Project 5, FYS4150

Candidate 55

December 6, 2012

Contents

1 About the problem

In this project we will look at the transportation of neurotransmitter molecules across the so called synaptic cleft separating a brain cell and a target cell. After an "action-potential" is recieved (in the axon terminal) vesicles inside the axon terminal merge with the presynaptic membrane, releasing the neurotransmitter molecules into the synaptic cleft. A vesicle is a kind of "bubble" inside a cell. Here we are talking about vesicles containing neurotransmitter moleules located in the axon (or nerve fibers) of a brain cell. The molecules then diffuse across the synaptic cleft, and are picked up by receptors on the target cell (or the postsynaptic side of the synaptic cleft if you want). We want to modell this particular event using a continuum modell, and the mathematical expression is the diffusion equation.

$$\frac{\partial u}{\partial t} = D\nabla^2 u$$

Where D is a diffusion coefficient. An illustrative figure of the problem can be found in figure (1).

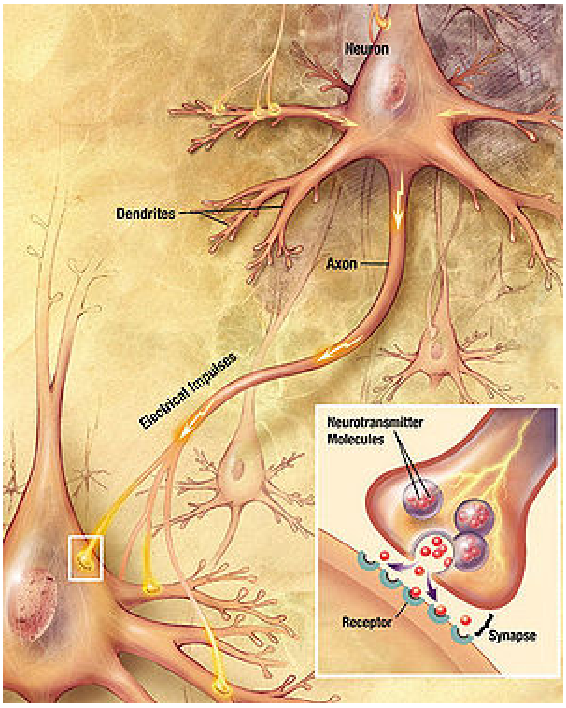

Figure 1: Illustration of communication between two brain cells and a close-up of the synaptic cleft separating the axon (presynaptic side) and the target cell (postsynaptic side). (From wikipedia search Chemical synapse)

We will assume that the synaptic cleft is of roughly equal width, and that the area of the synaptic cleft is large compared to its width. The concentration of neurotransmitters therefore only vary in one direction, from presynaptic to postsynaptic side. The diffusion equation then reduces to

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2}$$

which we can rescale to be dimensionless by introducing $x = \alpha\tilde{x}$

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\alpha^2 \partial \tilde{x}^2}$$

and define $\alpha^2 = D$; $\tilde{x} = x$ giving us the diffusion equation in its simplest form.

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

We will say that at some time $t = 0$ the "action potential" is recieved, and the vesicles merge with the presynaptic membrane meaning that the initial distribution of neurotransmitters is $u(x, 0) = \delta(x)$, that is 1 at $x = 0$ and 0 everywhere else. Furthermore we say that the neurotransmitters which reach the postsynaptic membrane (or the receptors) are removed from the synaptic cleft (and our system). This means that at the far side ($x = d$) $u(d, t) = 0$. We are now ready to solve the equation, see sections 2 and 3.
We will also extend the diffusion equation to two spatial dimensions, also using dimensionless variables. The

boundary and initial conditions of the two dimensional equation are quite different from the one dimensional case. We use $x, y \in [0, 1]$,

$$u(x, y, 0) = (1 - y)e^x$$

and

$$\begin{aligned} u(0, y, t) &= (1 - y)e^t \\ u(1, y, t) &= (1 - y)e^{1+t} \\ u(x, 0, t) &= e^{x+t} \\ u(x, 1, t) &= 0 \end{aligned}$$

1.1 Notation

For finite differences I will use the the notation

$$u(t_n, x_i, y_j) = u^n_{i,j}$$

where $t_n = t_0 + n \cdot \Delta t$, $x_i = x_0 + i \cdot \Delta x$ and $y_j = y_0 + j \cdot \Delta y$. Thus it is important not to confuse u^n (u to the power of n) with $u^n_{i,j}$ (u evaluated at timestep n and position i,j). I will at some points use the matrix B to simplify writing. B is defined to be

$$\mathbf{B} = \begin{pmatrix} 2 & -1 & 0 & \dots & & 0 & 0 \\ -1 & 2 & -1 & 0 & \dots & & 0 \\ 0 & \ddots & \ddots & \ddots & & & 0 \\ 0 & \dots & & 0 & -1 & 2 & -1 \\ 0 & \dots & & & 0 & -1 & 2 \end{pmatrix}$$

2 The algorithm

We will solve the 1+1 dimensional difusion equation by three different finite differnce schemes in this project. Using the standard approximation of the second derivative in space, we use successively more elaborate approximations to the time derivative.

