

# FYS3150 - Project 4

Alfred Alocias Mariadason

10.11.2014

## Introduction

The aim of this project is mainly to solve the one-dimensional diffusion equation, that is, find a closed-form solution and solve it numerically with implicit numerical methods for partial differential equations.

The methods used are the Explicit Forward Euler, Implicit Backward Euler and Implicit Crank-Nicolson Scheme. All the methods are tested up against each other and the truncation errors are also documented.

Link to programs: <https://github.com/0o1Insane1o0/project4>

## Abstract

The reason for solving the diffusion equation is because we want to model the dominant way of transporting signals between neurons in the brain. This is achieved by diffusion of signal molecules called neurotransmitters across the synaptic cleft which separates the cell membranes of two cells.

## Theory

As mentioned the process of transporting the signal is governed by diffusion. Mathematically we can model this by:

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

Here  $u$  is the concentration of the particular neurotransmitter (the mentioned molecule).  $D$  is the diffusion coefficient determined by the solvent in the synaptic cleft, so  $D$  is a coefficient which changes by environment.

To make this problem easier we can make some assumptions.

- 1.) The neurotransmitters are released roughly equally.
- 2.) The synaptic cleft is roughly equally wide across its whole.

With these assumptions we can assume that the neurotransmitter concentration only varies in one direction, from presynaptic to postsynaptic. We can choose this to be our  $x$ -direction and the diffusion equation reduces to:

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

Now, in a real diffusion process the neurotransmitters will occasionally bump into the presynaptic cleft and be absorbed(temporally) by the receptor at the postsynaptic cleft. This imposes following boundary and initial conditions:

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

With this the solution to the process is given by:

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

The initial conditions are given by:

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

And the boundary conditions are:

$$\begin{aligned} u(0, t) &= 1, \quad t > 0 \\ u(d, t) &= 0, \quad t > 0 \end{aligned}$$

## Closed-form Solution

For the closed-form solution we can start by looking at the differential equation itself:

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

The steady-state solution is given, by the above boundary conditions, as a linear expression:

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

With this we can make a new function  $v(x, t)$  and solve for  $v$  rather than  $u$ :

$$v(x, t) = u(x, t) - u_s(x) = u(x, t) + x - 1$$

The boundary conditions for  $v$  is:

$$v(0) = v(d) = 0$$

Assuming a form of seperable variable( $v(x, t) = F(x)G(t)$ ) we have that:

$$F(x) = A \cos kx + B \sin kx, \quad G(t) = C e^{-k^2 t}$$

The coefficient A, B and C can be found by the boundary conditions.

$$v(x, t) = 0 \Rightarrow (A \cos kx + B \sin kx)C = 0$$

We observe that  $A = 0$ ,  $C = 1$ , and  $k = \frac{n\pi}{d}$  fulfills the boundary conditions, however B now has  $n$  possibilities and we have:

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

The initial conditions for  $v$  gives:

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

To solve this we can use theory on Fourier series and obtain  $B_n$ .  $B_n$  is then given as the coefficient of the function  $(x - 1)$ :

$$\begin{aligned} B_n &= \frac{2}{d} \int_0^d (x - 1) \sin\left(\frac{n\pi}{d}x\right) dx \\ &= \frac{2}{d} \int_0^d \sin\left(\frac{n\pi}{d}x\right) dx + \frac{2}{d} \int_0^d -\sin\left(\frac{n\pi}{d}x\right) dx \end{aligned}$$

Integration by parts gives:

$$\begin{aligned} B_n &= \frac{2}{d} \left[ -x \frac{d}{n\pi} \cos\left(\frac{n\pi}{d}x\right) \right. \\ &\quad \left. + \int \frac{d}{n\pi} \cos\left(\frac{n\pi}{d}x\right) dx \right]_0^d + \frac{2}{d} \left[ \frac{d}{n\pi} \cos\left(\frac{n\pi}{d}x\right) \right]_0^d \\ &= -\frac{2d}{n\pi} \cos(n\pi) + \frac{2}{d} \sin(n\pi) + \frac{2}{n\pi} \\ &= -\frac{2d}{n\pi} \cos(n\pi) + \frac{2}{d} \sin(n\pi) + \frac{2}{n\pi} \cos(n\pi) - \frac{2}{n\pi} \end{aligned}$$

