Design of an Electron Gun using Computer Optimization

B. M. Lewis [*] H. T. Tran[†]
Department of Mathematics
Center for Research in Scientific Computation
North Carolina State University
Raleigh, NC 27695

M. E. Read [‡] R. L. Ives [§]
Calabazas Creek Research, Inc.
Saratoga, CA 95070

Abstract

This paper considers an optimization technique in which the objective is attained via alterations to the physical geometry of the system. This optimization framework, to be considered in the context of electron guns, is known as optimal shape design. Optimal shape design has been used in a number of applications including wing design, magnetic tape design, and nozzle design, among others. In this investigation, we use the methods of shape optimization to design the cathode of an electron gun. The dynamical equations modeling the electron particle path as well as the generalized shape optimization problem will be presented. Illustrative examples of the technique on gun designs that were previously limited to spherical cathodes will be given.

1 Introduction

The electron gun is used in a number of devices, including radar guns, CRTs (television and computer monitors), and TWT amplifiers. The name gun is indicative of the function of the device in that it shoots off a continuous stream of electrons. The amount of emission

[*] bmlewis2@unity.ncsu.edu
[†] tran@control.math.ncsu.edu
[‡] mike@calcreek.com
[§] RLIves@calcreek.com

can be more than 1 billion electrons per second. Along with the advance of 3D visualization software and the ability to prototype new designs for electron guns has come the need to increase a designers ability to quickly alter the components of an electron gun. The process of designing a gun usually involves finding an existing gun, with characteristics similar to those desired in the new gun, and altering it by hand until the new characteristics have been attained. This involves tedious manipulation that would be better left to an optimization routine.

In this work, we perform a feasibility study for using the methods of optimal shape design on an electron gun. Optimal shape design, otherwise known as structural optimization or redesign, is a process by which an engineer or designer can use mathematical optimization methods to design and determine the shape of a structure. The advance of graphical user tools (such as CAD design programs) has resulted in an increase in the ability to visualize structural designs. However, as is often the case, the design of the physical components and layout of a system or composite structure is goal oriented. For instance, the goal when designing a nozzle might be prescribing a certain velocity at the exit. Hence, finding the shape of the structure (for instance, the curvature of a nozzle) to achieve the said goal(s) can be a time intensive task. The goal in optimal shape design problems is to take some of the guess work out of the design process and allow computational routines to alter the geometry in a directed manner. Specifically, if one seeks to attain the desired goal through altering the geometry of a structure and the behavior of the structure can be modeled mathematically, then the goal can be attained using optimization techniques.

Section 2 includes an explanation of a simple electron gun and describes some of the characteristics that an objective function might emphasize. We introduce the general optimal shape design problem in Section 3, and we specify the design problem for the electron gun in Section 4. In the latter section we also present results for our feasibility study.

2 Electron Gun Basic Elements

We begin by describing the basic components and functions of an electron gun. As depicted in Figure 1(a), there are three basic components associated with the electron gun: the cathode, anode, and the focusing electrode. Anode and cathode are commonly used terms in electricity and refer to the positive and negative electrode. The cathode emits the electrons that later will make up the beam. The primary characteristics that the ideal cathode will adhere to are [3]:

1. emits electrons freely without any sort of influence (heating, bombardment, etc.),

2. emits abundantly so as to supply an unlimited current density,

3. electron emission continues unimpaired as long as it is needed,

4. emits electrons uniformly with practically zero initial velocity.

Of course, these characteristics are not within the realm of possibility. First of all, although electrons move freely in conductors like metals, they normally do not leave the metal without some manipulation. In fact, heating and bombardment are the two primary ways in which electrons are emitted through the use of a heating element behind the cathode (termed *thermionic emission*) or as a result of bombardment with a beam of electrons, ions, or metastable atoms (termed *secondary emission*). The guns in which we are interested use heating elements so we focus our attention on thermionic emission. In thermionic devices the cathode is heated in the neighborhood of 1000 degrees Celsius, and this allows electrons to escape from the surface into an electron cloud. The second major component, the anode, is a positively charged electrode. The function of the anode is to provide the potential energy to the emitted electrons. Thus, it attracts the electrons and causes them to accelerate. The final component of the electron gun that we discuss is the focusing electrode. This device bends the equipotential lines to cause uniform emission from cathode and focus the beam.

Figure 1(b) details two important aspects of an electron gun: the beam minimum, b_m, and the point where the beam minimum occurs in the z direction, $z = z_m$. In a beam with no magnetic fields, the electron beam is converging when it approaches z_m. However, the forces between the electrons (the *space charge* forces) cause the beam to stop converging at this point, and they then begin to diverge. Hence, an initial analysis can be performed on an electron gun without magnetic fields to try and attain a desired beam minimum.

