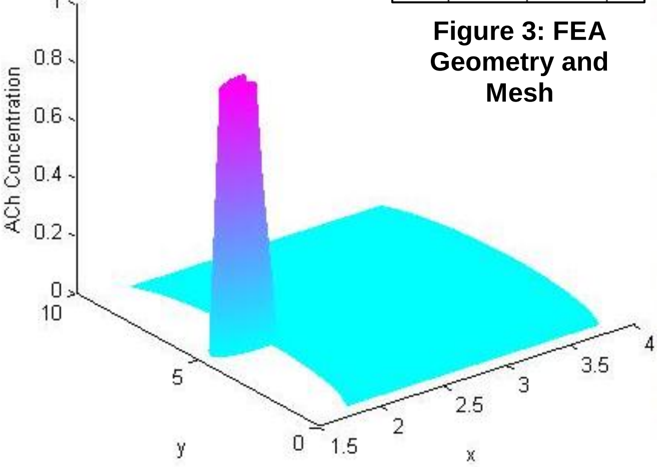


Figure 4: FEA Initial Conditions

implementing the construction and analysis of the FEA solution.

Solution

Analytical Solution

See Appendix A for the derivation of this solution.

$$C_{(t,x)} = \frac{2C_s}{L}\sum_{n=1}^{\infty}\sqrt{\left[1-\frac{1}{2\lambda_n}\sin(2\lambda_n L)\right]}\,e^{-(D\lambda_n^2+k)t}\cos(\lambda_n x) \tag{9}$$

$$-\lambda_n sin(\lambda_n L)+\frac{\beta}{D}cos(\lambda_n L)=0 \tag{10}$$

Numerical Solution & Observations: PDEPE

A three-dimensional plot illustrating how the concentration of acetylcholine in a normal synapse fluctuates with space and time can be seen in Figure 5.

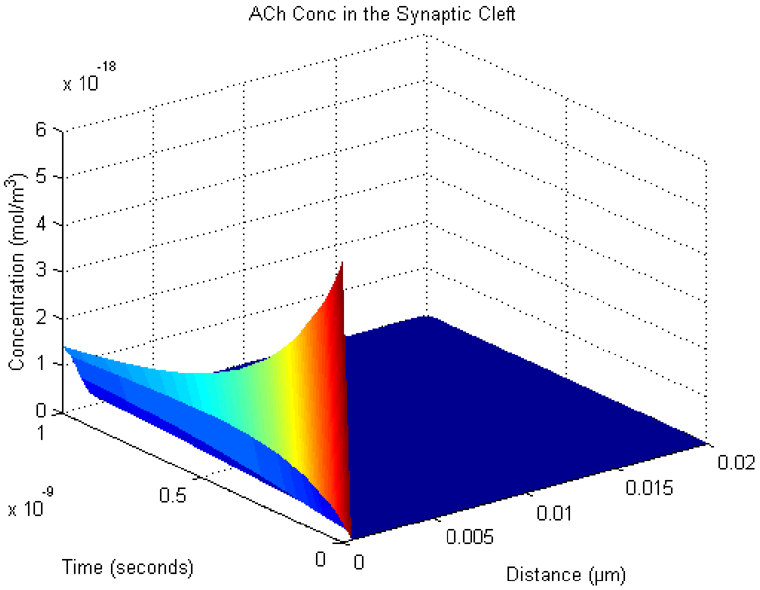


Figure 5: 3D plot of the concentration of acetylcholine as a function of distance and time in a normal synapse

The pathological study in which the presence of sarin gas inhibits the degradation of acetylcholine was also simulated by setting the consumption term coefficient in the PDE equal to zero. When the two cases were simulated, it was clear that the presence of sarin gas results in a much larger number of acetylcholine molecules bound to receptors on the postsynaptic membrane. Also, the time required to reach steady state was greater with sarin present than in the normal state. The number of acetylcholine molecules per unit area that bound the receptors in a normal synapse and a sarin-affected synapse can be seen in Figure 6.