2.1 Forward Euler

Starting off with the Forward Euler (FE) approximation we get the following scheme

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

$$u_i^{n+1} = \frac{\Delta t}{\Delta x^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) + u_i^n \tag{1}$$

So to solve the equation all we have to do is loop over the two variables and we are done.

```
for n = 0,1, ... , N
    for i = 0,1, ... , N_x
        u_new[i] = dtdx2*(u_prev[i+1]-2*u_prev[i] + u_prev[i-1]) + u_prev[i];
    u_prev = u_new;
end;
```

A discussion of stability and the error in this scheme can be found in the "Stability and precision" section.

2.2 Backward Euler

The Backward Euler (BE) approximation gives us a slightly more elaborate scheme seeing at it is an implicit one. The discretization gives

$$\begin{aligned} \frac{u_i^n - u_i^{n-1}}{\Delta t} &= \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} \\ u_i^n - \frac{\Delta t}{\Delta x^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) &= u_i^{n-1} \\ u_i^n\left(1 + 2\frac{\Delta t}{\Delta x^2}\right) - u_{i-1}^n\frac{\Delta t}{\Delta x^2} - u_{n+1}^n\frac{\Delta t}{\Delta x^2} &= u_i^{n-1} \end{aligned}$$

If we insert for a few steps we see that this takes the form of

$$\begin{pmatrix} \left(1+2\frac{\Delta t}{\Delta x^2}\right) & -\frac{\Delta t}{\Delta x^2} & 0 & \dots & & & 0 & 0 \\ -\frac{\Delta t}{\Delta x^2} & \left(1+2\frac{\Delta t}{\Delta x^2}\right) & -\frac{\Delta t}{\Delta x^2} & 0 & \dots & & & 0 \\ 0 & \ddots & \ddots & \ddots & 0 & \dots & & 0 \\ 0 & \dots & & & 0 & -\frac{\Delta t}{\Delta x^2} & \left(1+2\frac{\Delta t}{\Delta x^2}\right) & -\frac{\Delta t}{\Delta x^2} \\ 0 & \dots & & & & 0 & -\frac{\Delta t}{\Delta x^2} & \left(1+2\frac{\Delta t}{\Delta x^2}\right) \end{pmatrix} \mathbf{u}^n = \mathbf{u}^{n-1} \quad (2)$$

$$\mathbf{A}\mathbf{u}^n = \mathbf{u^{n-1}}$$

So for each timestep we need to multiply the inverse of $\mathbf{A}$ with a vector $\mathbf{u}^{n-1}$ containig the solution at the previous timestep. Looking a bit closer at equation (2) we realize two things. First of all, the matrix $\mathbf{A}$ does'nt change at all throughout the computation so we can get away with invertig it once. Second, and perhaps more important, we allready have a very efficient solver for this kind of problem from project 1.

A discussion of stability and the error in this scheme can be found in the "Stability and precision" section.

2.3 Crank Nicolson

Finally, we can use the Crank Nicolson approximation of the time derivative which is a special case of the so called theta -rule with $\theta = 0.5$.

$$\begin{aligned} \frac{u_i^n - u_i^{n-1}}{\Delta t} &= \frac{1}{2\Delta x^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) + \frac{1}{2\Delta x^2}\left(u_{i+1}^{n-1} - 2u_i^{n-1} + u_{i-1}^{n-1}\right) \\ u_i^n - \frac{\Delta t}{2\Delta x^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) &= u_i^{n-1} + \frac{\Delta t}{2\Delta x^2}\left(u_{i+1}^{n-1} - 2u_i^{n-1} + u_{i-1}^{n-1}\right) \\ u_i^n(2+2C) - Cu_{i+1}^n - Cu_{i-1}^n &= u_i^{n-1}(2-2C) + Cu_{i+1}^{n-1} + Cu_{i-1}^{n-1} \end{aligned}$$

where $C = \frac{\Delta t}{\Delta x^2}$. This can also be expressed as a linear algebra problem

$$(2\mathbf{I} + C\mathbf{B})\mathbf{u}^n = (2\mathbf{I} - C\mathbf{B})\mathbf{u}^{n-1} \quad (3)$$

Again we observe that we will need to invert a matrix, but this time we will also need to do a matrix-vector multiplication.

$$\begin{aligned} (2\mathbf{I} + C\mathbf{B})\mathbf{u}^n &= (2\mathbf{I} - C\mathbf{B})\mathbf{u}^{n-1} = \tilde{\mathbf{u}}^{n-1} \\ \mathbf{u}^n &= (2\mathbf{I} + C\mathbf{B})^{-1}\tilde{\mathbf{u}}^{n-1} \end{aligned}$$

This is the same problem as we had in the BE case, only with a modified right-hand side

2.4 2+1 dimensional diffusion equation

The Jacobi algorithm for solving linear systems is an iterative scheme very well suited for solving the Poisson and Laplace equations in 2 or 3 dimensions. If we breefly look at the discretization of the Laplace equation in 2D (4) we see that there is no obvious way to solve for a new point since we do not know 3 out of 5 points needed to calculate a new one. The solution is then (since there is no initial condition either) to start out with an educated guess, and use the scheeme (5) point by point on the entire grid several times until the solution (hopefully) converges.