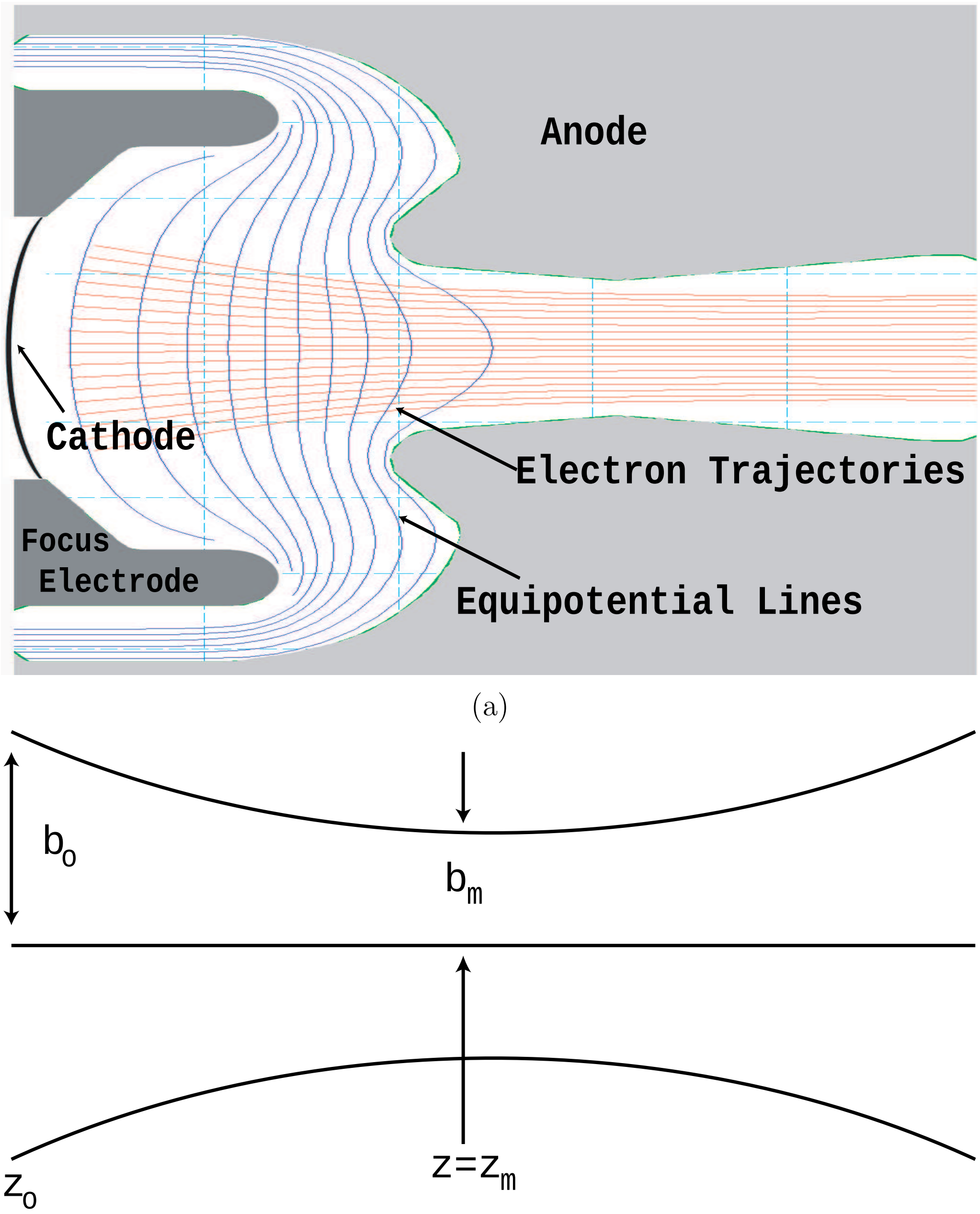
3 Shape Optimization

We now seek to formulate the general mathematical description for an optimal shape design problem, detailed explicitly in [4] and [8]. Let Υ denote domains that are admissible to the problem at hand. In other words, each element $\Omega \in \Upsilon$ is a domain over which the partial differential equation of interest has a solution, u. We define a real valued cost functional $J(\Omega, u)$ that is indicative of the objectives we wish to obtain within the set of admissible domains $\Omega \in \Upsilon$. The corresponding shape optimization problem is to find $\widehat{\Omega} \in \Upsilon$ such that

$$J(\widehat{\Omega}, u) \leq J(\Omega, u)$$

for all $\Omega \in \Upsilon$.

Thus, upon defining a reasonable goal and a cost functional that reflects this goal, we can use numerical optimization to find the optimal domain. For a summary of some of the methods for finding $\widehat{\Omega}$ we refer the interested reader to [7], [4], and [8]. In this work, we assume that the designer wishes either to slightly alter a domain from one which is optimal for a relatively close objective function or to put in the bounds of a feasible domain and find a "global" solution from within the defined bounds (a more time consuming task). We describe here the methods by which we will attempt to obtain local and global solutions. We do not use gradient based methods for reasons to be provided in the sequel. Therefore, for local optimization we compare an implicit filter with a Nelder-Mead routine, and to find



(a)

(b)

Figure 1: Nomenclature associated with an electron gun.

a global solution we incorporate the use of the Direct method (also, Direct followed by the implicit filter or Nelder-Mead).

3.1 Nelder-Mead Algorithm

The Nelder-Mead algorithm [9] is a simplex based method. Define a cost functional $f : \mathbb{R}^n \to \mathbb{R}$ and assume that we have a simplex of $n+1$ points $\{x^0, x^1, \ldots, x^n\}$ where for each i, $x^i \in \mathbb{R}^n$. Let the minimum of the functional evaluated at the points in this simplex be denoted by

$$f^{lb} = \min\{f\left(x^0\right), f\left(x^1\right), \ldots, f\left(x^n\right)\}$$

and x^{lb} be the point where this minimum occurs (lb is the value of the index). Likewise, let the maximum of the functional evaluated at the points in this simplex be denoted by

$$f^{ub} = \max\{f\left(x^0\right), f\left(x^1\right), \ldots, f\left(x^n\right)\}$$

and x^{ub} be the point where this maximum occurs (ub is the value of the index). Then, the goal of the algorithm is to find the minimum over the space by modifying the worst point of the simplex.

Algorithm 3.1. *Nelder-Mead*

1. Locate the centroid, c, of the simplex without the point x^{ub} where

$$c = \frac{\sum_{i=0, i \neq ub}^{n} x_i}{n}.$$

2. Compute $r = c + \alpha(c - x^{ub})$ where $\alpha > 0$ is the step size in the direction (relative to the centroid) that is opposite to the direction of x^{ub}.

3. If $f(r) < f^{ub}$ then look beyond x^{ub} by computing (with $\beta > 1$) $\tilde{x} = c + \beta(c - x^{ub})$.
   - If $f(\tilde{x}) < f(r)$, replace x^{ub} with $\tilde{x}$.
   - Else, replace x^{ub} with r.

4. Else, if $f(r) > f^{ub}$
   - If $\max\{f(x^i) | i \neq ub\} \geq f(r) \geq f^{lb}$ then $x^{ub} = r$.
   - If $\max\{f(x^i) | i \neq ub\} < f(r)$ then step back toward the centroid by computing (with $0 < \eta < 1$) $\hat{x} = c + \eta(x^{ub} - c)$ and

     (a) If $f(\hat{x}) \leq f^{ub}$ replace x^{ub} with $\hat{x}$.

     (b) Else, replace all of the elements of the simplex, x^i, with

$$\bar{x}^i = \frac{x^i + x^{lb}}{2}.$$

Stopping criteria can be based on a number of things including tolerance between the functional values at each iteration or the total number of iterations.

3.2 Implicit Filter Method

Implicit filtering is a projection based quasi-Newton optimization technique that uses difference approximations to the gradient [7]. As the optimization progresses, the difference increment decreases so as to filter out the initial oscillations and avoid local minima. Thus, if the cost functional is of the form

$$f(x) = g(x) + g_n(x),$$

where $g(x)$ is smooth and $g_n(x)$ is noise with low amplitude, then it has been shown that the implicit filter is quite effective. We now discuss, in a simplified manner, the implicit filtering routine. Assume that x_c is the current minimum and that $f_c = f(x_c)$. The iterate starts by creating a stencil about x_c at the current step size h_c. That is, let the $2n$ dimensional stencil be given by

$$S(x_c, h_c) = \{x_c \pm h_c e_i\},$$

where e_i are the unit vectors. Then, we evaluate f at each of the points in the stencil. Let

$$f^* = \min\{f(z)|z \in S(x_c, h_c)\}$$

and denote the element of the stencil where the minimum occurs as x^*. If $f^* < f_c$, then we let $x_o = x_c$ and set

$$x_c = x_o - \lambda \nabla_{h_c} f(x_o),$$

where λ is a parameter used to assure sufficient decrease (see [1]) and

$$(\nabla_{h_c} f(x))_i = \frac{f(x + h_c e_i) - f(x - h_c e_i)}{2h_c}.$$

Note that if $f^* \geq f_c$ then we refer to this as stencil failure and reduce h_c. This process repeats until h_c reaches a minimum user defined value.

