Midterm Project
TFY4240 Electromagnetic Theory
Assignment 1

Aksel Wilhelm Wold Eide

November 13, 2011

Abstract

Seperation of variables works well for solving for the electrical potential in a long, square, hollow tube, where the potential is zero on three of the edges and any general function on the last edge. A general, analytical solution can be obtained as a Fourier series, but the coefficients must in general be determined numerically. This can be done in a straightforward manner, allowing description of the potential and field as graphic plots. There are some minor numerical nuisances.

1 Introduction

The electric potential function, by nature of being a potential, must obey the Laplace equation. Otherwise, a given point would not have a well defined potential, but would require different values depending on which path was taken to arrive at it. This fairly strict mathematical condition means giving only boundary conditions for the potential of some space may uniquely define the potential at all points. In the following, this will be examined in a two dimensional space with specific boundary conditions.

2 Theory

2.1 Analytical work

2.1.1 Problem description

The problem to be solved is to determine the potential in a long, square, hollow tube, where the four walls have different potentials. The boundary conditions are as follows:

$$\begin{aligned} V(x=0,y) &= 0 \\ V(x=L,y) &= 0 \\ V(x,y=0) &= 0 \\ V(x,y=L) &= V_0(x) \end{aligned} \tag{1}$$

Where $V_0(x)$ can be any function. Being a potential function, V must respect Laplace's equation:

$$\nabla^2 V = 0$$

Which in two dimensions reduces to the following equation:

$$V_{xx} = -V_{yy} \tag{2}$$

2.1.2 Seperation of variables

We can introduce new variables, $u = \frac{x}{L}$ and $v = \frac{y}{L}$. We can also assume seperation of variables for the function V, such that:

$$V(x,y) \equiv F(u)G(v)$$

Inserting our separated function into Laplace's equation (2):

$$F''(u)G(v) = -F(u)G''(v)$$

Dividing on both sides by V yields:

$$\frac{F''(u)}{F(u)} = -\frac{G''(v)}{G(v)}$$

Evaluating this equation at the point $(u,v) = (a, \frac{1}{2})$ we get:

$$\frac{F''(a)}{F(a)} = -\frac{G''(\frac{1}{2})}{G(\frac{1}{2})}$$

Where the right hand side is obviously a constant. Thus, all possible choices of a must yield the same left hand side, i.e. a constant. We can then introduce the constant k:

$$\frac{F''(u)}{F(u)} = -\frac{G''(v)}{G(v)} = k \tag{3}$$

Studying in particular F, the constant curvature gives it the general form:

$$F(u) = A\exp(\sqrt{k}u) + B\exp(-\sqrt{k}u)$$

Where $\sqrt{k}$ will be a complex number for negative values of k. Inserting our function $V = F \cdot G$ into the boundary conditions (1), we obtain:

$$V(x = 0, y) = F(0)G(v) = 0$$

$$V(x = L, y) = F(1)G(v) = 0$$

If $G(v) = 0$ in these equations, then $V \equiv 0$ for all (x, y). We thus assume $G(v) \neq 0$, which reduces these equations to:

$$\begin{aligned} F(0) = A + B &= 0 \\ F(1) = A\exp(\sqrt{k}) + B\exp(-\sqrt{k}) &= 0 \end{aligned}$$

The first equation yields $A = -B$. Assuming $k > 0$, we obtain:

$$F(1) = 2A\sinh(\sqrt{k}) = 0$$

Which implies that $k = 0$, which is a direct contradiction with the assumption. Furthermore, $k = 0$ implies $V \equiv 0$ for all (x, y). Thus, we are left with $k < 0$, which means we have:

$$F(1) = 2|A|\sin(\sqrt{-k}) = 0$$

Where we have that A is a complex constant such that F will be real. It is also clear that, to allow the sine function to evaluate to zero:

$$\sqrt{-k} = n\pi, n \in \mathcal{N}$$

$$k = -(n\pi)^2$$

We have then shown that:

$$F(u) = \sum_{n=1}^{\infty} F_n \sin(n\pi u) \tag{4}$$

Using the exact same approach for G, we can show that:

$$G(v) = D\exp(\sqrt{-k}v) + E\exp(-\sqrt{-k}v)$$
$$G(0) = 0$$

Which results in, by evaluating the actual value of k and $D = -E$:

$$G(v) = \sum_{n=1}^{\infty} G_n \sinh(n\pi v) \tag{5}$$

In total, with (4) and (5) and superposition, we have obtained the most general function satisfying all but one of the boundary conditions as well as the Laplace equation:

$$V(u,v) = \sum_{n=1}^{\infty} V_n \sin(n\pi u)\sinh(n\pi v) \tag{6}$$

Where V_n are unspecified constants. These must be adjusted to satisfy the last boundary condition:

$$V(u,1) = \sum_{n=1}^{\infty} V_n \sin(n\pi u)\sinh(n\pi) = V_0(u) \tag{7}$$

2.1.3 Fourier coefficients

Due to orthogonality of sine functions when integrated over an integer number of periods, we can calculate the m'th constant V_m. This is done by multiplying (7) by $\sin(m\pi u)$ and integrating as follows:

$$\int_{u=-1}^{1} du\ \sin(m\pi u)V_0'(u) = V_m \sinh(m\pi) \tag{8}$$

Here, $V_0'(u)$ is an odd extension of $V_0(u)$, such that $V_0'(-u) = -V_0(u)$. We can define the error ϵ in V, as a solution to the boundary condition, with m Fourier terms, as:

$$\epsilon = \int_{u=-1}^{1} du\ (\sum_{n=1}^{m} V_n \sin(n\pi u)\sinh(n\pi) - V_0(u))^2 \tag{9}$$

2.1.4 Electrical field

Also, from V, the electrical field E is given by:

$$\vec{E} = \Delta V)$$

Which in the two dimensional case reduces to:

$$\vec{E}(u,v) = \left[\frac{\delta V}{\delta u}, \frac{\delta V}{\delta v}\right]$$

Inserting (6) into this equation yields componentwise equations:

$$\frac{\delta V}{\delta u} = \sum_{n=1}^{\infty} V_n n\pi \cos(n\pi u)\sinh(n\pi v) \tag{10}$$

$$\frac{\delta V}{\delta v} = \sum_{n=1}^{\infty} V_n n\pi \sin(n\pi u)\cosh(n\pi v) \tag{11}$$

2.2 Numerical work

2.2.1 Calculation

Using the formula for the m'th Fourier coefficient (8), evaluating the integrals numerically allows us to determine the values of V_m for any given $V_0(u)$. Inserting these coefficients into (6), we have a function describing the potential V at all points (u, v). It is also straightforward to calculate $\vec{E}$ from (10) and (11). The only issue is that our description of V involved an infinite sum. Using the error formula (9) allows us to simply include terms until the accuracy reaches the desired level.

