The Capacitance of Long Tubes

David-Alexander Robinson Sch.[*]
08332461

13th December 2011

Contents

Abstract

The *Gauss-Seidel Algorithm* was used to solve *Poisson's Equation* for the electrostatic potential inside a coaxial tube as implemented with C++ programing code. A `void` function was written to calculate the *Electric Field* at all points between the tubes, and a graph of the x-component of the electric field along the y direction was plotted, which was found to agree with Gauss' Law for electric charge. A `double` function was written to integrate the square magnitude of the electric field throughout the tube for a rage of values of b, and again a graph of the resulting data was plotted. Finally the *Over-Relaxed Gauss-Seidel Algorithm* was used to calculate the electrostatic potential for a range of values of the relaxation parameter ω and a graph of ω versus the number of iterations required was plotted with $a = 0.50$ and $b = 0.50$.

1 Introduction & Theory

1.1 The Poisson Equation

Poisson's Equation in two dimensions over some region $\mathcal{D}$ with boundary conditions

$$u(x, y) = 0$$

on $\partial\mathcal{D}$ is

$$\frac{\partial^2 u}{\partial x^2}(x, y) + \frac{\partial^2 u}{\partial y^2}(x, y) = 0 \tag{1}$$

Over a finite grid $[i, j]$ for $i, j = 0, 1, 2, ..., n$ this becomes

$$u(i + 1, j) + u(i - 1, j) + u(i, j + 1) + u(i, j - 1) - 4u(i, j) = 0 \tag{2}$$

1.2 The Gauss-Seidel Algorithm

Solving this equation reduces to resolving the matrix equation

$$A\, u(i, j) = 0$$

where here A is a sparse matrix. Then rewriting

$$P\, u(i, j) = -(A - P)u(i, j)$$

where P is an invertible matrix and taking

$$A = L + D + U$$

where D is the diagonal of A, and U and L are the upper and lower triangular matrices of A respectively, for the *Gauss-Seidel Algorithm*

$$P = D + L$$

which gives

$$u^{k+1}(i, j) = \frac{1}{4}\left(u^k(i + 1, j) + u^{k+1}(i - 1, j) + u^k(i, j + 1) + u^{k+1}(i, j - 1)\right) \tag{3}$$

1.3 The Electric Field

The *Electric Field* $\vec{E}$ is given by

$$\begin{aligned}\vec{E} &= \vec{\nabla}u(i,j) \\ &= \frac{\partial u(i,j)}{\partial x}\hat{i} + \frac{\partial u(i,j)}{\partial y}\hat{j}\end{aligned} \tag{4}$$

In a discretised grid system of size n×n the limit definition of a derivative may be used whence

$$E_x(i,j) = \frac{u(i+1,j) - u(i,j)}{h} \quad \text{and} \quad E_y(i,j) = \frac{u(i,j+1) - u(i,j)}{h} \tag{5}$$

where $h = 1/n$.

Gauss' Law for electric charge states that

$$\vec{\nabla} \cdot \vec{E} = \rho \tag{6}$$

This may also be expressed as

$$\phi = \oint_S \vec{E} \cdot \mathrm{d}\vec{A} \tag{7}$$

1.4 Darboux Sums and Darboux Integrals

For a partition P of an interval $[a,b]$ with

$$P = (x_0, x_1, ..., x_n)$$

the *Upper* and *Lower Darboux Sums* of $f : [a,b] \to \mathbb{R}$ with respect to P are defined as

$$U_{f,P} = \sum_{i=1}^{n}(x_i - x_{i-1})M_i$$

and

$$L_{f,P} = \sum_{i=1}^{n}(x_i - x_{i-1})m_i$$

respectively, where

$$M_i = \sup_{x\in[x_{i-1},x_i]} f(x)$$

and

$$m_i = \inf_{x\in[x_{i-1},x_i]} f(x)$$

The Upper and Lower Darboux Integrals of f are then

$$U_f = \inf\{U_{f,P} : P \text{ is a partition of } [a,b]\}$$

and

$$U_f = \sup\{L_{f,P} : P \text{ is a partition of } [a,b]\}$$

If $U_f = L_f$ then f is *Darboux Integrable* and the integral of $f(x)$ is given by

$$\int_a^b f(x)\,\mathrm{d}x = U_f = L_f \tag{8}$$

In this way a continuous integral can be discretised into a series sum by reversing the above arguments.

