2D SOLVER CODE DESCRIPTION: VERSION 1

Part 1: Introduction

This document provides a brief description of the code developed for solving the electrostatic EHD equations in 2D. These are the following:

$$\nabla \cdot \vec{E} = \rho/\epsilon \tag{1}$$

$$\nabla \vec{j} = 0 \tag{2}$$

where

$$\nabla \times \vec{E} = 0 \tag{3}$$

$$\vec{j} = \rho\mu\vec{E} \tag{4}$$

and thus the first two equations can be reduced to

$$\nabla^2\phi = \rho/\epsilon \tag{5}$$

$$\nabla\rho \cdot \nabla\phi = \rho^2/\epsilon \tag{6}$$

The first equation is simply Poisson's and is elliptic. The second equation is a nonlinear hyperbolic equation, which is also first order. In Cartesian coordinates, these equations become the following:

$$\phi_{xx} + \phi_{yy} = \rho/\epsilon \tag{7}$$

$$\phi_x\rho_x + \phi_y\rho_y = \rho^2/\epsilon \tag{8}$$

Date: 10 September 2013.

Thus, we have a a nonlinear system of coupled equations in ϕ and ρ, the potential and space charge, respectively.

Part 2: General Algorithm

To solve these equations, we modify and implement the algorithm of Davis and Hoburg 1983, an algorithm also used by Admaniak in several of his EHD studies. The algorithm has the following general structure:

(1) Pick an initial space charge concentration at the emitter.
(2) Solve Laplace's equation for an initial electric field.

$$\phi_{xx} + \phi_{yy} = 0 \tag{9}$$

(3) Solve for space charge distribution using the first order hyperbolic equation derived above and the Method of Characteristics (MOC).
(4) Solve for the the new electric field based on Poisson's equation.
(5) Repeat (3) and (4) until solution is sufficiently self-consistent.
(6) Check to see if electric field at emitter matches Peek criteria.
(7) Iterate on assumed charge distribution.

The code that has been constructed that implements this algorithm was programmed in MATLAB and uses a variety of build-in functions as well as customized numerical processes. In the subsequent sections, I will discuss each step of the general algorithm and describe how it was implemented for our use.

Step 1: Boundary Condition

Boundary conditions for Laplace/Poisson's equations are straightforward. A positive potential corresponding to the total desired potential difference across the electrodes is applied to the emitter surface and a 0 potential, or ground, is applied to the collecting electrode. Normal component of the electric fields are assumed to be 0 at the domain boundaries.

The boundary condition for the space charge distribution is not as straightforward. Currently, the code assumes a uniform charge density at the emitter (i.e. $\rho = C$ on the surface of the emitter). In reality, this is not completely accurate, as will be discussed later. Further improvements to the code can expand this boundary condition capability.

Step 2: Solving Laplace's Equation

MATLABs built-in PDE solver is used for this step, specifically the *pdenonlin* function. This function takes in several arguments, including those related to the mesh geometry, which is automatically generated using *initmesh*. A boundary condition function is also used to specify in the format required by the function. This function is currently called *pdebound_1d*, where the 1d refers to the fact that the geometry has only 1 emitter/collector pair.

In general, MATLAB's PDE solver can only solve elliptic equations of a particular format, which is specified within their documentation. In general, a will be set to 0, c will be set to 1, and f will be set to 0 for this initial solution. *pdenonlin* returns the potential field. The electric field can be solved for using the function *pdegrad*, which will be relevant for force calculations.

Step 3: Using the MOC

Now that we have an initial electric field to work with, it is now possible to apply MOC to determine the space charge distribution within the solution domain. The equation specified by charge conservation, while nonlinear, is first order and has the form of a "semilinear" equation, which in general is the following:

$$a(x,y)u_x + b(x,y)u_y = c(x,y,u) \tag{10}$$

where a and b of functions of space, only, and c is a function of both space and the solution, where the solution, u for our case is the charge density. As such, a is the x-component of the electric field, b is the y-component of the electric field, and c is $-\rho^2/\epsilon$.

As can be shown, if we think of this problem as being time-dependent, the solution will propagate in time in a very convenient manner, i.e.

$$\frac{dx}{dt} = a(x,y) \tag{11}$$

$$\frac{dy}{dt} = b(x,y) \tag{12}$$

$$\frac{du}{dt} = c(x,y) \tag{13}$$

or for our specific problem, these equations are

$$\frac{dx}{dt} = \mu E_x(x,y) \tag{14}$$

$$\frac{dy}{dt} = \mu E_y(x, y) \tag{15}$$

$$\frac{du}{dt} = -\mu\rho^2/\epsilon \tag{16}$$

where the ion mobility has been substituted back in to provide velocity terms on the right hand sides for the first two equations. Thus, our nonlinear PDE has been converted to 3 ODEs. The first two show how the solution develops spatially in time and the third shows how this solution changes along the defined trajectory. Unfortunately, however, there is no simple analytical solution to this system as there is no concise expression for the electric field at an arbitrary point. Luckily, the third ODE is easily integrated analytically to the following form, as presented in Davis and Hoburg 1983:

$$\rho = [1/\rho_0 + (\mu/\epsilon)t]^{-1} \tag{17}$$

where ρ_0 is the initial charge density at the emitter. Thus, all that we need is the electric field, which is provided directly from the MATLAB solution of Laplace's equation.