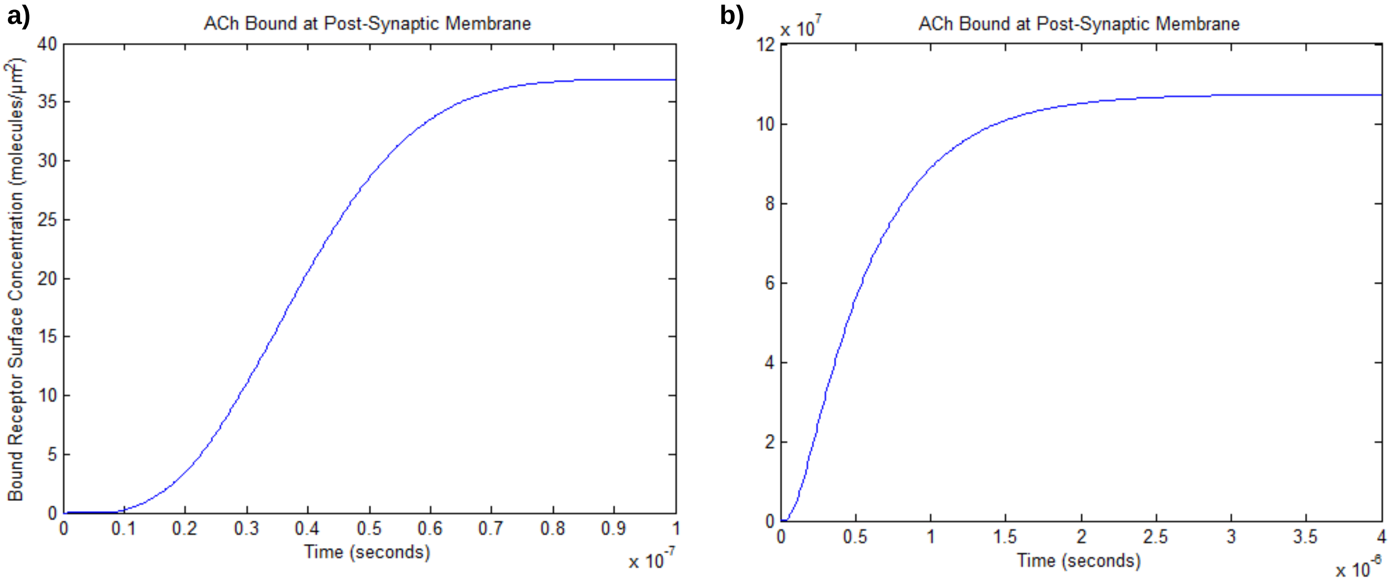


Figure 6: Number of bound acetylcholine molecules over time in the a) normal and b) sarin gas synapse

Numerical Solution & Observations: Finite Element Analysis

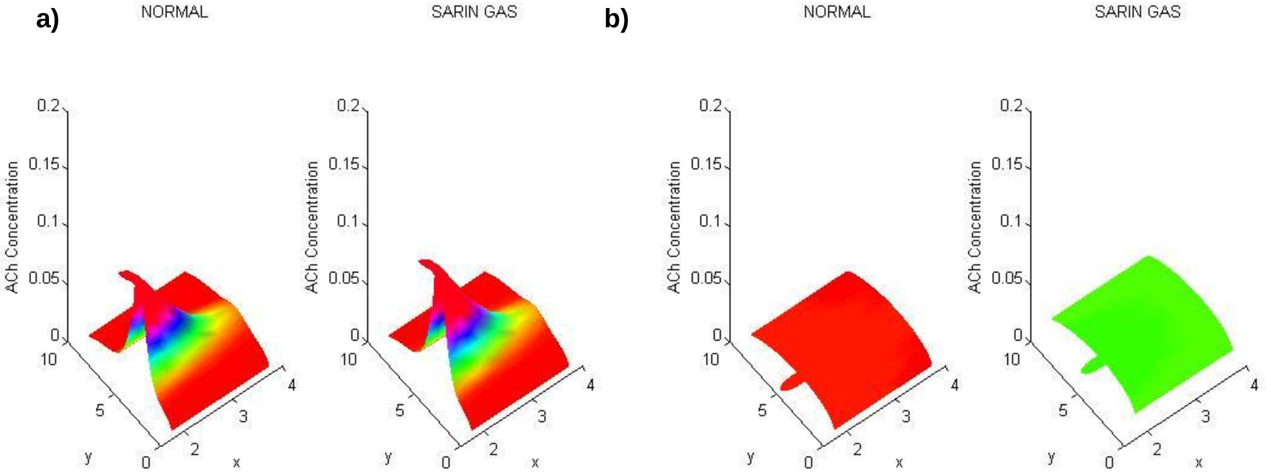


Figure 7: FEA solution a) during the first action potential and b) after the fourth action potential

Appendix E (the attached CD) contains a movie of FEA solution, and Figure 7 contains two screenshots from that movie. In Figure 7a the high concentration of acetylcholine in the vesicle can be seen diffusing throughout the synaptic cleft. Figure 7b shows acetylcholine concentration returning to zero in the synaptic cleft for the normal simulation and building up in the synaptic cleft of the sarin gas simulation. This result is consistent with pathophysiology of sarin gas. The post synaptic cell will fire more rapidly due to the high acetylcholine concentration and will eventually saturate the cell.