$$\nabla^2 u(x,y) = 0$$

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2} = 0 \quad (4)$$

$$u_{i,j} = \frac{1}{4}\left(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}\right) \quad (5)$$

Where we have assumed that $\Delta x = \Delta y$. Seeing as we have a time dependence, we also have an initial condition on our system. Therefore we allready know all values at previous timesteps (since the sceme is recursive), and we do not have to start with a quess of what the solution might look like. In stead, but still with the mentality of the Jacobi algorithm of using the previous value to get the new one, we can simply derive an explicit scheme by inserting our approximate derivatives. For example if we approximate the time derivative using the FE method we get

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

This is the same scheme as in 1 only with one more dimension. As we will se in the "stability and precision" section the centered difference or "Leap Frog" scheme has a better truncation error with respect to time than the FE scheme, and this will therefore (most likely) yield better results. The Leap Frog scheme is

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

We could also make an explicit scheme by the mentality in the Gauss-Seidel algorithm. That is to use some of the new values to get a new value. In one dimension this is known as the Euler-Chromer method and is known to be quite good for its simplicity. If we look closer at the FE or Leap Frog scheme we notice that we allready know the values $u_{i-1,j}^{n+1}$ and $u_{i,j-1}^{n+1}$. Using these values might give us better accuracy (at least compared to the FE case)

3 Analytic solution

As a comparison we can find the analytic solution to this problem as follows.

$$\frac{\partial u}{\partial t}=D\frac{\partial^2 u}{\partial x^2},\quad u(x,0)=u(d,t)=0 \tag{6}$$
$$x\in[0,d],\quad D=d=u(0,t)=1 \tag{7}$$

We see right away that the boundary $x=0$ could give us some problems, so we start off with a small trick

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

where the $u_s=1-x$ term is the steady-state solution to equation 6. This trick leaves us with new boundary conditions un $u(x,t)$

$$v(0,t)=u(0,t)-u_s(0,t)=1,\ \ u_s(0,t)=1\implies u(0,t)=0$$

which makes the whole procedure much simpler. We now assume that $u(x,t)$ can be separated into factors

$$u(x,t)=F(x)G(t)\implies\frac{\partial u}{\partial t}=\frac{\partial^2 u}{\partial x^2}\to F(x)\frac{\partial G}{\partial t}=G(t)\frac{\partial^2 F}{\partial x^2}$$
$$\frac{1}{G(t)}\frac{\partial G}{\partial t}=-k^2=\frac{1}{F(x)}\frac{\partial^2 F}{\partial x^2}$$

We start with the simplest of the equations which is the time dependence

$$\frac{1}{G(t)}\frac{\partial G}{\partial t}=-k^2$$
$$G(t)=Ce^{-k^2 t}$$

and leave it like this for now. The x-dependent equation is somewhat more complicated

$$\frac{1}{F(x)}\frac{\partial^2 F}{\partial x^2}=-k^2$$
$$F(x)=A\sin(kx)+B\cos(kx)$$

$$F(0)=A\sin(0)+B\cos(0)=0\implies B=0$$
$$F(d)=A\sin(kd)=0\implies k=\frac{m\pi}{d}=\pi m,\quad A=A_m$$

we now combine all the equations to determine $v(x,t)$

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

$$v(x,0) = 1 - x + \sum_{m=1}^{\infty} B_m \sin(m\pi x)$$

$$\implies \int_0^1 \sin(m\pi x)\sin(n\pi x)dx = \delta_{mn} = \int_0^1 (x-1)\sin(m\pi x)dx = -\frac{2}{m\pi} = B_m$$

This gives us the full analytical solution

$$v(x,t) = 1 - x - \frac{2}{\pi}\sum_{m=1}^{\infty} \frac{1}{m} e^{-(m\pi)^2 t} \sin(m\pi x) \tag{8}$$

which satisfies all initial and boundary conditions.

4 Results/Verification

From section 3 we have the analytic solution to our problem, and we can use this to verify our numerical schemes. Now we know from section 5 that both the FE and the BE schemes have errors going like $\mathcal{O}(\Delta t)$, and that the FE scheme has a stability criterion of $\Delta t \leq \frac{\Delta x^2}{2}$. From experience I know that using exactily this criterion can be bad, so we will use half of this and we will use the same for all the schemes so that we can easily compare results. In the analytic solution we have an infinite sum, which we of course have to cut at some point. Looking at our solution (8) we see that $t = 0$ will have the slowest convergence because there will be no additional dampening term in the exponential term. From figure (2) we see that there is very little change in afer 25 terms. It will then be very safe to terminate the sum after some 100-200 terms to make sure no “invisible” roundoffs are made.

