FYS3150 - Computational Physics

Project 4

Diffusion of neurotransmitters in the synaptic cleft

Filip Henrik Larsen

Date: 6. november 2014

Introduction

In this problem we had a look at different methods of solving diffusion equations, and as a specific problem we considered diffusion of neurotransmitters in the synaptic cleft.

Diffusion equations

A diffusion equation is a partial differential equation on the form

$$\frac{\partial u(r,t)}{\partial t} = D\nabla u(r,t)$$

where D is the diffusion constant. In this project we had D=1 and we only considered a one dimensional problem. The equation we were to solve therefore looked like

$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial^2 u(x,t)}{\partial x^2}$$

These equations can be solved both analytically and numerically, as we will do later on. A solution to the equation is a function $u(x,t)$ which satisfy the equation above. However in order to do so we must have initial- and boundary conditions. The solution must obey them as well. In this project $u(x,t)$ represented the distribution of neurotransmitters in the synaptic cleft at a time t. We defined the length of the synaptic cleft to be L so that the presynaptic and postsynaptic membranes were positioned at $x=0$ and $x=L$ respectively.

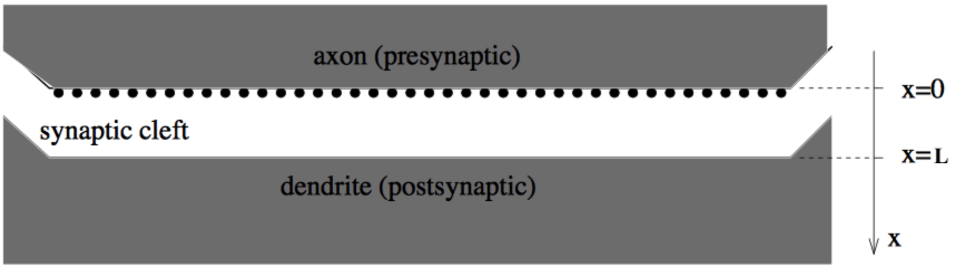

Figure 1: Schematic drawing of the synaptic cleft in our model. The black dots represent neurotransmitter molecules, and the situation shown corresponds to $t=0$.

The initial condition we were given was that all the neurotransmitters were to be at the presynaptic membrane, as Figure 1 illustrates nicely. We also had boundary conditions at both ends of the spatial interval. At the presynaptic membrane ($X=0$) the concentration of neurotransmitters should always be equal one. At the other end, at the postsynaptic membrane, it should always equal zero. Written mathematically the conditions were

$$u(x,0) = 0 \quad ,x>0 \tag{1}$$
$$u(0,t) = 1 \tag{2}$$
$$u(L,t) = 0 \tag{3}$$

After a long timethe distribution function $u(x,t)$ has diverged to a so-called steady state solution. At this point it will no longer change with time. For a system such as here, the steady state solution will take a linear form

$$u_s(x) = Ax + b$$

Boundary condition (2) gives $b=1$, and (3) gives $A=-1/L$. Thus the steady state is

$$u_s(x) = 1 - \frac{x}{L} \tag{4}$$

We now define another function

$$v(x,t) = u(x,t) - u_s(x) \tag{5}$$

Which is the difference between the state we're in and the steady state. The reason we do this is because we obtain dirichlet boundary conditions, which is easier to work with both analytically and numerically. We obtain the initial- and boundary conditions

$$v(x,0) = x/L - 1 \tag{6}$$
$$v(0,t) = 0 \tag{7}$$
$$v(L,t) = 0 \tag{8}$$

Analytical solution

We will now try to derive the solution to the diffusion equation of the function, as this also gives the solution of $u(x,t)$. We have the equation

$$\frac{\partial v(x,t)}{\partial t} = \frac{\partial^2 v(x,t)}{\partial x^2} \tag{9}$$

We use a technique called separation of variables to solve this, meaning we assume that

$$v(x,t) = X(x)T(t)$$

Which transforms (9) into

$$\frac{1}{T(t)}\frac{\partial T(t)}{\partial t} = \frac{1}{X(x)}\frac{\partial^2 X(x)}{\partial x^2}$$

We allow this to equal some constant $-\lambda^2$. The equations we have is thus

$$\frac{\partial^2 X(x)}{\partial x^2} + X(x)\lambda^2 = 0$$

with the solution

$$X(x) = A\sin(\lambda x) + B\cos(\lambda x$$

The boundary conditions will now restrict the constants.