Conclusions

Three models of increasing complexity were created to examine the diffusion of acetylcholine in the synaptic cleft and how the concentration levels were affected by sarin gas. While the analytical solution showed that a closed form solution to this problem is possible, it was laborious to derive and did not provide much insight into the dynamics of acetylcholine diffusion in the normal and sarin gas states. The numerical simulations were approximations of the solution, but they allowed for multiple iterations and additional analysis to examine the effects of sarin gas. By integrating the flux at the postsynaptic membrane, the PDEPE solution demonstrated that acetylcholine builds up at the postsynaptic membrane faster for the sarin gas simulation. The FEA solution demonstrated that sarin gas causes the acetylcholine to build up in the synaptic cleft over multiple action potentials. Both of these results are consistent with the pathophysiology of sarin gas exposure. The build-up of acetylcholine over-stimulates the postsynaptic cell, causing it to rapidly depolarize first, before eventually saturating.

Future Work

While the models provided some insight into acetylcholine diffusion, they were very simple portrayals of an extremely complicated process. Much more sophisticated models would be required to be useful in developing novel hypothesis and testing potential pharmaceutical interventions *in-silico*.

Future work could relax some of the assumptions to provide a more realistic model. One assumption was that neurotransmitters cannot diffuse out of the synaptic cleft. This diffusion out of the synaptic cleft and absorption by the glial cells play a large role in the regulation of neurotransmitter concentration. Thus, to approach a more realistic model, this would have to be included. This would however preclude the use of a one dimensional slab model and involve at least a two-dimensional problem with two additional Robin boundary conditions at the additional boundaries to model the glial cell uptake of neurotransmitters. With the proper constant values, this could be incorporated into the FEA solution. This would be particularly important in the sarin gas simulations where the acetylcholine has no other means of leaving the synaptic cleft.

Additionally, this problem could be adapted to incorporate the fate of the neurotransmitters after activating the postsynaptic receptor. This model stopped once the molecules adhered to the receptor surface. However, in reality, these molecules are eventually released from the receptor and either reabsorbed by the presynaptic membrane for later use or enzymatically degraded within the cleft.

One important assumption was the use a continuous model of diffusion for neurotransmitters in the synaptic cleft. A more realistic model would employ the use of individual molecule models with a random walk distribution that follows Brownian motion. However, this would not be an addition to the previously proposed model, but would require completely new design. If this model were created, the time required for a postsynaptic receptor to reach the threshold value for activation could be ascertained.

A sophisticated simulation of this process would likely require a multi-scale model. On the global level this model could incorporate additional relevant molecules and biochemical pathways in a more realistic three dimensional geometry using Brownian motion to dictate their movements. On the protein level, the model could use statistical properties to dictate interactions between molecules. A model of this complexity could have enormous potential as a way to preliminarily investigate novel pharmaceutical agents for diseases like Alzheimer's, myasthenia gravis and depression.

References

1. http://www.onlineanaesthesia.com/images/nmjunction.jpg.
2. http://www.ncbi.nlm.nih.gov/bookshelf/br.fcgi?book=neurosci&part=A321.
3. http://www.chemistryexplained.com/A-Ar/Acetylcholine.html.
4. http://www.biopsychiatry.com/alzheim.htm.
5. http://www.bt.cdc.gov/agent/sarin/basics/facts.asp.
6. *http://www.gulfweb.org/bigdoc/report/appgb.html#General Information.*
7. Tai, K., et al., *Finite Element Simulations of Acetylcholine Diffusion in Neuromuscular Junctions.* Biophysical Journal, 2003. 84(4): p. 2234-2241.
8. http://en.wikipedia.org/wiki/Chemical_synapse.
9. Land, B.R., et al., *Kinetic Parameters for Acetylcholine Interaction in Intact Neuromuscular Junction.* Neurobiology, 1981. 78(11): p. 7200-7204.
10. Cheng, Y., et al., *Continuum Simulations of Acetylchoine Diffusion with Reaction-determined Boundaries in Neuromuscular Junction Models.* Journal of Biophysical Chemistry, 2007. 127(3): p. 129-139.
11. Garris, P.A., et al., *Efflux of Dopamine from teSynaptic Cleft in the Nucleus Accumbens of the Rat Brain.* The Journal of Neuroscience, 1994. 14(10): p. 6084-6093.
12. Deen, William M., *Analysis of Transport Phenomena*. 1998: Oxford University Press.