Since  $n = 1, 2, \dots$ , we have that  $\sin(n\pi) = 0$  and  $\cos(n\pi) = (-1)^n$ .

$$\begin{aligned} B_n &= -\frac{2d}{n\pi}(-1)^n + \frac{2}{n\pi}(-1)^n - \frac{2}{n\pi} \\ B_n &= \frac{2}{n\pi}((-1)^n(1 - d) - 1) \end{aligned}$$

For our case  $d = 1$ , which gives:

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

So the closed-form solution is:

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

And for  $d = 1$  we have:

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

## Explicit Forward Euler

For the explicit method we approximate the function slope with a forward formula:

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

And discretized we have:

$$u_t = \frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{\Delta t} = \frac{u_{i,j+1} - u_{i,j}}{\Delta t}$$

$$u_{xx} = \frac{u(x_{i+1}, t_j) - 2u(x_i, t_j) + u(x_{i-1}, t_j))}{\Delta x^2} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2}$$

So the diffusion equation becomes:

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

Further we can assign  $\frac{\Delta t}{\Delta x^2}$  to a variable, for instance  $\alpha = \frac{\Delta t}{\Delta x^2}$ . This basically rearranges the equation such that the advancement of the time-step is on the left side giving the final discretized expression:

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

Note that we can express this on matrix form with a tridiagonal matrix with the factors  $\alpha$  alongs its upper and lower diagonal and  $1 - 2\alpha$  along its diagonal. A linear matrix expression is much more convenient when we study the stability of the scheme as predicated below.

## Implicit Backward Euler

For the implicit scheme we use the backward formula (rather than the forward formula used above) to arrive at our solution.

$$\frac{\partial u}{\partial t} = \frac{u_{i,j} - u_{i,j-1}}{\Delta t}$$

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

Using the same  $\alpha$  as above ( $\alpha = \frac{\Delta t}{\Delta x^2}$ ) we arrive at a discretized scheme with the function value at previous time-step as the only unknown:

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

We observe that this can be expressed in form of a linear system like this:

$$V_j = \begin{bmatrix} u_{1,j} \\ \vdots \\ u_{n,j} \end{bmatrix} \Rightarrow V_j = (I + \alpha A)^{-1} V_{j-1}$$

Where:

$$A = \begin{bmatrix} 2 & -1 & & & & 0 \\ -1 & 2 & -1 & & & \\ & -1 & 2 & & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & \\ 0 & & & & 2 & -1 \\ & & & & -1 & 2 & -1 \\ & & & & & -1 & 2 \end{bmatrix}$$

And now we can just use a tridiagonal solver(like the one from project 1) to integrate the equation through time and obtain the appoximated solution.

## Implicit Crank-Nicolson Scheme

The Crank-Nicolson Scheme is a combination of the two above schemes, the algorithm is as follows:

$$\begin{aligned}\frac{\partial u}{\partial t} &= \frac{u_{i,j+1} - u_{i,j}}{\Delta t} \\ \frac{\partial^2 u}{\partial x^2} &= \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{2\Delta x^2} + \frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{2\Delta x^2}\end{aligned}$$

Note that this is a time-centered scheme with  $t + \Delta t/2$  as center.

Again we define the same  $\alpha = \frac{\Delta t}{\Delta x^2}$  as previously and rearrange the equation. The discretized equation yields:

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

This can also be expressed in matrix form as:

$$(2I + \alpha A)V_j = (2I - \alpha A)V_{j-1}$$

Vector  $V_j$  is the same as in the implicit scheme and matrix  $A$  is the same as above. We can rewrite this as follows:

$$V_j = (2I + \alpha A)^{-1}(2I - \alpha A)V_{j-1}$$

## Truncation errors:

The truncation error of both the implicit and explicit schemes can be expressed with the remainder in a Taylor expansion(or Taylor polynomial). It is simple to convince ourselves that the truncation error of Backward Euler is the same as Forward Euler so only the Forward scheme will be expressed here. Now we Taylor expand the three parts of the explicit scheme and get:

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

Which gives:

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

Summing this gives that the truncation error for both the Forward and Backward scheme the truncation error is:

$$\mathcal{O}(\Delta t, \Delta x^2)$$

For the Crank-Nicolson scheme we proceed in the same way as above, but with the expansion centered at  $t + \Delta t/2$ . Using the expansions above and solving the expansions for  $u(x + \Delta x, t + \Delta t)$  and  $u(x - \Delta x, t + \Delta t)$  at  $t + \Delta t/2$  we get:

$$\begin{aligned} \left[ \frac{\partial u(x, t + \Delta t/2)}{\partial t} \right]_{approx} &= \frac{\partial u(x, t + \Delta t/2)}{\partial t} + \mathcal{O}(\Delta t^2) \\ \left[ \frac{\partial^2 u(x, t + \Delta t/2)}{\partial x^2} \right]_{approx} &= \frac{\partial^2 u(x, t + \Delta t/2)}{\partial x^2} + \mathcal{O}(\Delta x^2) \end{aligned}$$

Which gives a summed truncation error of:

$$\mathcal{O}(\Delta t^2, \Delta x^2)$$

## Stability Conditions

The stability condition for the explicit scheme is given by:

$$\frac{\Delta t}{\Delta x^2} \leq \frac{1}{2}$$

This basically means that a low  $\Delta x$  gives a very low  $\Delta t$  which will obviously give unwanted consequences if the chosen time-intervall is large. We can study this scheme by requiring that the solution approaches a definite value as we increase our time-step. This means that by theory on iterative schemes we require the spectral radius of our matrix to satisfy the following condition:

$$\rho(A) < 1$$

The spectral radius is essentially interpreted as the smallest number such that a circle with radius centered at origin of the complex plane contains all eigenvalues of  $A$ . Note that if  $A$  is positive-definite the above condition is always satisfy.

If we want to we can solve the eigenvalues of  $A$  and use those to find the conditions defined by the spectral radius. In order to do this we can start of by expressing  $A$  in terms of a matrix  $B$  defined by the coefficients of the element of  $A$  we get that  $A$  can be expressed as:

$$A = I - \alpha B$$

The eigenvalues of  $A$  is then given as:

$$\lambda_i = 1 - \alpha \mu_i$$

Where  $\mu_i$  is the eigenvalues of  $B$ . Using the fact that the elements of  $B$  can be expressed as:

$$b_{ij} = 2\delta_{ij} - \delta_{i+1j} - \delta_{i-1j}$$

We can now find the eigenvalues of  $B$  with the set of eigenequations:

$$(Bx)_i = \mu_i x_i$$

If we now assume that  $x$  can be expanded in a basis of  $x = \sin(n\theta)$  with  $\theta = l\pi/n + 1$  and the endpoints are zero. The set can then be rewritten as:

$$2 \sin i\theta - \sin((i+1)\theta) - \sin((i-1)\theta) = \mu_i \sin(i\theta)$$

$$2(1 - \cos(\theta))x_i = \mu_i x_i$$

$$\mu_i = 2(1 - \cos(\theta))$$

With the required condition defined by the spectral radius above we have that  $\alpha$  must have the following conditions:

$$-1 < 1 - 2\alpha(1 - \cos(\theta)) < 1$$

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

Which results in:

$$\alpha \leq \frac{1}{2} \Rightarrow \frac{\Delta t}{\Delta x^2} \leq \frac{1}{2}$$

As mentioned above, if the matrix at hand is positive-definite the condition given by the spectral radius,  $\rho(A) < 1$  is always satisfied. This basically means that all the eigenvalues are positive. It is easy to convince ourselves that this is true for the Implicit Backward scheme since  $A = I + \alpha B$  gives eigenvalues  $\lambda_i = 1 + 2\alpha(1 - \cos(\theta)) > 1$  and since we use the inverse we have that  $\rho(A^{-1}) < 1$  and the method is always stable.

The same argument of stability goes for the Implicit Crank-Nicolson scheme. The essential part is that we get:

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

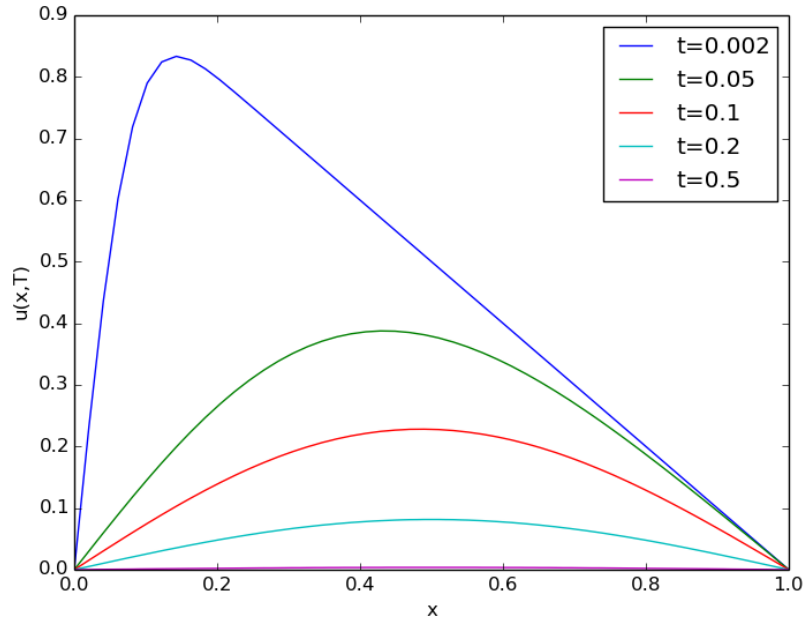
From:

$$\left| (2 + \alpha\mu_i)^{-1}(2 - \alpha\mu_i) \right| < 1$$

## Results

Using the programs mentioned in the introduction section above the results of the implementations of these three schemes are as follows:

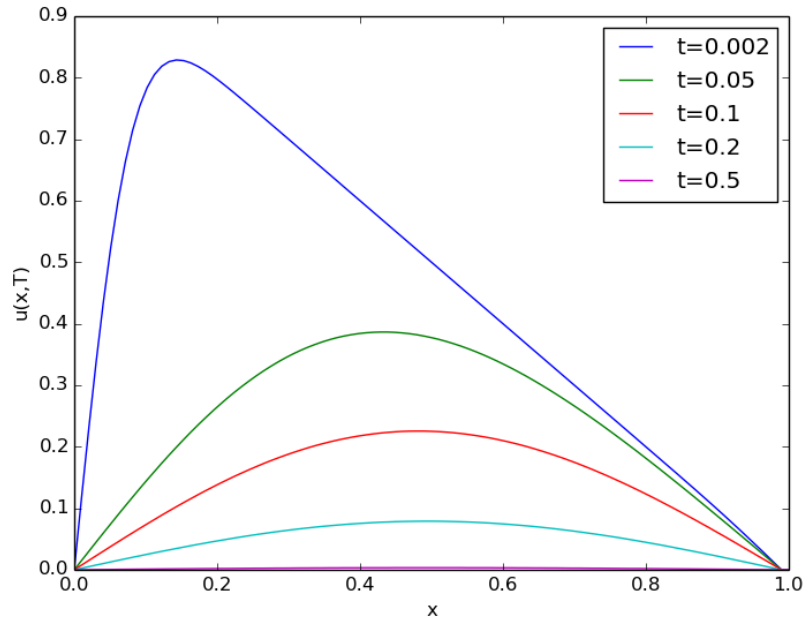
### Explicit Forward Euler Scheme at different times