3.3 DIRECT Method

DIRECT is a global optimization routine that does not require a gradient and is based on the Lipschitzian approach [6, 2]. The main difference is that DIRECT does not require input of the Lipschitz constant. Instead the algorithm considers values between zero and "infinity" for the Lipschitz constant. The name DIRECT stands for DIviding RECTangles and this is pretty indicative of how the method works. We assume that we are minimizing a cost functional over some bounded parameter space Ξ. That is, we are trying to find the solution to

$$\min_{x \in \Xi} f(x),$$

where $\Xi \subset \mathbb{R}^n$ and $f(x)$ is continuous in at least some region of the global optimum $\hat{x}$ (of the parameter space). Since Ξ has hard bounds, we refer to Ξ as a hyper-rectangle in $\mathbb{R}^n$. Let

S denote a set of hyper-rectangles, initially with a single element Ξ. Upon calculating the functional value at the center, c, of Ξ, the first iteration of DIRECT consists of evaluating $f(x)$ at $2n$ points, $c \pm \delta e_i$, $i = 1, \ldots, n$, where e_i are the unit vectors in Ξ. Then the region Ξ is trisected in every direction creating smaller hyper-rectangles with centers given by $c \pm \delta e_i$, $i = 1, \ldots, n$, and these hyper-rectangles are made elements of S. In each subsequent iteration of DIRECT, potentially optimal hyper-rectangles in S are identified and divided in a similar fashion. For an in depth description of the method we refer the interested reader to [6]. At each iteration, the DIRECT method balances the global and local searches by identifying potential optimal hyper-rectangles based upon not only the functional value of the center but also the size of the hyper-rectangle. We note that the method also normalizes the parameter space Ξ to the unit hypercube in the first iteration and uses scaling factors to identify the functional values.

4 Shape Optimization of an Electron Gun

We have developed the general optimal shape design problem and described numerical methods for finding a local optimum and a global optimum. Now we detail how we are going to apply these methods to an electron gun and what tools we have available to complete the task. Put simply, we wish to alter the shape of the cathode in an attempt to attain a specified beam minimum and laminar flow at the beam minimum. Historically, cathodes are spherical due to limitations in fabrication and designs based on legacy models. However, due to the implementation of numerically controlled machines for parts fabrication, it is now feasible that the cathode take on different shapes. This leads to some interesting problems in that it could radically alter the layout and function of electron guns (such as eliminating the need for focus electrodes, etc.). Because of the geometric nature of the optimization, we treat this as an optimal shape design problem.

The optimal shape design problem requires two main parts, a solution to the PDE and a cost functional based on this solution. There are two main components to the mathematics involved in the electron gun; the setup of the static field and the particle pusher. We initiate this study by first simplifying the electron gun problem. To do so, we assume that we are dealing with a gun that has no magnetic components to aid in beam formation. The electric field in the gun is first calculated using Poisson's equations, derived from Maxwell's equations. The electron trajectories are then calculated using the Lorentz force equations.

For statics, the governing equations of the fields are the time-independent Maxwell's equations given by

$$\nabla \cdot E = \frac{\rho}{\varepsilon_0},$$

where E is the electric field, ρ is the charge density, and $\varepsilon_0 = 8.85 \times 10^{-12}\ F/m$ is the permittivity of vacuum. When a magnetic field is induced there is a magnetic field equation accompanying this equation. However, we assume that the magnetic field is zero. We then

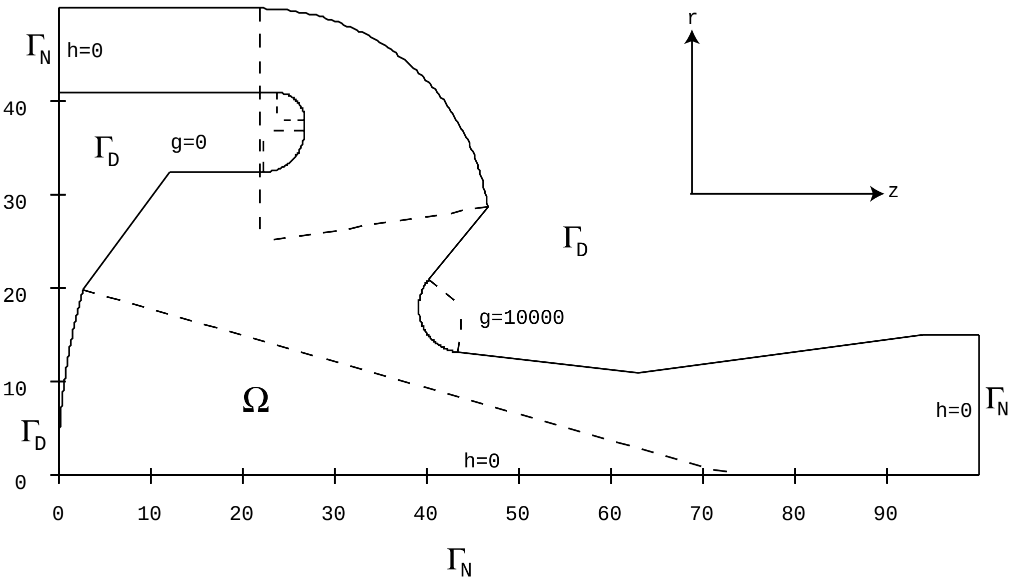


Figure 2: The classic form for the steady Maxwell's equations.

define the electric potential (scalar) Φ as

$$E = -\nabla \Phi$$

so that

$$\nabla^2 \Phi = -\frac{\rho}{\varepsilon_0}.$$

This can be rewritten in strong form as:

> Let $\Psi \subset \mathbb{R}^2$ be open with a piecewise smooth boundary Γ. Let $\Omega = \Psi \cup \Gamma$ and denote the unit outward normal of Γ as n_Γ. The boundaries that have a Dirichlet condition we subscript with D and the boundaries that have a Neumann condition we subscript with N so that $\Gamma = \Gamma_D \cup \Gamma_N$. Given $f = \rho/\varepsilon_0 : \Omega \to \mathbb{R}$, $g : \Gamma_D \to \mathbb{R}$, and $h : \Gamma_N \to \mathbb{R}$ find $\phi : \Omega \to \mathbb{R}$ such that
>
> $$-\nabla^2 \phi = f \qquad \text{in } \Psi, \qquad (4.1)$$
>
> $$\phi = g \qquad \text{on } \Gamma_D, \qquad (4.2)$$
>
> $$-n_\Gamma \cdot \nabla \phi = h \qquad \text{on } \Gamma_N. \qquad (4.3)$$

In Figure 2, we present a diagram for the classic form of the electron gun that serves as a test bed for these shape optimization studies. This gun, designed at the University of Maryland (included with example files in [5]), is a simple gun that is radially symmetric so we represent it in 2D with a (z, r) axis imposed. Here, g is piecewise constant and defined by

$$g = \begin{cases} 10000 & \text{on the anode} \\ 0 & \text{otherwise} \end{cases}$$

and $h = 0$.