Appendix A: Analytical Solution

The following differential equation (Eq. 1) with the given boundary conditions (Eq. 2 and Eq. 3) and initial condition (Eq. 4) is solved using Finite Fourier Transforms[12]. Notice that this equation incorporates a Robin boundary condition (Eq. 3).

$$\frac{\partial C_{(t,x)}}{\partial t} = D\frac{\partial^2 C_{(t,x)}}{\partial x^2} - kC_{(t,x)} \tag{1}$$

$$\left.\frac{\partial C}{\partial x}\right|_{(t,x=0)} = 0 \tag{2}$$

$$-D\left.\frac{\partial C}{\partial x}\right|_{(t,x=L)} = \beta C_{(x=L,t)} \tag{3}$$

$$C_{(t=0,x)} = C_s\delta_{(x)} \tag{4}$$

The solution is first written as a Fourier series (Eq. 5).

$$C_{(t,x)} = \sum_n c_{n(t)}\Phi_{n(x)} \tag{5}$$

The orthonormal basis set expansion function is then found by solving the eigenvalue problem in *x* (Eq. 6) using the boundary conditions of the original differential equation but for the homogenous case (Eq. 7 and Eq. 8). Equation 7 is used to evaluate one constant of integration, while Equation 8 is used to obtain an expression forλ_n. The other constant of integration is evaluated using the definition of an orthonormal function.

$$\frac{d^2\Phi_{n(x)}}{dx^2} = -\lambda_n^2\Phi_{n(x)} \tag{6}$$

$$\left.\frac{d\Phi_{n(x)}}{dx}\right|_{(x=0)} = 0 \tag{7}$$

$$\left.\frac{d\Phi_{n(x)}}{dx}\right|_{(x=L)} + \frac{\beta}{D}\Phi_{n(x=L)} = 0 \tag{8}$$

Solving and applying the boundary conditions:

$$\Phi_{n(x)} = Asin(\lambda_n x) + Bcos(\lambda_n x) \tag{9}$$

$$\frac{d\Phi_{n(x)}}{dx} = \lambda_n Acos(\lambda_n x) - \lambda_n Bsin(\lambda_n x) \tag{10}$$

Applying Equation 7:

$$\frac{d\Phi_{n(0)}}{dx} = \lambda_n Acos(0) - \lambda_n Bsin(0) = 0 \tag{11}$$

$$A = 0 \tag{12}$$

$$\Phi_{n(x)} = Bcos(\lambda_n x) \tag{13}$$

$$\frac{d\Phi_{n(x)}}{dx} = -\lambda_n Bsin(\lambda_n x) \tag{14}$$

Applying Equation 8 with Equations 13 and 14:

$$\frac{d\Phi_{n(L)}}{dx} + \frac{\beta}{D}\Phi_{(L)} = -\lambda_n Bsin(\lambda_n L) + \frac{\beta}{D}Bcos(\lambda_n L) = 0 \tag{15}$$

Thus, the values of λ_n are given by Equation 16, which must be solved numerically, where *n* corresponds to the nth positive root (n=1,2,3,…):

$$-\lambda_n sin(\lambda_n L) + \frac{\beta}{D}cos(\lambda_n L) = 0 \tag{16}$$

The constant B must still be evaluated and is done so by making $\Phi_{n(x)}$ orthonormal:

$$\int_0^L \Phi_{n(x)}\Phi_{n(x)}dx = 1 \tag{17}$$

$$B^2\int_0^L \cos(\lambda_n x)\cos(\lambda_n x)dx = 1 \tag{18}$$

$$\frac{1}{2}B^2\int_0^L [1+\cos(2\lambda_n x)]dx = 1 \tag{19}$$

$$\frac{1}{2}B^2 L + \int_0^L \cos(2\lambda_n x)dx = 1 \tag{20}$$

$$\frac{1}{2}B^2 L + \left[\frac{1}{2\lambda_n}\sin(2\lambda_n x)\right]_0^L = 1 \tag{21}$$

$$\frac{1}{2}B^2 L + \frac{1}{2\lambda_n}\sin(2\lambda_n L) = 1 \tag{22}$$

$$B = \sqrt{\left(\frac{2}{L}\right)\left[1 - \frac{1}{2\lambda_n}\sin(2\lambda_n L)\right]} \tag{23}$$

$\Phi_{n(x)}$is now specified by combining Equation 23 with Equation 13:

$$\Phi_{n(x)} = \sqrt{\left(\frac{2}{L}\right)\left[1 - \frac{1}{2\lambda_n}\sin(2\lambda_n L)\right]}\ cos(\lambda_n x) \tag{24}$$

Next, $c_{n(t)}$ must be evaluated by first multiplying both sides of Equation 5 by $\Phi_{n(x)}$ (an orthonormal function), and then integrating both sides from 0 to L:

$$c_n(t) = \int_0^L C_{(t,x)}\Phi_{n(x)} \tag{25}$$

To obtain a differential equation for $c_{n(t)}$, the original differential equation (Eq. 1) and initial condition (Eq. 4) are multiplied by $\Phi_{n(x)}$ and integrated over *x* from 0 to L. This multiplication and integration is performed term by term.