2.2.2 Plotting

Python's plotting library, *matplotlib* provides functionality for plotting the electrical potential V and electrical field $\vec{E}$. Defining a square grid over the area $(u = 0, v = 0)$ to $(u = 1, v = 1)$ and sampling the functions makes plotting them straightforward. A fairly coarse $50 \cdot 50$ grid was used with satisfactory results.

2.2.3 Coding

The complete code for Python in given in the appendix, 6.

3 Results

3.1 Plotting

For each trial boundary function, $V_0(u)$, the following plots were generated:

- 1) The boundary function, $V_0(u)$ and $V(u, 1)$ with successively more Fourier terms.
- 2) A contour plot of the potential function, $V(u, v)$, for the most accurate Fourier approximation from (1).
- 3) A quiver plot of the electrical field, $E(u, v)$, for the same Fourier approximation.

The plots, being fairly numerous and large, have been placed at the end of the text.

4 Discussion

4.1 General observations

The plots show that the resulting potential follows some general trends, regardless of $V_0(u)$. The potential is largest in absolute value for $v = 1$, and drops relatively quickly as the value of v diminishes. Similarily, the potenial is generally largest near $u = \frac{1}{2}$, and drops towards both edges, $u = 0$ and $u = 1$. This is a consequence of the boundary conditions, which force the potential to drop off as it approaches the boundary where $V = 0$. The only significantly different plot, even though the boundary functions are widely different, can be seen in figure 5. The sine function has multiple nodes, which in turn causes the potential function to have multiple nodes. The plot in figure 4 sticks out because the boundary potential is negative. The colors are simply reversed, and there is no other effect. It is obvious from the equations given in the theory section that multiplying the boundary function $V_0(u)$ by any postiive or negative constant will only result in the same multiplication in the potential $V(u, v)$ and electrical field $\vec{E}(u, v)$, while the general shape of the solution will remain unchanged.

4.2 Numerical aberrations

There is a significant difference between figures 1, 4, 5, which all converge very well for $m < 10$, and figures 2 and 3, none of which convege within the maximum allowed terms, $m = 50$. This is in both cases caused by points where the function cannot be differentiated, which is a well known problem for the convergence of Fourier series. In the case of figure 2, this is a result of the unit step in the heaviside function, whereas in figure 3, the culprit is $V_0(0) \neq 0$, which gives an ambigious value for $V(1, 0)$. The plot of the electrical field in figure 2 shows a numerical aberration. Exactly where the unit step function kicks in, we have an electrical field approaching infinity, as seen by the extremely long vector arrows in this region. Exactly the same sort of behaviour is present in figure 3, for the same reason. These behaviours are acceptable; in both of the cases, discontinuity in the potential function is introduced not by our solution, but by the boundary conditions, and the electrical field will inevitably be ill defined. There is also a slight aberration in figure 5, where the positive and negative regions of the potential have alternating colors. This is simply a result of the plotting tool showing a difference between extremely slight, positive and negative numbers, even when their magnitude is neglicible.

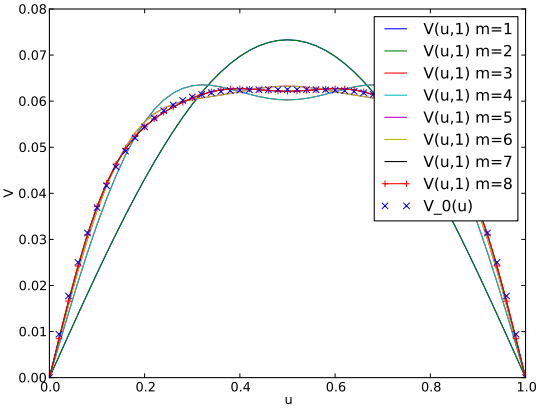
5 Conclusions

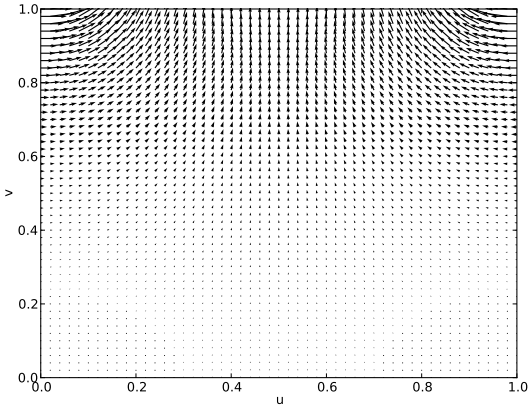
The results are in good accordance with theory, and shows that the hollow, square tube can be modelled with straightforward mathematics and numerics. The solutions can be calculated to very good approximation with neglicible runtime, unless discontinuity in the boundary conditions cause problems for the Fourier approximation. Even in this case, including a sufficient amount of terms produces decent results.

6 Appendix

Attached is the complete source code for Python used to produce the results in 3. The code has only been slightly modified to plot for different functions, $V_0(u)$ as commented in the code itself. Also included is the the complete LaTeX code to produce this pdf-document. Some minor revisions in the