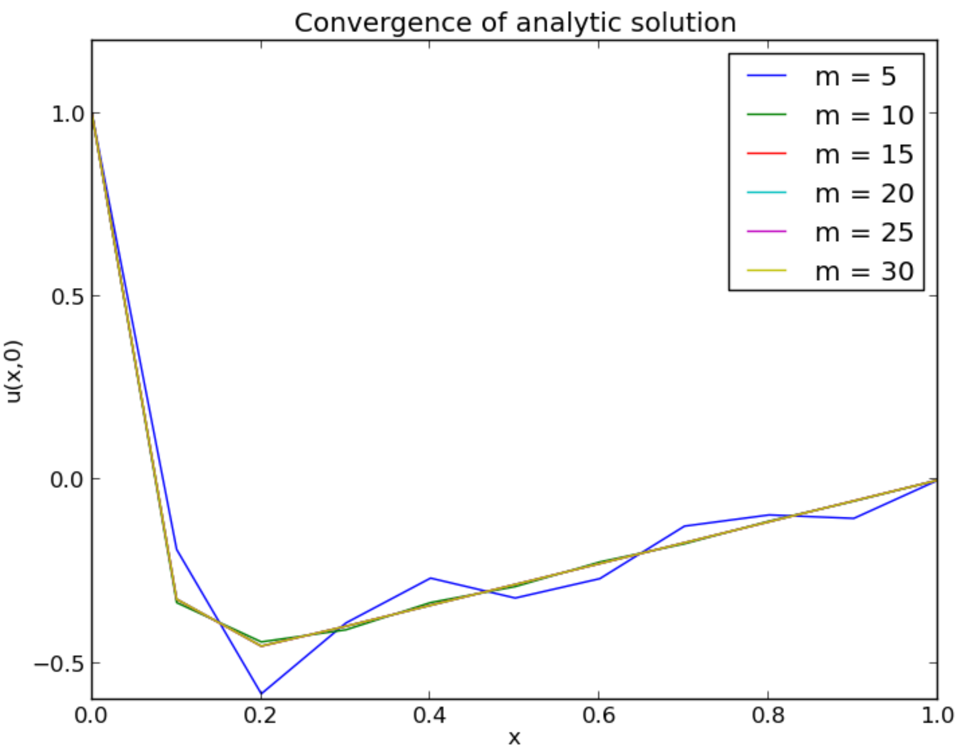

Figure 2: Plot of the analytic solution at $t = 0$ while adding more terms.

The errors we get from the different schemes have been visualized in figures (3) and (4). As mentioned the error should go like $\mathcal{O}(\Delta t)$.

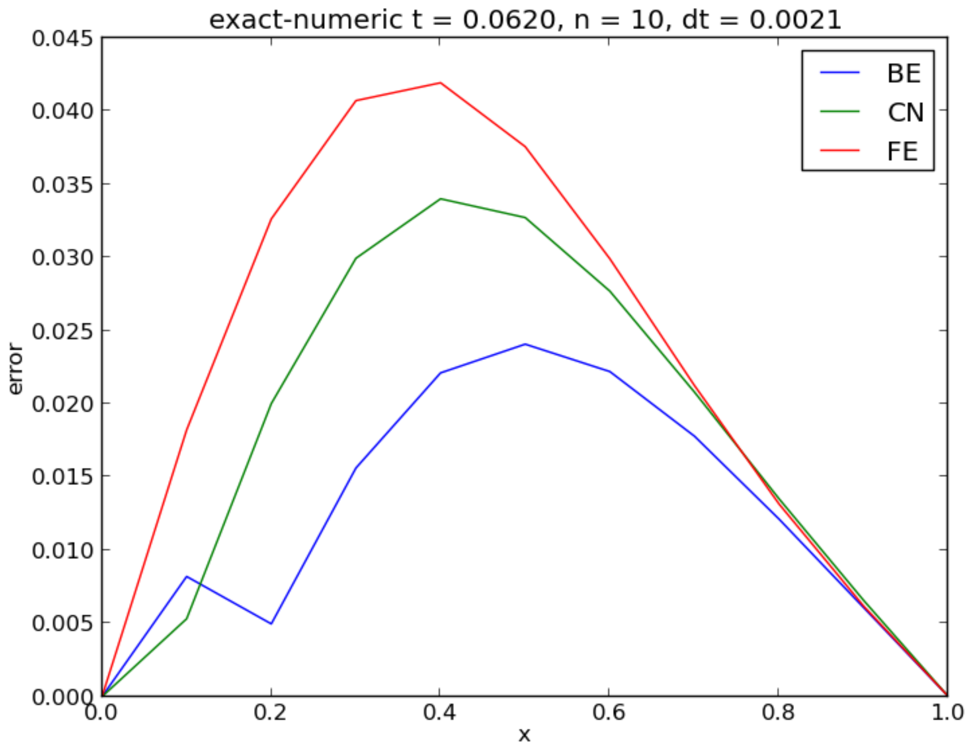

Figure 3: The absolute error for $\Delta x =^1 /_{10}$.

