# An Advanced Simulation Code for Modeling Inductive Output Tubes

## Phase I Final Report

Calabazas Creek Research, Inc.
690 Port Drive
San Mateo, CA 94404
(650) 312-9575, Fax: (650) 312-9536
RLI@CalCreek.com

Principle Investigator: Dr. R. Lawrence Ives

Topic Number: 46
Subtopic Number: b
SBIR Contract Number: DE-SC0001846

July 19, 2010

## Abstract

During the Phase I program, CCR completed several major building blocks for a 3D large signal, inductive output tube (IOT) code using modern computer language and programming techniques. These included a 3D, Helmholtz, time-harmonic, field solver with a fully functional graphical user interface (GUI), automeshing and adaptivity. Other building blocks included the improved electrostatic Poisson solver with temporal boundary conditions to provide temporal fields for the time-stepping particle pusher as well as the self electric field caused by time-varying space charge. The magnetostatic field solver was also updated to solve for the self magnetic field caused by time changing current density in the output cavity gap. The goal function to optimize an IOT cavity was also formulated, and the optimization methodologies were investigated.

## Introduction

Calabazas Creek Research, Inc. (CCR), in collaboration with Science Applications International Corporation (SAIC), proposed to develop an advanced, large signal computer code for rapid, accurate design of inductive output tubes. The code would include a user-friendly, intuitive, graphical user interface (GUI) and computer optimization. The program would use the SAIC code Nemesis as the starting point for code development **[1]**. Nemesis is a time-domain formulation that relies on integration of equivalent circuit equations coupled with the Lorentz force equations for particle trajectories. The code employs a 2D model for the circuit fields and a 2D Poisson solver. This proposed program would extend the formulation for fully 3D operation. This would allow simulation over a wider parameter range and more complex geometries.

Nemesis does not incorporate a graphical user interface, which limits marketability and use. CCR is a leader in CAD-based GUI creation. CCR's Beam Optics Analyzer (BOA) program imports geometric data from commercial, solid modeling packages with parametric input using state of the art graphics features, including text boxes, radio buttons, pull-down menus, and pick boxes. CCR also developed GUIs for the computer codes CASCADE, SURF3D/LOT, and OOPIC.

CCR is also a leader in integrating computer optimization into design codes. CCR integrated optimization routines for designing RF waveguides and windows [**2**], RF antenna launchers [**4**], electron guns [**3**, **6**], and RF circuits [**5**]. In each case, the time and cost to design these components was dramatically reduced while simultaneously achieving higher performance. In some cases, designs were achieved that could not previously be obtained [**7**, **8**]. Optimization routines can explore a much larger parameter space than practical with manual simulation. Computer optimization was instrumental in commercialization of these codes.

The objective of the program was to develop an advanced, 3D, large-signal design code for modeling inductive output tubes. The code would include a user-friendly, CAD-based graphical user interface and computer optimization. This Phase I program defined the theoretical basis for 3D simulation of the fields and particles. The GUI requirements were identified and input screens defined. This included integration with 3D solid modeling software. The optimization requirements were defined, including identification of optimization parameter and generation of goal functions.

Following tasks were planned for this Phase I program:

1. Investigate methods to improve and extend Nemesis to a Full 3D Code
2. Integrate CAD with Nemesis and Automeshing
3. Develop Goal Functions and Constraints
4. Design a Graphical User Interface
5. Nemesis Validation

The following sections describe the accomplishments in each task, and if program directions were altered for better and more productive goals, they will be explained. In addition, we will also provide future plans to be built upon the successes of this Phase I program. Some concluding remarks will be provided at the end of this report.

## Accomplishments

### 1. *Investigate methods to improve and extend Nemesis to a Full 3D Code*

For this task, the emphasis was on finding techniques and implementation to extend Nemesis to a full 3D code. Nemesis was specifically developed for IOT's. It is a large signal code, which avoids solving the full Maxwell equations by combining an equivalent circuit model for the output cavity, a model for the circuit fields that scales with the cavity voltage and a particle pusher. The particle pusher includes the circuit fields, the external focusing magnetic fields and the beam space-charge fields The code determines beam characteristics and the output power at a specified resonant frequency [1]. It currently uses the analytic 2D field model by Kosmahl and Branch [9] in the output gap. The 2D field model is assumed to be axisymmetric, and the azimuthal magnetic flux density, and axial and radial electric fields are analytically derived as functions of the gap voltage, length and radius. A full 3D field is needed to design cavities for multiple beam IOTs. To address this shortcoming CCR proposed to develop a time-harmonic field solver. Due to the limited resources of the Phase I program, only a 2D version without a GUI was planned. However, , a fully 3D, time-harmonic, field solver with a fully functional GUI was actually implemented [10, 11].