Thus, we have all the information necessary to solve for a space charge distribution. First, starting points are generated on the emitter surface. From these starting points, electric field lines are traced from the electric field information in time (*getRho*). These are also known as the characteristic lines (hence method of characteristics). Once the characteristic lines have reached the surface of the collector, equation 17 is used to determine the space charge along the line. Given that we have to take discrete steps in space (and as such time) we have a discrete number of space charge points. These locations as well as corresponding space charges are placed in a storage variable for later use. To obtain more points, I also lay down a uniform grid over the entire solution domain at which end points are generated and characteristic lines are traced backwards in time (*getRhoGrid*). In total, I end up generating characteristic lines that look like the following, shown below.

In addition, values of zero space charge are stored along the edges of the boundary, although the legitimacy of this approach is questionable. This is by far the slowest part of the code due to the interpolation required at each step for the electric field. I built a custom interpolation function *tri2gridCKG* that uses barycentric (triangular) interpolation at the point specified, but his also requires finding the triangle that this point resides in. Future work needs to focus on speeding up this interpolation. This can be done by redesigning the code for optimizing speed or by increasing the step size taken along each characteristic line as I have hardcoded a certain spatial

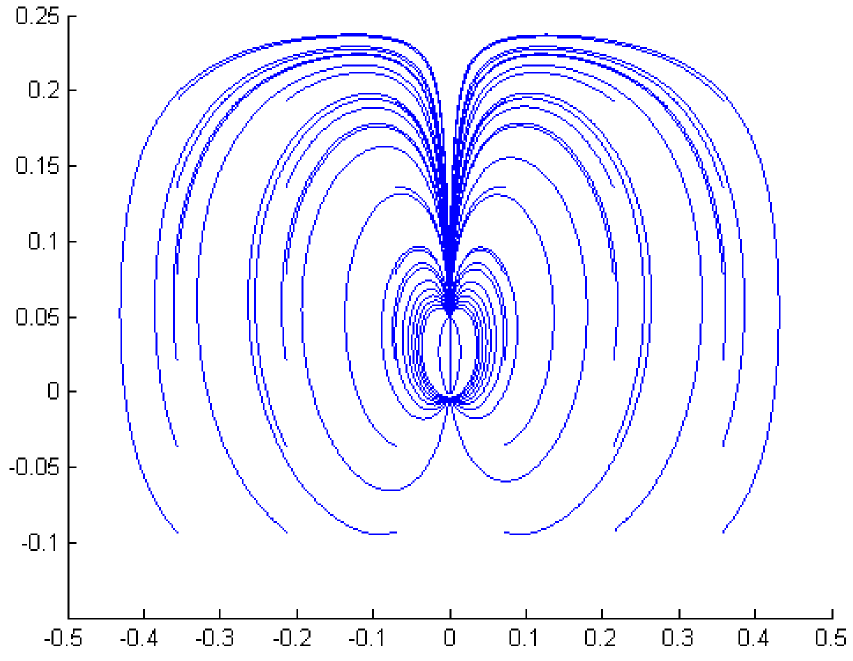

FIGURE 1. Generated characteristic lines.

step size, which is quite small. Larger steps might be possible in regions far away from the emitter or collector.

Also notice that the entire domain is not covered by the characteristic lines, thus the need for zero values on the domain in order to bound the problem. Future work can focus on generating a more uniform and full distribution of space charge in the domain.

From here, it is necessary to generate a space charge value at each triangular mesh for use in solving Poisson's equation (discussed next). This is done so in the *interpf* function, where *TriScatteredInterp* is used to perform this interpolation as we do not have a uniform grid from which to sample. A sample space charge density mesh is shown below, which has been generated using this interpolation scheme.

As you can see, there is still some local fluctuations in the charge distribution, implying that the resolution of the determined space charge needs to be increased. This should be looked at in future work as the grid I assigned is still very coarse.