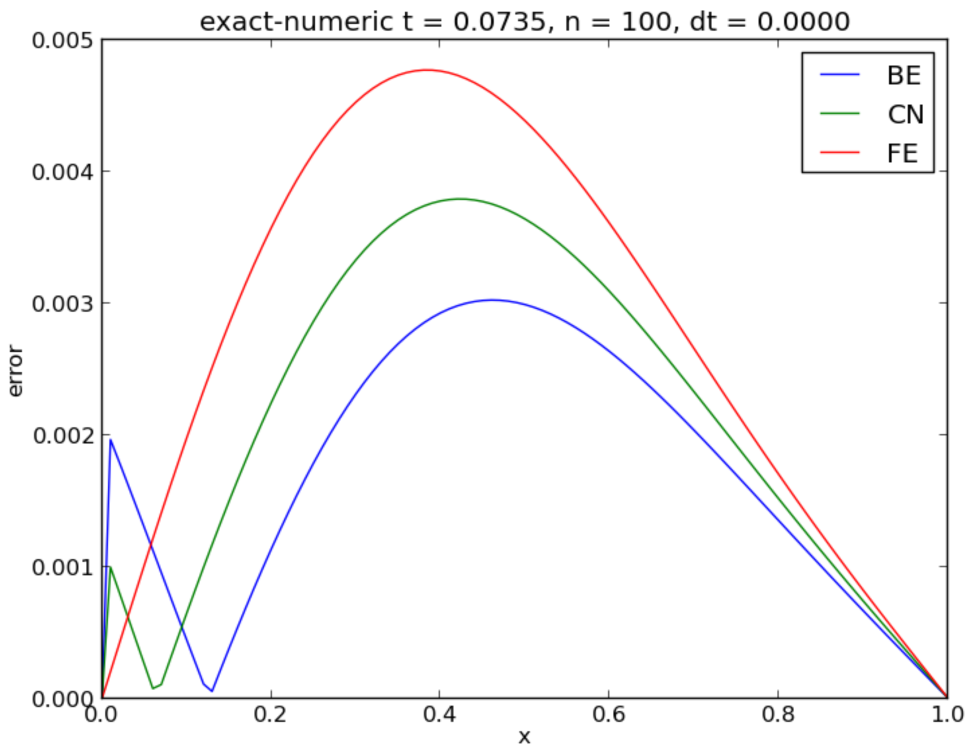

Figure 4: The absolute error for $\Delta x =^1 /_{100}$.

We notice that contrary to the results of the truncation error the BE scheme gives the best results. As another verification we could let the simulation run untill the steady state is reached. Since the steadu state is a first order polynomial, this should be represented to machine precision, which means that the error should go to 0 as the number of time-steps gets large. However, as is also expected in a diffusion equation, the convergence is very slow after the simulation has run for a while. we can see the results of this experiment in figures (5) and (6).

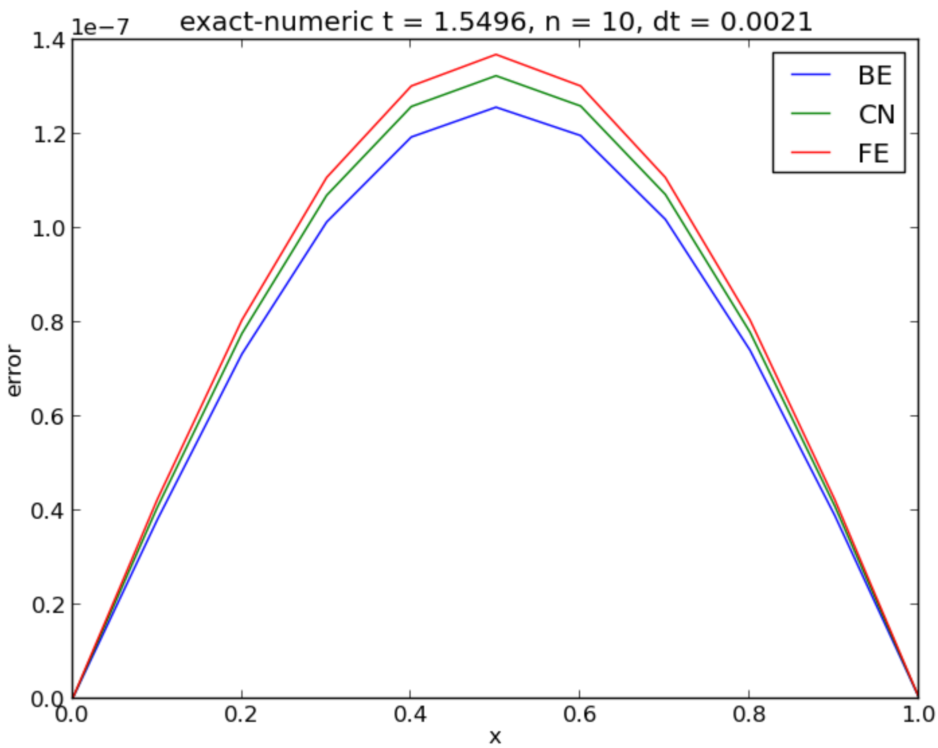

Figure 5: The absolute error for $\Delta x =^1 /_{100}$.