Applying these operations to the first term on the left of the equals sign in Equation 1:

$$\int_0^L \frac{\partial C_{(t,x)}}{\partial t}\Phi_{n(x)}dx = \frac{\partial}{\partial t}\int_0^L C_{(t,x)}\Phi_n(x)dx = \frac{\partial c_{n(t)}}{\partial t} \tag{26}$$

Sturm- Liouville theory[12] (Equation 27) is used to perform these operations on the first term to the right of the equals sign in Equation 1:

$$D\int_0^L \frac{\partial^2 C_{(t,x)}}{\partial x^2}\Phi_{n(x)}dx = D\left[\Phi_{n(x)}\frac{\partial C_{(t,x)}}{\partial x} - C_{(t,x)}\frac{\partial \Phi_{n(x)}}{\partial x}\right]_0^L - D\lambda_n^2 c_{n(t)} \tag{27}$$

$$D\int_0^L \frac{\partial^2 C_{(t,x)}}{\partial x^2}\Phi_{n(x)}dx = D\left[\Phi_{n(L)}\frac{\partial C_{(t,x)}}{\partial x}\bigg|_L - C_{(t,x=L)}\frac{\partial \Phi_{n(x)}}{\partial x}\bigg|_L\right] - D\lambda_n^2 c_n(t) \tag{28}$$

$$D\int_0^L \frac{\partial^2 C_{(t,x)}}{\partial x^2}\Phi_{n(x)}dx = D\left[\Phi_{n(L)}\left(\frac{-\beta}{D}\right)C_{(t,x=L)} - C_{(t,x=L)}\frac{\partial \Phi_{n(x)}}{\partial x}\bigg|_L\right] - D\lambda_n^2 c_{n(t)} \tag{29}$$

$$D\int_0^L \frac{\partial^2 C_{(t,x)}}{\partial x^2}\Phi_{n(x)}dx = -DC_{(t,x=L)}\left[\left(\frac{\beta}{D}\right)\Phi_{n(L)} + \frac{\partial \Phi_{n(x)}}{\partial x}\bigg|_L\right] - D\lambda_n^2 c_n(t) \tag{30}$$

Applying Equation 8 to Equation 30:

$$D\int_0^L \frac{\partial^2 C_{(t,x)}}{\partial x^2}\Phi_{n(x)}dx = -D\lambda_n^2 c_{n(t)} \tag{31}$$

Applying these operations to the second term to the right of the equals sign in Equation 1:

$$\int_0^L -kC_{(t,x)}\Phi_{n(x)}\,dx = -kc_{n(t)} \tag{32}$$

Applying these operations to the initial condition:

$$c_{n(t=0)} = \int_0^L C_{(t=0,x)}\Phi_{n(x)}dx = \int_0^L C_s\delta_{(x)}\Phi_{n(x)}dx = C_s\Phi_{n(0)} \tag{33}$$

$$c_{n(t=0)} = C_s\sqrt{\left(\frac{2}{L}\right)\left[1 - \frac{1}{2\lambda_n}\sin(2\lambda_n L)\right]} \tag{34}$$

Assembling the differential equation (Equation 35 – includes terms evaluated in Eq. 26, Eq. 31, and Eq. 32) and initial condition (Equation 36 – includes the term evaluated in Eq. 34) which have been multiplied by $\Phi_{n(x)}$ and integrated over the *x* from 0 to L:

$$\frac{dc_{n(t)}}{dt} = -D\lambda_n^2 c_{n(t)} - kc_{n(t)} \tag{35}$$

$$c_{n(t=0)} = C_s\sqrt{\left(\frac{2}{L}\right)\left[1 - \frac{1}{2\lambda_n}\sin(2\lambda_n L)\right]} \tag{36}$$

Solving for $c_{n(t)}$:

$$c_{n(t)} = C_s\sqrt{\left(\frac{2}{L}\right)\left[1 - \frac{1}{2\lambda_n}\sin(2\lambda_n L)\right]}\,e^{-(D\lambda_n^2+k)t} \tag{37}$$

