FYS3150 - Project 5

Kandidatnummer

December 9, 2013

Introduction

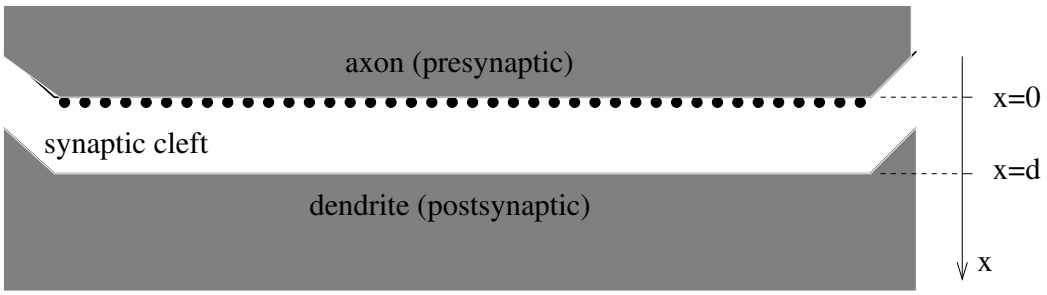
This project will be about solving the diffusion equation in both 1 and 2 spacial dimensions both analytically and numerically. This equation is widelly used in both thermodynamics, statistical physics and quantum mechanics to name a few. Therefore, I will use different schemes, and compare it with the analytical solutions to find the most efficient and stable solution. In this specific case, I simulate the concentration of neurotransmitters between two nerve cells, neurons. This concentration can be simulated by the diffusin equation.

1+1 dimensional problem

First I will look at the 1+1 dimensional problem. That means one spacial dimension, x, and one time dimension. In this simulation we set the amount of neurotransmitters equal to one at the presynaptic, $x = 0$. This will not change throughout the simulations, because the axon produces the neurotransmitters. The postsynaptic absorbs all the neurotransmitters, so the concentration will always be 0 at $x = 1$. The concentration throughout the cleft is given by the diffusion equation:

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

In our case, we set $D = 1$.



1+1 dimensional problem, analytical solution

We want the boundary conditions equal to 0 at both $x = 0$ and $x = 1$, so we define a new function

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

and solve the diffusion equation for this function. This equation can be solved by separation of variables. Setting $v(x,t) = X(x)T(t)$ gives:

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

The soltion for the time equation is

$$T(t) = e^{-\lambda^2 t}$$

while the spacial solution is

$$X(x) = Asin(\lambda x) + Bcos(\lambda x)$$

By the boundary conditions $v(0,t) = v(1,t) = 0$ we see that $B = 0$ and that $\lambda = n\pi$. So we get

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

The initial condition is given on the form $Ax+b$, so $u_s(x) = 1-x$ fullfills the boundary conditions. We have defined $v(x,t) = u(x,t) - u_s(x)$ so $u(x,t) = v(x,t) + u_s(x)$,

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

To find the C_ns, I solve the equation for time equal to 0. I know that $u(x,0) = 0$ for $0 < x < 1$, so