$dx=0.01, dt=0.00005$

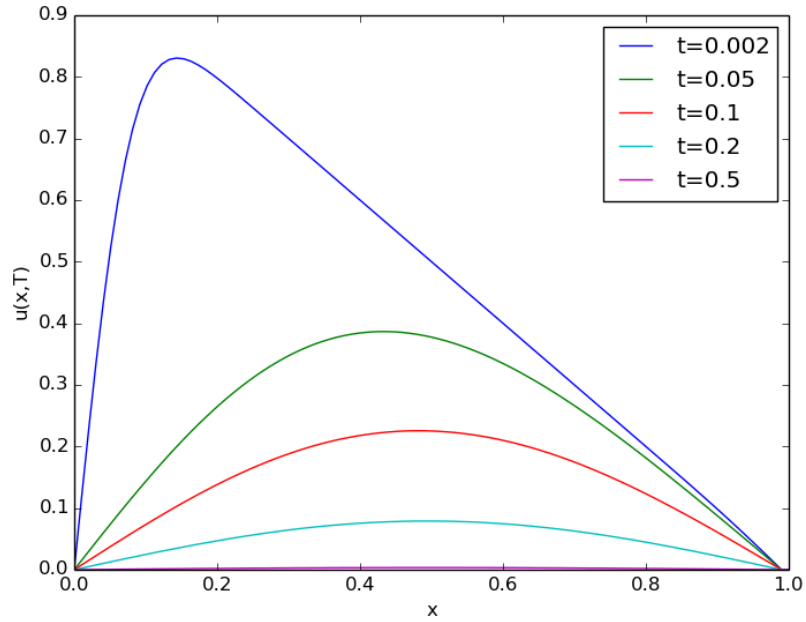


### Implicit Backward Euler Scheme at different times



$dx=0.01, dt=0.00005$

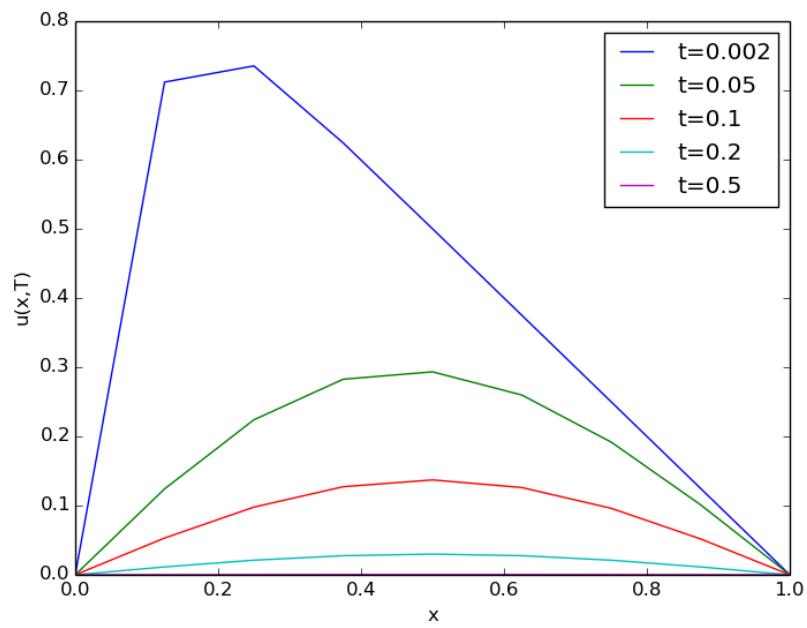
### Implicit Crank-Nicolson Scheme at different times



$dx=0.01, dt=0.00005$

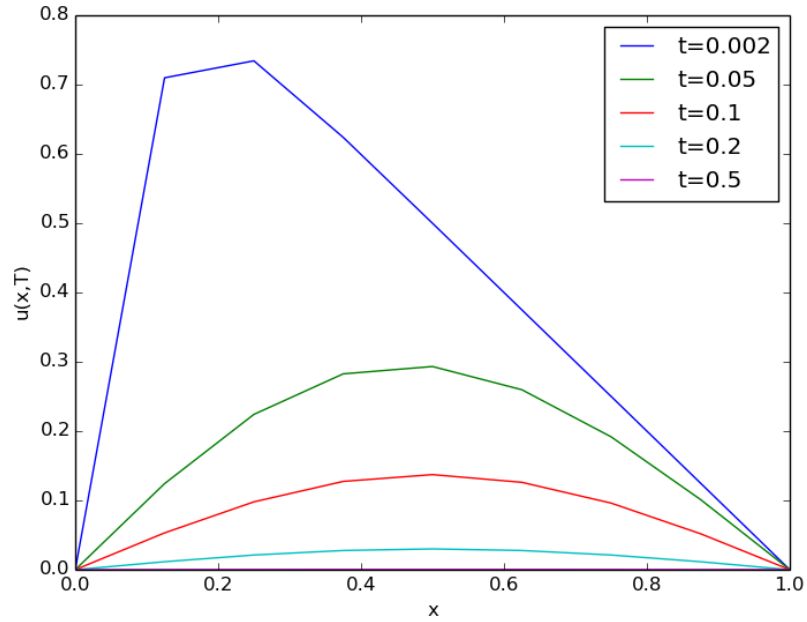
In order to test the schemes we run the simulations with a  $dx = 0.1$ .