The time-harmonic fields in the cavity gap are governed by inhomogeneous vector wave equations, also known as inhomogeneous Helmholtz electromagnetic wave equations [12]. Each equation depends only on one type of field, electric or magnetic. We selected to work with the magnetic field since its form is almost identical to the curl-curl equation of magnetostatics. The inhomogeneous Helmholtz wave equation in term of the magnetic field is

$$\nabla \times \left( \frac{1}{\varepsilon_r} \nabla \times \mathbf{H} \right) - k_0^2 \mu_r \mathbf{H} = \nabla \times \frac{\mathbf{J}}{\varepsilon_r} \quad \text{in} \quad \Omega \tag{1}$$

where the wave number in free space $k_0 = \omega\sqrt{\varepsilon_0 \mu_0}$ , $\varepsilon_r$ and $\mu_r$ are the relative permittivity and permeability respectively. The current density vector $\mathbf{J}$ is produced by the particles travelling by the gap. We assume lossless cavities and employ magnetically and electrically conducting boundary conditions to obtain the following weak formulation to be discretized by the finite element method

$$\int_{\Omega}\left[\frac{1}{\varepsilon_r}(\nabla\times\mathbf{W})\cdot(\nabla\times\mathbf{H})-k_0^2\mu_r\mathbf{W}\cdot\mathbf{H}\right]d\Omega=\int_{\Omega}\mathbf{W}\cdot\nabla\times\frac{\mathbf{J}}{\varepsilon_r}\,d\Omega \qquad (2)$$

where $\mathbf{W}$ is an arbitrary vector function that satisfies the magnetically conducting condition. Solving this weak form by the finite element method requires interpolation functions for the weighting function $\mathbf{W}$ and trial solution $\mathbf{H}$. The standard finite element method uses scalar interpolation functions for $\mathbf{W}$ and $\mathbf{H}$. Unfortunately, this enforces all-directional continuity and cannot treat dielectric and conducting surfaces where the transverse field components are singular. Thus, we implemented the vector finite element method in which the interpolation functions are vectors, and only the tangential continuity is strongly enforced.

To test this new solver we solved for the fields inside a box caused by a current flowing along a straight cylindrical conductor uniform in magnitude but varied sinusoidal in space and tim, as shown in Figure 1. A sinusoidal in space and time current flow is a very good approximation of the actual electron beam in the cavity gap region.