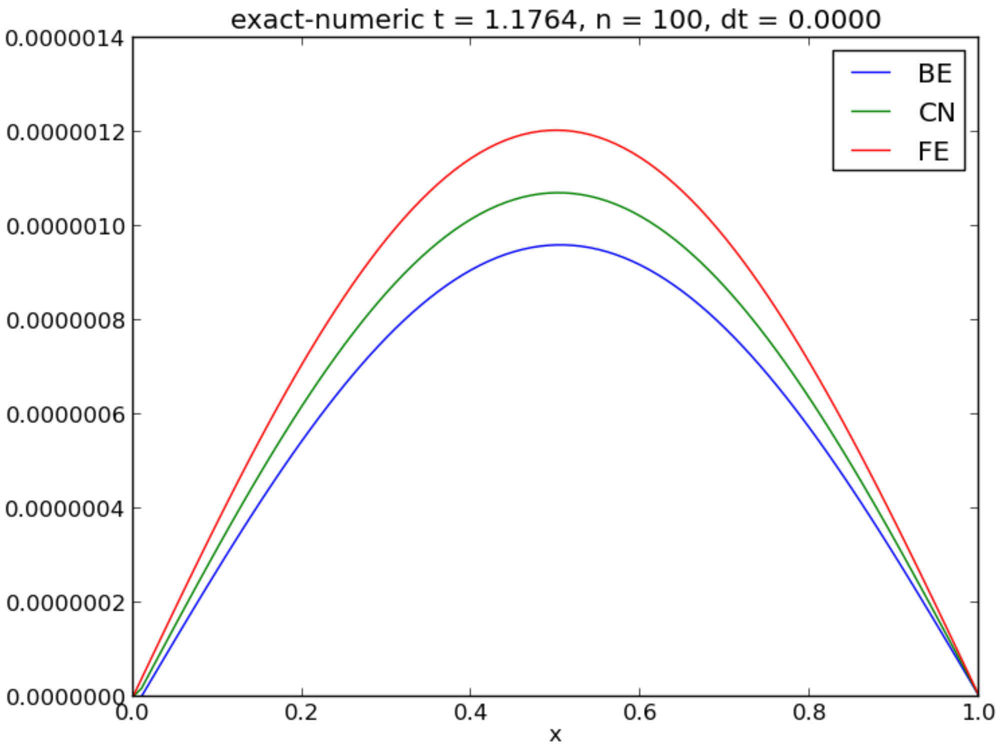

Figure 6: The absolute error for $\Delta x =^1 /_{100}$.

5 Stability and precision

To get a feeling of how good the numerical schemes are we will analyze their errors and stability criterions indivudually.

5.1 Forward Euler

Let us fist look at the truncation error of this scheme. If we do a Taylor expansion of $u(x+\Delta x,t)$, $u(x-\Delta x,t)$ and $u(x,t+\Delta t)$, assuming $\Delta x, \Delta t \to 0$ we get the following

$$\begin{aligned}
u(x+\Delta x,t) &= u(x,t) + \frac{\partial u(x,t)}{\partial x}\Delta x + \frac{\partial^2 u(x,t)}{2\partial x^2}\Delta x^2 + \mathcal{O}(\Delta x^3) \\
u(x-\Delta x,t) &= u(x,t) - \frac{\partial u(x,t)}{\partial x}\Delta x + \frac{\partial^2 u(x,t)}{2\partial x^2}\Delta x^2 + \mathcal{O}(\Delta x^3) \\
u(x,t+\Delta t) &= u(x,t) + \frac{\partial u(x,t)}{\partial x}\Delta t + \mathcal{O}(\Delta t^2)
\end{aligned}$$

Theese are the local errors, meaning that our complete scheme gives the error of

$$\begin{aligned}
\frac{u^{n+1}-u^n}{\Delta t} &\approx \frac{\partial u(x,t)}{\partial t} + \mathcal{O}(\Delta t) \\
\frac{u_{i+1}-2u_i+u_{i-1}}{\Delta x^2} &\approx \frac{\partial^2 u(x,t)}{\partial x^2} + \mathcal{O}(\Delta x^2)
\end{aligned}$$

Let us also look at the stability of the scheme, but by in slightly different way than what has been done in the rest of the course. Let us assume that the slution of the diffusion equation takes the form

$$u(x,t) = A^n e^{ikq\Delta x} \tag{9}$$

Where A^n is a damping factor which is time dependent. If we insert this expression 9 into the scheme 1 we get

$$\begin{aligned}
A^n e^{ikq\Delta x}(A-1) &= CA^n e^{ikq\Delta x}(e^{i(q+1)k\Delta x} - 2e^{iqk\Delta x} + e^{i(q-1)k\Delta x}) \\
A-1 &= C(e^{ik\Delta x} - 2 + e^{-ik\Delta x}) \\
A &= 1 - 4C(cos(k\Delta x)-1) = 1 - 4C\sin^2(k\Delta x/2)
\end{aligned}$$

