

COMPUTATIONAL PHYSICS, PROJECT 4

VILDE EIDE SKINGEN, JOHN-ANDERS STENDE AND KARI ERIKSEN

Abstract

The transport process of signals in the synaptic cleft in the brain is governed by diffusion. This diffusion process can be modeled by simulations, and be described mathematically by the diffusion equation. In this project we will solve the diffusion equation using three different methods for solving differential equations. By imposing initial and boundary conditions to the mathematical expression, the full solution to the diffusion equation can be found in closed form. We will use this analytical solution to test our numerical solutions.

We want to study the numerical stability of three methods for partial differential equations; the explicit Forward Euler scheme, the implicit Backward Euler and the implicit Crank-Nicolson.

Diffusion of neurotransmitters in the synaptic cleft

Introduction

In this project we wish to model the diffusion of neurotransmitters in the synaptic cleft. The dominant way of transporting signals between nerve cells in the brain is by means of diffusion across the synaptic cleft separating the cell membranes of the two cells. These neurotransmitter molecules are initially inside vesicles located in the pre-synaptic axon terminal. When an action potential reaches the axon terminal, the vesicles release neurotransmitters into the synaptic cleft. These molecules then diffuse toward the post-synaptic membrane and bind to receptors in the membrane.

Theory

The transport process in the synaptic cleft is governed by diffusion, and thus we can describe it mathematically by

$$\frac{\partial u}{\partial t} = D\nabla^2 u, \quad (1)$$

where u is the concentration of the particular neurotransmitter, and D is the diffusion coefficient of the neurotransmitter in the solvent in the synaptic cleft.

The diffusion equation describes the evolution in time of the density u of a quantity. The quantity of interest is the total flux at time t of molecules across the post-synaptic membrane.

We assume that the neurotransmitter is released roughly equally on the pre-synaptic side and that the synaptic cleft is roughly equally wide across the whole synaptic terminal. Given the large area of the synaptic cleft compared to its width we assume that the neurotransmitter concentration only varies in the direction across the synaptic cleft. We choose this direction to be the x -direction. With this choice of coordinate our diffusion equation reduces to

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2}. \quad (2)$$

A synaptic vesicles burst at a time $t = 0$, releasing the neurotransmitters into the cleft. Immediately after the release the concentration profile in the x -direction is given by

$$u(x, t = 0) = N \delta(x), \quad (3)$$

where N is the number of particle released into the synaptic cleft per area of membrane.

Diffusion is strongly linked with random walks. To get an idea over the time-dependence of the neurotransmitter concentration at the post-synaptic side, we look at the solution of a "free" random walk. We then assume that there are no obstacles or particle absorbers in either direction. This is a simplification; in the true diffusion process the molecules will, for instance, bump into the pre-synaptic membrane and be absorbed by receptors at the post-synaptic membrane. The random walk gives rise to a Gaussian distribution in time and space. We assume that our distribution is given by a normal distribution. The solution of equation (2) with the initial condition in equation (3) is given by

$$u(x, t) = \frac{N}{\sqrt{4\pi Dt}} e^{-x^2/4Dt} . \quad (4)$$

The concentration at the postsynaptic side $u(d, t)$ approaches 0 in the limit $t \rightarrow 0$ and $t \rightarrow \infty$.

In our mathematical model we will give the following boundary and initial conditions to approach the situation

$$u(x = 0, t > 0) = u_0, \quad u(x = d, \text{all } t) = 0, \quad u(0 < x < d, t < 0) = 0 . \quad (5)$$

We will set $d = 1$. Thus there are no neurotransmitters in the synaptic cleft when $t < 0$, for $t > 0$ the concentration at the pre-synaptic boundary of the synaptic cleft is kept fixed at $u = u_0 = 1$, and that the post-synaptic receptors immediately absorb nearby neurotransmitters so that $u = 0$ at the post-synaptic side of the cleft.