Assembling the solution for $C_{(t,x)}$ using Equations 5, 24, and 37; where λ_n is defined by Equation 16:

$$C_{(t,x)} = \frac{2C_s}{L}\sum_{n=1}^{\infty}\sqrt{\left[1-\frac{1}{2\lambda_n}\sin(2\lambda_n L)\right]}\,e^{-(D\lambda_n^2+k)t}\cos(\lambda_n x) \tag{38}$$

Appendix B: MATLAB Code for Numerical Solution Using PDEPE

```
functionSynapseandflux

%Parameters
L=0.02; %µm
tfinal=1e-7; %seconds
D=400; %µm^2/s

tpoints=501;
xpoints=41;

%m=0 denotes that our problem is in a slab
m = 0;
%Create time and distance vectors
x = linspace(0,L,xpoints);
t = linspace(0,tfinal,tpoints);

%Call pdepe to solve the equation
sol = pdepe(m,@pdex1pde,@pdex1ic,@pdex1bc,x,t);
%Solution is first component of sol
C = sol(:,:,1);

%Estimating flux at boundary
Flux=zeros(1,length(t));
for i=2:501
dFlux=-D*((C(i,41)-C(i,40))/(L/xpoints))*(tfinal/tpoints);
Flux(i)=Flux(i-1)+dFlux;
end

%Plotting Flux vs time
figure
plot(t,Flux*6.022e23)
xlabel('Time (seconds)')
ylabel('Bound Receptor Surface Concentration (molecules/µm^2)')
title('ACh Bound at Post-Synaptic Membrane')

%Create 3d plot
figure
surf(x,t,C,'Edgecolor','None')
title('AChConc in the Synaptic Cleft')
xlabel('Distance (µm)')
ylabel('Time (seconds)')
zlabel('Concentration (mol/m^3)')

%PDE function (define our PDE)
function [c,f,s] = pdex1pde(x,t,Conc,DCDx)
```

```
D=400; %µm^2/s
k=23e7; %1/s, kcat of AChE
%Time derivative coefficient
c = 1;
%x derivative coefficient (includes first derivative to make a second
%derivative in x)
f = D*DCDx;
%Forcing function coefficient
G=-k*Conc; %mol/µm^3 s
s = G;

%IC function
function C0 = pdex1ic(x)
L=0.02;
Cs=6e-13; %mol/µm^2, concentration pulse into synapse
%Delta function initial condition, concentration is Cs at x=0 and 0
%elsewhere
if x==0
    C0=Cs;
else
    C0=0;
end

%BC function
function [pl,ql,pr,qr] = pdex1bc(xl,ul,xr,ur,t)
D=400; %µm^2/s;
beta=4.7e22; %µm^3/mol*s Receptor binding rate
%dCdX=0 at x=0
pl = 0;
ql = 1;
%-beta*Conc-D*dCdx=0 at x=L (Robin BC)
pr = -beta*ur;
qr = -1;
```

Appendix C: MATLAB PDETOOL set-up of FEA solution

The PDE Toolbox can be opened by calling `pdetool` in the command window. This opens a GUI to guide the user through setting up a PDE in two spatial dimensions and numerically approximating the solution using FEA. The general steps of the process are outlined in the command bar (Figure 8): draw the geometry, define the boundary conditions, set the PDE parameters, mesh the geometry, solve the PDEs, and plot the solution. First, the desired two-dimensional geometry was constructed in the GUI from a library of basic shapes (Figure 10). Next, the boundary conditions and problem specific diffusion equation parameters were entered. This included a source term correlating to acetlycholinesterase in the normal condition and no source term in the sarin gas case. The PDE Toolbox then meshed the defined geometry (Figure 3). The final step in the PDE Toolbox is to export all of the parameters to the MATLAB workspace for further analysis.

