# Numerical solutions to electrostatic problems: Efficient Algorithm Design, Project Structure, and Auxilliary Classes

David Muir

March 14th, 2013

## Introduction

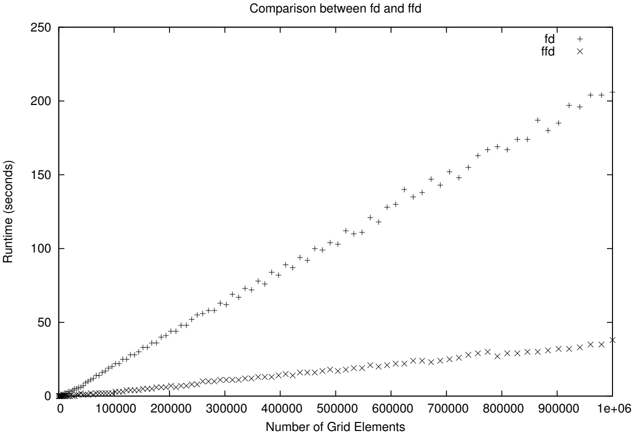
- Optimising an existing implementation of the Finite Difference algorithm.
- Defining the project structure and ensuring proper separation of concerns
- Creating auxilliary classes
    - To read pixel data from bitmap image files
    - To output grid information to the gnuplot graphing program

## Fast Finite Difference

The Fast Finite Difference algorithm was an attempt to optimise the existing Finite Difference algorithm by Karl Nordstrom.
The requirements for this algorithm were as follows,

- To reduce the runtime of the Finite Difference algorithm by an appreciable degree so as to allow more time-effective computations.
- To arrive at exactly the same approximation as the previous algorithm.

## Performance



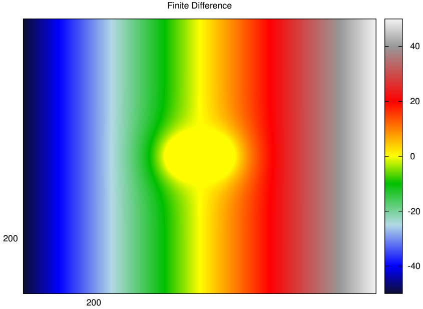
*Run on an i5-3317U@1.7GHz for 10,000 iterations per grid size and Ofast compiler optimsations on both algorithms.*

## Performance Optimisations

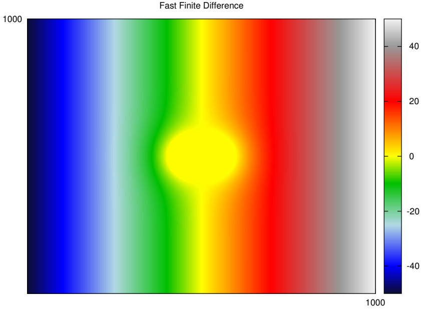
To attain the performance increases desired these traits were selected.

- Reduce repeated calculations
- Reduce memory access
- Reduce object creation
- Pass by reference rather than value

# Non-diminishing Accuracy



Finite Difference



Fast Finite Difference

## Project Structure

For a medium size project such as this a well defined structure is an important concern. Without the proper seperation of concerns the project would soon become an unmaintainable mess.

The code is split up into the following directories:

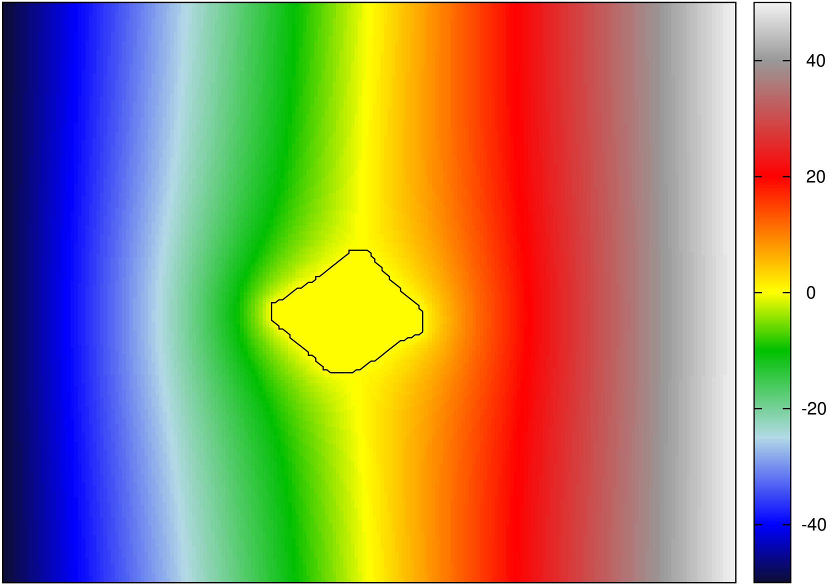
- Algorithms
- Errors
- Structures
- Utils

## Gnuplot

- The Gnuplot class was created to facilate in-app usage of the Gnuplot program. With an aim to allow more automation and reduce our dependancy on additional scripting.
- The class opens a connection to Gnuplot and allows passing of commands, comments, and Grid data directly to the program.
- There is also the option to read commands from a script and save to a script

## Bmp Reader

- The Bmp Reader class was created to allow us to input user defined shapes as a standard format
- The class reads in the pixel data of the bitmap and takes darker sections as referring to boundaries.
- The class will then create and return a grid populated by the bitmap


40
20
0
-20
-40