We are thus looking at a one-dimensional problem

$$\frac{\partial^2 u(x, t)}{\partial x^2} = \frac{\partial u(x, t)}{\partial t}, \quad t > 0, x \in [0, d]$$

or

$$u_{xx} = u_t,$$

with initial conditions, i.e., the conditions at $t = 0$,

$$u(x, 0) = 0 \quad 0 < x < d$$

with $d = 1$ the length of the x -region of interest. The boundary conditions are

$$u(0, t) = 1 \quad t > 0,$$

and

$$u(d, t) = 0 \quad t > 0.$$

We will solve the partial differential equation by

1. The explicit forward Euler algorithm

$$u_t \approx \frac{u(x_i, t_j + \Delta t) - u(x_i, t_j)}{\Delta t}$$

and

$$u_{xx} \approx \frac{u(x_i + \Delta x, t_j) - 2u(x_i, t_j) + u(x_i - \Delta x, t_j)}{\Delta x^2}.$$

2. The implicit Backward Euler with

$$u_t \approx \frac{u(x_i, t_j) - u(x_i, t_j - \Delta t)}{\Delta t}$$

and

$$u_{xx} \approx \frac{u(x_i + \Delta x, t_j) - 2u(x_i, t_j) + u(x_i - \Delta x, t_j)}{\Delta x^2},$$

3. Finally the implicit Crank-Nicolson scheme with a time-centered scheme at $(x, t + \Delta t/2)$

$$u_t \approx \frac{u(x_i, t_j + \Delta t) - u(x_i, t_j)}{\Delta t}.$$

With the corresponding spatial second-order derivative

$$u_{xx} \approx \frac{1}{2} \left(\frac{u(x_i + \Delta x, t_j) - 2u(x_i, t_j) + u(x_i - \Delta x, t_j)}{\Delta x^2} + \frac{u(x_i + \Delta x, t_j + \Delta t) - 2u(x_i, t_j + \Delta t) + u(x_i - \Delta x, t_j + \Delta t)}{\Delta x^2} \right).$$

Method

In this project we want as mentioned simulate the concentration of a particular neurotransmitter in the synaptic cleft. More specific between the presynaptic membrane and the postsynaptic membrane. In order to compare our simulation, using the three different methods Backward Euler, Forward Euler and the Crank-Nicolson scheme, we need something we know to be an exact solution or a closed form solution. One of our task in this project is to find this solution. Which in this case is on closed form.

We start with the diffusion equation in 1 dimension

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2}. \quad (6)$$

We are given the solution of the steady-state

$$u_s(x) = 1 - x,$$

which obeys the boundary conditions for Eq. (??). We therefore do not need to worry about these previous conditions on the solution u . We make a new function

$$v(x, t) = u(x, t) - u_s(x)$$

with boundary conditions $v(0) = v(d) = 0$ and initial condition $v(x, 0) = u(x, 0) - u_s(x)$. And this function $v(x, t)$ is easier to solve. By first solving $v(x, t)$ it is easy to find $u(x, t)$. $u(x, t)$ is simply $v(x, t) + u_s(x)$ where $u_s(x)$ is time-independent, and therefore can just be added after we know v . To find v we first use separation of variables.

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

$$\frac{\partial v(x, t)}{\partial t} = X(x) \frac{\partial T(t)}{\partial t}$$

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

$$X(x)\dot{T}(t) = T(t)\ddot{X}(x)$$

$$\frac{\dot{T}(t)}{T(t)} = \frac{\ddot{X}(x)}{X(x)}$$

For this to be true for all t and x , both the rhs and the lhs must be constant.

$$\frac{\dot{T}(t)}{T(t)} = k \Rightarrow \dot{T}(t) - kT(t) = 0$$

$$\frac{\ddot{X}(x)}{X(x)} = k \Rightarrow \ddot{X}(x) - kX(x) = 0$$

First we solve:

$$\ddot{X}(x) - kX(x) = 0.$$

Using the abc-method we find that $X = \pm\sqrt{k}$. We use $k = \mu^2$ and get $\sqrt{k} = \mu$. For $\mu^2 > 0$ the general solution becomes

$$X(x) = Ae^{\mu x} + Be^{-\mu x}.$$

Using boundary conditions $v(0) = v(d) = 0$

$$\begin{aligned} X(0) &= Ae^0 + Be^0 = 0 \\ \Rightarrow A + B &= 0 \end{aligned}$$

and

$$\begin{aligned} X(d) &= Ae^{\mu d} + Be^{-\mu d} = 0 \\ \Rightarrow A &= B = 0. \end{aligned}$$

This solution is not of interest. We instead look at $k = 0$. The general solution in this case is

$$X(x) = ax + b.$$

With boundary conditions we get

$$\begin{aligned} X(0) &= b = 0 \\ X(d) &= ad = 0 \Rightarrow a = 0. \end{aligned}$$

The only case of interest is then $k = -\mu^2$, when $-\mu^2 < 0$. We get the general solution

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

Boundary conditions give

$$X(0) = A \cos(0) + B \sin(0) = A = 0$$

$$X(d) = B \sin(\mu d) \Rightarrow \sin(\mu d) = 0.$$

We know that \sin is zero when $x = 0, \pi, 2\pi, \dots$, we therefore need $\mu d = n\pi$. We then get the solution for X ,

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

which of course is zero in the end points $x = 0$ and $x = d = 1$.