1.5 The Over-Relaxed Gauss-Seidel Algorithm

The *Over-Relaxed Gauss-Seidel Algorithm* is given by

$$u_R^{k+1}(i,j) = (1+\omega)u_{G-S}^{k+1}(i,j) - \omega u_R^n(i,j) \tag{9}$$

where $u_{G-S}^n(i,j)$ is the usual Gauss-Seidel sequence entry. Here $\omega = 0$ corresponds to the usual Gauss-Seidel algorithm.

2 Experimental Method

2.1 The Electrostatic Potential

A new `class` was written called `Potential` to represent an n×n matrix as an n×n array such that objects declared in this class could be arguments of any new functions to be written.

A `void` function `initial_data` was then written to set up the initial data, as a function of a and b, the variables which define the lower left corner of the inner pole. An `if` − `else` `if` − `else` loop was used to specify the initial data for the three cases of the outer boundary, the inner pole, and all other points.

The `bool` function `update_gauss_seidel` was modified to implement the Gauss-Seidel algorithm for the Poisson equation with no sources.

This code was then iterated using a `do` − `while` loop to converge the electrostatic potential to a specified accuracy.

Graphs of the potential were then plotted as an `splot` using `gnuplot`.

2.2 The Electric Field

A `class` function was written to calculate the x- and y-components of the electric field $\vec{E}$. A double nested `for` loop was used to calculate the gradient by iterating over all points (i,j) in the grid, where the derivative is taken in the opposite direction for points on the boundaries $i = n-1$ or $j = n-1$. The x- and y-components of the electric field were also plotted using `gnuplot`. In addition, a graph of the x-component of the electric field versus y was plotted at $x = 1/2$.

A `double` function was then written to calculate the integral

$$L = \int_0^1 \int_0^1 \vec{E} \cdot \vec{E} \, \mathrm{d}x \, \mathrm{d}y \tag{10}$$

where for the discrete system $\mathrm{d}x = 1/n$ and $\mathrm{d}y = 1/n$. This was done by using a double nested `for` loop to iterate over all (i,j) which added together the squares of both components of the electric field.

Moreover, a `do` − `while` loop was inserted in the `main` function to calculate L for the range of values of b

$$0.05 \leq b \leq 0.95$$

A graph of L versus b was then plotted for this range.

2.3 The Over-Relaxed Gauss-Seidel Algorithm

Finally, the `update_gauss_seidel` function was edited to be a function of the relaxation parameter ω and allow it to implement the over-relaxed Gauss-Seidel algorithm. Another `for` loop was added to the `main` function to repeat the calculation of the electrostatic potential for the range of values of ω

$$0.00 \leq \omega \leq 1.00$$

A graph of the number of iterations required versus the over-relaxation parameter was then plotted.

Furthermore, code was added to the `main` function to read the arguments `integrate` and `relax` from the command line. Two sets of `if` loops were used to distinguish between the two cases, in order to perform one form of calculation or the other, to save on computational expense.

3 Results & Analysis

3.1 The Electrostatic Potential

The following graphs of the electrostatic potential were plotted for values of $a = 0.2$ and $b = 0.3$ and $a = 0.5$ and $b = 0.5$ respectively