The electron trajectories are determined by the Lorentz force (conservation of momentum) equations given by

$$\frac{dP}{dt} = q(E + v \times B)$$

which simplify to (since there is no magnetic field)

$$\frac{dP}{dt} = qE,$$

where

$$P = m\gamma v$$

is the momentum. Here, m is the particle's mass,

$$\gamma = \frac{1}{\sqrt{1 - v^2/c^2}}$$

is a scaling factor, v is the velocity of the particle, and c is the velocity of light. The behavior of the electron gun is simulated via a particle pusher. A particle pusher using a *self-consistent* approach generally proceeds as follows (assuming no magnetic field):

Algorithm 4.1. *Particle Pusher*

1. Determine the electric field without space charge.
2. Assign appropriate initial conditions and launch a specified number of particles from the emitters.
3. Track the velocities and positions of the particles under the influence of the electric field (this modifies the electric field).
4. Recalculate the field with the presence of particle charges with space charge.
5. With new fields, calculate the particle motion.
6. Repeat until the system converges (current densities at the emitters are unchanging and the potential field is invariant between iterations).

One of the widely used simulation programs for 2D electron guns is EGUN, created by W. B. Herrmannsfeldt at the Stanford Linear Accelerator Center. EGUN is a finite difference based particle pusher that can be used for very complex electron guns. For numerical proof of concept, we use this program to generate the electron gun dynamics. This provides a nice starting point because it allows for analysis that is independent of the actual code so that it relies on "black box" information. This will allow for easy adaptation once the 3D particle pusher (Beam Optics Analysis by Calabazas Creek Research, Inc.) is complete. EGUN is an executable DOS based program. The output, a text file, provides a number of useful

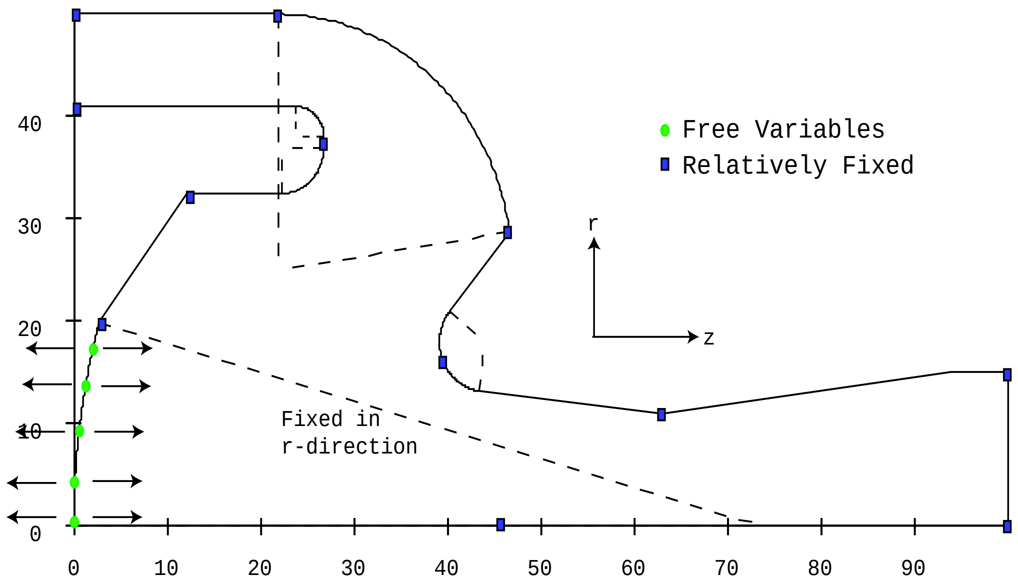


Figure 3: Free variables for the optimal shape design problem.

things, including (but not limited to) 6 fully traced rays (electron trajectories) with the (z,r) coordinates and the radial derivative values of each ray at each time step. We denote the ith electron trajectory as $e^i(z,r)$ and the radial derivative of the electron trajectory at (z,r) as $\partial e^i(z,r)/\partial r$. EGUN also comes with a program, POLYGON, which takes limited information about the gun and creates a boundary file that can be used with EGUN.

Therefore, the remaining issues are defining the free variables for the optimization and a cost functional that will achieve the design goal. POLYGON takes ordered pairs, in part, for input. We can thus treat the cathode as a discrete number (N_c) of ordered pairs given by

$$C = \{(z_i^c, r_i^c) : i = 1, \ldots, N_c\}.$$

The desired gun will still maintain relative size and scaling in the r direction so that the distance between the first point of the focus electrode and the r axis remains the same. Thus, the free variables are the N_c z values $(z_i^c, i = 1, \ldots, N_c)$ of the cathode, where the last point of the cathode is relatively fixed since it is also the first point of the focus electrode. For a graphical depiction of the free variables, see Figure 3. POLYGON has the ability to take the new cathode z values, along with the fixed boundary points, on each iteration and compose a boundary input file.