Now we solve:

$$\dot{T}(t) - kT(t) = 0.$$

We solve for T , and get $T = k$. The general solution then becomes

$$T(t) = Be^{-k2t} = Be^{-\mu^2 t}.$$

Since $\mu d = n\pi$,

$$T(t) = Be^{-\left(n\frac{\pi}{d}\right)^2 t}.$$

Now we can write out the solution of v .

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

$$A_n B_n = C_n$$

We use the initial condition

$$v(x, 0) = \sum_{n=1}^{\infty} C_n \sin\left(n\frac{\pi}{d}x\right) = f(x) = x - 1.$$

We recognize C_n as the inverse fourier coefficient to the function $f(x)$. Thus we get

$$C_n = 2 \int_0^1 (x - 1) \sin(n\pi x) dx.$$

$$\int_0^1 x \sin n\pi x \, dx = \frac{1}{n\pi}$$

$$- \int_0^1 \sin \pi x \, dx = \frac{2}{n\pi}$$

$$C_n = 2 \left(\frac{1}{n\pi} - \frac{2}{n\pi} \right) = -\frac{2}{n\pi}$$

Here we have set $d = 1$. Now the final solution becomes

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

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

Now it is easy to find u , when it is simply

$$u(x, t) = v(x, t) + u_s(x),$$

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

Now we have the closed form solution to our problem, but we also want to solve it numerically by using three different methods. The implicit Backward-Euler scheme, the explicit Forward-Euler scheme and the implicit Crank-Nicolson scheme.

The algorithms for these three methods goes as follows:

Backward Euler

$$u_t \approx \frac{u(x_i, t_j) - u(x_i, t_j - \Delta t)}{\Delta t}$$

$$u_{xx} \approx \frac{u(x_i + \Delta x, t_j) - 2u(x_i, t_j) + u(x_i - \Delta x, t_j)}{\Delta x^2}$$

If we now define $\alpha = \frac{\Delta t}{\Delta x^2}$, we get the implicit scheme

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

Now this can be written into a tridiagonal matrix system.

$$\mathbf{A} = \begin{pmatrix} 1 + 2\alpha & -\alpha & 0 & \cdots & 0 \\ -\alpha & 1 + 2\alpha & -\alpha & \cdots & 0 \\ 0 & -\alpha & 1 + 2\alpha & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & -\alpha & 1 + 2\alpha \end{pmatrix}$$

What we need to solve is

$$\mathbf{A}\mathbf{V}_j = \mathbf{V}_{j-1},$$

where \mathbf{V}_j is the unknown, and \mathbf{V}_{j-1} is our initial vector containing the values from our equation for the stationary-state system, $u_s(x) = 1 - x$. Now this is the same problem as in project 1. Using forward and backward substitution, we can find \mathbf{V}_j .

Forward Euler

$$u_t \approx \frac{u(x_i, t_j + \Delta t) - u(x_i, t_j)}{\Delta t}$$

$$u_{xx} \approx \frac{u(x_i + \Delta x, t_j) - 2u(x_i, t_j) + u(x_i - \Delta x, t_j)}{\Delta x^2}$$

Again we define $\alpha = \frac{\Delta t}{\Delta x^2}$ and get

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

$$\mathbf{A} = \begin{pmatrix} 1 - 2\alpha & \alpha & 0 & \cdots & 0 \\ \alpha & 1 - 2\alpha & \alpha & \cdots & 0 \\ 0 & \alpha & 1 - 2\alpha & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \alpha & 1 - 2\alpha \end{pmatrix}$$

$$\mathbf{A}\mathbf{V}_j = \mathbf{V}_{j+1}$$

Forward Euler is much easier than the implicit scheme, now all we need to do is a matrix-vector multiplication, and we find \mathbf{V}_j .