$$v(x,0) = \sum_{n=1}^{\infty} C_n sin(n\pi x) = x - 1$$

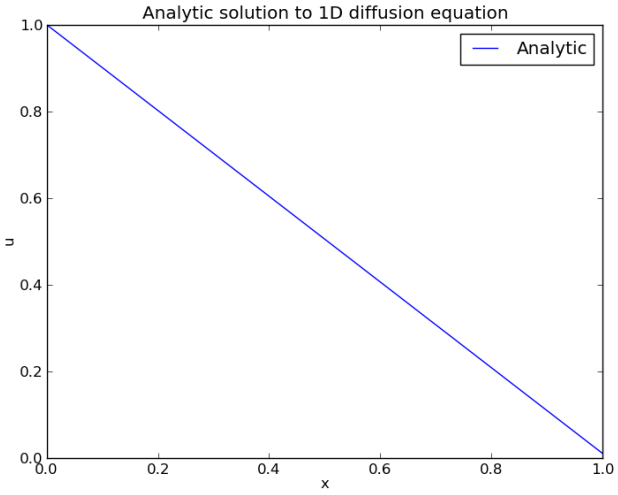
Using the fourier expansion we see that

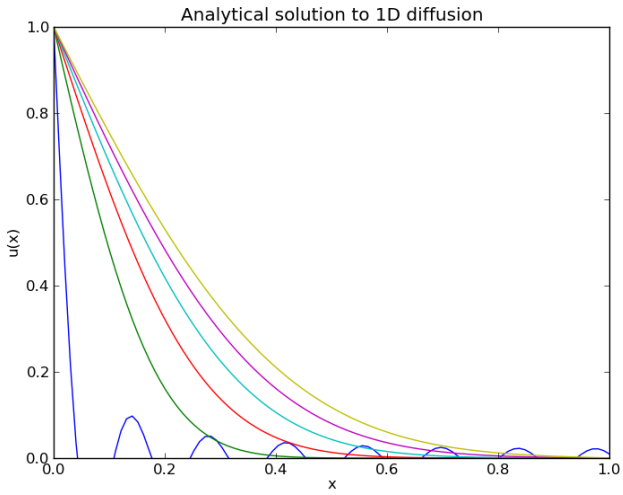
$$C_n = \frac{-2}{\pi n}$$

and we get the full solution

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

Which we see approaches the steady state solution after some time. Here I used 100 points along the x-axis and i let the sum go from $n = 1$ to $n = 15$. A movie showing how the solution develops in time is on github. The right plot is the first 6 analytic solutions. As time passes, the solution moves more and more towards the steady state solution.





1+1 dimensional, numerical solution

Markov Chains

In this project I will solve the 1+1 dimensional diffusion equation by Monte Carlo methods using Markov Chains. I will simulate the diffusion equation by programming the following random walk steps:

1. Set initial number of particles, N_0, that start at $x = 0$. This number will be conserved for $x = 0$ throughout the whole simulation.
2. Set up a vector, u, that contain positions of each individual particle. The positions will be split up into Nx different bins with width dx.
3. For each particle, draw a random number, r. If $r < 0.5$ then $pos_{new} = pos_{old} - l_0$ else $pos_{new} = pos_o ld + l_0$.
4. If a particle moves beyond $x = 1$, remove particle.
5. If a particle moves from $x = 0$, add a new at $x = 0$ to maintain N_0. In addition: If the move is negative, remove particle from vector.
6. Repeat 2-5 for all time steps until final time is reached.

I will implement two different algorithms. The first algorithm will have constant steplength

$$l_0 = \sqrt{2\Delta t}$$

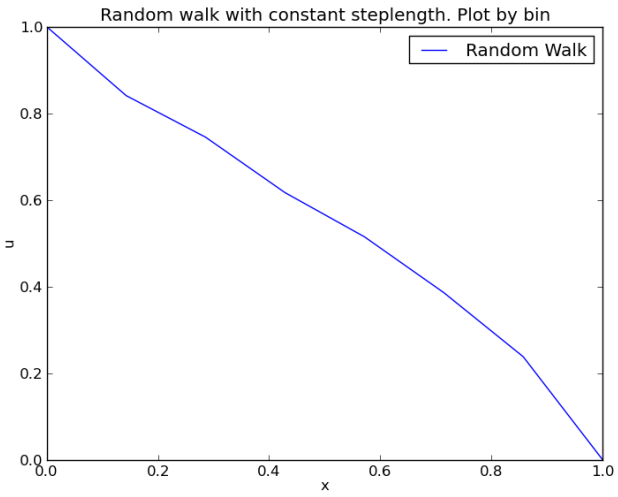
and the other will have

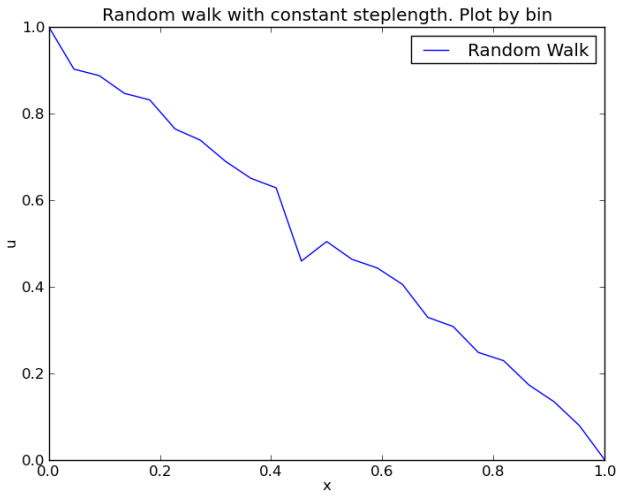
$$l_0 = \sqrt{2\Delta t}\xi$$

Where ξ is a random number generated by a Gaussian distribution with mean value 0 and standard deviation $\sigma = \frac{1}{\sqrt{2}}$.