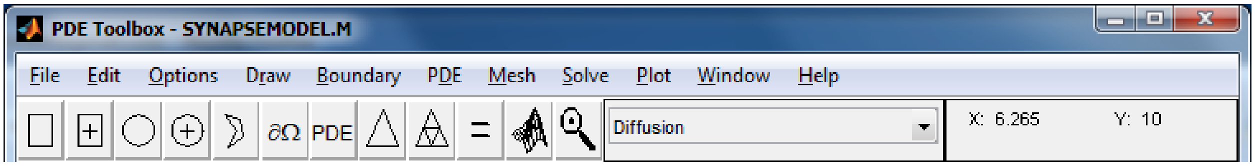
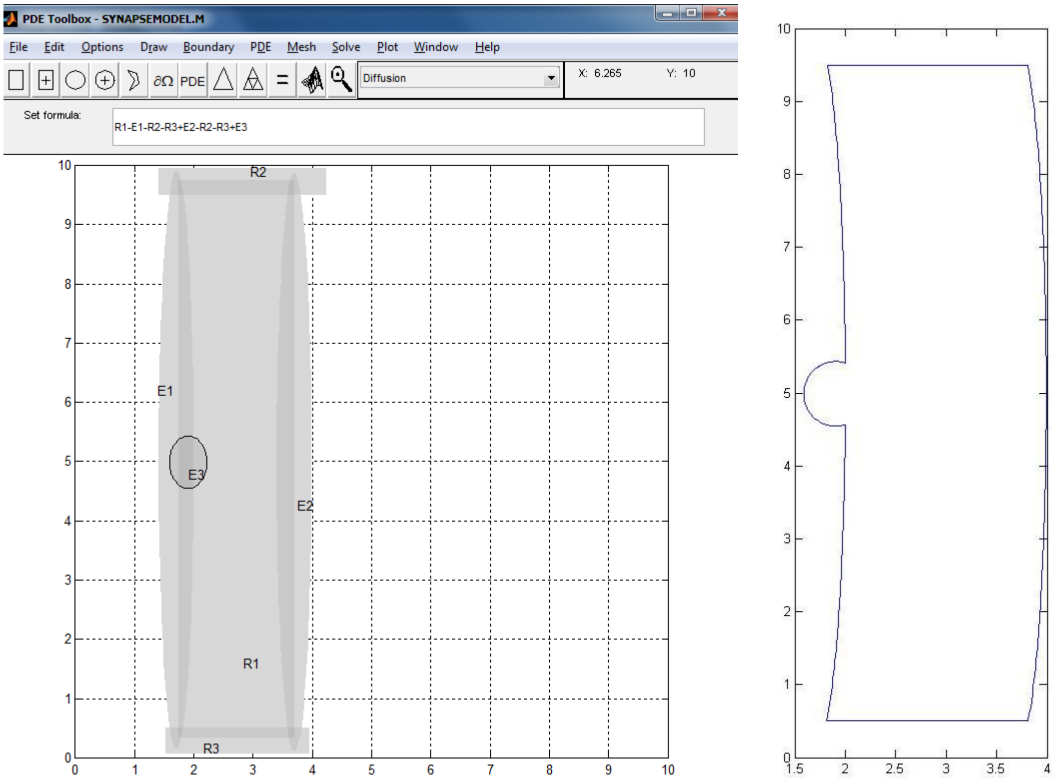

PDE Toolbox - SYNAPSEMODEL.M
File Edit Options Draw Boundary PDE Mesh Solve Plot Window Help
∂Ω PDE =
Diffusion
X: 6.265 Y: 10

Figure 8: PDE Toolbox command bar


PDE Toolbox - SYNAPSEMODEL.M
File Edit Options Draw Boundary PDE Mesh Solve Plot Window Help
∂Ω PDE =
Diffusion
X: 6.265 Y: 10

Set formula: R1-E1-R2-R3+E2-R2-R3+E3

R2
E1
E3
E2
R1
R3
0
1
2
3
4
5
6
7
8
9
10
0
1
2
3
4
5
6
7
8
9
10
0
1
2
3
4
5
6
7
8
9
10
1.5
2
2.5
3
3.5
4

Figure 9: a) Generating the geometry in the GUI using shape b) the final geometry