Crank-Nicolson

$$u_t \approx \frac{u(x_i, t_j + \Delta t) - u(x_i, t_j)}{\Delta t}$$

$$u_{xx} \approx \frac{1}{2} \left(\frac{u(x_i + \Delta x, t_j) - 2u(x_i, t_j) + u(x_i - \Delta x, t_j)}{\Delta x^2} + \frac{u(x_i + \Delta x, t_j + \Delta t) - 2u(x_i, t_j + \Delta t) + u(x_i - \Delta x, t_j + \Delta t)}{\Delta x^2} \right)$$

Using α as previous, we end up with

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

This gives

$$(2\mathbf{I} - \alpha\mathbf{B})\mathbf{V}_{j-1} = \mathbf{V}_j,$$

and

$$\mathbf{A} = \begin{pmatrix} 2 + 2\alpha & \alpha & 0 & \cdots & 0 \\ \alpha & 2 + 2\alpha & \alpha & \cdots & 0 \\ 0 & \alpha & 2 + 2\alpha & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \alpha & 2 + 2\alpha \end{pmatrix}.$$

And we can find \mathbf{V}_{j-1} by using Gaussian Elimination as we did for Backward Euler.

Now we have all we need to solve this problem numerically. But there are some things worth knowing about, as we use these methods. Such as what the truncation error we do, or how stable these methods are. This we can find deriving the expressions for the derivatives using Taylor expansion. First we look at the explicit scheme and find the new value of u on step forward in time.

$$u_{i,j+1} = u(x, t + \Delta t) = u(x, t) + \Delta t u'(x, t) + \frac{\Delta t^2}{2} u''(x, t)$$

$$u_t = u'(x, t)\Delta t = u(x, t + \Delta t) - u(x, t) - \frac{\Delta t^2}{2}u''(x, t)$$

$$u_t = u'(x, t) = \frac{u(x, t + \Delta t) - u(x, t)}{\Delta t} - \frac{\Delta t}{2}u''(x, t)$$

We see that the truncation error goes as $O(\Delta t)$.

To find the truncation error for the second derivative of x we must find an expression both for the value on step back in the position-space, and on step forward. We get

$$u_{i+1,j} = u(x + \Delta x, t) = u(x, t) + \Delta x u'(x, t) + \frac{\Delta x^2}{2}u''(x, t) + \frac{\Delta x^3}{3!}u'''(x, t) + \frac{\Delta x^4}{4!}u''''(x, t)$$

$$u_{i-1,j} = u(x - \Delta x, t) = u(x, t) - \Delta x u'(x, t) + \frac{\Delta x^2}{2}u''(x, t) - \frac{\Delta x^3}{3!}u'''(x, t) + \frac{\Delta x^4}{4!}u''''(x, t)$$

To find an expression for the second derivative, we add these to expression together.

$$u(x + \Delta x, t) + u(x - \Delta x, t) = 2(u(x, t) + \Delta x^2 u''(x, t) + \frac{2}{4!}\Delta x^4 u''''(x, t))$$

$$u''(x, t) = \frac{u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)}{\Delta x^2} - \frac{2}{4!}\Delta x^2 u''''(x, t)$$

And the truncation error goes as $O(\Delta x^2)$. This can be shown for the explicit scheme as well, but the Crank-Nicolson goes a bit different.

$$u(x, t) = u(x, t + \Delta t/2) - \frac{\partial u(x, t + \Delta t/2)}{\partial t} \frac{\Delta t}{2} + \frac{\partial^2 u(x, t + \Delta t/2)}{2\partial t^2} \Delta t^2 + O(\Delta t^3)$$

Then the derivatives become

$$\left[\frac{\partial u(x, t + \Delta t/2)}{\partial t} \right] = \frac{\partial u(x, t + \Delta t/2)}{\partial t} + O(\Delta t^2),$$

$$\left[\frac{\partial^2 u(x, t + \Delta t/2)}{\partial x^2} \right] = \frac{\partial^2 u(x, t + \Delta t/2)}{\partial x^2} + O(\Delta x^2).$$

And the truncation errors are $O(\Delta t^2)$ and $O(\Delta x^2)$. Both Backward Euler and Crank-Nicolson are stable for all Δx and Δt , but for Forward Euler, we must demand $\Delta t < \frac{1}{2}\Delta x^2$. Meaning we must have $\alpha < \frac{1}{2}$.

Results

We've tested the solution for each of the three methods for $\Delta x = 0.1$ and $\Delta x = 0.01$. The value of Δt is set on the basis of the stability limit of the Forward Euler scheme, i.e. $\Delta t = 0.5(\Delta x)^2$.