When I'm plotting the random walk, I see if the particle is inside the bin $x + dx$. If it is, I count it as 1, and I add it to the solution vector. This do not give very smooth curves for the constant steplength part unless one makes the right amount of bins. Since the steplength is constant, there will only be a few points that the particle may be in. By using a wrong amount of bins, the solution will be wrong. The steplength is $\sqrt{2dt}$, so there will be $\frac{1}{\sqrt{2dt}}$ different positions that are legal. For $dt = 0.01$, there will be 7.07 legal positions, so 8 bins will include all positions the particles might have.

The program did not allow $dt = 0.0001$ or smaller probably due to too many elements in array. For $dt = 0.001$, the amount of bins are 23, and this gives the following results below: First the case for $dt = 0.01$, then the case for $dt = 0.001$.





In both cases, the amount of particles at position 0 are 1000. Exactly as I wanted it to be. Although the solutions are not very smooth, they respect the boundary conditions perfectly.

When looking at the Markov chain with randomly generated steplength, I must be a little careful with the boundary condition at $x = 0$. I must already here decide the bin size, and make all particles within the first bin be equal to 1000. Not only particles with value 0. I want to make 20 bins, So all particles within $0 + \frac{1}{20}$ are counted as particle at boundary, and this amount cannot exceed 1000.

I will need to map the uniform random distribution to the normal distribution. The normal distribution looks like

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

In our case $\mu = 0$ and $\sigma = \frac{1}{\sqrt{2}}$ so we get:

$$f(x) = \frac{1}{\sqrt{\pi}} e^{-x^2}$$

So we have our new probability distribution

$$\frac{1}{\sqrt{\pi}} e^{-y^2} dy = dx$$

but we cannot perform the following integral analytically.

$$\frac{2}{\sqrt{\pi}} \int_0^{\infty} e^{-y^2} dy = x$$

To get around this, I look at a distribution dependent on two uniform generated numbers.

$$f(y, z) = \frac{1}{\sqrt{\pi}} e^{-(y^2+z^2)}$$

Changing to polar coordinates with

$$r = \sqrt{y^2 + z^2} \quad \theta = tan^{-1} \frac{x}{y}$$

giving

$$f(r,\theta) = \frac{1}{\sqrt{\pi}} r e^{-r^2} dr d\theta$$

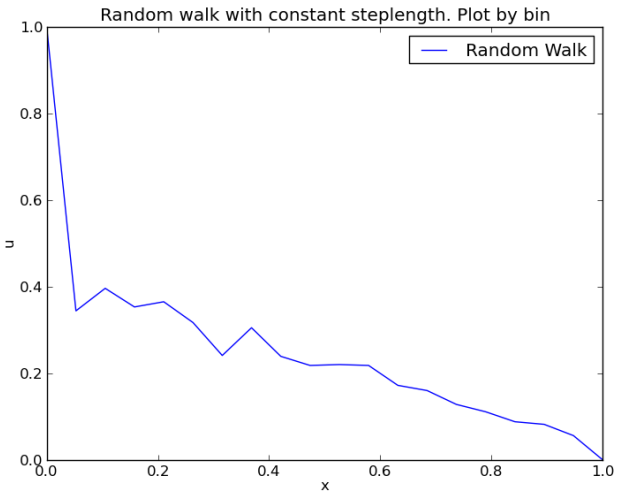
θ can be gotten by multiplying a random number by 2π. To find r, we must change variables and use that for exponential distribution, the mapping will look like $y(x) = -ln(1-x)$ Set $u = r^2$ and defining a new PDF: $e^{-u}du$. Now we see that

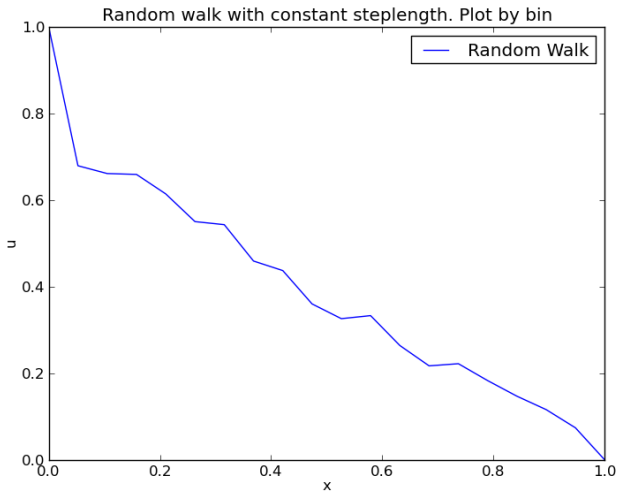
$$y = rcos(\theta) = \sqrt{u}cos(\theta) = \sqrt{-ln(1-y')}cos(\theta)$$

$$z = rsin(\theta) = \sqrt{u}sin(\theta) = \sqrt{-ln(1-z')}sin(\theta)$$

Where y' and z' are random numbers from uniform distribution.