The objectives of the optimal shape design problem for the electron gun are related to the beam minimum, b_m (see Section 2). The engineer usually, as a first step, tries to obtain a specified beam minimum, b_d. Hence, the cost functional that we use takes a user defined desired beam minimum b_d as input. As mentioned above, EGUN can fully trace six electron trajectories throughout the domain. However, the simulation can be run with up to 100 rays (their trajectories are not reported). Let N_r be the total number of electron trajectories. To find the beam minimum, we must track the outermost electron. The beam minimum is then the minimum of the r values for the outermost ray. Hence, a cost functional that we could use is

$$F(z^c) = (b_m - b_d)^2.$$

However, there are several other aspects to consider. Let z_m denote the z value where the beam minimum occurs. It is of practical importance that all of the rays attain their minima in a neighborhood of z_m. In particular, the six rays that are fully tracked must attain their minima in a small neighborhood of z_m. This implies that we must add a term (here $m = 6$) such as

$$\sum_{i=1}^{m} (z_m - z_m^i)^2,$$

where z_m^i is the z location of the ith tracked electron minimum (in the r direction), to our cost functional. Finally, we require laminar flow for each ray in a neighborhood of the beam minimum. We do so by assuming that in a neighborhood of z_m the radial derivatives, with respect to time, are close to zero. Therefore, we take n_t values of the radial derivatives around z_m for each tracked ray and require that the sum of these values squared is close to zero. Thus, the cost functional that we choose to minimize in the shape optimization problem is given by

$$F(z^c) = 500(b_m - b_d)^2 + \sum_{i=1}^{6} (z_m - z_m^i)^2 + \sum_{i=1}^{6} \sum_{k=1}^{n_t} \left(\frac{\partial e^i(z_k, r_k)}{\partial r} \right)^2 . \qquad (4.4)$$

We first present some local results. For comparison, we use both the Nelder-Mead and implicit filter algorithms since there is no analytic gradient available. The flowchart for the local minimization is found in Figure 4. We now describe each block of the flow chart in detail.

Algorithm 4.2. *Local cathode shape optimization*

- **Initial Cathode:** We initiate the process by providing an initial guess. In the two optimization examples that we present, we use $N_c = 5$ z values for the free variables coupled with the fixed r values at $r^c = 0, 4, 8, 12, 16$, respectively. The sixth (last) point defining the cathode is the same as the first point defining the focus electrode at $r = 19.9$. Thus, our initial guess is for the five free z values $(z_i^c, i = 1, \ldots, 5)$. Note that increasing the number of points in EGUN for the cathode definition might not necessarily result in a better shape. This is due to the fact that if the points are more than 2 mesh units apart then EGUN uses a fitting curve to connect the points. Otherwise, EGUN will connect the points with a straight line. Hence, we use these five r^c values so that the cathode is piecewise curved.

- **Create Boundary:** To create the boundary we define the cathode using the current guess for the z^c values along with the fixed r^c values. The rest of the points remain

	Spherical	IG		NM	IF	NM	IF
b_d	(4.2)	4.2	4.4	4.2	4.2	4.4	4.4
b_m	4.198	4.284		4.479	4.194	4.556	4.432
$z_m = z_m^9$	46.003	43.551		41.179	44.398	40.794	41.979
z_m^1	40.911	34.108		39.310	44.539	39.710	42.915
z_m^2	37.332	33.333		38.135	44.964	38.535	41.341
z_m^4	37.837	39.057		42.660	44.694	43.460	41.874
z_m^6	40.416	43.296		41.316	44.543	42.515	42.923
z_m^8	42.666	42.494		39.724	43.344	39.733	40.926
$F(z^c)$	210.180	218.484	221.684	56.009	1.5778	29.641	3.806

Table 1: Local results for optimal shape design with $b_d = 4.2$ and $b_d = 4.4$ using Nelder-Mead (NM) and implicit filter (IF).

unaltered and are defined by ordered pairs in a POLYGON data file. POLYGON will use this file in an attempt to create the boundary. The Matlab program calls the POLYGON executable and then opens the resulting output file. We then scan the output file for errors and if there is a boundary error we return $f(z^c) = 100000$ and iterate. Otherwise, we proceed.

- **Finite Difference:** EGUN is now called by the Matlab routine. EGUN is a DOS executable that uses the newly created data file for the boundary and attempts Algorithm 4.1. If the solution does not converge or if another error occurs then there are errors in the output and the data needed to calculate F is not written to file. Hence, we must scan the EGUN output file for the data and for errors simultaneously. If the Matlab routine finds errors in the output, $F(z^c) = 100000$. Otherwise, all of the data relevant to the cost functional is stored in arrays. We run the simulation with $N_r = 9$ rays. As mentioned, we can only track six. The rays that we track (from the bottom to the top) are rays 1, 2, 4, 6, 8, and 9.

- **Find Cost:** We then calculate the value of $F(z^c)$ given by (4.4). We take all $\partial e^i(z,r)/\partial r$ values in a neighborhood of $z_m \pm 3$ for the cost functional.

- **New Cathode:** Iterate using Nelder-Mead or implicit filter until convergence criteria are met or the maximum number of iterations has been exceeded.

The electron gun that we have been considering was designed with a spherical cathode. The column labelled *Spherical* in Table 1 displays the beam minimum radius (b_m), the z location where the ith tracked ray attains its minimum (z_m^i), and the cost functional value. Notice that the beam minimum is 4.198. Therefore, we can assume that the optimization routine will find a local solution for a specified desired beam radius of $b_d = 4.2$ given a good