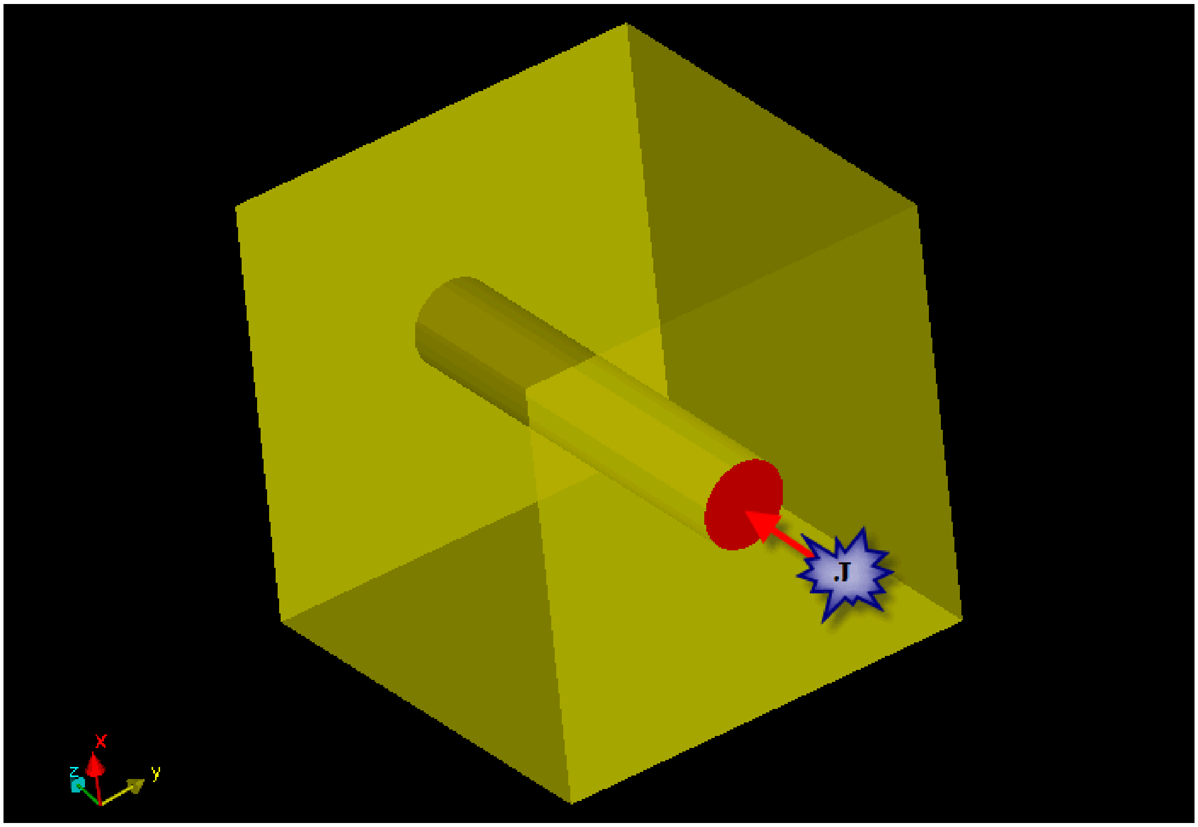


**Figure 1** Cylindrical conductor in a box, uniform current but sinusoidal in space and time.

The mesher is automated and generated unstructured meshes as seen in Figure 2. This mesh included 135,489 tetrahedra, 274,348 faces, 164,661 edges and 25,803 vertices. Since the vector finite element is the method of choice, in this time-harmonic field solver, the number of mesh edges is the number of unknowns. The finite element field solver solves for the magnetic field then obtains the electric field from

$$\mathbf{E} = \frac{1}{j\omega\varepsilon}\left(\nabla \times \mathbf{H} - \mathbf{J}\right) \tag{3}$$

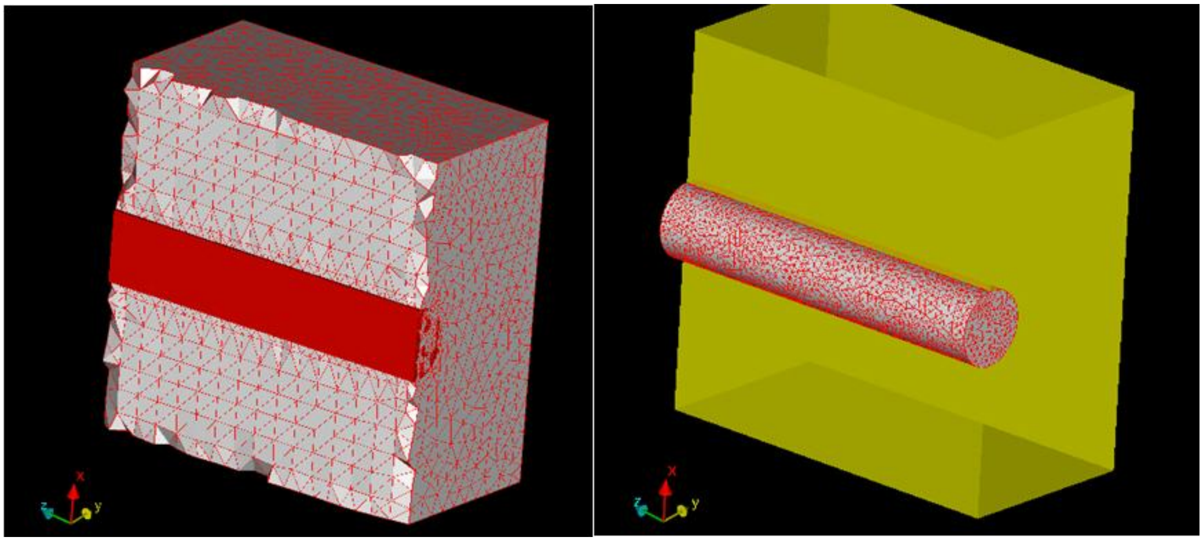


**Figure 2** Finite element, unstructured mesh for a cylindrical conductor in a box.

The electric field as shown in Figure 3 exhibits the sinusoidal characteristics along the z axis. An output gap diameter then must be sufficiently small within the field amplitude to experience any cavity field and particle interaction.