### Error-estimate Explicit Forward Euler Scheme



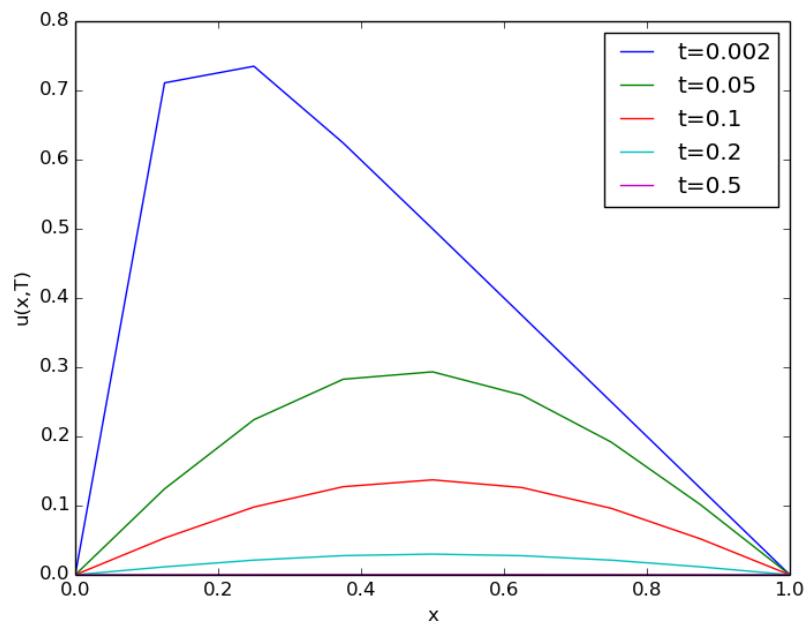
$dx=0.1, dt=0.00005$

### Error-estimate Implicit Backward Euler Scheme



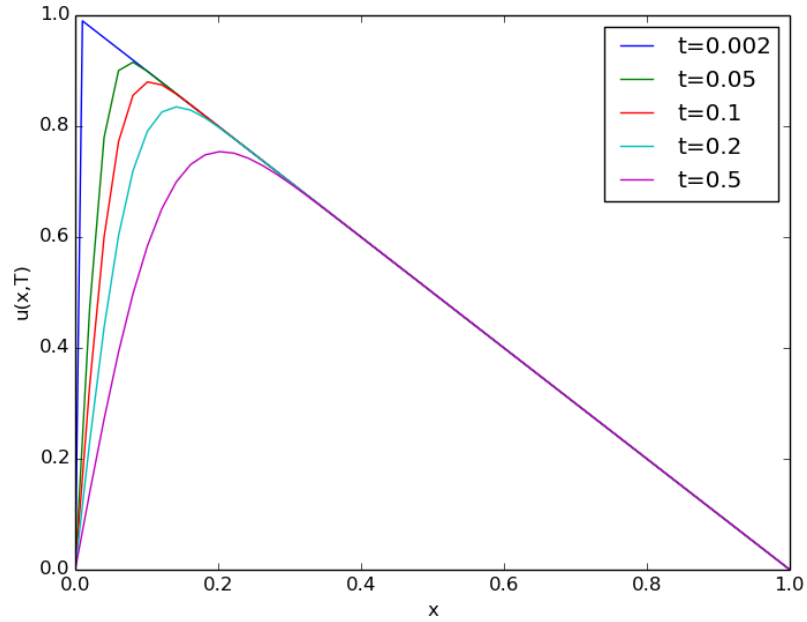
$dx=0.1, dt=0.00005$

### Error-estimate Implicit Crank-Nicolson Scheme



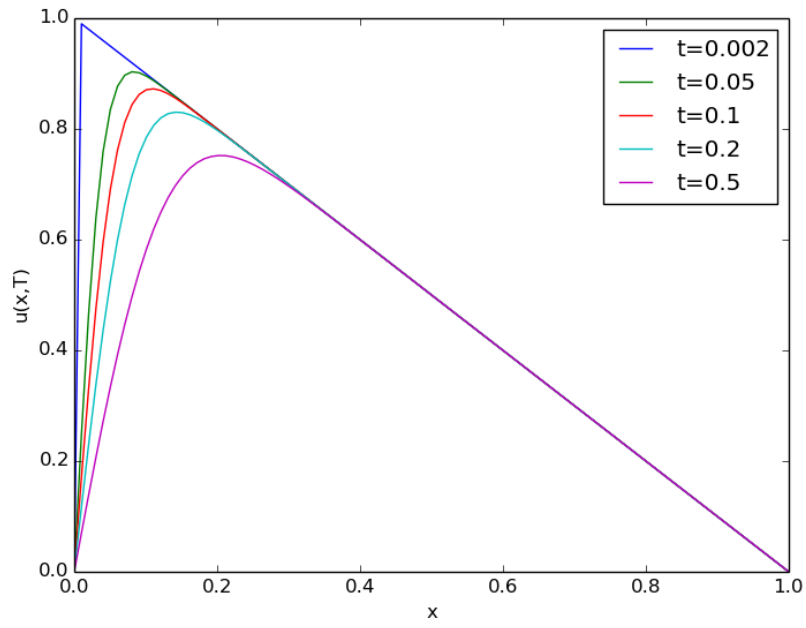
$dx=0.1, dt=0.00005$