Figure 1: $V_0(u) = \frac{1}{16} - (u - \frac{1}{2})^4$





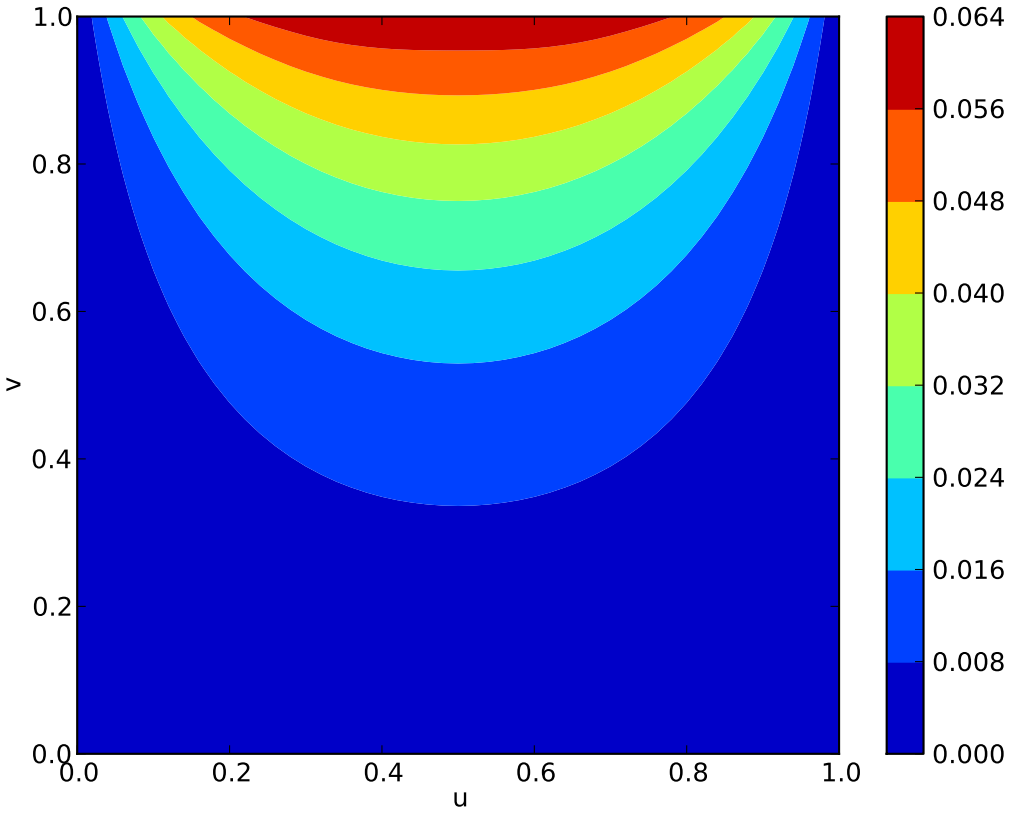
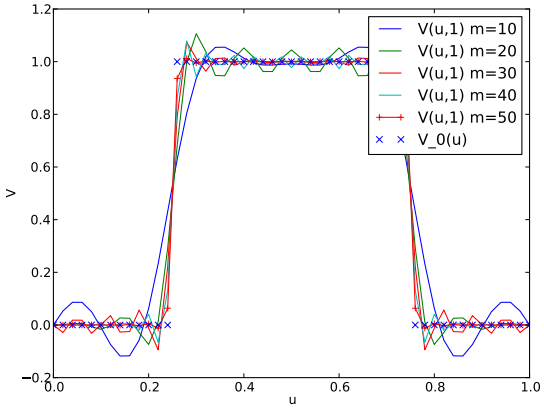
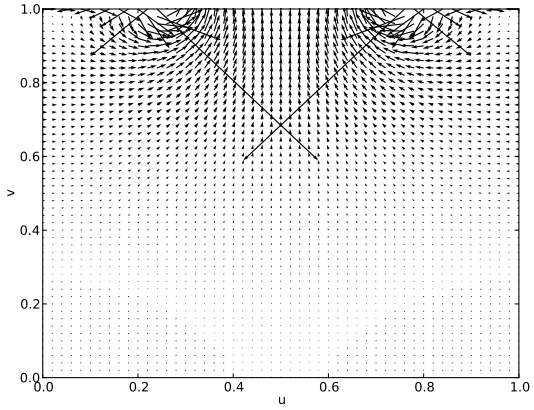


Figure 2: $V_0(u) = \theta(u - \frac{1}{4})\theta(\frac{3}{4} - u)$, where θ is the heaviside function





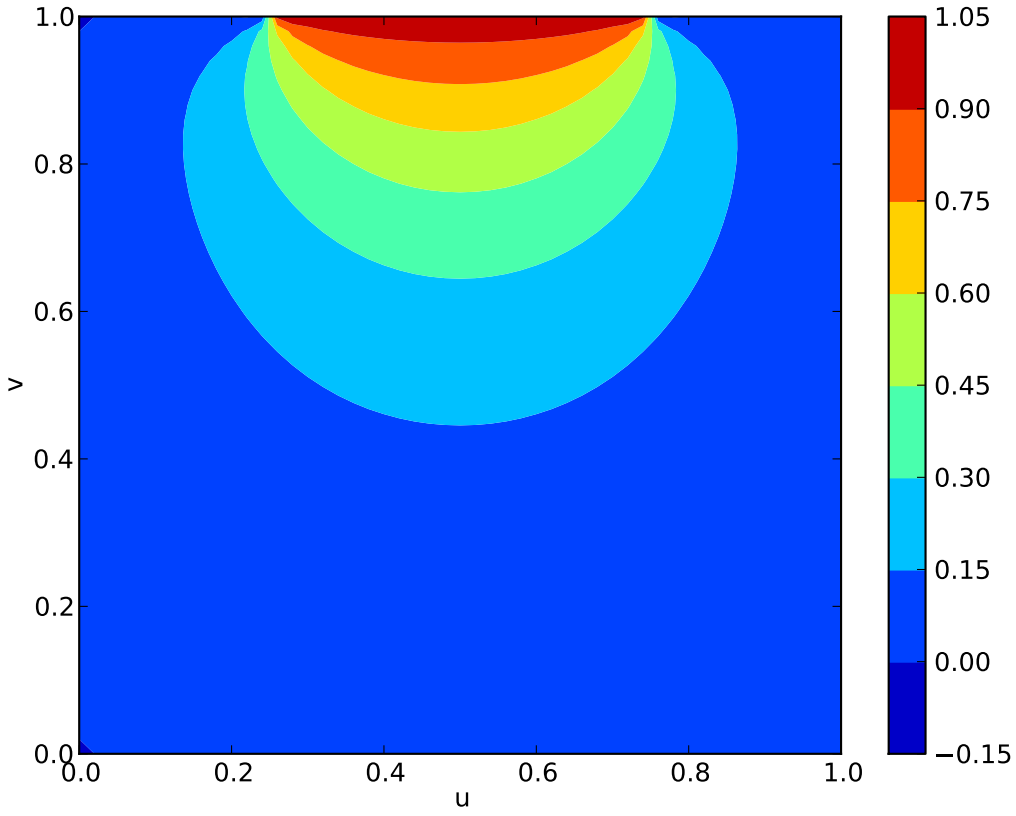
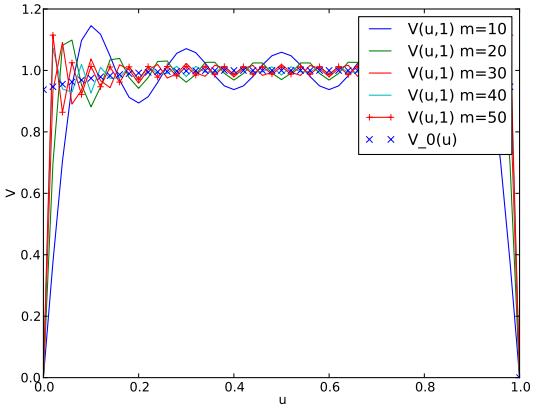
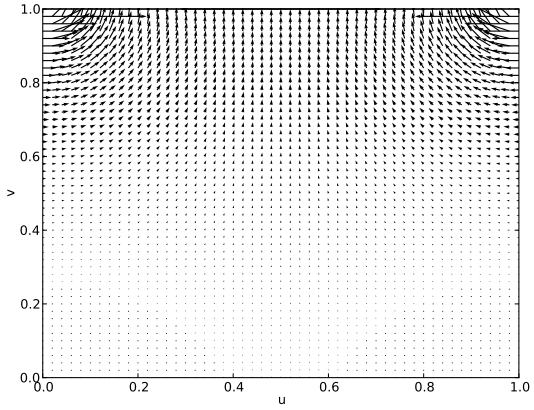