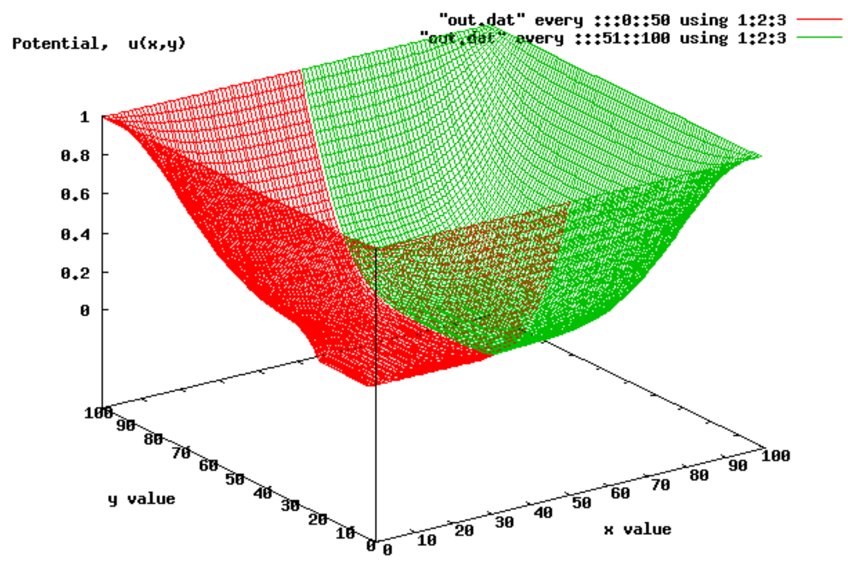

Figure 1: Electrostatic potential for $a = 0.2$ and $b = 0.2$

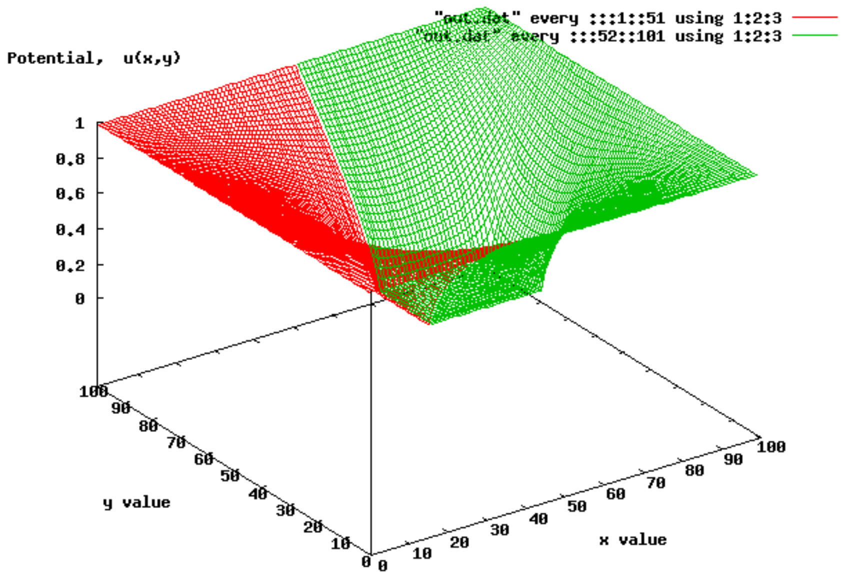

Figure 2: Electrostatic potential for $a = 0.5$ and $b = 0.5$

3.2 The Electric Field

The following graphs were obtained for the x- and y-components of the electric field $\vec{E}$

