Universitetet i Oslo

Computational physics

FYS4150

Exercise 1

Heine H. Ness

11. september 2016

Introduction

In this assignment we are going to solve a linear second-order differential equation numerically.

The programming language of choice is c++ and all programming was done in Qt.

Metod

Numerical method

In the real world we se time and space as continuous where all the real numbers are uniqu and can have infinit properties such as decimal points.

Computers store its numers on fysical mytes of 8 bits each. This leads to a fysical limit of how many digits that can be stored for any given number. And only a set of real numbers can be represented.

In practise the numberline becomes descret and not continuous. So if x is the varible of a continuous function $f(x)$ the numerical representation of $f(x)$ is also descrete.

The step-length h can be as little as the computer allows [1]. In a computer x and $f(x)$ would be represented as follows

$$
\begin{aligned}
x &= x_i = x_0 + ih \\
f(x) &= f(x_i) = f_i
\end{aligned}
$$

here x_0 is the starting point if one exists. And i is the step with h being the step-length.

Differential equations

Alot of physics problems involve solving a linear differential equations. Examples of these are the Schrödinger equation, diffusion equation and poisson's equation.

All these equations have the form

$$\nabla u(\mathbf{r}) = cf(\mathbf{r})$$

where $\nabla = \frac{\partial^2}{\partial x^2}$ the second order partial derivative. c is a constant, $f(\mathbf{r})$ is known as the inhomogeneous term and $u(\mathbf{r})$ is a real or complex valued function. $\mathbf{r}$ is a position vector.

In this assignment we deal with a one dimentional Poisson's equation and for our case the equation is written as follows.

$$-\frac{d^2}{dx^2}u(x) = f(x)$$

Nummerical derivation

Derivation decribes the curvature of a function and the second order derivative decribes its slope. The general formula for the first order derivative is

$$\frac{\partial}{\partial x} = \frac{x_i - x_{i-1}}{h}$$

Numerical error estimate

LU decomposition

-

References

[1] Computational physics Lecture Notes Fall 2015 section 2.3.2, Morten Hjorth-Jensen