Figure 3: $V_0(u) = 1 - (u - \frac{1}{2})^4$





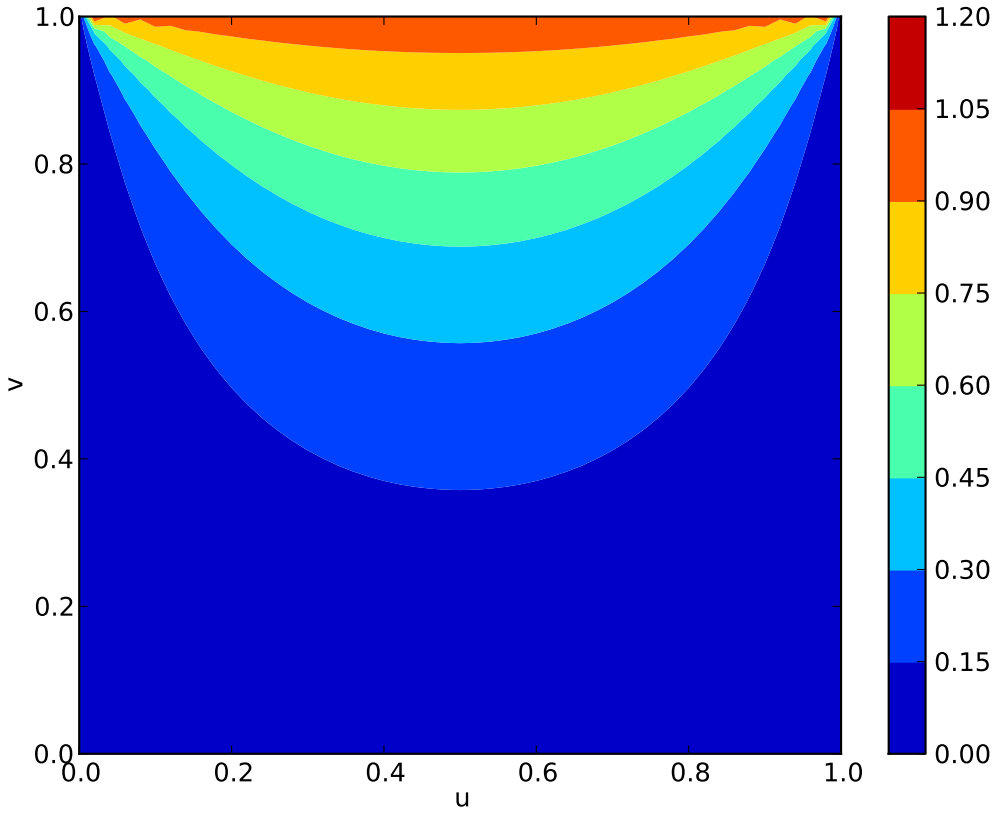


Figure 4: $V_0(u) = \sin(u(u-1)\cosh(u))^3$