Note that x_i has been replaced by x_q. This means that the final solution is on the form

$$u_q^n = \left(1 - 4C\sin^2(k\Delta x/2)\right)^n e^{ikq\Delta x}$$

It is obvious that we need $-1 \leq A \leq 1$ to avoid divergence. $A \leq 1 \implies C \geq 0$ is not of to much interest, however if we insert for

$$\begin{aligned} -1 \leq A \implies 1 - 4C\sin^2(k\Delta x/2) &\geq -1 \\ C &\leq \frac{1}{2\sin^2(k\Delta x/2)} \end{aligned}$$

and $max(\sin^2(k\Delta x/2)) = 1$ we are left with

$$C = \frac{\Delta t}{\Delta x^2} \leq \frac{1}{2} \tag{10}$$

5.2 Backward Euler

The truncation error of the BE scheme is very similar to the FE case. In fact it is obvious that the spacial part is identical, so we only need to redo the time dependence.

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

And so the complete scheme gives an error of

$$\begin{aligned} \frac{u^n - u^{n-1}}{\Delta t} &\approx \frac{\partial u(x,t)}{\partial t} + \mathcal{O}(\Delta t) \\ \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} &\approx \frac{\partial^2 u(x,t)}{\partial x^2} + \mathcal{O}(\Delta x^2) \end{aligned}$$

We can investigate the stability of the BE scheme in the same way as we did for the FE scheme. Assume again that we have a general solution on the form 9. Insert it in 2 as follows

$$\begin{aligned} A^n e^{ikq\Delta x}(1 - A^{-1}) &= CA^n \left(e^{ik(q+1)\Delta x} - 2e^{ikq\Delta x} + e^{ik(q-1)\Delta x}\right) \\ A^{-1} &= 1 - 4C\sin^2(k\Delta x/2) \\ A &= \frac{1}{1 - 4C\sin^2(k\Delta x/2)} \end{aligned}$$

giving us the general term

$$u_q^n = \left(\frac{1}{1 - 4C\sin^2(k\Delta x/2)}\right)^n e^{ikq\Delta x}$$

but now we have that $0 \leq A \leq 1$.

$$\begin{aligned} \frac{1}{1 - 4C\sin^2(k\Delta x/2)} &\leq 1 \\ 1 \leq 1 - 4C\sin^2(k\Delta x/2) &\implies C \geq 0 \end{aligned}$$

The other limitation ($A \geq 0$) gives us nothing and thus the scheme is stable for all values of Δt and Δx.

5.3 Crank Nicolson

The truncation error of the CN scheme is slightly more intricate to calculate because we need to Taylor expand around $t' = t + \Delta t/2$

$$\begin{aligned}
u(x+\Delta x, t+\Delta t) &= u(x,t') + \frac{\partial u(x,t')}{\partial x}\Delta x + \frac{\partial u(x,t')}{\partial x}\Delta t/2 + \frac{\partial^2 u(x,t')}{2\partial x^2}\Delta x^2 + \frac{\partial^2 u(x,t')}{2\partial t^2}\Delta t^2/4 \\
&+ \frac{\partial^2 u(x,t')}{\partial x \partial t}\Delta x \Delta t/2 + \mathcal{O}(\Delta x^3) \\
u(x-\Delta x, t+\Delta t) &= u(x,t') - \frac{\partial u(x,t')}{\partial x}\Delta x + \frac{\partial u(x,t')}{\partial x}\Delta t/2 + \frac{\partial^2 u(x,t')}{2\partial x^2}\Delta x^2 + \frac{\partial^2 u(x,t')}{2\partial t^2}\Delta t^2/4 \\
&- \frac{\partial^2 u(x,t')}{\partial x \partial t}\Delta x \Delta t/2 + \mathcal{O}(\Delta x^3) \\
u(x+\Delta x, t) &= u(x,t') + \frac{\partial u(x,t')}{\partial x}\Delta x - \frac{\partial u(x,t')}{\partial x}\Delta t/2 + \frac{\partial^2 u(x,t')}{2\partial x^2}\Delta x^2 + \frac{\partial^2 u(x,t')}{2\partial t^2}\Delta t^2/4 \\
&- \frac{\partial^2 u(x,t')}{\partial x \partial t}\Delta x \Delta t/2 + \mathcal{O}(\Delta x^3) \\
u(x-\Delta x, t) &= u(x,t') - \frac{\partial u(x,t')}{\partial x}\Delta x - \frac{\partial u(x,t')}{\partial x}\Delta t/2 + \frac{\partial^2 u(x,t')}{2\partial x^2}\Delta x^2 + \frac{\partial^2 u(x,t')}{2\partial t^2}\Delta t^2/4 \\
&+ \frac{\partial^2 u(x,t')}{\partial x \partial t}\Delta x \Delta t/2 + \mathcal{O}(\Delta x^3) \\
u(x, t+\Delta t) &= u(x,t') + \frac{\partial u(x,t')}{\partial t}\Delta t/2 + \frac{\partial^2 u(x,t')}{2\partial t^2}\Delta t^2/4 + \mathcal{O}(\Delta t^3) \\
u(x, t) &= u(x,t') - \frac{\partial u(x,t')}{\partial t}\Delta t/2 + \frac{\partial^2 u(x,t')}{2\partial t^2}\Delta t^2/4 + \mathcal{O}(\Delta t^3)
\end{aligned}$$

