Computational Analysis of Electric Potentials
T. Dunn
Physics 193
date 4/5/2004

1 Introduction

I created a program in C++ to solve for the electric potentials inside of a closed box by means of the relaxation method on lattices. Then I created Perl script to solve for the electric fields of each simulation. More specifically I simulated the potentials and electric field of a box with varying set potentials on the perimeter, a box of set potential inside another box, and a circle inside another box. For the later two a third program was created for calculating the capacitance of each. All of these models can be thought of as two dimensional slices of a infinitely long three dimensional duct.

2 Theory: Potential Inside a Closed Box

For any problem pertaining to electric fields and potentials we know that Laplace’s equation must be satisfied. For this problem, this means that any equation satisfying our potential boxes must also satisfy:

\[ \nabla^2 V(x, y, z) = \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = -4\pi k \rho \]  \quad (1)

and since there is no charge inside out box:

\[ \nabla^2 V_{dv} = -4\pi k \rho = 0 \]  \quad (2)

we also know, due to the nature of our problem, that

\[ \frac{\partial^2 V}{\partial z^2} = 0 \]  \quad (3)

since our infinitely long duct is symmetric along in the z direction. Which gives us the equation:

\[ \nabla^2 V(x, y) = \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = 0 \]  \quad (4)
Keeping this in mind, we can approximate the values of the points surrounding our central point by Taylor expanding the partial derivatives in each directions as:

\[ V_N = V_0 + (\Delta x) \frac{\partial V(x, y)}{\partial x} + \frac{(\Delta x)^2 \partial^2 V}{2 \partial x^2} \]  
\[ V_S = V_0 - (\Delta x) \frac{\partial V(x, y)}{\partial x} + \frac{(\Delta x)^2 \partial^2 V}{2 \partial x^2} \]  
\[ V_W = V_0 + (\Delta y) \frac{\partial V(x, y)}{\partial y} + \frac{(\Delta y)^2 \partial^2 V}{2 \partial y^2} \]  
\[ V_E = V_0 - (\Delta y) \frac{\partial V(x, y)}{\partial y} + \frac{(\Delta y)^2 \partial^2 V}{2 \partial y^2} \]

and if we take the sum of these 4 we get:

\[ \sum_{N,S,E,W} V_i = 4V_0 + (\Delta y)^2 \frac{\partial^2 V}{\partial y^2} + (\Delta x)^2 \frac{\partial^2 V}{\partial x^2} \]  
\[ \sum_{N,S,E,W} V_i = 4V_0 + (\Delta x)^2 \nabla^2 V(x, y) \]

Which can be re-written as:

\[ V_0 = \frac{1}{4} \sum_{N,S,E,W} V_i + (\Delta x)^2 \nabla^2 V(x, y) \]

and since we know from Laplace’s equations:

\[ \nabla^2 V(x, y) = 0 \]

we get a final equation for each point in our box which says that the voltage at any point is equal to the average of the four adjacent points.

\[ V_0 = \frac{1}{4} \sum_{N,S,E,W} V_i \]

### 3 Simulation

By using the method described in equation 13 for finding the voltage at a given point based on it’s neighbors, and applying it to all the points in
a lattice with set potentials in certain areas, and iterate it over the lattice several times, eventually we will approach a steady set of values for the lattice. This method for solving lattices is known as the relaxation method.

Using the relaxation method I created a program in C++ to calculate the potentials inside a box of side length 1 with a box of length .2 inside. The following code excerpt shows the method for calculating the voltage values of the lattice:

```cpp
while ( e>=E){
    r = I/2;
    // resets error value for new sweep
    if (r==0){
        e = 0;
    }
    I++;
    for (z=2 ; z< N; z++){
        for (z1=2; z1< N; z1++){
            temp = V[z][z1];
            if (((z+z1)%2 == r){
                V[z][z1] = .25*(V[z+1][z1]+V[z-1][z1]
                +V[z][z1+1]+V[z][z1-1]);
            }
            temp = fabs(temp - V[z][z1]);
            if (temp > e){
                e = temp;
            }
        }
    }
}
```

The reason why my 'for' statements are "for (z=2 ; z< N; z++)" instead of the "for (z=1; z< (N-1); z++)" is because I set my lattice to be a (N+2)x(N+2) in order to make certain calculations easier later on in the program. Also it should be noted that E is my maximum allowed change for any one lattice value before the loop to the new lattice value after the loop has finished. Once the loop had successfully ran without getting and error values greater
than E the program produce output in a format capable of being read in
gnuplot's splot function.