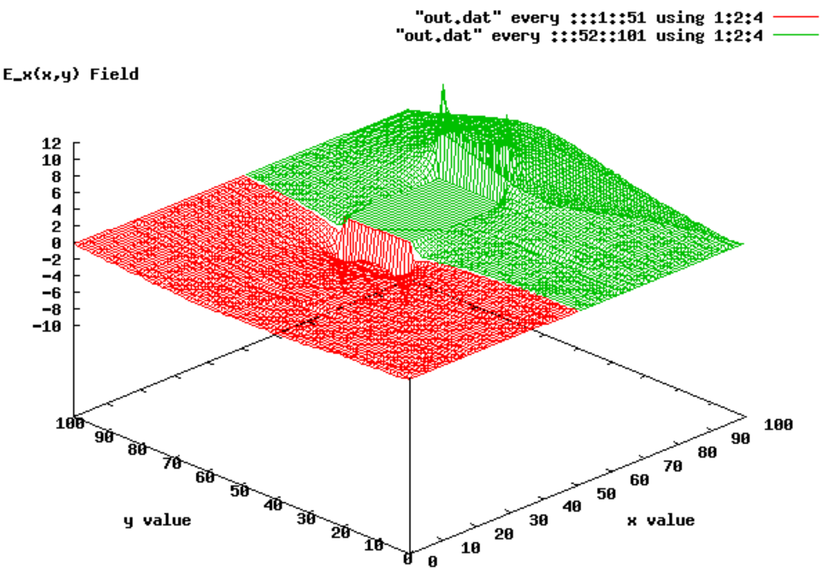

Figure 3: X-component of the Electric Field

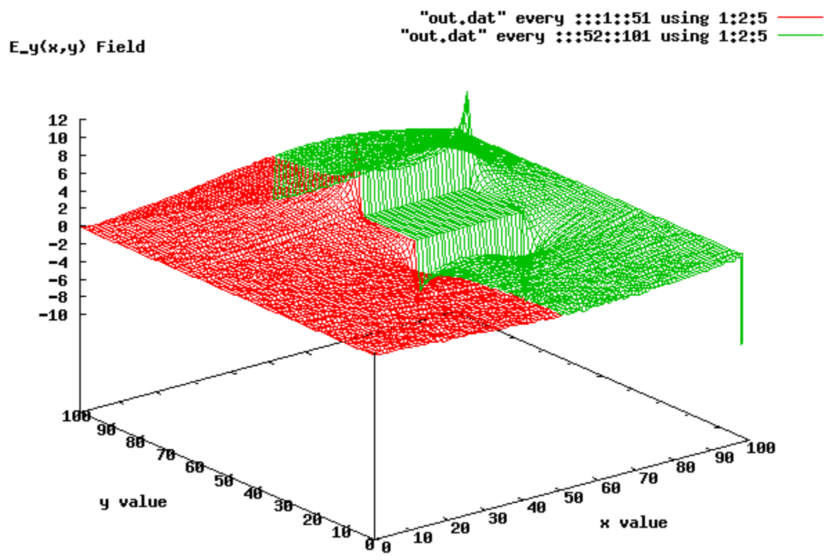

Figure 4: Y-component of the Electric Field

A graph of the x-component of the electric field versus y was then plotted at $x = 1/2$ for values of $a = 0.10$ and $b = 0.10$ to give