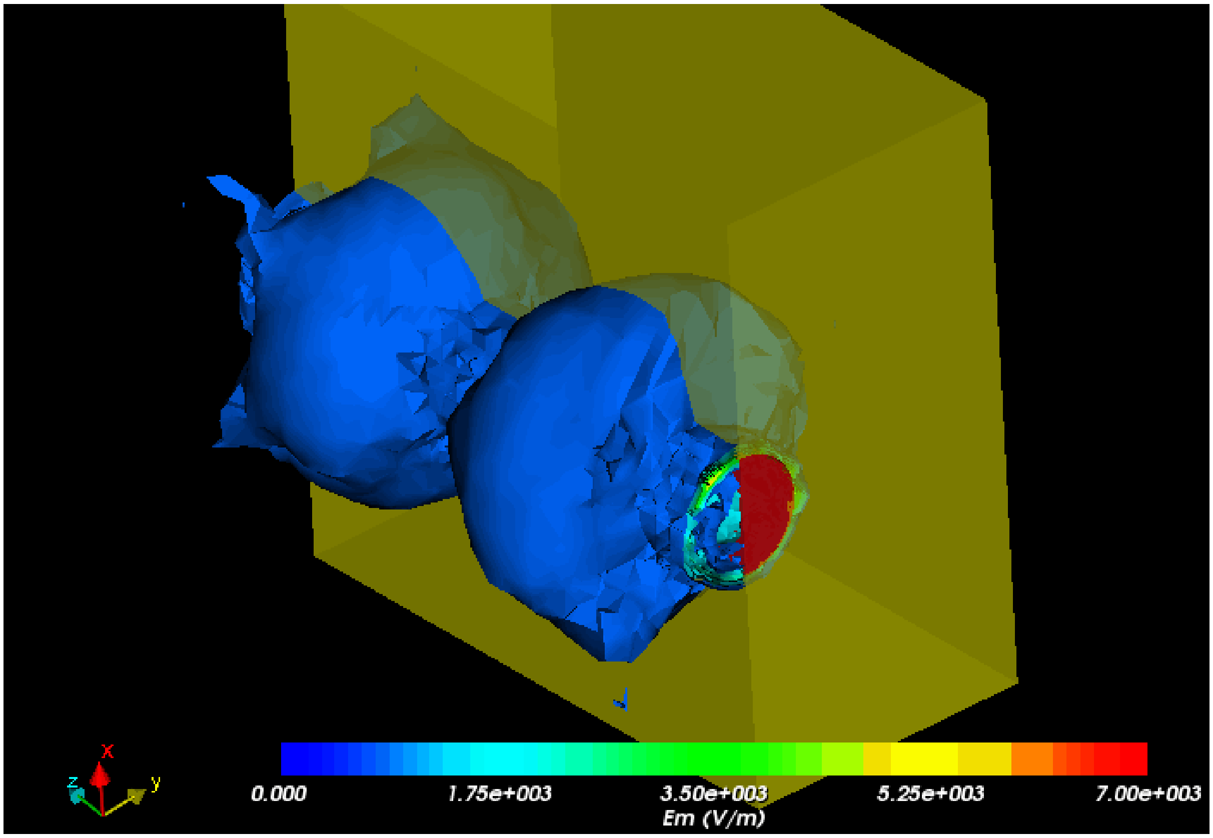


**Figure 3** The electric field around a cylindrical conductor with a current flow sinusoidal in space and time.

In Fig. 4, the xy plane slicing of the 3D field displays the gradient plot of the electric field on the left and contour plot on the right. Again the sinusoidal characteristics can be seen clearly, particularly in the gradient plot.