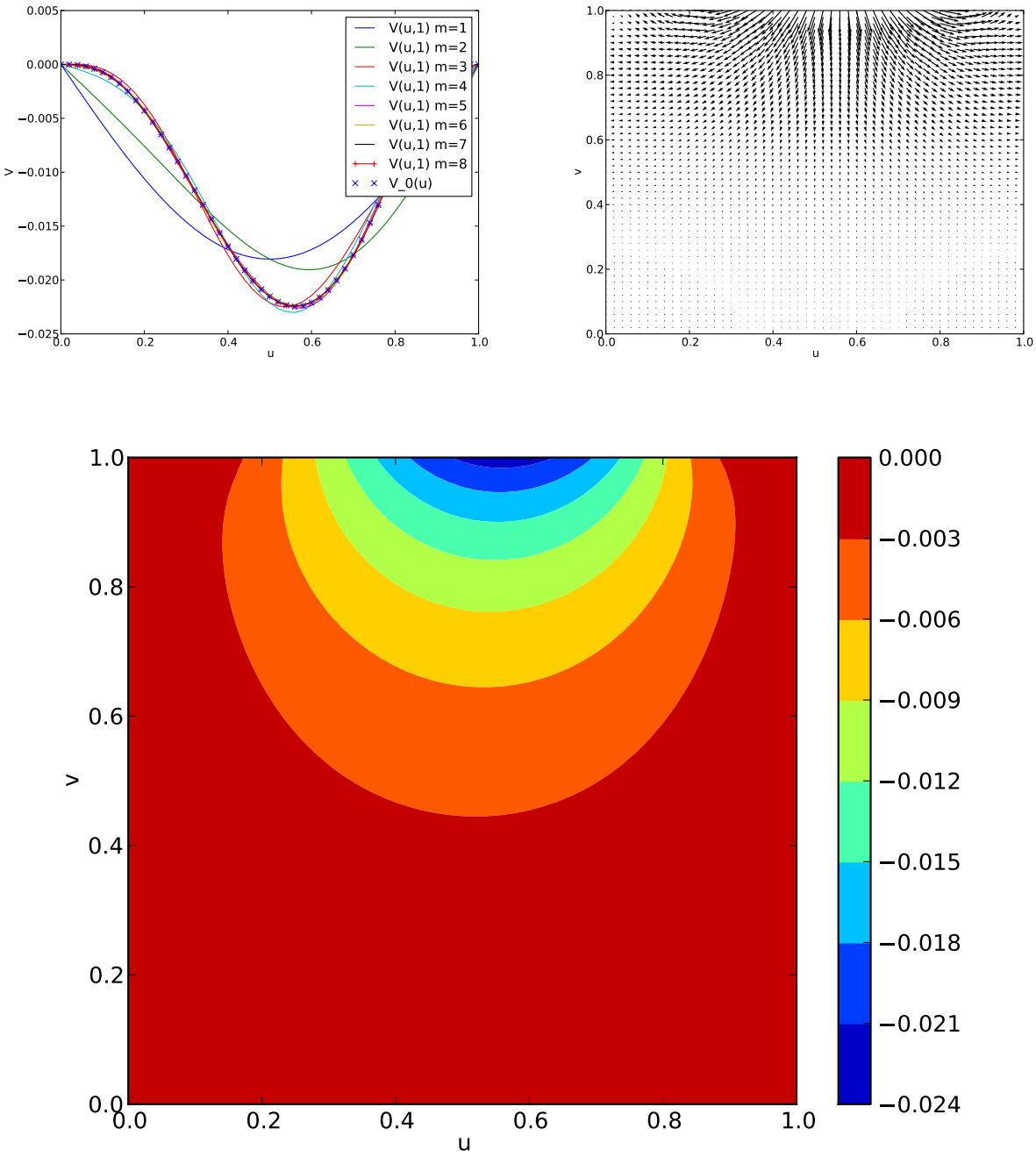
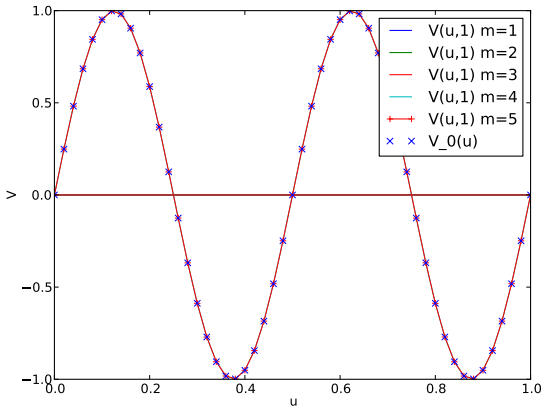
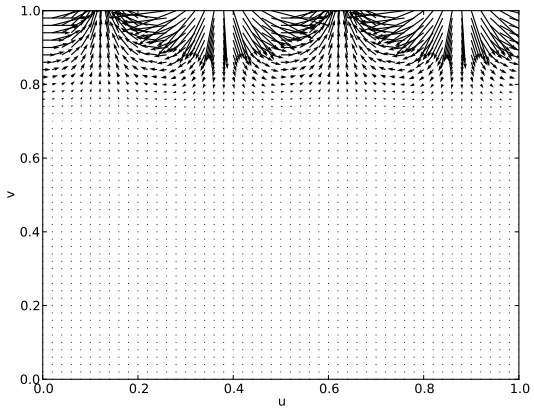
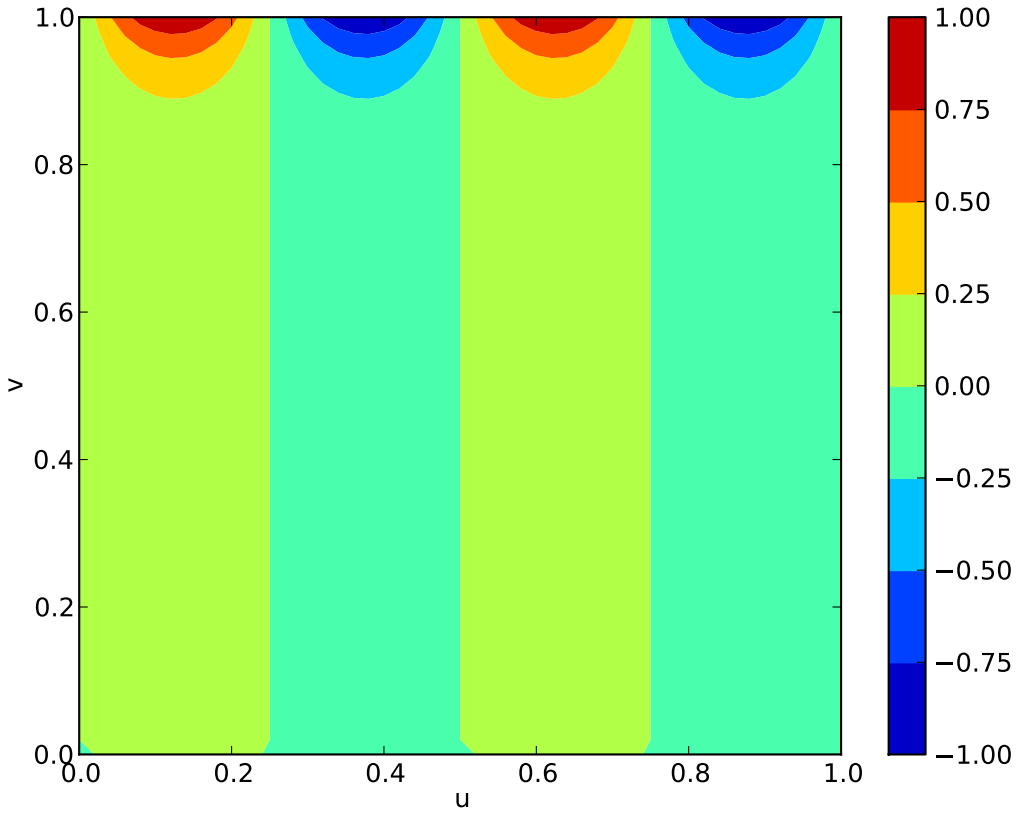