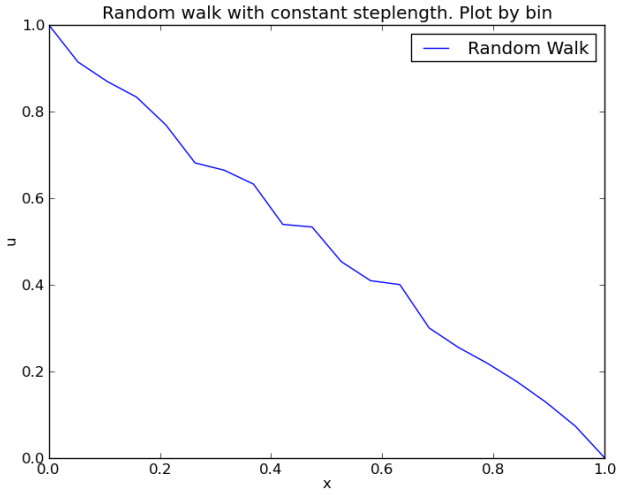
This gives the following plots. First for $dt = 0.01$ and then for $dt = 0.001$





which actually gives worse results than the constant steplength.

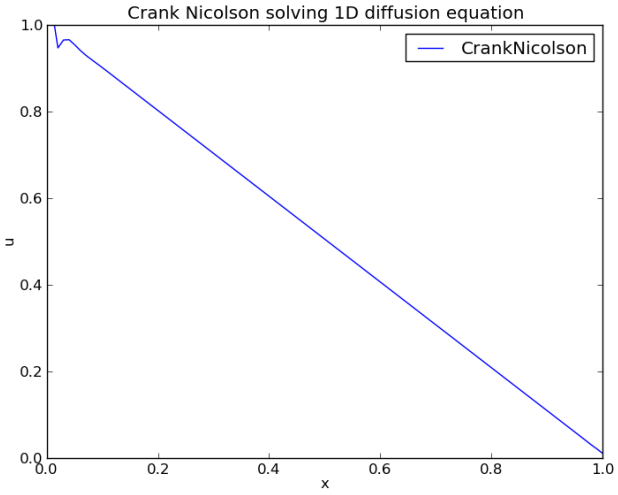
By reducing dt even further I get the following results. Left $dt = 0.0001$ and right $dt = 0.00001$:

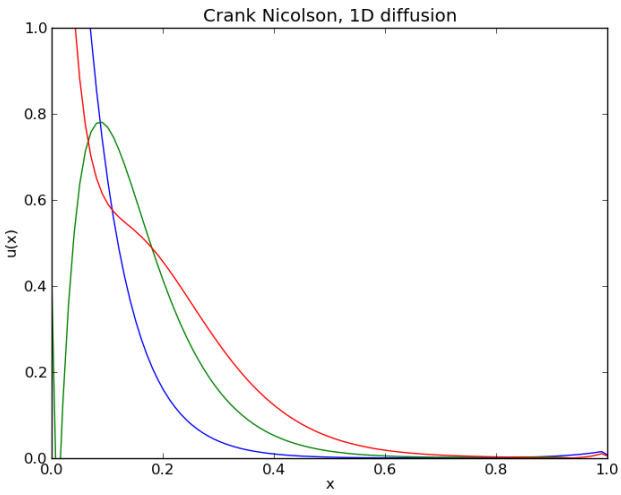


And this is starting to look more like the analytic / Crank Nicolson solution, even though it is still not too smooth. For $dt = 0.00001$, the computer needed 177 seconds, so this is starting to demand resources.