When $\Delta x = 0.1$, $\Delta t = 0.005$ and we've set $t_{fin} = 1.0$ (final value of t) so that the total number of time steps is 200. We will study the solution at two time points t_1 and t_2 where $u(x, t_1)$ is smooth but still significantly curved and $u(x, t_2)$ is almost linear. We've found that the curves $u(x, t_1 = 9\Delta t)$ and $u(x, t_2 = 49\Delta t)$ fit these descriptions well. Figure 1 and figure 2 respectively shows these curves for all the three methods together with the closed-form solution we found earlier.

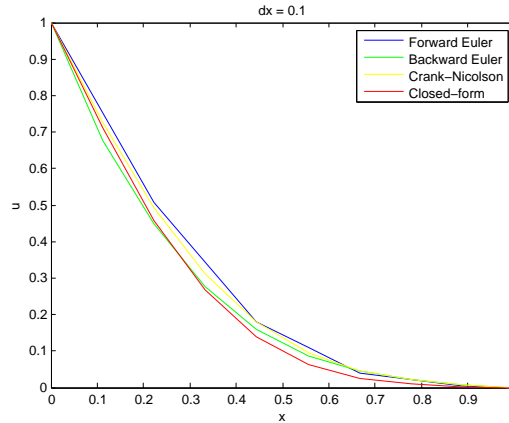


Figure 1: $u(x, t_1)$ for all three methods together with closed-form solution

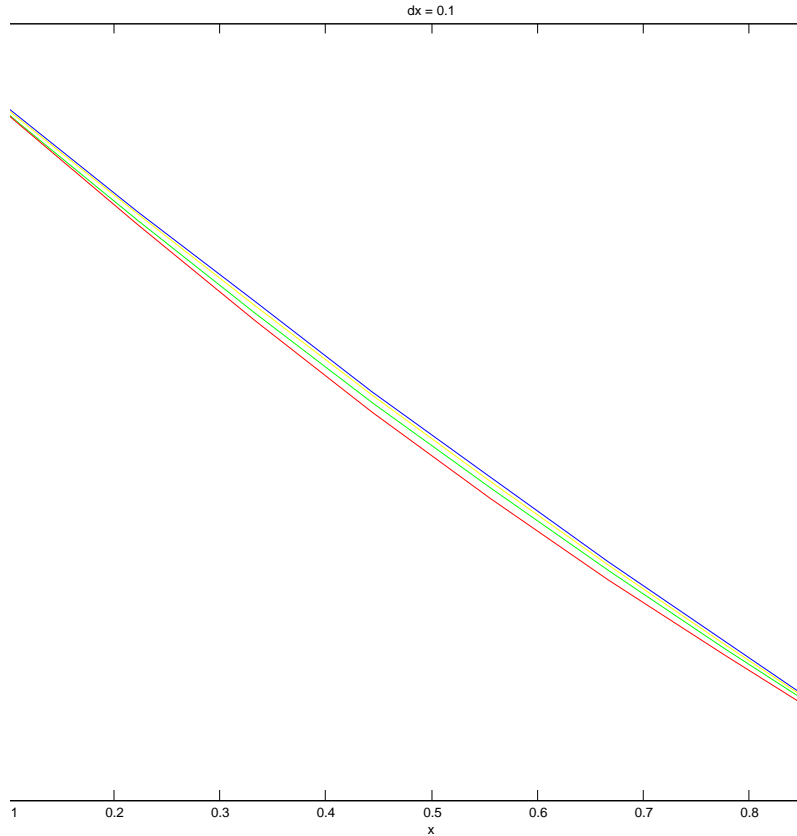


Figure 2: $u(x, t_2)$ for all three methods together with closed-form solution

We see that it's actually the Backward Euler curve that fits the closed-form solution best for both t_1 and t_2 . This is a bit strange considering that the Crank-Nicolson algorithm has a lower truncation error than the other two. There can be three explanations for this: There is either something wrong or inaccurate about our implementation of the C-N scheme, the Backward Euler method works better for this specific problem or there are other sources to loss of numerical precision at play.

Figure 3 and 4 shows the absolute error for each method, which clearly shows that Backward Euler has the smallest deviation from the closed-form solution. The error of Crank-Nicolson is a bit larger, but still lower than for the Forward Euler method. All three errors is of magnitude 10^{-2} .