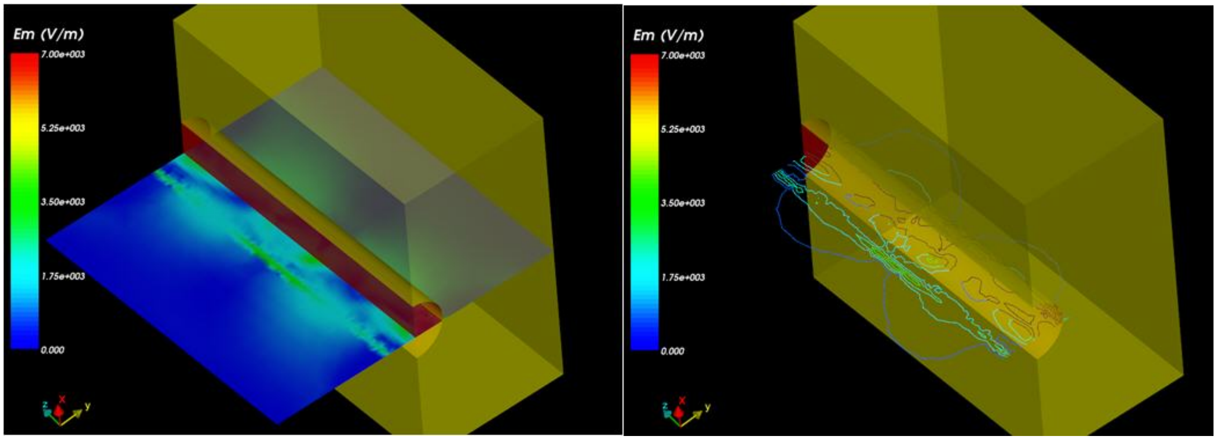


**Figure 4** Helmholtz electric field on the xy plane (a) Gradient (b) Contour plot.

This Helmholtz field solver is now fully integrated into BOA with a fully functional GUI for both preprocessing user's input data and postprocessing of the solution fields. the GUI is discussed in more detail in Task 4.

Nemesis uses a 2D Poisson solver to determine the self-electric field caused by the fluctuating space-charge field on every time step. Capturing the 3D particle characteristics requires a 3D Poisson solver. BOA has this capability to provide the 3D electric field from 3D space charge fields deposited by particles when they travel throughout the mesh [13, 14, 15, 16]. The Poisson field solver in BOA uses the standard, scalar finite element method and can use the same mesh for the Helmholtz solver to find the time-harmonic field. Thus, the field interpolation is very efficient when calculating the Lorentz force associated with the equation of motion.

Self-magnetic field induced by beam current is also included in Nemesis by integrating Ampere’s law, but the radial component of the current density is ignored. It is more accurate to use a fully 3D magnetostatic solver to calculate the self-magnetic field. Fortunately, such a solver, using the vector finite element method similar to the Helmholtz solver, is also available in BOA [17]. Again all three solvers: magnetostatic, Helmholtz and Poisson, can use the same mesh, resulting in very efficient field interpolation.

Nemesis does not simulate the actual bunching of the beam at the cathode and grid gap but rather assumes the beam has a specific form as it enters the cavity. Both the spatial distribution of the momentum and the temporal variation of beam current due to bunching at the injecting plane are assumed. Realistic and accurate models of the beam conditions at the injecting plane are very important for accurate spent beam data to correctly design an efficient collector. There are specific models for the momentum space distributions currently available in Nemesis: a rigid-rotor, a confined-flow, and a matched-beam to simulate the initial axial and transverse momenta. They assume the injected beam does not expand or contract upon entry; thus, the initial momenta do not contain any radial component.

In the IOT, the beam is density modulated at the grid with a periodic signal. The electron emission beyond the grid occurs when net potential at the grid is positive and the electrons reach the grid with sufficient momentum to overcome the net grid potential when it becomes negative. Previous investigations at CCR using ballistic theory [18, 19] and MAGIC simulations [20] indicated that the electron transit time at the grid gap has a major impact on the IOT performance and perhaps the shape of the bunched beam. Thus, assuming the bunch shape is endowed with periodic characteristics at the injecting plane, as Nemesis does, the beam current is calculated by,

$$I_b\left(t\right)=I_{peak}\left\{\begin{matrix}\sin^2\left(\dfrac{\pi t}{\tau_w}\right); & 0\le t\le\tau_w \\ 0; & \tau_w<t\le\tau_p\end{matrix}\right\} \tag{4}$$

where $I_{peak}$ is the peak current, $\tau_p = {}^1\!/_f$ the bunching period with the grid frequency *f*, and $\tau_w$ the bunch width. This is probably not sufficient to fully characterize the initial bunched beam at the injecting plane. This will lead to inaccurate spent beam data and consequently inefficient collector design or even a nonfunctional collector.

In this task, we investigated models to include the initial radial movement of the beam, and to improve the model of temporal beam current at the injected plane to include the transit time effects for a more realistic temporal model. However, after some further investigation, we felt that it was much better to improve BOA by modeling of the actual bunching of the beam from the cathode, in the cathode-grid gap travelling through the anode and entering the output gap. This more accurately captures the particle temporal and spatial characteristics than using the heuristic models described above.