$$(7) \;\Rightarrow\; B = 0$$

$$(8) \;\Rightarrow\; A\sin(\lambda x) = 0 \;\Rightarrow\; \lambda = \frac{\pi n}{L} \quad, n \in \mathbb{Z}$$

We assume $n > 0$ because the negative values are only linear combinations of the positive, and thus the positive values alone span the entire space.
The time dependent function has to obey

$$\frac{\partial T(t)}{\partial t} + T(t)\lambda^2 = 0$$

Which has the solution

$$Ce^{-\lambda^2 t}$$

where C is some constant. The solution to the diffusion equation (9) is thus

$$v(x,t) = \sum_{n=1}^{\infty} A_n \sin\left(\frac{\pi n}{L}x\right) e^{-\left(\frac{\pi n}{L}\right)^2 t}$$

Where the constant C in the time dependent function has been absorbed in A_n. The initial condition (6) gives

$$\frac{x}{L} - 1 = \sum_{n=1}^{\infty} A_n \sin\left(\frac{\pi n}{L}x\right)$$

A_n is nothing but the Fourier coefficients of the function $x/L - 1$, therefore

$$\begin{aligned} A_n &= \frac{2}{L}\int_0^L \left(\frac{x}{L} - 1\right)\sin\left(\frac{n\pi}{L}x\right)dx \\ &= \frac{2}{n\pi}\Big((1-L)\cos(n\pi) - 1\Big) \end{aligned}$$

Leaving the analytical solution

$$v(x,t) = \sum_{n=1}^{\infty} \frac{2}{n\pi}\Big[(1-L)\cos(n\pi) - 1\Big]\sin\left(\frac{\pi n}{L}x\right) e^{-\left(\frac{\pi n}{L}\right)^2 t}$$

In this project we set $L = 1$, and therefore the solution of the diffusion equation in this problem were

$$v(x,t) = \sum_{n=1}^{\infty} -\frac{2}{n\pi}\sin(\pi n x)\, e^{-(\pi n)^2 t} \tag{10}$$

Numerical solution

In order to solve this diffusion equation numerically there are several schemes we could use. In this project we focused on three in particular: the explicit, the implicit and the Crank-Nicolson. I will now present the algorithms for the schemes and their stability properties. Using the discretization

$$\begin{aligned} x_i &= i\Delta x \\ t_j &= j\Delta t \\ u_{i,j} &= u(x_i, t_j) \end{aligned}$$

The explicit scheme is written as

$$u_t = \frac{u_{i,j+1} - u_{i,j}}{\Delta t} \tag{11}$$

$$u_{xx} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} \tag{12}$$

and thus the equation reads

$$\frac{u_{i,j+1} - u_{i,j}}{\Delta t} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2}$$

Deffining $\alpha = \Delta t/\Delta x^2$ and rearranging a bit we have

$$u_{i,j+1} = \alpha u_{i-1,j} + (1 - 2\alpha)u_{i,j} + \alpha u_{i+1,j} \tag{13}$$

Which is the algorithm we use for this scheme. This scheme has a truncation error of order $O(\Delta t)$ for the time-derivative and $O(\Delta x^2)$ for the spatial part. The reason for this is that in the approximations for rate of change in time we do not include terms of higher order than Δt, and likewise for the double derivative of x, we do not include terms containing higher order of Δx than Δx^2.

The implicit scheme is written as

$$u_t = \frac{u_{i,j} - u_{i,j+1}}{\Delta t} \tag{14}$$

$$u_{xx} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} \tag{15}$$

resulting in

$$\frac{u_{i,j} - u_{i,j+1}}{\Delta t} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2}$$

We define α as for the explicit scheme and end up with the equation

$$u_{i,j-1} = -\alpha u_{i-1,j} + (1 - 2\alpha)u_{i,j} - \alpha u_{i+1,j} \tag{16}$$

If we define vectors

$$\mathbf{v}_j = \begin{pmatrix} u_{0,j} \\ u_{1,j} \\ \vdots \\ u_{n,j} \end{pmatrix}$$

the above equation can be expressed as a matrix-vector equation.