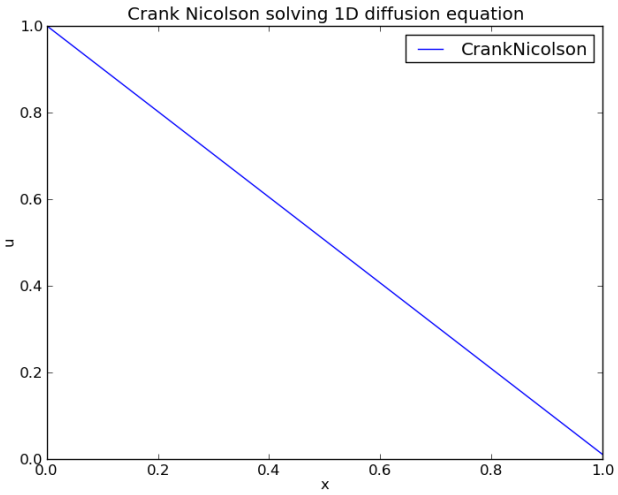
Crank Nicolson

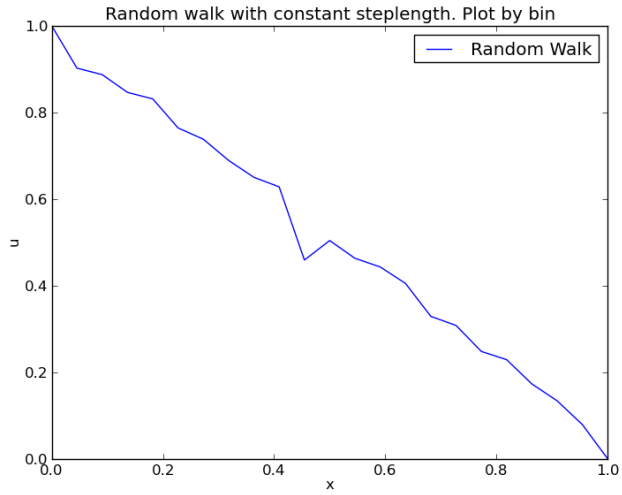
I solve the equation numerically using Crank Nicolson to compare with the Markov Chains. By setting $dx = 0.01\hat{A}$ and $dt = 0.01$ (the same values as for Markov Chain with constant steplength), I get the following results. The right plot shows the first 3 timesteps, one get a feeling how it evolvs in time. (blue: t0, red t1, green t2)





Decreasing dt by a factor 10^3 for Crank Nicolson and by a factor 10 for the random walk, one can see the difference in efficiency for these two methods. The random walk method used 7 seconds, while the Crank Nicolson scheme used 6 seconds. Which is clearly in favour of Crank Nicolson since dt was reduced much more. For these values, Crank Nicolson gives very high accuracy, while the Random walk method do not show significant improvements. Plots below:





Analytical solution of 2+1 dimensional problem

We have seen before that the solution of a 1+1 dimensional problem is

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

To find the solution for the 2+1 dimensional problem, I will use separation of variables for both $X(x)$, $Y(y)$ and $T(t)$. I need boundary conditions for X and Y and initial condition for T. The boundary conditions for X are given as

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

The initial condition is given as

$$u(x,y,0) = 0, \;\; 0 < x < 1$$

The boundary conditions for the Y-direction is not given, so I have to set them for myself. Since the synaptic cleft is finite and we only want the ions inside the cleft, the concentration of ions must be reduced to 0 in the x-direction, so a guess for boundary conditions might be

$$u(0,x,t) = u(1,x,t) = 0$$

The x-direction is along the synapses, and the y-direction is perpendicular to the synapses. So the density is highest at $y = 0$ and will be reduced as y increases.

Finding steady state solution

I need the steady state solution for my problem, and that is given by solving the Laplace equation:

$$\nabla^2 u_s = 0$$

by separation of variables that reduces to:

$$\frac{1}{X(x)}\frac{d^2X(x)}{dx^2} = -\lambda^2 \;\; \frac{1}{Y(y)}\frac{d^2Y(y)}{dy^2} = \lambda^2$$

And I get the solutions:

$$X(x) = \alpha cos(\lambda x) + \beta sin(\lambda x) \;\; Y(y) = \gamma cosh(\lambda y) + \delta sinh(\lambda y)$$

Need to check boundary conditions to solve the unknown variables.

$$u_s(0,y) = 0 = \alpha(\gamma cosh(\lambda y) + \delta sinh(\lambda y)) \tag{1}$$

$$u_s(x,0) = 1 = \gamma(\alpha cos(\lambda x) + \beta sin(\lambda x)) \tag{2}$$

So we see that for these equations to hold, (without resorting to the non-trivial solution), $\alpha = 0$ and therefore

$$\gamma\beta sin(\lambda x) = 1$$

$$u_s(1,y) = 0 = \beta cos(\lambda)[\gamma cosh(\lambda y) + \delta sinh(\lambda y)] \tag{3}$$