### Stability test Explicit Forward Euler Scheme



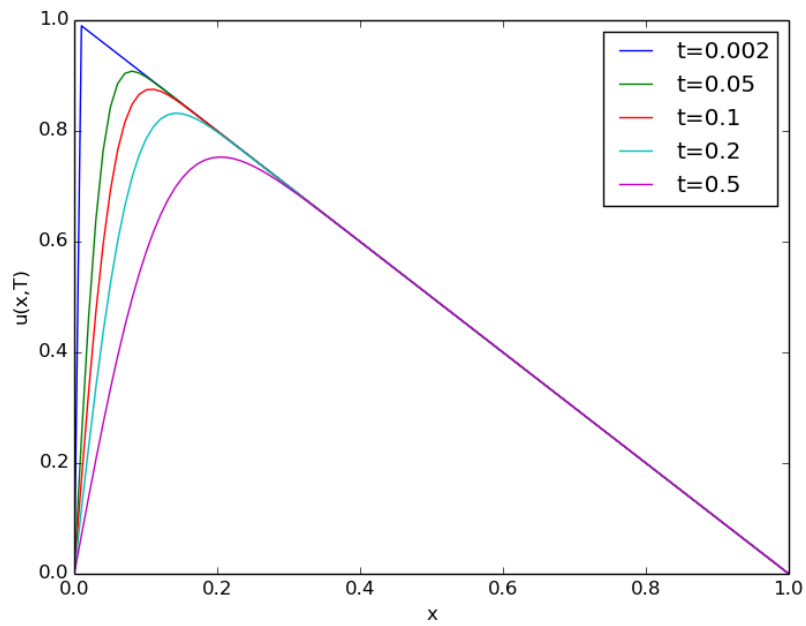
$dx=0.1, dt=0.005$

### Stability test Implicit Backward Euler Scheme



$dx=0.1, dt=0.005$

### Stability test Implicit Crank-Nicolson Scheme

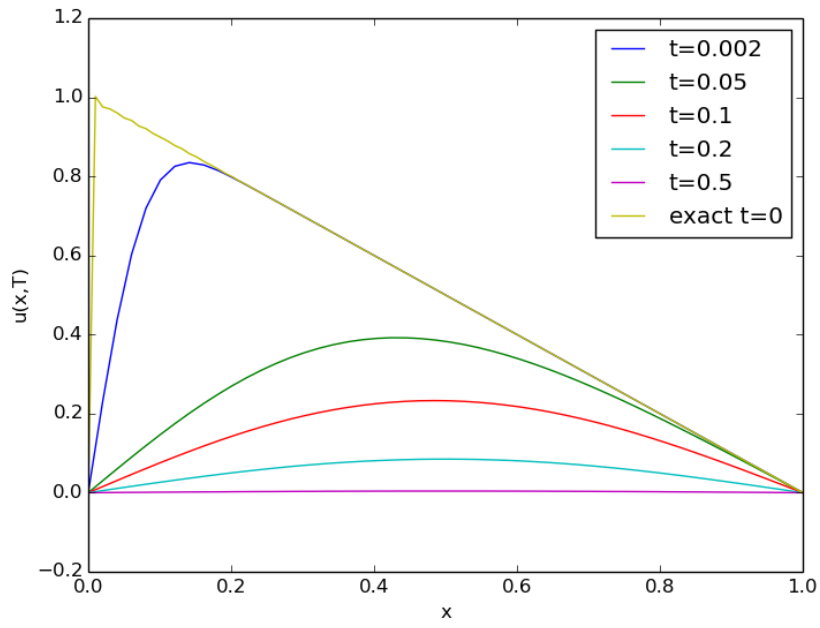


$dx=0.1, dt=0.005$

As we can see the forward scheme is a bit unstable when the stability condition is at its limit (see stability section above).