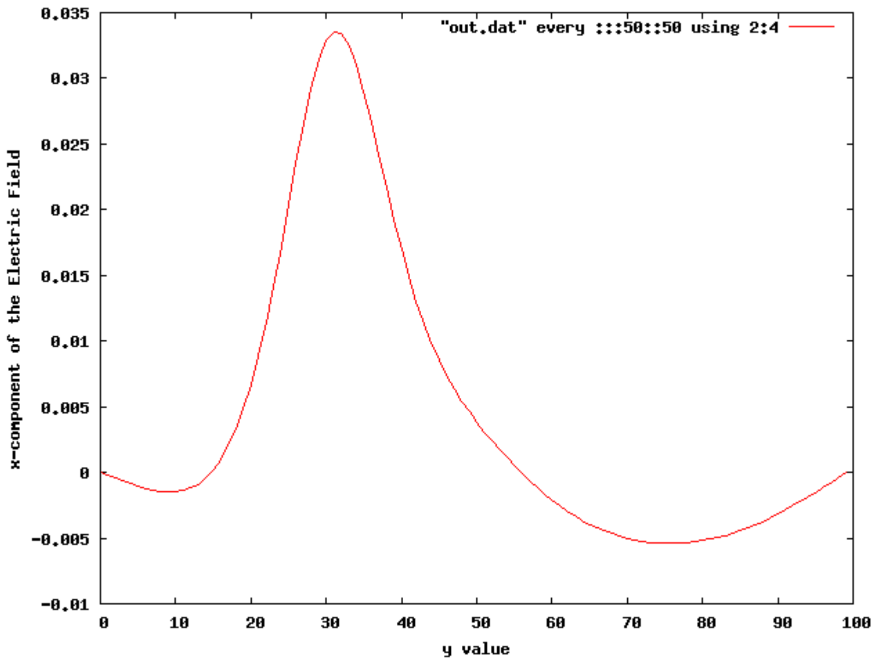

Figure 5: X-component of the Electric Field versus y at $x = 1/2$

In addition, a graph of the integral L versus the value of b was plotted

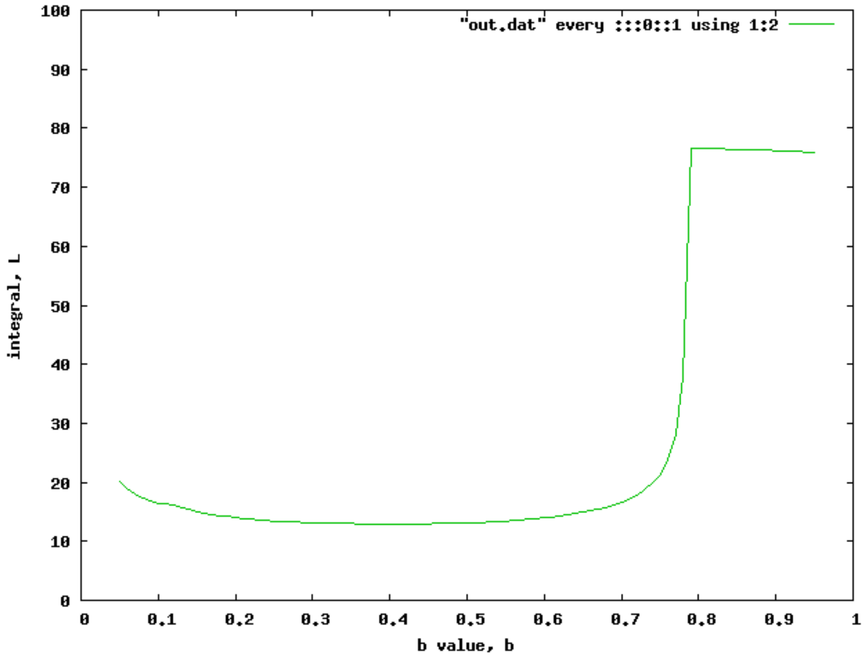

Figure 6: Integral L versus b

3.3 The Over-Relaxed Gauss-Seidel Algorithm

It was found that the electrostatic potential converged in the least number of steps for a relaxation parameter of $\omega = 0.90$, which required 80 iterations. A graph of the dependence of the number of iterations m versus the relaxation parameter ω was plotted

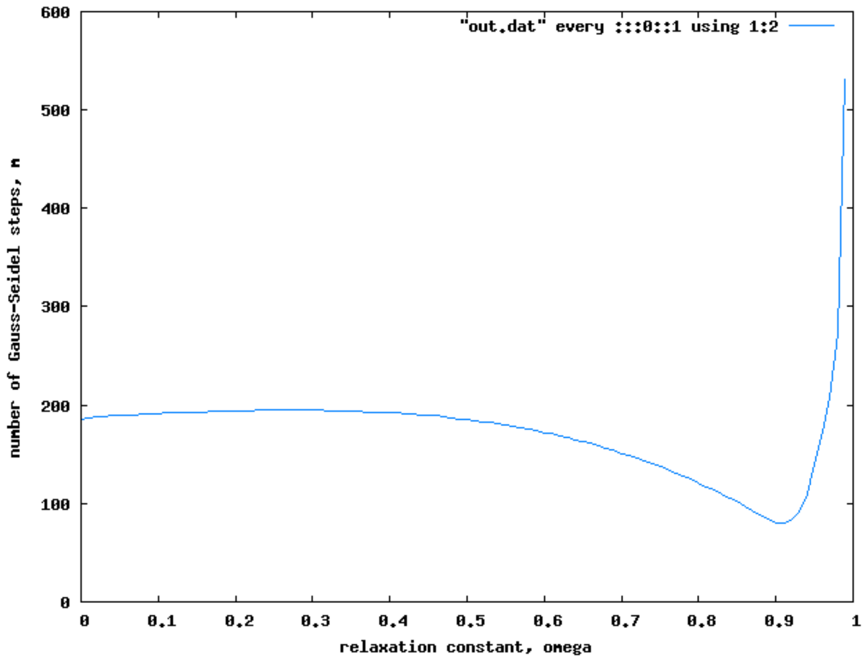

Figure 7: Number of Iterations m versus Relaxation Parameter ω

4 Conclusions

It was found that the Gauss-Seidel algorithm was able to be used to solve Poisson's equation for the electrostatic potential everywhere inside a coaxial tube with C++ programing code.

Furthermore, C++ code was able to be used to calculate the electric field at all points between the tubes from this electrostatic potential. It is clear that the graph obtained agrees with Gauss' law for electric charge, as the electric field is highest at the boundary of the inner tube, corresponding to a charge concentration, and goes to zero away from the inner tube as there are no charges present.

The dependence of the integral L was almost constant with the value of b, except that it has a jump discontinuity at $b = 0.7$, as this corresponds to the inner tube touching the edge of the outer tube.

Finally, it was seen that the over-relaxed Gauss-Seidel algorithm may effectively be used to calculate the electrostatic potential in a shorter amount of iterations than the usual Gauss-Seidel algorithm. For a range of values of the relaxation parameter ω the minimum number of iterations required was found to be $m = 80$, which corresponded to a value for the relaxation parameter of $\omega = 0.90$.