$$\mathbf{v}_{j-1} = \hat{\mathbf{A}}\mathbf{v_j} \tag{17}$$

With the matrix $\hat{\mathbf{A}}$ being the tridiagonal matrix

$$\mathbf{A} = \begin{pmatrix} 1+2\alpha & -\alpha & & & \\ -\alpha & 1+2\alpha & -\alpha & & \\ & \ddots & \ddots & \ddots & \\ & & -\alpha & 1+2\alpha & -\alpha \\ & & & -\alpha & 1+2\alpha \end{pmatrix}$$

Since we got the initial conditions of the system in this problem we set $\mathbf{v}_{j-1}$ to be this function. Thus the only unknown quantity in equation (17) is $\mathbf{v}_j$. This problem we have solved in project 1, and we used the same technique here. Look up my report on project 1 for an explanation on the algorithms. However, as we find $\mathbf{v}_j$, we have done one iteration in time. Using this new vector as the new $\mathbf{v}_{j-1}$ we can thus find the function at the next iteration and so on... Also this scheme has a truncation error of order $O(\Delta t)$ for the time-derivative and $O(\Delta x^2)$ for the spatial part.

The Crank-Nicolson scheme is a bit different from the others. This scheme make a better approximation of the time derivative by using the midpoint rule. As a result we evaluate the time derivative at $t' = t + \Delta t/2$. Therefore we must also calculate the double derivative with respect to x at this time. Meaning we must approximate

$$\frac{\partial u(x,t')}{dt} \qquad \text{and} \qquad \frac{\partial^2 u(x,t')}{dx^2}$$

The way we approximate these terms, is using the midpoint rule.

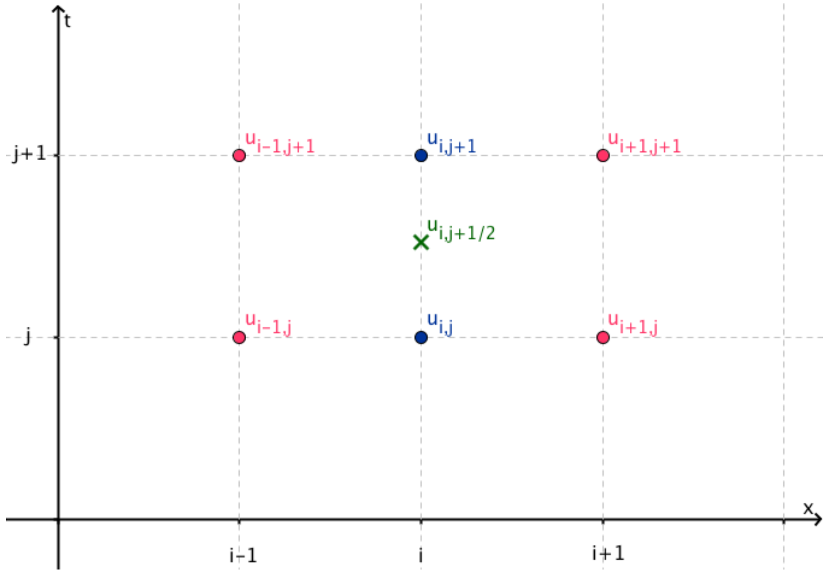

Figure 2: Calculation molecule for the Crank-Nicolson scheme.

Basically we approximate the time derivative of the element $u_{i,j+1/2}$ using the midpoint rule as

$$\frac{\partial u(x,t')}{dt} \approx \frac{u_{i,j+1} - u_{i,j}}{\Delta t}$$

And we also approximate the spatial part in the same way. We approximate as

$$\frac{\partial^2 u(x,t')}{dx^2} \approx \frac{1}{2}\left[\frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{\Delta x^2} - \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2}\right]$$

These have to equal. By introducing again the α and reorganizing this results in

$$-\alpha u_{i+1,j+1} + (2+2\alpha)u_{i,j+1} - \alpha u_{i-1,j+1} = \alpha u_{i+1,j} + (2-2\alpha)u_{i,j} - \alpha u_{i-1,j} \tag{18}$$

Which is the algorithm for the Crank-Nicolson method. This scheme has a truncation error of order $O(\Delta t^2)$ for the time-derivative and $O(\Delta x^2)$ for the spatial part. The reason for the improved approximation of the time-derivative is that we use the midpoint rule. All the algorithms shown here can be neatly summarized by the so-called theta method

$$\frac{u_{i,j} - u_{i,j-1}}{\Delta t} = \frac{\theta}{\Delta x^2}\left(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}\right) + \frac{1-\theta}{\Delta x^2}\left(u_{i+1,j-1} - 2u_{i,j-1} + u_{i-1,j-1}\right)$$

where $\theta = 0$ gives the explicit scheme as in (13), $\theta = 1$ gives the implicit scheme as in (16) and $\theta = 1/2$ gives the Crank-Nicolson scheme as in (18).