To this end, we implemented a symbolic interpreter in the GUI so that temporal boundary conditions can be imposed to the electrodes, and modified the Poisson solver to permit stepping in time. We are currently implementing the IOT input cavity equivalent *LCR* circuit algorithm to obtain the temporal charge loaded grid potential. This algorithm requires the input driver voltage, inductance, capacity and resistance of the input cavity to provide a required temporal grid voltage. We are also working on the particle pusher to model the beam bunching from the cathode, through the grid and to the output cavity.

During the course of learning Nemesis, we found the code was written in procedural language. It was structurally very difficult to efficiently build a robust, easy-to-maintain program when integrating Nemesis with object-oriented, generic programming 3D field solvers in BOA. We determined it was better to build the 3D infrastructures, such as the full 3D Helmholtz field solver described earlier, for a future large signal code programmed in modern programming language and techniques. Such a code would be more extensible, robust and easy-to-maintain. This future large signal code would be intimately integrated with BOA using the same GUI, the same mesh for all three finite element 3D field solvers as well as the particle pusher. We would need to implement the circuit part for the output cavity to interact with the 3D fields to extract the beam energy and produce the spent beam for the collector. BOA could then use this spent beam to design the collector. The whole process to design the complete IOT device, starting from the cathode to the output cavity to the collector, would be seamless. It would require minimal user's input in between analysis types and, more importantly, would provide a single code to maintain and support.

## 2. *Integrate CAD with Nemesis and provideAutomeshing*

After determining that integration of the finite element field solvers into Nemesis was impractical, we shifted our emphasis to the automeshing and adaptivity for the 3D Helmholtz solver. The mesh shown in Fig. 2 was auto-generated, and all three solvers use the same geometry imported directly from a CAD package such as SolidWorks (SW). When the GUI detects pre-specified geometry entities such as a conductor or secondary emitter, it will set the mesh on these entities to some default size. The user can then reset the mesh sizes or even disable them.

The finite element method produces the optimal approximation from solution spaces, which is the magnetic field in the Helmholtz field solver. However, we are also interested in the electric field given by Eq. (3) i.e. the gradient of the finite element approximation (magnetic field). Unfortunately, the discrete electric field is discontinuous across the element boundaries. Consequently, we incorporated a post-processing procedure whereby the discontinuous approximation to the electric field is smoothed. By post-processing the electric field ,we can estimate *a posteriori* the error for mesh adaptivity. A rather natural approach to the error estimation is based on measuring the difference between the direct and post-processed (recovered) approximation to the gradient. BOA implemented this approach using the procedure developed by Zienkiewicz and Zhu [21, 22, 23, 24]. Their so-called superconvergent patch recovery (*SPR*) procedure post-processes the finite element approximation to obtain values of the electric field at the nodes. These are the *recovered*, averaged and smoothed gradients plus the current density sampled from the centroids of all elements sharing a common node. These values are then used to obtain the globally reconstructed electric field producing a $C^0$-continuous field. The recovered field is compared with the unprocessed field to obtain the posteriori error estimate. Having this error estimate, the Helmholtz field solver refines or coarsens the mesh with the goal of reducing the error estimate below a prescribed value.

## 3. Develop Goal Functions and Constraints

The optimizer in BOA relies on SolidWorks (SW) for model geometry and design tables for optimizing the design geometry [25, 26]. The particle pusher and the field solvers provide the particle characteristics to compute objective functions, described shortly, and update the optimized parameters. The optimizer then updates the geometry by modifying the design tables from SW. The optimizer can choose one of two iterative methods to iterate on the objective function to obtain an optimal point. Both methods were specifically selected for their efficiency and deterministic property, and, more importantly, gradients of the objective functions were not the strict requirement. They both rely on the initial iterate, *N*-vector of *N* optimized parameter, which is constructed from the user-provided, constrained, design parameters. Each method constructs this initial simplex differently, and each has its own algorithm to alter the simplex during the optimization.