Figure 5: $V_0(u) = \sin(4\pi u)$







actual text may have occured in between exporting the code and the finalizing of the document.

```
#! /usr/bin/env python
# -*- coding: utf-8 -*-

#A bunch of includes I used in a Statistical Physics-excercies, which generally provides
what I need

from __future__ import division             # To assure that division between two integers
gives a sensible result
import sys                                  # Here we use the 'sys' library to handle
command line arguments
import numpy                                # A (big) library for doing array oriented
numerics
import matplotlib as mpl                    # A plotting framework
from matplotlib import rc                   # Configuration files
mpl.use('PDF')                              # Uncomment to generate figures in PDF format
import matplotlib.pyplot as pyplot
from scipy.integrate import odeint          # To solve systems of 1st order ordinary
differential equations
from scipy.optimize  import newton          # Rootfinder using Newton's method
from scipy.integrate import quad            # To do one-dimensional integrals (quadratures)

#V0 is the boundary function used
#Tweaking the fTerms-values it probably a good idea; difficult functions will require a
large number
#of Fourier terms, calling for larger steps to plot fewer functions

def V0(u):
    if u<0: return -V0(-u)                          #Creates an antisymmetric extension
    for Fourier purposes

    #if (u>1/4 and u<3/4): return 1                 #1 This is a heaviside function
    #return 0

    return numpy.sin(4*numpy.pi*u)                  #3 This is a sine function

    #return (1 - (u - 1/2)**4)                      #4 This is a sort of quadratic
    function with non-zero edges

    #return (1/16 - (u - 1/2)**4)                   #5 This is the same sort of function
    with zero edges

    #return numpy.sin(u*(u-1)*numpy.cosh(u))**3     #6 This is a fairly random function

def V0square(u):
    return V0(u)**2

#Vterm describes the m'th term in the Fourier series for V
def Vterm(u,v,m):
    return numpy.sin(m*numpy.pi*u)*numpy.sinh(m*numpy.pi*v)

#Euterm and Evterm describe the m'th term in the Fourier series for E's u and v-coordinates
def Euterm(u,v,m):
    return numpy.cos(m*numpy.pi*u)*numpy.sinh(m*numpy.pi*v)*m*numpy.pi

def Evterm(u,v,m):
    return numpy.sin(m*numpy.pi*u)*numpy.cosh(m*numpy.pi*v)*m*numpy.pi
```

```
#The m'th Fourier term, used in integration to determine the Fourier coefficients
def quadFunc(u,m):
    return numpy.sin(m*u*numpy.pi)*V0(u)

#Error function for estimating convergence
def errFunc(u,add):
    f = 0
    for n in xrange(1,add[0]):
        f += Vterm(u,1,n)*add[1][n]
    return (f-V0(u))**2


def main(argv):

    fTermsMax = 50                                      # Max number of Fourier terms to use
    fTermsStep = 1                                      # Number of new Fourier terms before
    reevaluating precision
    fTermsCurrent = 0                                   # Number of Fourier terms currently
    in use; dynamic, for storage purposes
    gridSize = 50                                       # Number of subdivisions in the u-
    and v-direction of the area
    fCoeff = range(0,fTermsMax+1)                       # Array for storing the coefficients
    error = 100                                         # Initializing error as larger than
    maxError
    maxError = 0.0001 * quad(V0square,-1,1)[0]          # Target for approximation error,
    relative to the norm of the target function


    xValues = numpy.linspace(0,1,gridSize+1)
    yValues = numpy.linspace(0,0,gridSize+1)

    #Complicated loop; it calculates the Fourier coefficients, while every ?-steps
    considering if convergence is already satisfactory.
    #It also plots the approximations as it goes along. It's most important function in
    going forwards is deciding the number of
    #Fourier coefficients to use and setting their value.
    while((error > maxError) and (fTermsCurrent < fTermsMax)):
        if (error != 100): pyplot.plot(xValues, yValues, '-', label="V(u,1) m=%d" %(
        fTermsCurrent))
        for m in xrange(fTermsCurrent+1,fTermsCurrent+1+fTermsStep):
            fCoeff[m] = quad(quadFunc, -1, 1, m)[0] / numpy.sinh(m*numpy.pi)
            for x in range(gridSize):
                yValues[x] += Vterm(xValues[x],1,m)*fCoeff[m]
        fTermsCurrent += fTermsStep
        error = quad(errFunc, -1, 1, [fTermsCurrent, fCoeff])[0]

    #Plots the final approximation
    pyplot.plot(xValues, yValues, '+-r', label="V(u,1) m=%d" %(fTermsCurrent))

    #Calculates and pltos the actual boundary function
    for x in range(gridSize):
        yValues[x] = V0(xValues[x])
    pyplot.plot(xValues, yValues, 'xb', label="V_0(u)")

    #More plotting related stuff
    #pyplot.title("Comparison of given and approximated functions at the non-zero boundary")
    pyplot.legend()
```

```
    pyplot.xlabel('u')
    pyplot.ylabel('V')
    pyplot.savefig("boundary.pdf")
    pyplot.clf()

    #Initializes xValues and yValues as a grid, and also variables to store the valeus of
    the potential the electrical field components on the grid
    xValues = numpy.linspace(0,1,gridSize+1)
    yValues = numpy.linspace(0,1,gridSize+1)
    zValues = [[0]*(gridSize+1) for x in xrange(gridSize+1)]
    EuValues = [[0]*(gridSize+1) for x in xrange(gridSize+1)]
    EvValues = [[0]*(gridSize+1) for x in xrange(gridSize+1)]

    #Caluclates the potential and electrical field components for each point on the grid,
    with m incrementing over Fourier terms
    for m in range(1,fTermsCurrent+1):
        for x in range(gridSize+1):
            for y in range(gridSize+1):
                zValues[y][x] += Vterm(xValues[x], yValues[y], m)*fCoeff[m]
                EuValues[y][x] += Euterm(xValues[x], yValues[y], m)*fCoeff[m]
                EvValues[y][x] += Evterm(xValues[x], yValues[y], m)*fCoeff[m]

    #Plotting for the potential
    CP = pyplot.contourf(xValues, yValues, zValues)
    pyplot.colorbar(CP)
    #pyplot.title('V(u,v)')
    pyplot.xlabel('u')
    pyplot.ylabel('v')
    pyplot.savefig("contourV")
    pyplot.clf()

    #Plotting for the electrical field
    pyplot.quiver(xValues,yValues,EuValues,EvValues)
    #pyplot.title('E(u,v)')
    pyplot.xlabel('u')
    pyplot.ylabel('v')
    pyplot.savefig("quiverE")
    pyplot.clf()

if __name__ == "__main__":
    main(sys.argv[1:])
```

```
\title{Midterm Project \\ TFY4240 Electromagnetic Theory \\ Assignment 1}
\author{
Aksel Wilhelm Wold Eide \\
}
\date{\today}
\documentclass[12pt]{article}
\usepackage{amsmath, amsthm, amssymb}
\usepackage{graphicx}
\begin{document}
\maketitle
\begin{abstract}
Seperation of variables works well for solving for the electrical potential in a long, square, hollow tube, where the potential is zero on three of the edges and any general function on the last edge. A general, analytical solution can be obtained as a Fourier series, but the coefficients must in general be determined numerically. This can be done in a straightforward manner, allowing description of the potential and field as graphic plots. There are some minor numerical nuisances.
\end{abstract}
\section{Introduction}
The electric potential function, by nature of being a potential, must obey the Laplace equation. Otherwise, a given point would not have a well defined potential, but would require different values depending on which path was taken to arrive at it. This fairly strict mathematical condition means giving only boundary conditions for the potential of some space may uniquely define the potential at all points. In the following, this will be examined in a two dimensional space with specific boundary conditions.
\section{Theory}
\subsection{Analytical work}
\subsubsection{Problem description}
The problem to be solved is to determine the potential in a long, square, hollow tube, where the four walls have different potentials. The boundary conditions are as follows:
\begin{align}
V(x=0,y) &= 0 \notag \\
V(x=L,y) &= 0 \notag \\
V(x,y=0) &= 0\notag \\
V(x,y=L) &= V_0(x) \label{eq:boundary}
\end{align}
Where $V_0(x)$ can be any function. Being a potential function, V must respect Laplace's equation:
$$\nabla^2 V = 0$$
Which in two dimensions reduces to the following equation:
\begin{equation}\label{eq:laplace}V_{xx} = - V_{yy}\end{equation}
\subsubsection{Seperation of variables}
We can introduce new variables, $u=\frac{x}{L}$ and $v=\frac{y}{L}$. We can also assume seperation of variables for the function $V$, such that:
$$V(x,y) \equiv F(u)G(v)$$
Inserting our separated function into Laplace's equation \eqref{eq:laplace}:
$$F''(u)G(v) = -F(u)G''(v)$$
Dividing on both sides by $V$ yields:
$$\frac{F''(u)}{F(u)} = -\frac{G''(v)}{G(v)}$$
Evaluating this equation at the point $(u,v) = (a,\frac{1}{2})$ we get:
$$\frac{F''(a)}{F(a)} = -\frac{G''(\frac{1}{2})}{G(\frac{1}{2})}$$
Where the right hand side is obviously a constant. Thus, all possible choices of $a$ must yield the same left hand side, i.e. a constant. We can then introduce the constant $k$:
```