To further analyze the final data from my simulations i created the fol-
lowing Perl script to take the data output produced by my c++ program and
calculate the electric field vectors at each point in my lattice and put them
into a file that could then be loaded into gnuplot.

#!/usr/bin/perl
while(<>){
    if(/\s+\(/) {next; }
    if(/(\d+)+/s+([\d.\-]++)/) {
        $x = $1;
        $y = $2;
        $v = $3;
        $V[$x][$y] = $v;
    }
}

for( $j=0; $j<$y; $j++) {
    for( $i=0; $i<$x; $i++) {
        $Ex = $V[$i+1][$j] - $V[$i][$j];
        $Ey = $V[$i+1][$j] - $V[$i][$j];
        $E = sqrt($Ex*$Ex + $Ey*$Ey);
        print " # $Ex $Ey at $i $j\n";
        if ($E == 0)){
            $Exhat = $Ex/$E;
            $Eyhat = $Ey/$E;
            $iEx = $i + $Exhat;
            $jEy = $j + $Eyhat;
            print "set arrow from $i,$j,2.5 to $iEx,$jEy,2.5\n";
        } else {
            print "# |E| = 0 at $x $y\n";
        }
    }
}
4 Data

I ran my simulation for lattice spacing of .1 (shown in figure 1) and .05 (shown in figure 2). In order to test the validity of these graphs though I compared a Δx equal to .0333 to a graph of an analytically solved box of the same resolution and overlaid them. The results are shown in figure 3. As

Figure 1: Shows the potential values of a box with side potentials 5 Volts and 10 Volts and lattice spacing of .1. Also shows the equal voltage contours and the electric field vectors as well. The minimum allowed error for each loop in the program is 0.0001 Volts

this figure shows the two are in excellent agreement at this size.
Figure 2: Shows the potential values of a box with side potentials 5 Volts and 10 Volts and lattice spacing of 0.05. Also shows the equal voltage contours and the electric field vectors as well. The minimum allowed error for each loop in the program is 0.0001 Volts
Figure 3: Overlays the results from a $\Delta x = 0.0333$ (accualcompare.dat) to that of the actual solution of the same $\Delta x$ (exact.dat). The edge potentials are 10 Volts and 5 Volts. The minimum allowed error for each loop in the program is 0.0001 Volts.
5 Theory: Capacitance of a Box Within a Box

In order to calculate the capacitance per unit length of the box within a box potential problem we can use the relationship:

\[ C = \frac{Q}{V} \]  

(14)

or

\[ \frac{C}{L} = \frac{\sigma}{V} \]  

(15)

But in order to solve for this we must find \( \sigma \). We can solve for \( \sigma \) if we approximate each surface as an plane of infinite size having an charge density \( \sigma \) and electric field \( \vec{E} \). If we make this approximation we can calculate \( \sigma \) by using the equation:

\[ \vec{E} = \frac{\sigma}{2\varepsilon_0} \]  

(16)

In order to solve do this we must first calculate the electric field vector \( \vec{E} \). We can find the electric field vector \( \vec{E} \) on the edge of the box by solving for the partial derivatives of the Voltage, \( \frac{\partial V(x,y)}{\partial x} \) and \( \frac{\partial V(x,y)}{\partial y} \), and finding the resulting vector. Since we have not solved for an analytical solution to our Voltage differential we cannot calculate the partial derivatives of our Voltage equation analytically either. So instead we must approximate our partial derivatives. The approximation of the partial derivatives is:

\[ \partial x \approx V(x + \Delta x, y) - V(x - \Delta x, y) \]  

(17)

\[ \partial y \approx V(x, y - \Delta y) - V(x, y - \Delta y) \]  

(18)

by taking the difference between the nearest neighbors.