The first iterative method, Nelder-Mead (NM), maintains a $N+1 \times N$ simplex approximation to an optimal point [27, 28]. Essentially, the strategy is to bracket a minimum so that the subsequent isolation can be guaranteed. Nelder-Mead attempts to replace the worst vertex with a new point by expanding, contracting or shrinking the simplex. It does not explicitly use the approximate gradient information. Although the original NM algorithm is unconstrained, CCR modified it for bound constrained problems by constructing an initial simplex bounded by design parameters. During the expansion phases, the derived vertex is also constrained [25, 26]. Nelder-Mead is not guaranteed to converge, even for smooth problems; and it can stagnate at a non-optimal point (local minima). In practice, however, the Nelder-Mead algorithm performs well. When faced with stagnation, a restart with the current best point and a different set of simplex directions often helps [28]. The method converges when the simplex norm is reduced to a specified size, or the number of objective function evaluations exceeds a budgeted number.

The second iterative method, implicit filtering (IF), originated from the coordinate search method [28]. It starts with an initial iterate of optimized variables and a scale *h* to construct both right and left simplex from the initial iterate with edges having length *h*. The coordinate search then samples the entire $2N\times N$ simplex and replaces the initial iterate with the new point whose objective function value is the minimum. If this is not met (i.e. one has a stencil failure), the algorithm shrinks the simplex by reducing *h*. The process repeats until the number of function evaluations exceeds a specified limit. Implicit filtering uses the same functional values as coordinate search, and similarly reduces the scale when stencil failure takes place. However, it improves the coordinate search method by using the objective function values on the simplex to approximate a gradient to build a local model. Using this model, implicit filter then searches for a better point by augmenting coordinate search with a steepest descent. Thus, IF is a projected, quasi-Newton method that uses difference gradients. The method converges when the number of objective function evaluations exceeds a budgeted number, as mentioned above, but also when the norm of the gradient meets a specified limit. Implicit filtering's reduction of the difference increment as the optimization progresses is essential for the method to step over noisy objective functions.

The iterative method, either NM or IF, optimizes on the objective function to achieve an optimal configuration. The goal of the optimizer is to search for optimal design parameters by minimizing an objective function. This goal function is different for different optimization problems. The objective function must be a function of optimized parameters. To design an optimally functional output cavity for the IOT requires optimizing the gap geometry, as well its voltage, after a period of time τ; thus, the objective function is the Q factor. The lower the Q factor the more efficient the beam energy reduction and the lower the spent beam energy. The Q factor can be defined as

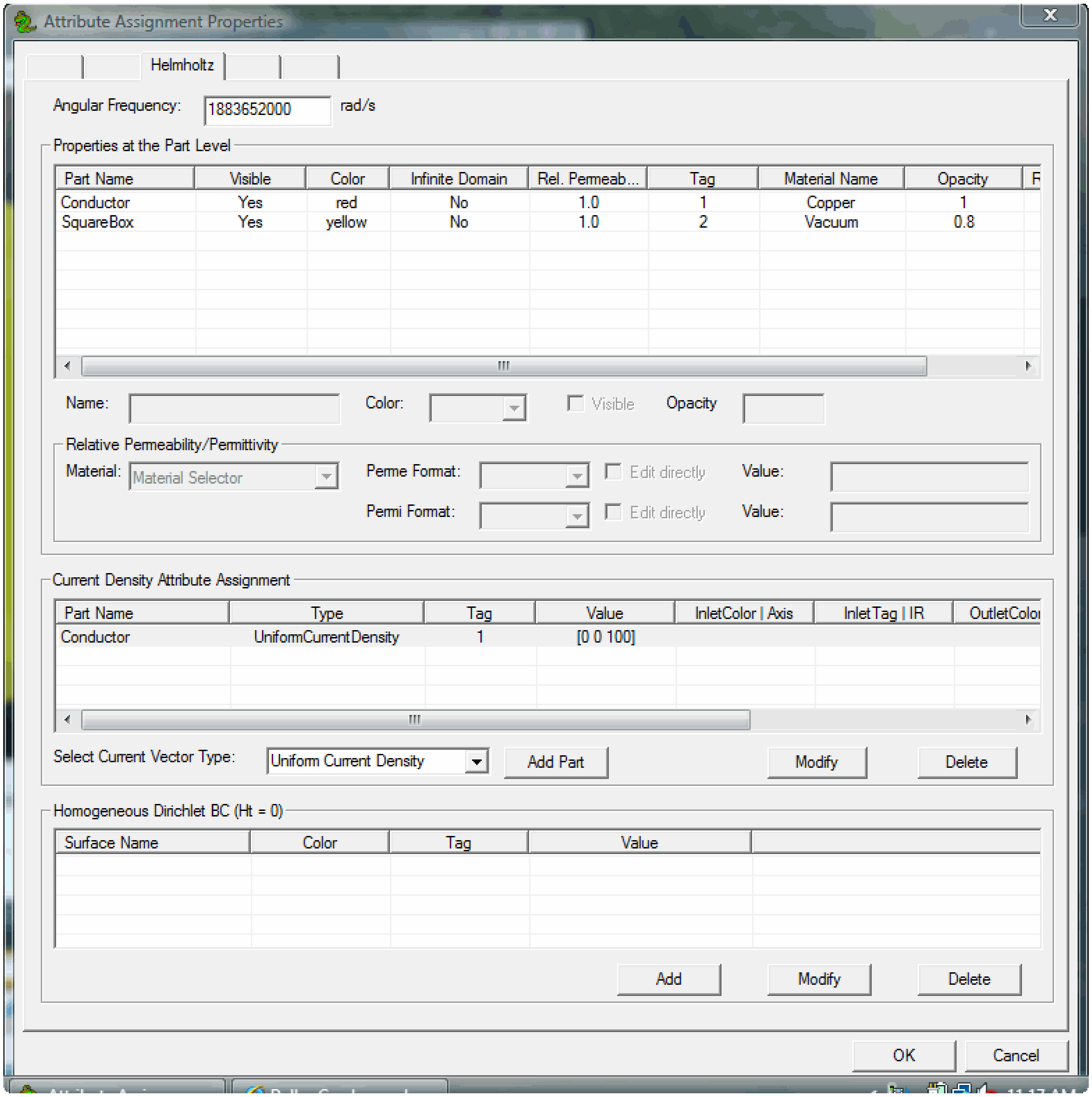
$$Q \equiv \frac{\int_0^\tau \sum_{j=1}^{N_o} ke_j^o \, dt}{\int_0^\tau \sum_{j=1}^{N_i} ke_j^i \, dt} \tag{5}$$