The stability of the schemes can be analyzed by inspection of the matrices we get when rewriting the algorithms as matrix-vector equations. We did this for the implicit scheme, and using the same definition of the vectors $\mathbf{v}_j$ we can show that

$$
\begin{aligned}
&(13) \quad \Rightarrow \quad \mathbf{v}_j = \hat{\mathbf{A}}\mathbf{v}_{j-1} \quad && , \hat{\mathbf{A}} = \hat{\mathbf{I}} - \alpha\hat{\mathbf{B}} \\
&(16) \quad \Rightarrow \quad \mathbf{v}_j = \hat{\mathbf{A}}^{-1}\mathbf{v}_{j-1} \quad && , \hat{\mathbf{A}} = \hat{\mathbf{I}} + \alpha\hat{\mathbf{B}} \\
&(18) \quad \Rightarrow \quad \mathbf{v}_j = \hat{\mathbf{A}}\mathbf{v}_{j-1} \quad && , \hat{\mathbf{A}} = \left(2\hat{\mathbf{I}} + 2\alpha\hat{\mathbf{B}}\right)^{-1}\left(2\hat{\mathbf{I}} - 2\alpha\hat{\mathbf{B}}\right)
\end{aligned}
$$

where

$$
\hat{\mathbf{B}} = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}
$$

It can be shown that the eigenvalues of this matrix are $2 - 2\cos\theta$.

The stability of the schemes is given by the spectral radius, $\rho(\hat{A})$ of the matrices $\hat{\mathbf{A}}$ above. The spectral radius is defined as

$$
\rho(\hat{A}) = \max\left\{|\lambda| : det\left(\hat{A} - \lambda\hat{I}\right) = 0\right\}
$$

Meaning that the spectral radius of a matrix, $\hat{A}$, equals the largest (in magnitude) eigenvalue of this matrix. The criteria for the schemes to be stable is

$$
\rho(\hat{A}) < 1
$$

If this criteria is pleased, the resulting vector will converge towards a finite value. In our case toward the so-called steady state solution. The criteria of the spectral radius tells us that for the explicit scheme we have

$$
\rho(\hat{I} - \alpha\hat{B}) < 1
$$

$$
\begin{aligned}
-1 < 1 - \alpha 2(1 - \cos\theta) < 1 \\
0 < 2\alpha(1 - cos\theta) < 2
\end{aligned}
$$

By definition $\alpha > 0$.

$$
\alpha < \frac{1}{1 - \cos\theta}
$$

$$
\alpha < \frac{1}{2}
$$

This means that the explicit scheme is stable as long as $\alpha < 1/2$.
For the implicit case we have

$$
\rho((\hat{I} + \alpha\hat{B})^{-1}) < 1
$$

$$
-1 < \frac{1}{1 + \alpha 2(1 - \cos\theta)} < 1
$$

Which obviously is true for all values of α (because $\alpha > 0$), meaning the implicit scheme is stable for any value of α!
For the Crank-Nicolson method we have

$$
\rho((2\hat{I} + 2\alpha\hat{B})^{-1}(2\hat{I} - 2\alpha\hat{B})) < 1
$$

$$
-1 < \frac{2 - 2\alpha(1 - \cos\theta)}{2 + 2\alpha(1 - \cos\theta)} < 1
$$

Lets look at the criteria separately

$$
\begin{aligned}
-(2 + 2\alpha(1 - \cos\theta)) < 2 - 2\alpha(1 - \cos\theta) \qquad & \qquad 2 - 2\alpha(1 - \cos\theta) < 2 + 2\alpha(1 - \cos\theta) \\
-4 < 0 \qquad & \qquad 4\alpha(1 - \cos\theta) > 0
\end{aligned}
$$

As both of these always are true, again because of the definition of α, the Crank-Nicolson scheme is stable for any value of α.