Since $\beta = 0$ leads to a nontrivial solution, we must set $cos(\lambda) = 0$. That leads to

$$\lambda = n\pi \quad n = 1, 2, 3, ...$$

The last equation

$$u_s(x,1) = 0 = \beta sin(n\pi x)[\gamma cosh(n\pi) + \delta sinh(n\pi)] \tag{4}$$

leads to

$$\gamma cosh(n\pi) = -\delta sinh(n\pi)$$

$$\delta = -\gamma coth(n\pi)$$

To find out the solution for (2):

$$1 = \gamma\beta sin(n\pi x)$$

I must use Fourier expansion theory:

$$1 = \sum_{n=1}^{\infty} \gamma_n sin(n\pi x) \quad \gamma_n = 2\int_0^1 sin(n\pi x)dx$$

$$\gamma_n = \frac{2}{n\pi}(1 - cos(n\pi))$$

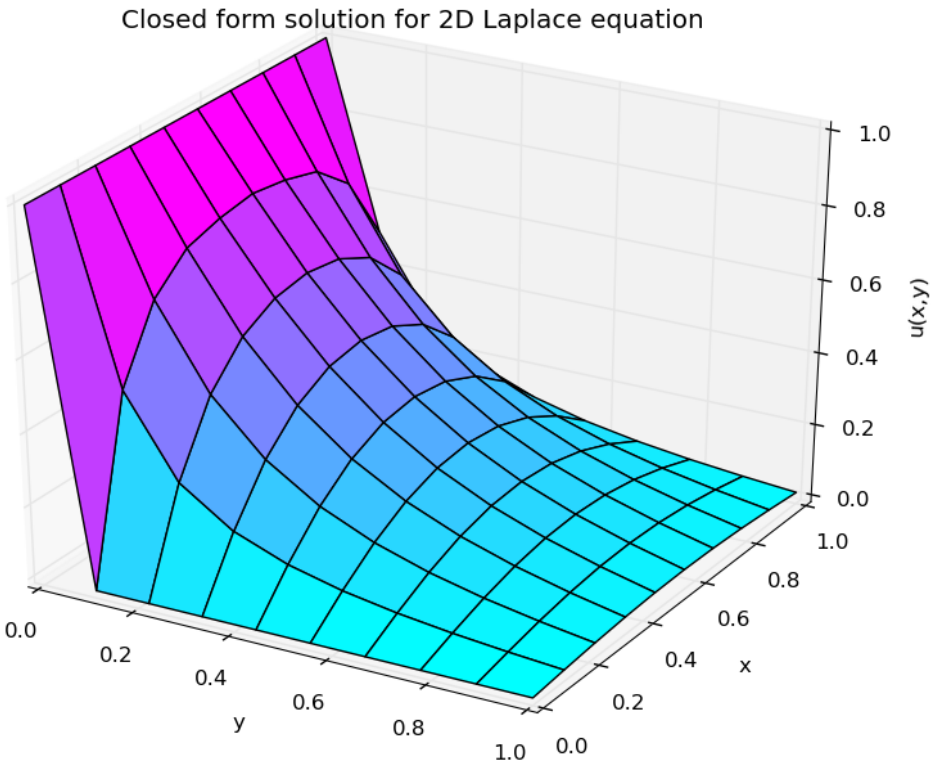
Now I can put everything back together, and I find that

$$u_s(x,y) = \beta sin(n\pi x)[\gamma cosh(n\pi y) - \gamma coth(n\pi)sinh(n\pi y)]$$

Leading to the final solution

$$u_s(x,y) = \sum_{n=1}^{\infty} \frac{2}{n\pi}(1 - cos(n\pi))[cosh(n\pi y) - coth(n\pi)sinh(n\pi y)] \tag{5}$$

By plotting the steady state solution, I get the same results as for the explicit and implicit methods.



which is not weird, since the Laplace equation is the solution that the diffusion equation will approach, but one loses the possibility of seeing the development through time.

2+1 dimensional problem. Numerical solution

Explicit method

I will use two different schemes to simulate the 2-dimensional diffusion equation. First I will look at the explicit scheme and its stability.