Step 4: Solving Poisson's Equation

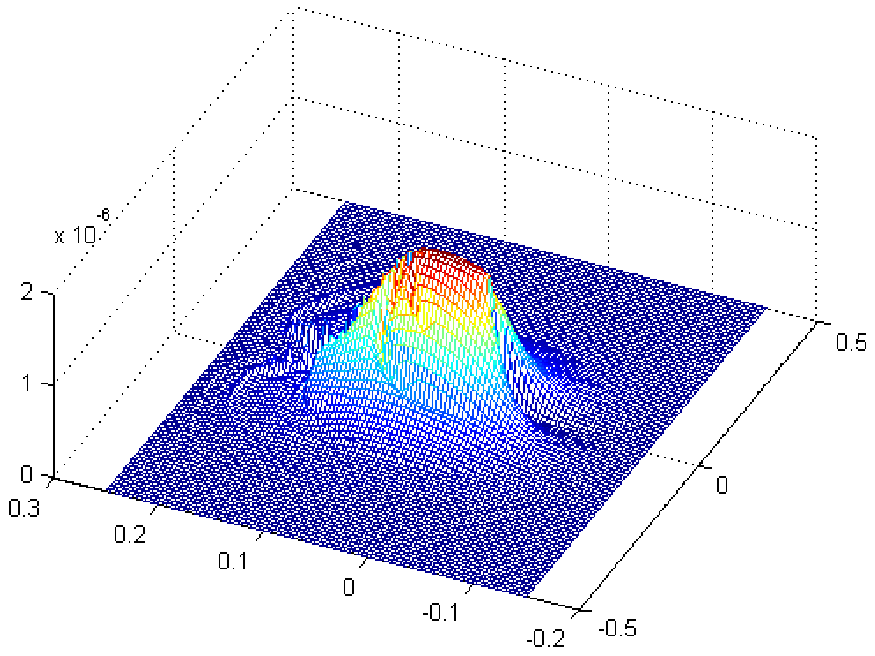

FIGURE 2. Generated space charge density.

This step is identical to Step 2 in which Laplace's equation was solved, with the only exception being that space charge is taken into account. Keep in mind that the space charge is interpolated at the centroids of each triangle that make up the mesh. Thus, f is a $1 \times n_t$ vector, where n_t is the number of triangles in the mesh.

Step 5: Iterating for Self-Consistency

After performing Step 4, we now have a new electric field from which to calculate the space charge distribution. Thus, Steps 3 and 4 can be repeated until convergence of the solution (although I doubt convergence is ever guaranteed using this type of iteration, this could be worth looking into). Right now, the solution is considered to converge given the following criteria is met:

$$\frac{||\phi_i - \phi_{i-1}||_2}{||\phi_i||_2} < 0.001 \qquad (18)$$

that is the L2 norm of the residual between successive solutions divided by the L2 norm of the solution vector is less than 0.001. This is achieved after about 5-8 iterations. This convergence criteria can probably be improved, but seems to be sufficient here as the current from the emitter is generally within 1-3% of the current into the collector, thus showing that charge conservation is approximately satisfied.

Step 6: Checking Peek's Criteria

Once the solution has converged to self-consistency (or the inner loop has converged using the terminology from Davis and Hoburg 1983), it is necessary to check if Peek's criteria has been met at the surface of the emitter. In reality, the plasma region has a non-negligible thickness such that assigning space charge to the emitter surface is not strictly correct, but most modeling studies I have come across neglect this plasma region. Most studies performed by Admaniak apply what is called the Kaptzov hypothesis, which states that after corona ignition, the electric field at the surface of the emitter will stay at the critical electric field value required for ignition, which is governed roughly by Peek's law (which are mostly empirical in nature).

Right now, I am determining the average electric field around the emitter and comparing this value to the value determined from Peek's law with the following form:

$$E_{cr} = 30\delta(1 + 0.301/\sqrt{\delta r}) \tag{19}$$

This equation, however, is for two parallel wires and is not strictly correct given that the wires are not of equal diameter. Application of Peek's criteria/Kaptzov hypothesis needs to be improved in the code. Current is still being largely under predicted using the current methodology (by a factor of 4 or 5). In addition, a nonuniform charge density on the emitter could be implemented so that Peek's law is satisfied at select points. This, however, will add computational complexity.

Step 7: Iterate on Initial Space Charge

Currently, Newton-Raphson is used to iterate the initial space charge and convergence to within 0.1% is achieved usually within 3 or 4 iterations (known as the outer loop). To reduce the number of inner-loop iterations, Laplace's equation is only solved for the first inner-loop iteration and the solution resulting from that initial space charge value is used to initialize all future outer-loop iterations. This may cause issues in the future if the solution is not very well-behaved.

Part 3: Sample Solution

Here I provide a sample solution based on the following parameters:

$$d = 0.05\ m \tag{20}$$

$$V = 15\ kV \tag{21}$$

where the emitter and collector are sized based on the experimental setup. The 1D F/P ratio is 16.667. The numerically determined value is 18.44. Current at the emitter is found to be 0.3798 A/m and at the collector 0.3887 A/m, which is about a 2% difference. The critical electric field estimated by Peek's law is 1.2085×10^7 V/m, where the average electric field at the emitter has converged to almost exactly that.

As a comparison, if the electric field is not converged according to space charge, current differs between the emitter and collector by about 30%. In addition, F/P is much closer to the 1D case with a value of 16.93. Overall, however, the F/P predicted by this code is not all that different from the 1D case, further supporting the idea that 2D effects have little impact and the thrust to power ratio is quite robust to other factors.

Part 4: Intended Future Work

My intention is to use this code to first compare numerical results to the experimental ones obtained by Kento. After that, we should be able to apply this code to the issue of thrust density to try and determine more favorable thrust configurations that maximize force per unit area.

In the short term, the code certainly needs to be improved, particularly regarding the outer iteration loop (i.e. space charge). If the Katpzov hypothesis ends up being an issue, we could always iterate the space charge until the current matches experimental results, but if possible I would like to avoid this to remove any bias associated with our experimental setup on the numerical results.