Results

In my script i implemented the schemes in a class in such a way that calling upon it with a function as input it will do one iteration in time, and return a new vector. In the main function we can tweak the variables (limits, number of points, time period...). Setting the spatial step length to $\Delta x = 0.1$ and time step length $\Delta t = 0.005$ we are at the limit of the stability criteria for the explicit scheme. First we took a look at the system in an early stage. We then got the result bellow

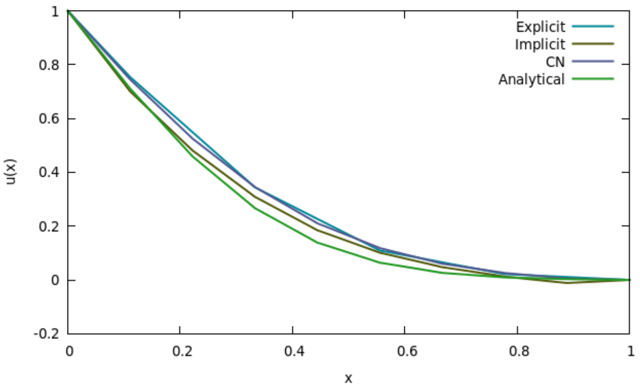

Figure 3: Calculated state of the system in an early stage. ($t = 0.05$)

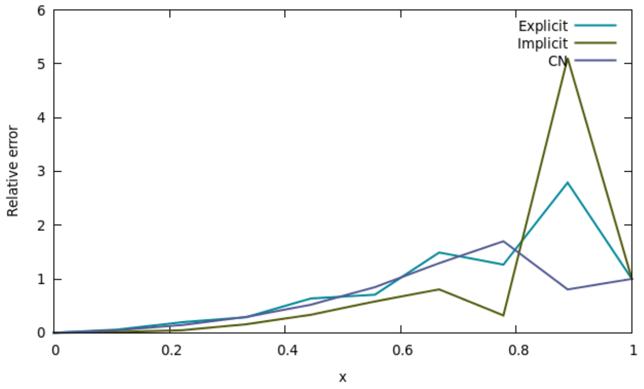

Figure 4: Relative error compared to the analytical solution, in an early stage ($t = 0.05$).

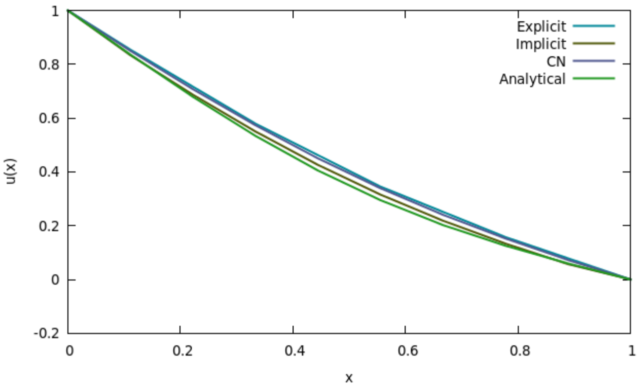

Figure 5: Calculated state of the system at a later stage. ($t = 0.1$)

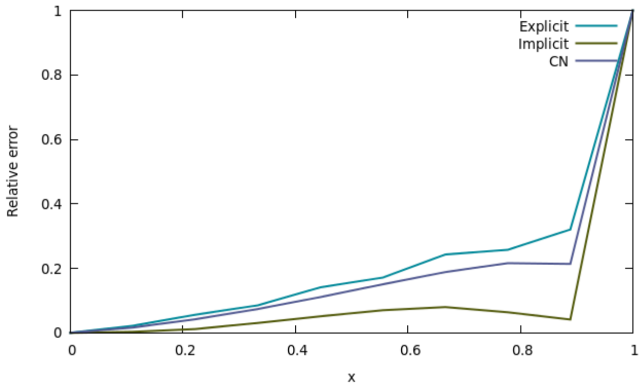

Figure 6: Relative error compared to the analytical solution, at a later stage ($t = 0.1$).

We see that as time develops the state converges toward the steady state, as expected. I think the figures showing the relative error is the most interesting. We clearly see that the explicit scheme has the greatest relative error, however I was surprised that the Crank-Nicolson wasn't far superior to the implicit as well. It does have a better approximation for the time-derivative, and so I expected a better approximation than the implicit scheme, however for the late stage the implicit is pretty much dead on, and Crank-Nicolson is not! In the early stage the implicit is also the best approximation in the major part of the interval, but close to the end the relative error escalates. The thing is, the error isn't even that big, but the relative error is. With this as my evidence I would say that the implicit scheme is the best one, though this does not make sense to me.

Github repository

`https://github.com/filiphl/FYS3150.git`