6 Simulation

In order to calculate the capacitance per unit length of my boxes I first needed to find the charge density at each point on the edges of my boxes. So in order to do that I added a subroutine to my code that would calculate the magnitude of the electric field at each point by taking the approximate partial derivatives as outlined earlier.
Q = 0.0;
Qi = 0.0;
Ii = 0;
I = 0;
for (z=1; z<N+1; z++){
    for (z1=1; z1<N+1; z1++){
        if (V[z][z1] == 10.0){
            VE[z][z1] = fabs(V[z-1][z1] - V[z+1][z1])
                + fabs(V[z][z1-1] - V[z][z1+1]);
            Q += fabs(V[z-1][z1] - V[z+1][z1]);
            Qi += fabs(V[z][z1-1] - V[z][z1+1]);
            Ii += 1;
        }
        else if (V[z][z1] == 5.0){
            VE[z][z1] = fabs(V[z-1][z1] - V[z+1][z1])
                + fabs(V[z][z1-1] - V[z][z1+1]);
            Qi += fabs(V[z-1][z1] - V[z+1][z1]);
            Qi += fabs(V[z][z1-1] - V[z][z1+1]);
            Ii += 1;
        }
    }
}
cout << "#Iterations: " << I << endl;
// find the average capacitence per unit length
cout << "#Charge outer box: " << Q << " charge inner box: " << Qi << endl;
Qi = Qi/5;
Q = Q/5;
cout << "#Capacitance outside: " << Q << " Capacitance inside: " << Qi << endl;

   In this program Q and I correspond to the outer box while Qi and Ii correspond to the inner box.

7 Data

If we look at the graphs of the potentials from my simulations we see potentials that seem to behave as we would expect them too. That is with Potentials that decrease from the outside box to the inside box as shown in figures 4 and 5.
Figure 4: Shows the potential values of a box within a box with lattice spacing of .1. Also shows the equal voltage contours and the electric field vectors as well. The outside potential is 10 Volts and the inside potential is 5 Volts. The minimum allowed error for each loop in the program is 0.0001 Volts

From the accuracy of the program in solving for the potentials of the first box it is reasonable to assume that the voltages inside these simulations are also accurate as well. So from that it is reasonable to assume that we should be able to calculate the capacitance per unit length of our duct from this data. As a check for this we should find that the capacitance per unit length of the inside box and the outside box are equal as well as the charge per unit length of the boxes. In fact one would assume that the smaller the $\Delta x$ of the lattice the more accurate, and therefore the closer the two would be. And for the first few simulations this seems to be the case. However as we get finer and finer lattices the separation between the two values ceases to get smaller and starts to get much greater. This anomaly can be seen in table 1 and graph 6
Figure 5: Shows the potential values of a box within a box with lattice spacing of .05. Also shows the equal voltage contours and the electric field vectors as well. The outside potential is 10 Volts and the inside potential is 5 Volts. The minimum allowed error for each loop in the program is 0.0001 Volts.

But what could be causing this to happen? Why would it work on less finely meshed lattices and not on more finely meshed ones. Well the only thing that I had yet to take into consideration was the value of my maximum allowed error. And logically this makes sense, that a finer lattice would require a smaller allowed error. I had noticed in figure 7 that my answers seemed to start diverging around $\Delta x = .01$. So I decided to run one of my earlier simulations, where my values matched up with the exact value very closely, but change the error value and see if I got similar results. Thankfully this is exactly what I saw as shown in figure 8.

So finally had found the cause of my anomalies. My code hadn’t been outputting the bad data, it had just been outputting data too early. Since the errors had first started to show up around $\Delta x = .01$ with a maximum
Figure 6: Shows a profile view of a lattice of $\Delta x = .0025$ (profile1.dat) and of the actual solution with $\Delta x = .0333$ (laplaceexact.dat). Notice how the simulation data has a greater concavity than the actual solution.
Figure 7: Compairs the calculated capacitence value of the inside box (zap.dat u 1:3) and the outside box (zap.dat) at various Δx values
Figure 8: Shows a profile view of a lattice of $\Delta x = .0333$ (profile2.dat) and that of the actual solution with $\Delta x = .0333$ (laplaceexact.dat). The error value was set $E = .1$ instead of the usual .0001.
error value $E = .0001$ it seemed that in order to get accurate data i would need $E$ to satisfy this relationship:

$$ E \propto (\Delta x)^2 $$  \hspace{1cm} (19)