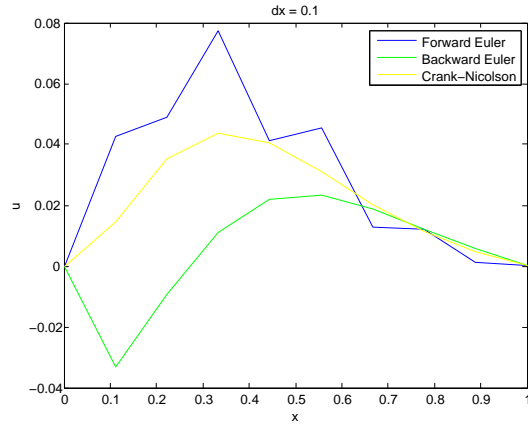


Figure 3: Absolute error (numerical - analytical), $t = t_1$

We will now test the methods for $\Delta x = 0.01$. Now, $\Delta t = 0.00005$ and for $t_{fin} = 0.1$ we have a total of 20000 time steps. We therefore increase t_1 and t_2 by a factor of 1000. Plots like figure 1 and 2 are now not that interesting since the curves practically lies on top of each other, thus we rather show the absolute error for both t in figure 5 and 6.

Now we see even more clearly that Backward Euler works best for this problem. It has the lowest error except for the jump in the beginning of the interval for both t_1 and t_2 .

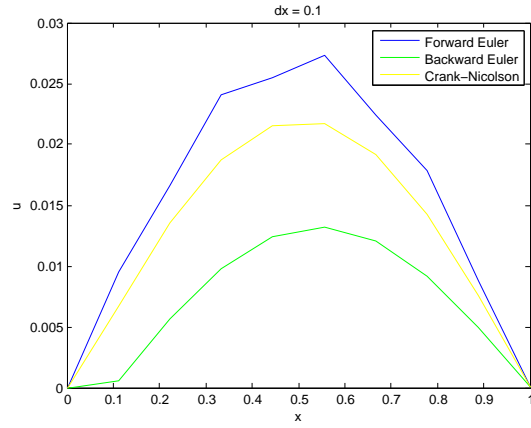


Figure 4: Absolute error (numerical - analytical), $t = t_2$

Conclusion

In this project we've simulated the diffusion of neurotransmitters in the synaptic cleft using the diffusion equation in one dimension. This partial differential equation have been solved using three approaches: Forward Euler, Backward Euler and Crank-Nicolson. After testing the solutions of these three methods by comparing them to the closed-form solution, we have found that the Backward Euler method is the one that works best for this problem, despite Crank-Nicolson having the lowest truncation error. This is probably due to the fact that this method has the largest number of floating-point operations, which can lead to loss of numerical precision. All three methods do nevertheless have an error of magnitude 10^{-3} when $\Delta x = 0.01$, which is a quite good result.

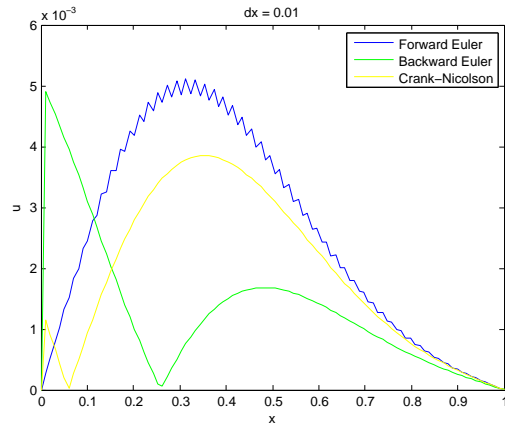


Figure 5: Absolute error (numerical - analytical), $t = t_1$

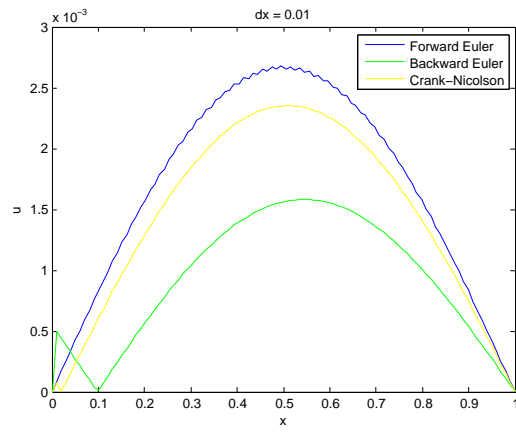


Figure 6: Absolute error (numerical - analytical), $t = t_2$