\begin{equation}\label{eq:seperation}\frac{F"(u)}{F(u)} = -\frac{G"(v)}{G(v)} = k\end{equation}
Studying in particular F, the constant curvature gives it the general form:
$$F(u) = A\exp(\sqrt{k}u) + B\exp(-\sqrt{k}u)$$
Where $\sqrt{k}$ will be a complex number for negative values of k. Inserting our function $V = F\cdot G$ into the boundary conditions \eqref{eq:boundary}, we obtain:
$$ V(x=0,y) = F(0)G(v) = 0$$
$$ V(x=L,y) = F(1)G(v) = 0$$
If $G(v) = 0$ in these equations, then $V \equiv 0$ for all (x,y). We thus assume $G(v) \neq 0$, which reduces these equations to:
\begin{align*}F(0) &= A + B &= 0 \\
F(1) &= A\exp(\sqrt{k}) + B\exp(-\sqrt{k}) &= 0\end{align*}
The first equation yields $A=-B$. Assuming $k>0$, we obtain:
$$F(1) = 2A\sinh(\sqrt{k}) = 0$$
Which implies that $k=0$, which is a direct contradiction with the assumption. Furthermore, $k=0$ implies $V \equiv 0$ for all (x,y). Thus, we are left with $k<0$, which means we have:
$$F(1) = 2|A|\sin(\sqrt{-k}) = 0$$
Where we have that A is a complex constant such that F will be real. It is also clear that, to allow the sine function to evaluate to zero:
$$\sqrt{-k} = n\pi, n \in \mathcal{N}$$
$$k = -(n\pi)^2$$
We have then shown that:
\begin{equation}\label{eq:F}F(u) =\sum_{n=1}^{\infty} F_n \sin(n\pi u)\end{equation}
Using the exact same approach for G, we can show that:
$$G(v) = D\exp(\sqrt{-k}v) + E\exp(-\sqrt{-k}v) = $$
$$G(0) = 0$$
Which results in:
\begin{equation}\label{eq:G}G(v) = \sum_{n=1}^{\infty} G_n \sinh(n\pi v)\end{equation}
In total, with \eqref{eq:F} and \eqref{eq:G} and superposition, we have obtained the most general function satisfying all but one of the boundary conditions as well as the Laplace equation:
\begin{equation}\label{eq:V}V(u,v) = \sum_{n=1}^{\infty} V_n \sin(n \pi u) \sinh(n\pi v)\end{equation}
Where V_n are unspecified constants. These must be adjusted to satisfy the last boundary condition:
\begin{equation}\label{eq:V0}V(u,1) = \sum_{n=1}^{\infty} V_n \sin(n\pi u) \sinh(n\pi) = V_0(u)\end{equation}
\subsubsection{Fourier coefficients}
Due to orthogonality of sine functions when integrated over an integer number of periods, we can calculate the m'th constant V_m. This is done by multiplying \eqref{eq:V0} by $\sin(m\pi u)$ and integrating as follows:
\begin{equation}\label{eq:Vm}\int_{u=-1}^{1}du \; \sin(m\pi u) V'_0(u) = V_m \sinh(m\pi)\end{equation}
Here, $V'_0(u)$ is an odd extension of $V_0(u)$, such that $V'_0(-u) = -V_0(u)$. We can define the error ϵ in V, as a solution to the boundary condition, with m Fourier terms, as:
\begin{equation}\label{eq:error}\epsilon = \int_{u=-1}^{1} du \; (\sum_{n=1}^{m} V_n \sin(n\pi u) \sinh(n\pi)-V_0(u))^2\end{equation}
\subsubsection{Electrical field}
Also, from V, the electrical field E is given by:
$$\vec E = \Delta V)$$
Which in the two dimensional case reduces to:
$$\vec E(u,v) =\left[\frac{\delta V}{\delta u}, \frac{\delta V}{\delta v}\right]$$
Inserting \eqref{eq:V} into this equation yields componentwise equations:
\begin{equation}\label{eq:Eu}\frac{\delta V}{\delta u} = \sum_{n=1}^{\infty} V_n n \pi \cos(n \pi