Which means that the error of the whole scheme scales like

$$\begin{aligned}
\frac{\partial u(x,t')}{\partial t} &\approx \frac{\partial u(x,t')}{\partial t} + \mathcal{O}(\Delta t^2) \\
\frac{\partial^2 u(x,t')}{\partial x^2} &\approx \frac{\partial^2 u(x,t')}{\partial x^2} + \mathcal{O}(\Delta x^2)
\end{aligned}$$

We can check the stability of the CN scheme in the same way that we checked it for FE and BE schemes by assuming the same solution 9 as we did for the FE and BE case, and insert it in the CN scheme.

$$\begin{aligned}
A^n e^{ikq\Delta x}(1 - A^{-1}) &= A^n \frac{C}{2}\big(e^{i(q+1)k\Delta x} - 2e^{iqk\Delta x} + e^{i(q-1)k\Delta x}\big) + A^{n-1}\frac{C}{2}\big(e^{i(q+1)k\Delta x} - 2e^{iqk\Delta x} + e^{i(q-1)k\Delta x}\big) \\
(1 - A^{-1}) &= \frac{C}{2}\left(-4\sin^2(k\Delta x/2) - 4A^{-1}\sin^2(k\Delta x/2)\right) \\
1 + 2C\sin^2(k\Delta x/2) &= A^{-1}\big(1 - 2C\sin^2(k\Delta x/2)\big) \\
A &= \frac{1 - 2C\sin^2(k\Delta x/2)}{1 + 2C\sin^2(k\Delta x/2)}
\end{aligned}$$

And the limitations are $-1 \leq A \leq 1$

$$\begin{aligned}
\frac{1 - 2C\sin^2(k\Delta x/2)}{1 + 2C\sin^2(k\Delta x/2)} &\leq 1 \\
1 - 2C\sin^2(k\Delta x/2) &\leq 1 + 2C\sin^2(k\Delta x/2) \\
\implies C &\geq 0
\end{aligned}$$

The other limitation ($A \geq -1$) gives us nothing and thus the CN scheme is stable for all choises of Δt and Δx.

5.4 Schemes for diffusion in multiple dimensions

We can try the same approach to investigate the stability of the schemes in multiple dimesions. We start out with a general solution on the form

$$A^n e^{ik(p\Delta x + q\Delta y)}$$

and insert it in the respective schemes.

$$\begin{aligned}
&A^n e^{ik(p\Delta x + q\Delta y)}\big(A - 1\big) = CA^n e^{ikq\Delta y}\big(e^{ik(q+1)\Delta x} - 2e^{ikq\Delta x} + e^{ik(q-1)\Delta x}\big) + \\
&CA^n e^{ikp\Delta x}\big(e^{ik(p+1)\Delta y} - 2e^{ikp\Delta y} + e^{ik(p-1)\Delta y}\big) \\
&(A - 1) = -4C\sin^2(k\Delta x/2) - 4C\sin^2(k\Delta y/2)
\end{aligned}$$

Plugging in for A we see that the limitations on A are $-1 \leq A \leq 1$. We will also insert the maximum value for the sine terms.

$$-8C + 1 \geq -1$$
$$8C \leq 2 \implies C \leq \frac{1}{4}$$

We see that the other limit only gives us $C \geq 0$ which really is not that exciting.
For the leap frog method we get

$$A^n e^{ik(p\Delta x + q\Delta y)}\big(A - A^{-1}\big) = 2CA^n e^{ikq\Delta y}\big(e^{ik(q+1)\Delta x} - 2e^{ikq\Delta x} + e^{ik(q-1)\Delta x}\big) +$$
$$2CA^n e^{ikp\Delta x}\big(e^{ik(p+1)\Delta y} - 2e^{ikp\Delta y} + e^{ik(p-1)\Delta y}\big)$$
$$(A - A^{-1}) = -8C(\sin^2(k\Delta x/2) + \sin^2(k\Delta y/2))$$

To avoid som unnececarily nasty expressions we will from here on look at the "worst case" where $\sin^2(k\Delta x/2) = \sin^2(k\Delta y/2) = 1$

$$A^2 - 16CA - 1 = 0$$
$$A = \frac{1}{2}\big(-16C \pm \sqrt{16^2C^2 + 4}\big) = -8C \pm \sqrt{64C^2 + 1}$$

That is A will be a linear combination of the two roots $A_1 = -8C - \sqrt{64C^2 + 1}$ and $A_2 = -8C + \sqrt{64C^2 + 1}$. We notice that A_1 is a negative root which will cause the solution to oscillate.

$$A^n = \alpha_1 A_1^n + \alpha_2 A_2^n$$

we can determine the coeficcients α_1 and α_2 from the initial condition $u(x, y, 0) = I$

$$A^0 = \alpha_1 + \alpha_2 = I \implies \alpha_1 = I - \alpha_2$$

HER TROR JEG DET ER NOE FEIL..

6 Solving by Finite Element Methods

7 Final comments