Appendix D: MATLAB Code for FEA Solution

The following code will replicate the work performed in the pdetool GUI

```
function pdemodel
[pde_fig,ax]=pdeinit;
pdetool('appl_cb',10);
set(ax,'DataAspectRatio',[10 10 1]);
set(ax,'PlotBoxAspectRatio',[1 1 1]);
set(ax,'XLim',[0 10]);
set(ax,'YLim',[0 10]);
set(ax,'XTick',[ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,]);
set(ax,'YTick',[ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,]);
pdetool('gridon','on');
% Geometry description:
pdeellip(1.7056396148555697,5.0309491059147193,0.29573590096286106,4.87964236
58872072,0,'E1');
pdeellip(3.6960110041265466,4.9896836313617605,0.29573590096286106,4.87964236
58872072,0,'E2');
pderect([1.7400275103163696 3.7070151306740033 9.745529573590094
0.33700137551581744],'R1');
pderect([4.2434662998624484 1.4099037138927111 9.4979367262723517
9.9518569463548818],'R2');
pderect([3.9546079779917465 1.5199449793672624 0.068775790921597135
0.50206327372764825],'R3');
pdeellip(1.9050894085281982,4.9896836313617614,0.31636863823933936,0.44360385
144429149,0,'E3');
set(findobj(get(pde_fig,'Children'),'Tag','PDEEval'),'String','R1-E1-R2-
R3+E2-R2-R3+E3')
% Boundary conditions:
pdetool('changemode',0)
pdetool('removeb',[1 2 10 14 15 18 19 ]);
pdesetbd(14,'neu',1,'0','0')
pdesetbd(13,'neu',1,'0','0')
pdesetbd(12,'neu',1,'0','0')
pdesetbd(11,'neu',1,'0','0')
pdesetbd(10,'neu',1,'1','0')
pdesetbd(9,'neu',1,'1','0')
pdesetbd(8,'neu',1,'0','0')
pdesetbd(7,'neu',1,'0','0')
pdesetbd(6,'neu',1,'0','0')
pdesetbd(5,'neu',1,'0','0')
pdesetbd(4,'neu',1,'0','0')
pdesetbd(3,'neu',1,'0','0')
pdesetbd(2,'neu',1,'0','0')
pdesetbd(1,'neu',1,'0','0')
% Mesh generation:
setappdata(pde_fig,'Hgrad',1.3);
setappdata(pde_fig,'refinemethod','regular');
setappdata(pde_fig,'jiggle',char('on','mean',''));
pdetool('initmesh')
% PDE coefficients:
pdeseteq(2,'4000','0.0','0','1.0','0:100','0','0.0','[0 100]')
setappdata(pde_fig,'currparam',['4000';'0'])
% Solve parameters:
setappdata(pde_fig,'solveparam',...
str2mat('0','1000','10','pdeadworst','0.5','longest','0','1E-
4','','fixed','Inf'))
% Plotflags and user data strings:
setappdata(pde_fig,'plotflags',[1 1 1 1 1 1 1 1 0 0 0 14 1 0 0 0 0 1]);
setappdata(pde_fig,'colstring','');
setappdata(pde_fig,'arrowstring','');
setappdata(pde_fig,'deformstring','');
setappdata(pde_fig,'heightstring','');
```

After running the pdetool GUI and exporting the parameter values, the following code was run to generate a movie comparing the normal and sarin gas condition. Note that the mesh points inside the vesicle used for the initial conditions were determined manually in the PDE Toolbox.

```
%to run first load SynapseModel.m in pdetool and export the parameters
%must export the variables b,p,e,t,c,a,f,d from pdetool
clf
%determine length of simulation and time step
tlist=0:.01:.5;
%Define initial conditions
u0=zeros(size(p,2),1);  %set everything to zero
n=11;   %number of nodes the initial concentration is spread over
Cs=10;   %initial concentration
set=[181,141,140,175,105,133,176,139,104,180,138]; %data points in vescicle
for i=1:size(set,2)
    u0(set(i))=Cs/n;
end
%u0=ones(size(p,2),1);
%define global sink term due to cleavage in cleft
a1='0';
a2='5';
c='10';

%solve PDE
u1=parabolic(u0,tlist,b,p,e,t,c,a1,f,d);
u2=parabolic(u0,tlist,b,p,e,t,c,a2,f,d);

%run multiple iterations
for i=1:3
   ui1=u0+u1(:,size(u1,2));
   ui2=u0+u2(:,size(u2,2));
   uf1=parabolic(ui1,tlist,b,p,e,t,c,a1,f,d);
   uf2=parabolic(ui2,tlist,b,p,e,t,c,a2,f,d);
   u1=[u1,uf1];
   u2=[u2,uf2];
end

clear M
%sum(ul(:,1))
%sum(ul(:,2))
for j=1:size(u1,2)
    figure(1)
    subplot(1,2,2)
    pdesurf(p,t,u1(:,j))
    xlabel('x')
    ylabel('y')
    zlabel('ACh Concentration')
    title('SARIN GAS')
    colormap('hsv')
    caxis([0 .05])
    axis([1.5 4 0 10 0 .2])
    subplot(1,2,1)
    pdesurf(p,t,u2(:,j))
    xlabel('x')
    ylabel('y')
    zlabel('ACh Concentration')
    title('NORMAL')
    colormap('hsv')
    caxis([0 .05])
    axis([1.5 4 0 10 0 .2])
    M(j)=getframe(figure(1));
end
movie2avi(M,'moviecompressed','compression', 'Cinepak')
```

Appendix E: FEA Solution Video

See provided CD.