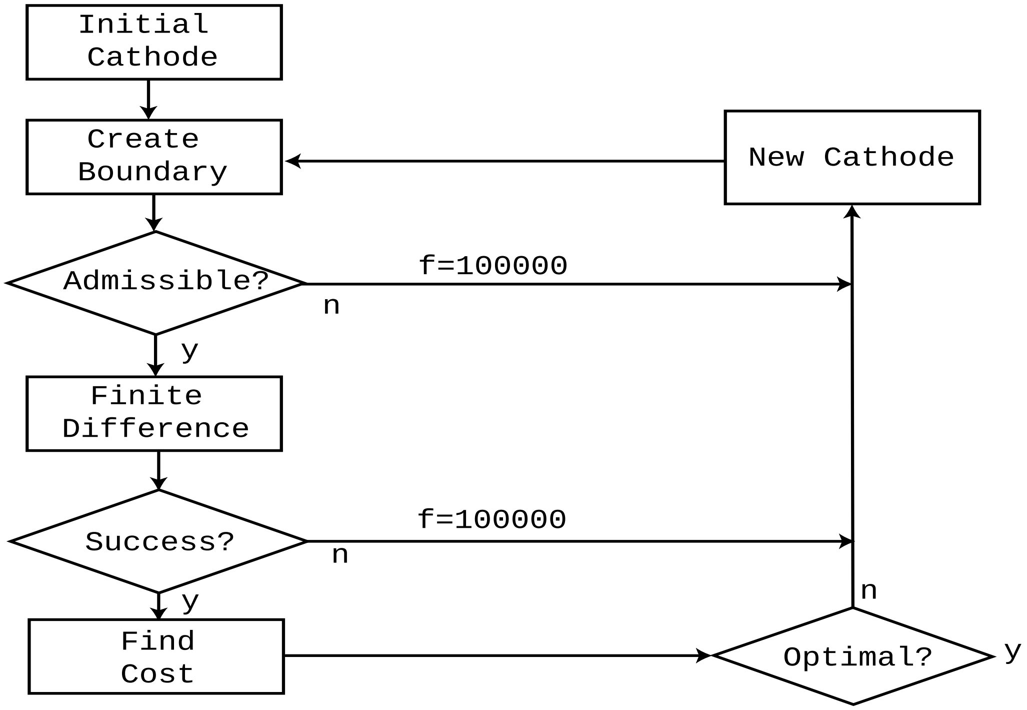
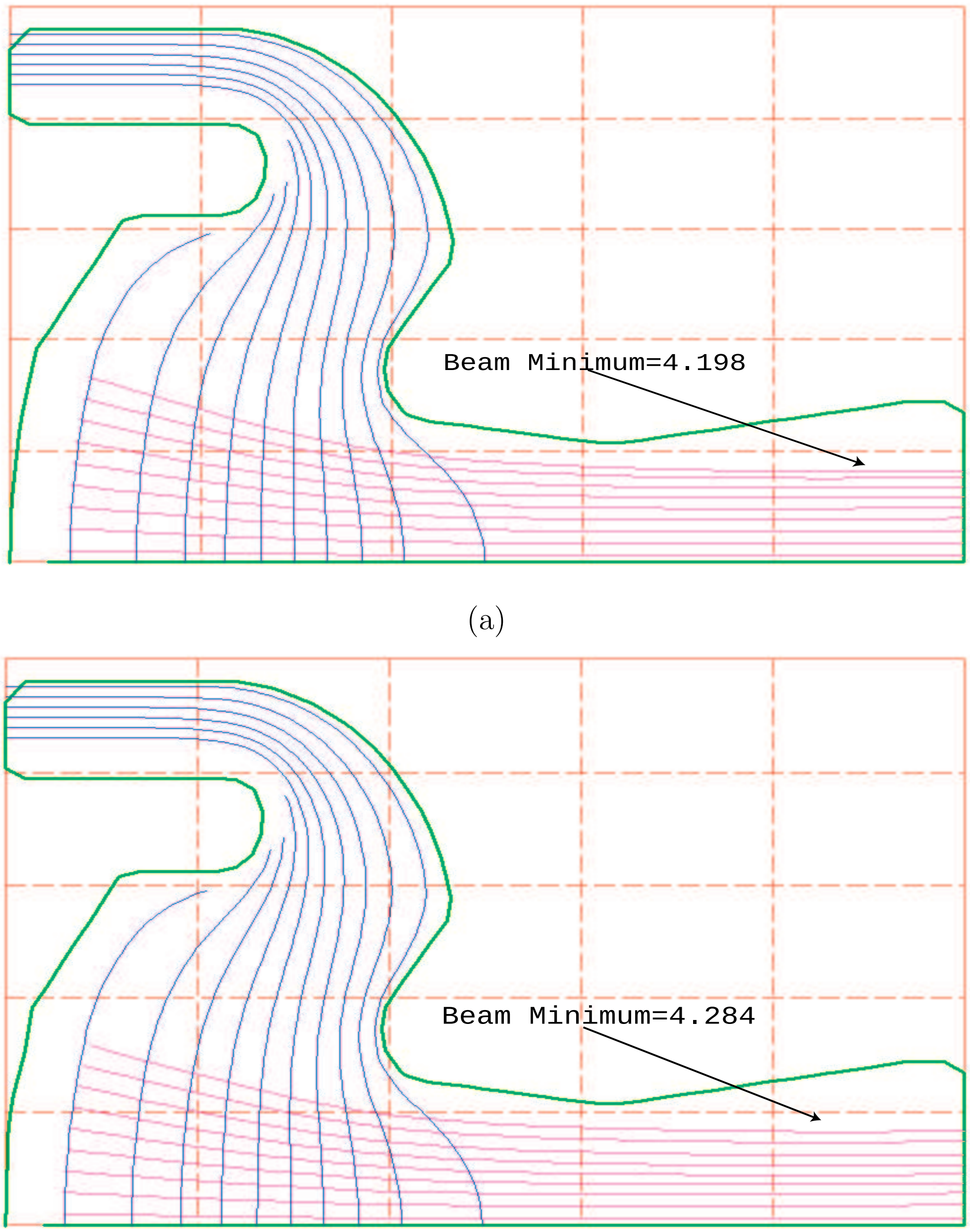


Figure 4: Flowchart for the local optimization routine.

initial guess. The cost functional value for the spherical cathode is 210.180 (with $b_d = 4.2$). This higher cost functional value is due to the disparity between the location of the ray local minima and the location of the beam minimum z_m. The dynamics of the particle pusher for the spherical cathode are depicted in Figure 5(a). In Figure 5(b), we display the initial guess for the optimization problem. Here, we have made a good initial guess (in the table, the column labelled IG) for the five z values of the cathode (see Algorithm 4.2). The cost functional value as well as the beam minimum are given in Table 1.

The results for Algorithm 4.2 using a Nelder-Mead and implicit filter search algorithms are given in columns 3 and 4 of Table 1 for $b_d = 4.2$. Here, it is clear that the implicit filter routine finds a much better set of parameters. The graphical representation for the implicit filter optimal solution can be found in Figure 6(a). If we specify the desired radius of the beam minimum to be $b_d = 4.4$ then we expect the cathode to have less curvature, given that all other components of the electron gun remain unchanged. Figure 6(b) is a graphical depiction of the local optimal solution (found via the implicit filter) with $b_d = 4.4$ and the cathode, indeed, has less curvature than that of the electron gun found with $b_d = 4.2$. Columns 6 and 7 represent the beam minima, location of the ray minima, and the functional values for $b_d = 4.4$ when using Nelder-Mead and implicit filter, respectively, to find the optimal cathode shape. Again, the implicit filter finds a cathode shape that results in a more desirable electron gun.