The algorithm is pretty straight-forward

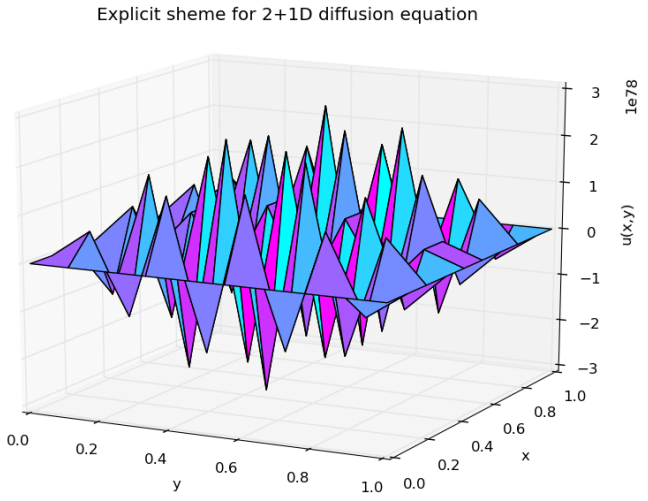
$$u_{i,j}^{l+1} = u_{i,j}^{l} + \alpha[u_{i+1,j}^{l} + u_{i-1,j}^{l} + u_{i,j+1}^{l} + u_{i,j-1}^{l} - 4u_{i,j}^{l}]$$

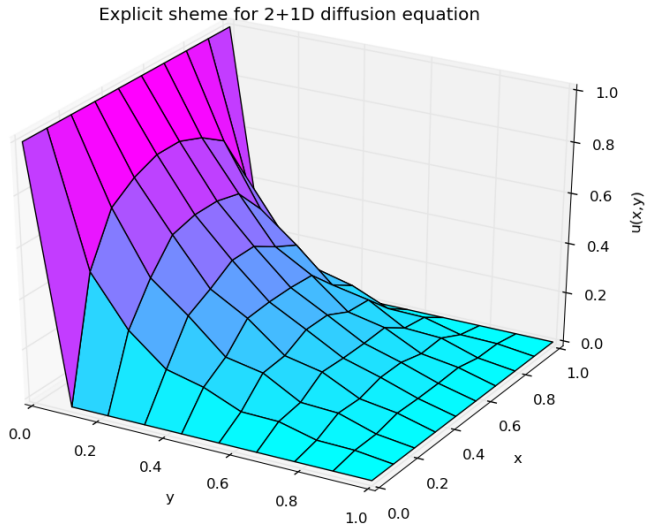
which is derived by applying the numerical approximation for the double derivative and reorganizing the equation. $\alpha = \frac{\Delta t}{h^2}$ The scheme is explicit because it generates the value at the next time step based on the previous time step.

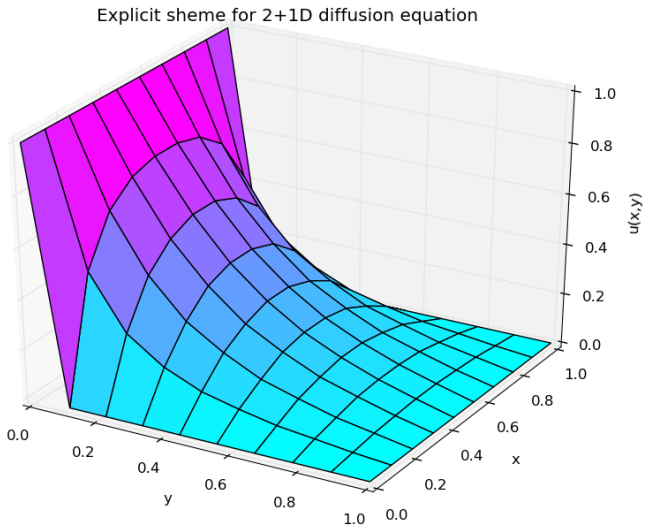
For the 1 dimensional explicit scheme, we had problems with stability if $\Delta t > \frac{1}{2}\Delta x^2$ For the two-dimensional case, I found that at $\Delta x = 0.1$ and $\Delta t = 1/370 = 0.0027$, the explicit scheme stabalized. This is barely more than $\frac{1}{4}\Delta x^2$, so based on this, I say that

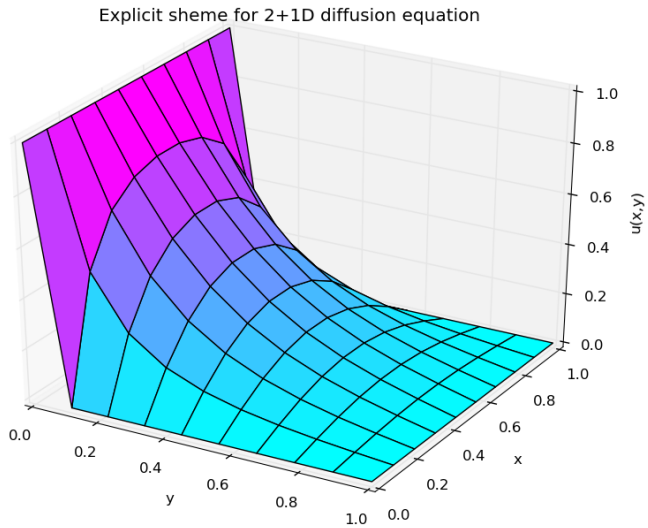
$$\Delta t \geq \frac{1}{4}\Delta x^2$$

If we want a stable solution. Following are four plots of the explicit solution:









Top left: $dt = \frac{1}{100}$, top right: $dt = \frac{1}{369}$, lower left: $dt = \frac{1}{370}$, lower right $dt = \frac{1}{1000}$

The results I get from the explicit scheme is very close to the analytic steady state solution.

Implicit method

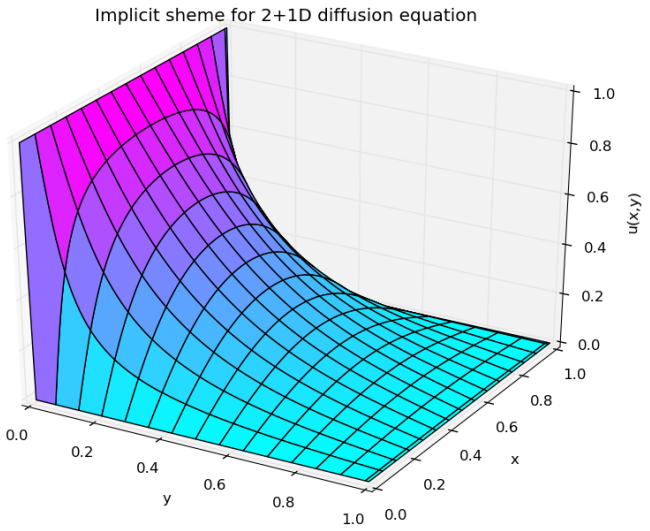
The implicit method is derived almost identically as the explicit scheme, but one uses the backward going Euler formula instead of the forward going formula for the time derivative on the diffusion equation. Reorganizing the equation, I end up with

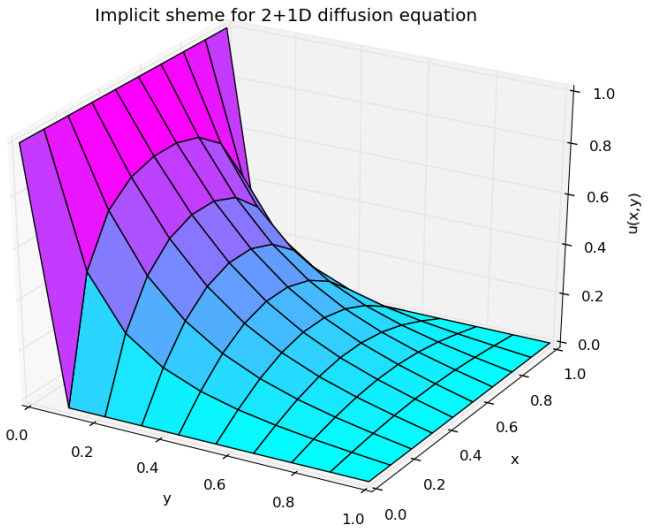
$$u_{i,j}^{l} + 4\alpha u_{i,j}^{l} - \alpha[u_{i+1,j}^{l} + u_{i-1,j}^{l} + u_{i,j+1}^{l} + u_{i,j-1}^{l}] = u_{i,j}^{l-1}$$

Which is an equation that has many unknown values at time l that are decided from a single point at an earlier time, $l-1$.

The way I solve this equation is by using Jacobi's method. I guess an initial u, then computing a new solution u_{new} by the given equation, and then I see if my new solution u_{new} is close enough to my initial guess u.

This scheme has no stability issues like the explicit scheme has. The error is the same as for the explicit scheme, since the backward and the forward Euler formula has the same error, but since I do not have to worry about the stability, I can increase the spacial resolution quite a bit without my results going bananas!





First plot has a $\Delta x = 0.01$ and $\Delta t = 0.01$. This made the explicit scheme unstable, but I could with no problem get a higher spacial resolution here without decreasing the time step. Second plot has $\Delta x = 0.1$ and $\Delta t = 0.001$. This looks exactly equal to the explicit solution, which would be natural. It also looks very close to the closed form solution for the steady state.