This is not an exact relationship between $E$ and $\Delta x$ but from trial and error this seems to work within the scope of my data. With this in mind i recreated my previous table of capacitance per unit length values and got the following results shown in table 2. From this we can see that the capacitance is converging nicely for both the inside box and outside box at a capacitance per unit length of about $6.7e^{-11}$ Farads/meter. But why is the inside capacitance per unit length not converging as well as the outside capacitance per unit length? By looking at a a graph of the charge density along the edges of the inside box (figure 10)and the outside box (figure 9) we can see why. It is because the inside box has a singularity at it’s corners. In fact if the simulation is correct we would see that the singularity behaves as

$$ \sigma \sim x^{-1/3} $$ \hspace{1cm} (20)

or

$$ \sigma = ax^{-1/3} $$ \hspace{1cm} (21)

where $x$ is the distance from the corner. This also means that if we take the
Figure 9: Shows a profile view of a lattice of $\Delta x = .0025$ and of the actual solution with $\Delta x = .0333$. Notice how the simulation data has a greater concavity than the actual solution.
Figure 10: Shows a profile view of a lattice of $\Delta x = .0025$ and of the actual solution with $\Delta x = .0333$. Notice how the simulation data has a greater concavity than the actual solution.
<table>
<thead>
<tr>
<th>$\Delta x$</th>
<th>Outside Capacitance</th>
<th>Inside Capacitance</th>
</tr>
</thead>
<tbody>
<tr>
<td>.1</td>
<td>5.80772e-11</td>
<td>5.80688e-11</td>
</tr>
<tr>
<td>.05</td>
<td>6.21734e-11</td>
<td>6.21239e-11</td>
</tr>
<tr>
<td>.02</td>
<td>6.51135e-11</td>
<td>6.47389e-11</td>
</tr>
<tr>
<td>.01</td>
<td>6.59595e-11</td>
<td>6.58064e-11</td>
</tr>
<tr>
<td>.006667</td>
<td>6.64166e-11</td>
<td>6.63547e-11</td>
</tr>
<tr>
<td>.005</td>
<td>6.64166e-11</td>
<td>6.63547e-11</td>
</tr>
<tr>
<td>.004</td>
<td>6.65436e-11</td>
<td>6.64468e-11</td>
</tr>
<tr>
<td>.003333</td>
<td>6.6641e-11</td>
<td>6.65015e-11</td>
</tr>
</tbody>
</table>

Table 2: Compares the calculated capacitance per unit length value for the outside box and inside box for different sized lattice spacing. The maximum allowed error was calculated using equation 19. All capacitance are measured in Farads.

natural log of both sides of this we end up with

\[
\ln(\sigma) = \ln(a) + \ln(x^{-1/3})
\]

\[
= \ln(a) - \frac{1}{3}\ln(x)
\]

So to verify this we can take the values for potential and position of one side of our inside box and plot them using the equation

\[
\ln(y) = \ln(x)
\]

where $y$ is the potential and $x$ is the position, we should get a line of slope $-\frac{1}{3}$ for the points very close to one of the corners. This is almost exactly what we see when when looking at data from a lattice of $\Delta x = .001$ as shown in 11.

This singularity at the corners of the inside should account for the difference between the charge density of the inner box and that of the outer box for $\Delta x \geq .02$.

If we apply the same analysis to the corners of the outer box we find that charge density should decrease linearly as we get very close to the corners. And this is exactly what we see as shown in figure 12.
Figure 11: Plots $\ln(y) = \ln(x)$, where $y$ is the potential and $x$ is the position along one edge of the inside box with one of the corners set as $x = 0$. Also plots the line $\ln(y) = \ln(0.077x^{-0.3})$ for comparison.
Figure 12: Plots the first 33 points of figure 9 and a line $f(x) = 0.00117088x - 0.000981205$
8 Analysis

Both the box with in a box and the box with varying side potentials showed output that was expected. The voltages gradually changed from one set potential to the other without any "bumps" or anomalies. The fact that the simulated potential and the actual potential agreed so well shows the validity of the simulation.

For calculating the capacitance of the box within a box I got values for the inside box and the outside box that were very close to each other when the proper $\Delta x$ and maximum allowed error value were set. Further evidence for the validity of the calculated capacitance value can be shown from the analysis of the charge density at the corners where the charge density behaved as we would expect from the theory.