u) \sinh(n\pi v)\end{equation}
\begin{equation}\label{eq:Ev}\frac{\delta V}{\delta v} = \sum_{n=1}^{\infty} V_n n \pi \sin(n \pi u) \cosh(n\pi v)\end{equation}
\subsection{Numerical work}
\subsubsection{Calculation}
Using the formula for the m'th Fourier coefficient \eqref{eq:Vm}, evaluating the integrals numerically allows us to determine the values of V_m for any given $V_0(u)$. Inserting these coefficients into \eqref{eq:V}, we have a function describing the potential V at all points (u,v). It is also straightforward to calculate $\vec E$ from \eqref{eq:Eu} and \eqref{eq:Ev}. The only issue is that our description of V involved an infinite sum. Using the error formula \eqref{eq:error} allows us to simply include terms until the accuracy reaches the desired level.
\subsubsection{Plotting}
Python's plotting library, $matplotlib$ provides functionality for plotting the electrical potential V and electrical field $\vec E$. Defining a square grid over the area $(u=0,v=0)$ to $(u=1,v=1)$ and sampling the functions makes plotting them straightforward.
\subsubsection{Coding}
The complete code for Python in given in the appendix, \ref{appendix}.
\section{Results}\label{results}
\subsection{Plotting}
For each trial boundary function, $V_0(u)$, the following plots were generated:
\begin{itemize}
\item 1) The boundary function, $V_0(u)$ and $V(u,1)$ with successively more Fourier terms.
\item 2) A contour plot of the potential function, $V(u,v)$, for the most accurate Fourier approximation from (1).
\item 3) A quiver plot of the electrical field, $E(u,v)$, for the same Fourier approximation.
\end{itemize}
The plots, being fairly numerous and large, have been placed at the end of the text.
\section{Discussion}\label{discussion}
\subsection{General observations}
The plots show that the resulting potential show some general trend, regardless of $V_0(u)$. The potential is largest in absolute value for $v=1$, and drops relatively quickly as the value of v diminishes. Similarily, the potenial is generally largest near $u=\frac{1}{2}$, and drops towards both edges, $u=0$ and $u=1$. This is a consequence of the boundary conditions, which force the potential to drop off as it approaches the boundary where $V = 0$. The only significantly different plot, even though the boundary functions are widely different, can be seen in figure \ref{fig5}. The sine function has multiple nodes, which in turn causes the potential function to have multiple nodes. The plot in figure \ref{fig4} sticks out because the boundary potential is negative. The colors are simply reversed, and there is no other effect. It is obvious from the equations given in the theory section that multiplying the boundary function $V_0(u)$ by any postiive or negative constant will only result in the same multiplication in the potential $V(u,v)$ and electrical field $\vec E(u,v)$, while the general shape of the solution will remain unchanged.
\subsection{Numerical aberrations}
There is a significant difference between figures \ref{fig1}, \ref{fig4}, \ref{fig5}, which all converge very well for $m<10$, and figures \ref{fig2} and \ref{fig3}, none of which convege within the maximum allowed terms, $m=50$. This is in both cases caused by points where the function cannot be differentiated. In the case of figure \ref{fig2}, this is a result of the unit step in the heaviside function, whereas in figure \ref{fig3}, the culprit is $V_0(0) \neq 0$, which gives an ambigious value for $V(1,0)$. The plot of the electrical field in figure \ref{fig2} shows a numerical aberration. Exactly where the unit step function kicks in, we have an electrical field approaching infinity, as seen by the extremely long vector arrows in this region. Exactly the same sort of behaviour is present in figure \ref{fig3}, for the same reason. Both of these behaviours are acceptable; in the first case, discontinuity in the potential function is introduced not by our solution,

but by the boundary conditions, and the electrical field will inevitably be ill defined.
\section{Conclusions}\label{conclusions}
The results are in good accordance with theory, and shows that the hollow, square tube can be modelled with straightforward mathematics and numerics. The solutions can be calculated to very good approximation with neglicible runtime.
\begin{figure}
\caption{$V_0(u) = \frac{1}{16} - (u-\frac{1}{2})^4$ }
\includegraphics[scale=0.4]{5boundary.pdf}
\includegraphics[scale=0.4]{5quiverE.pdf}
\includegraphics[scale=0.8]{5contourV.pdf}
\label{fig1}
\end{figure}
\begin{figure}
\caption{$V_0(u) = \theta(u-\frac{1}{4})\theta(\frac{3}{4}-u)$, where θ is the heaviside function}
\includegraphics[scale=0.4]{1boundary.pdf}
\includegraphics[scale=0.4]{1quiverE.pdf}
\includegraphics[scale=0.8]{1contourV.pdf}
\label{fig2}
\end{figure}
\begin{figure}
\caption{$V_0(u) = 1-(u-\frac{1}{2})^4$}
\includegraphics[scale=0.4]{4boundary.pdf}
\includegraphics[scale=0.4]{4quiverE.pdf}
\includegraphics[scale=0.8]{4contourV.pdf}
\label{fig3}
\end{figure}
\begin{figure}
\caption{$V_0(u) = \sin(u(u-1)\cosh(u))^3$}
\includegraphics[scale=0.4]{6boundary.pdf}
\includegraphics[scale=0.4]{6quiverE.pdf}
\includegraphics[scale=0.8]{6contourV.pdf}
\label{fig4}
\end{figure}
\begin{figure}
\caption{$V_0(u) = \sin(4\pi u)$}
\includegraphics[scale=0.4]{3boundary.pdf}
\includegraphics[scale=0.4]{3quiverE.pdf}
\includegraphics[scale=0.8]{3contourV.pdf}
\label{fig5}
\end{figure}
\section{Appendix}\label{appendix}
Attached is the complete source code for Python used to produce the results in \ref{results}. The code has only been slightly modified to plot for different functions, $V_0(u)$ as commented in the code itself. Also included is the the complete LaTeX code to produce this pdf-document. Some minor revisions in the actual text may have occured in between exporting the code and the finalizing of the document.
\end{document}