Here are some comparison plots with the closed-form result:

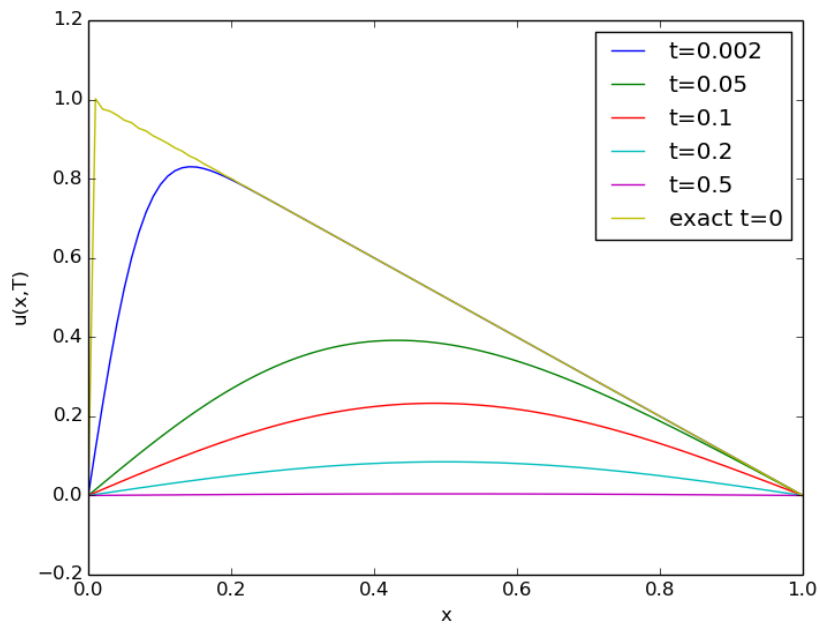
### Explicit Forward Euler



$dx=0.01, dt=0.005$

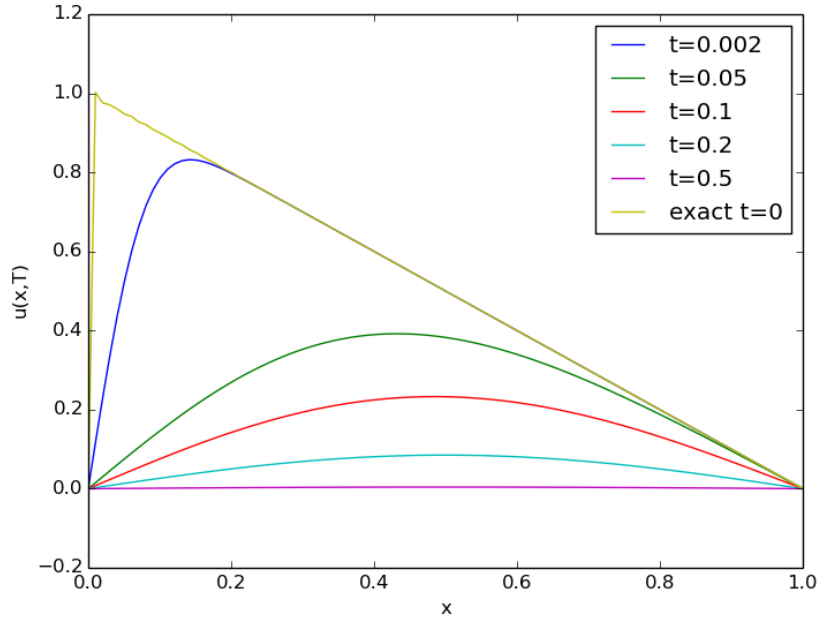
G

### Implicit Backward Euler



$dx=0.01, dt=0.005$

### Implicit Crank-Nicolson



$dx=0.01, dt=0.005$

Considering the stability properties and the truncation errors found in respective sections above and the resulting plots as seen above, one would classify the Implicit Backward scheme as the best. However for the problem at hand there aren't any significant problems with the Crank-Nicolson scheme.