Project 5 - Partial Differencial Equation

Solveig Andrea Devold Fjeld and Sarah Rastad

December 8, 2016

Abstract

1 Introduction

2 Theory

VET IKKE HVOR JEG SKAL SETTE LIGNINGEN

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

2.1 Equation

In this project we are solving the partial differencial equation:

$$\frac{\partial^2 u(x,t)}{\partial x^2} = \frac{\partial u(x,t)}{\partial t}, t > 0, x \in [0,1] \tag{2}$$

which can also be written

$$u_{xx} = u_t \tag{3}$$

This partial differencial equation can be seen as the temperature gradient in a rod of lenght L. This equation can be seen as being dimensionless since there are no constant multiplied to the equation and x goes from zero to one.

To solve this equation we are looking for a solution by seperating the variables:

$$u(x,t) = X(x)T(t) \tag{4}$$

If we take the partiall derivatives of this expression we get:

$$u_{xx} = X''(x)T(t), and u_t = X(x)T'(t) \tag{5}$$

So if we set put this in the equation (3) we get:

$$\frac{T'(t)}{T(t)} = \frac{X''(x)}{X(x)} = constant = -\lambda \tag{6}$$

We see that thsi must be equal to a constant and we see that this is an eigenvalue problem. We put a minus sign infront of the eigenvalue because of convention.

This gives uss the equations:

$$u(0,t) = X(0)T(t) = 0u(1,t) = X(1)T(t) = 0 \tag{7}$$

If we let $T(t) = 0$ we get the trivial solution which we are not interested

2.2 Algortihm

2.2.1 Forward Euler

In forward euler we are approximating the time derivative by:

$$u_t \approx \frac{u(x, t+\Delta t) - u(x,t)}{\Delta t} = \frac{u(x_i, t_j + \Delta t) - u(x_i, t_j)}{\Delta t} \tag{8}$$

This is an explicit scheme because it finds the current time step by looking at the (LES MER PÅ FORSKJELLEN AV IMPLICIT OG EXPLICIT)

We are also using a centered difference in space with the approximation as you can see in equation (1). So setting these to equtions equal to eachother gives:

$$\frac{u_{i,j+1} - u_{i,j}}{\Delta t} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} \Rightarrow u_{i,j+1} = \alpha u_{i-1,j} + (1-2\alpha)u_{i,j} + \alpha u_{i+1,j} \tag{9}$$

And this is the equation we use to solve this. We can implement this as a algorithm jus by looping over the timesteps, for so to loop over the x values where $x \in [0, 1]$.

2.2.2 Backward Euler

This is an implicit scheme where we approximating the time derivative by:

$$u_t \approx \frac{u(x,t) - u(x, t-\Delta t)}{\Delta t} = \frac{u(x_i, t_j) - u(x_i, t_j - \Delta t)}{\Delta t} \tag{10}$$

And by setting $u_t = u_x x$ we get the equation:

$$u_{i,j-1} = \alpha u_{i-1,j} + (1-2\alpha)u_{i,j} - \alpha u_{j+1,i} \tag{11}$$

We then introduce the matrix:

$$\begin{bmatrix} 1+2\alpha & -\alpha & 0 & 0 & \dots & 0 & \\ -\alpha & 1+2\alpha & -\alpha & 0 & \dots & 0 & \\ 0 & -\alpha & 1+2\alpha & -\alpha & \dots & 0 & \\ \vdots & \vdots & \vdots & \ddots & & \vdots & \\ 0 & 0 & 0 & \dots & & 1+2\alpha & \end{bmatrix}$$

Then we see that we can formulate this as a matrix multiplication problem:

$$\hat{A}V_j = V_{j-i} \tag{12}$$

Which means we can rewrite our differential equation problem to:

$$V_j = \hat{A}^{-1}V_{j_1} = \hat{A}^{-1}(\hat{A}^{-1}V_{j_2}) = ... = \hat{A}^{-j}V_0 \tag{13}$$

To solve this matrix equation we utilize the Gaussian elimination for tridiagonal matrixes which we solved in project 1.

2.2.3 Crank Nicolsen

3 Results

4 Discussion

5 Conclusion