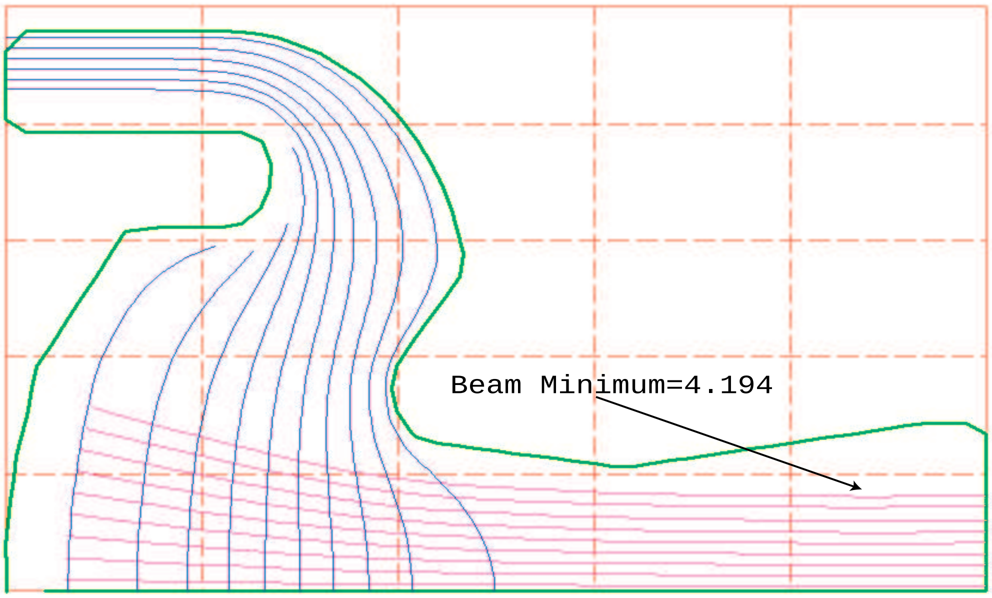
The local optimization routines are effective so long as the user has a good initial guess.



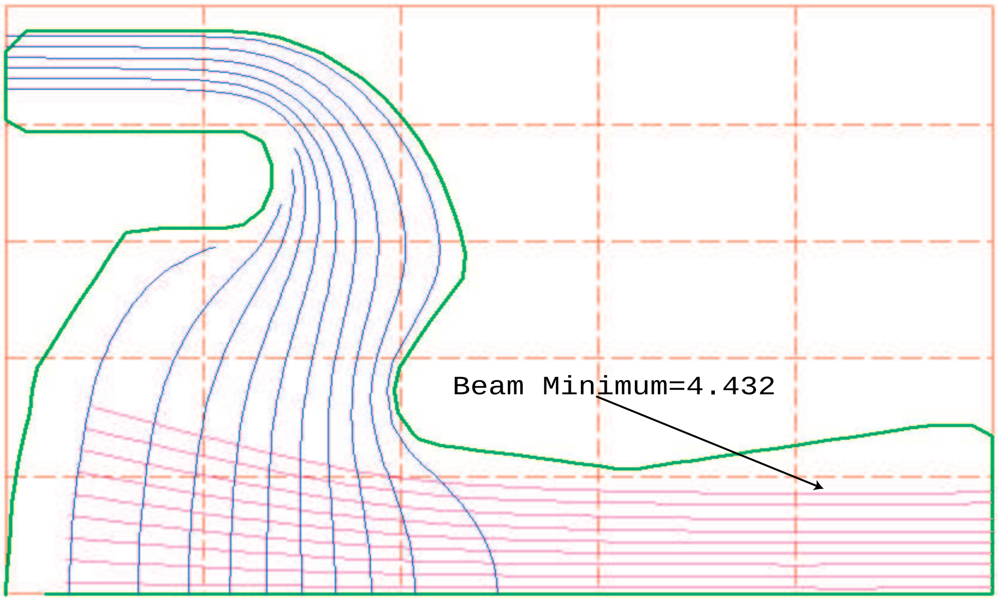
(a)

(b)

Figure 5: Gun design with (a) spherical cathode and (b) initial guess for cathode.



(a)



(b)

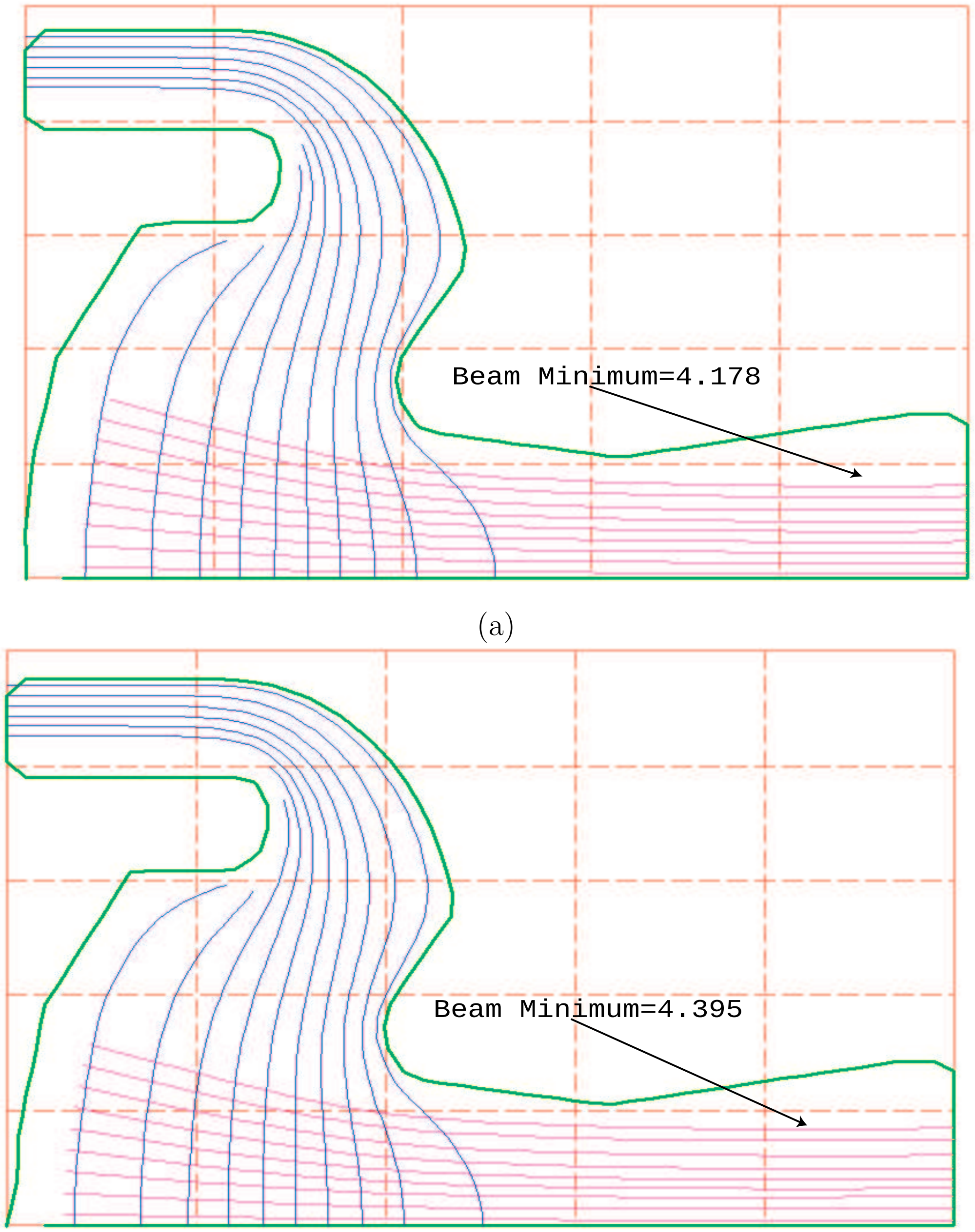
Figure 6: Local results for optimal shape design via the implicit filter with (a) $b_d = 4.2$ and (b) $b_d = 4.4$.

	D20	D20NM	D20IF	D40	D40NM	D40IF
b_d	4.2	4.2	4.2	4.2	4.2	4.2
b_m	4.233	4.233	4.233	4.172	4.189	4.178
$z_m = z_m^9$	43.189	43.189	43.189	44.410	44.414	44.417
z_m^1	44.145	43.745	44.145	43.747	43.747	43.752
z_m^2	42.577	42.577	42.577	44.172	44.172	44.177
z_m^4	42.303	42.303	42.303	43.503	43.904	44.308
z_m^6	45.331	45.332	45.331	45.349	45.350	44.956
z_m^8	42.534	42.534	42.534	43.752	43.755	43.760
$F(z^c)$	7.635	7.035	7.635	3.026	2.134	1.476

Table 2: Global results for optimal shape design with $b_d = 4.2$, DIRECT (D20, D40), DIRECT followed by Nelder-Mead (D20NM, D40NM), and DIRECT followed by implicit filter (D20IF, D40IF).

However, ideally one would prefer to design the gun (minus the cathode), input a beam current and desired beam minimum radius and position, and have the optimization routine find the cathode. For our feasibility study, we ignore the beam current requirement; however, it can be added to the cost function for a more complete design. Due to the requirement that our set of admissible boundaries be piecewise continuous and that the gun have a length requirement, there are hard bounds on the cathode z locations. Thus, since the information to compute f is "black box" and there are constraints on the free variables, DIRECT is a logical choice for optimization. We impose hard bounds of $z_l^c = -3$ and $z_u^c = 3$ for each of the 5 cathode z values. We then iterate DIRECT for either 20 or 40 iterations. Upon completion, we use the resulting z values for the local minimization routine. The results for a desired beam minimum radius of $b_d = 4.2$ are given in Table 2. We note that DIRECT followed by the Nelder-Mead local search algorithm found cathode shapes resulting in the closest beam minima for both 20 and 40 iterations of DIRECT. However, the lowest cost functional value came after using 40 iterations of DIRECT followed by the implicit filter routine. The graphical representation of the EGUN results using the cathode found via 40 iterations of DIRECT followed by the local implicit filter search is given in Figure 7(a).

The results for a desired beam minimum radius of $b_d = 4.4$ are given in Table 3. We see again that the global search followed by the Nelder-Mead algorithm found cathode shapes resulting in the closest beam minima for both 20 and 40 iterations of DIRECT. In fact, in both global optimizations, the implicit filter routine doesn't move the D20 guess. The lowest cost functional value came after using 20 iterations of DIRECT followed by the Nelder-Mead routine. The results for this optimization methodology (20 DIRECT iterations followed by Nelder-Mead) are displayed in Figure 7(b). As in the local optimization, we notice that the cathode displays less curvature than when $b_d = 4.2$.



(a)

(b)

Figure 7: Global results for optimal shape design via (a) 40 iterations of DIRECT followed by implicit filter for $b_d = 4.2$ and (b) 20 iterations of DIRECT followed by Nelder-Mead for $b_d = 4.4$.

	D20	D20NM	D20IF	D40	D40NM	D40IF
b_d	4.4	4.4	4.4	4.4	4.4	4.4
b_m	4.383	4.395	4.383	4.348	4.349	4.348
$z_m = z_m^9$	41.975	41.576	41.975	42.395	42.395	42.395
z_m^1	42.130	42.130	42.130	43.334	42.534	43.334
z_m^2	41.360	41.360	41.360	41.763	41.363	41.763
z_m^4	43.093	42.293	43.093	42.693	42.293	42.693
z_m^6	42.941	41.742	42.941	43.735	43.735	43.735
z_m^8	40.135	39.736	40.135	41.344	41.344	41.344
$F(z^c)$	6.115	4.293	6.115	5.622	5.295	5.622

Table 3: Global results for optimal shape design with $b_d = 4.4$, DIRECT (D20, D40), DIRECT followed by Nelder-Mead (D20NM, D40NM), and DIRECT followed by implicit filter (D20IF, D40IF).

5 Conclusions

In this work we have described another tool that is available to control and/or optimization practitioners. That is, we have exhibited how one can design the geometry of components of a system, so long as such freedom exists, to achieve goals defined a priori. We believe that this feasibility study successfully demonstrated that optimization techniques can be used to design the cathode of an electron gun. Both local and global methods were developed with the use of Nelder-Mead, implicit filter, and DIRECT search algorithms. Of the methods surveyed, we believe that the implicit filter works the best for local optimization and that 20 iterations of DIRECT followed by a Nelder-Mead algorithm to fine tune is the best method for the global optimization.

Acknowledgements

Research reported here was supported by Calabazas Creek Research, Inc. and U.S. Department of Energy grant number DE-FG03-00ER82966.

References

[1] T. D. Choi, O. J. Eslinger, C. T. Kelley, J. W. David, and M. Etheridge. Optimization of automotive valve train components with implicit filtering. *Optimization and Engineering*, 1:9–28, 2000. CRSC Technical Report CRSC-TR98-44, NCSU.

[2] D. E. Finkel. Direct optimization algorithm user guide. CRSC Technical Report CRSC-TR03-11, NCSU, Raleigh, North Carolina, Mar. 2003.

[3] A. S. Gilmour, Jr. *Principles of Traveling Wave Tubes.* Artech House, Boston, Massachusetts, 1994.

[4] J. Haslinger and P. Neittaanmäki. *Finite Element Approximation for Optimal Shape Design.* John Wiley and Sons, Chichester, Great Britain, 1988.

[5] W. B. Herrmannsfeldt. Egun–an electron optics and design program. SLAC Report 331, Stanford U., Stanford, California, Oct. 1988.

[6] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.

[7] C. T. Kelley. *Iterative Methods for Optimization.* SIAM Frontiers in Applied Mathematics. SIAM, Philadelphia, Pennsylvania, 1999.

[8] O. Pironneau. *Optimal Shape Design for Elliptic Systems.* Springer Series in Computational Physics. Springer-Verlag, New York, New York, 1984.

[9] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C, the art of scientific computing.* Cambridge University Press, Cambridge, England, 2nd edition, 1993.