where $ke^i$ is the injecting particle kinetic energy, $N_i$ the total number of injecting particles, $ke^o$ the exiting particle kinetic energy and $N_o$ the number of exiting particles. All particles are synchronically pushed, and ,the temporal integrals can be performed after the spatial sums. We also consider of upstreaming particles by giving them more weight in the calculation of the objective function. It is important to note that Eq. (5) is only one objective function for this type of optimization. Objective functions can be crafted to include other operating parameters that are important.

## 4. *Design a Graphical User Interface*

As we modified efforts to build infrastructures for a large 3D signal code, we designed and implement a fully functional GUI for the Helmholtz field solver. This included the preprocessor, shown in Fig. 5, for input of electromagnetic material properties and boundary conditions, including the harmonic angular frequency. The current density vector can be prescribed to a CAD part either uniformly or sinusoidally in space. A popup dialog allows the user to enter the current density components for a straight conductor (Fig. 6) or the current for coils (Fig. 7). For coils the user is not required to manually supply the coil geometry. The GUI will automatically detect and fill in the coil dimensions.

The postprocessor for the Helmholtz solver was also implemented. It processes the solution magnetic field to obtain the electric field. From its control panel, shown in Fig. 8 with the contour plot of the current density vector, one can plot the fields, magnetic, electric and current density fields, at a point, along an arbitrary line, on surfaces of CAD entities, or arbitrary cut planes. Starting from the control panel, line plotting of the fields begins with a popup window, as shown in Fig. 9a, for specifying the end points, and plot type. The line plot of the current density along the z axis is shown in Fig. 9b. The user can also click on the Plot Parameters to access another popup dialog to change the plot properties. The field slicer can display fields as gradients or contours on an arbitrary plane whose orientation can be specified from a popup dialog (Fig. 10a). Figure 10b shows a gradient plot of the current density vector field on a cut plane.



Angular Frequency: 1883652000 rad/s

Properties at the Part Level

| Part Name | Visible | Color | Infinite Domain | Rel. Permeab... | Tag | Material Name | Opacity | F |
|---|---|---|---|---|---|---|---|---|
| Conductor | Yes | red | No | 1.0 | 1 | Copper | 1 | |
| SquareBox | Yes | yellow | No | 1.0 | 2 | Vacuum | 0.8 | |

Name: Color: □ Visible Opacity

Relative Permeability/Permittivity

Material: Material Selector

Perme Format: □ Edit directly Value:

Permi Format: □ Edit directly Value:

Current Density Attribute Assignment

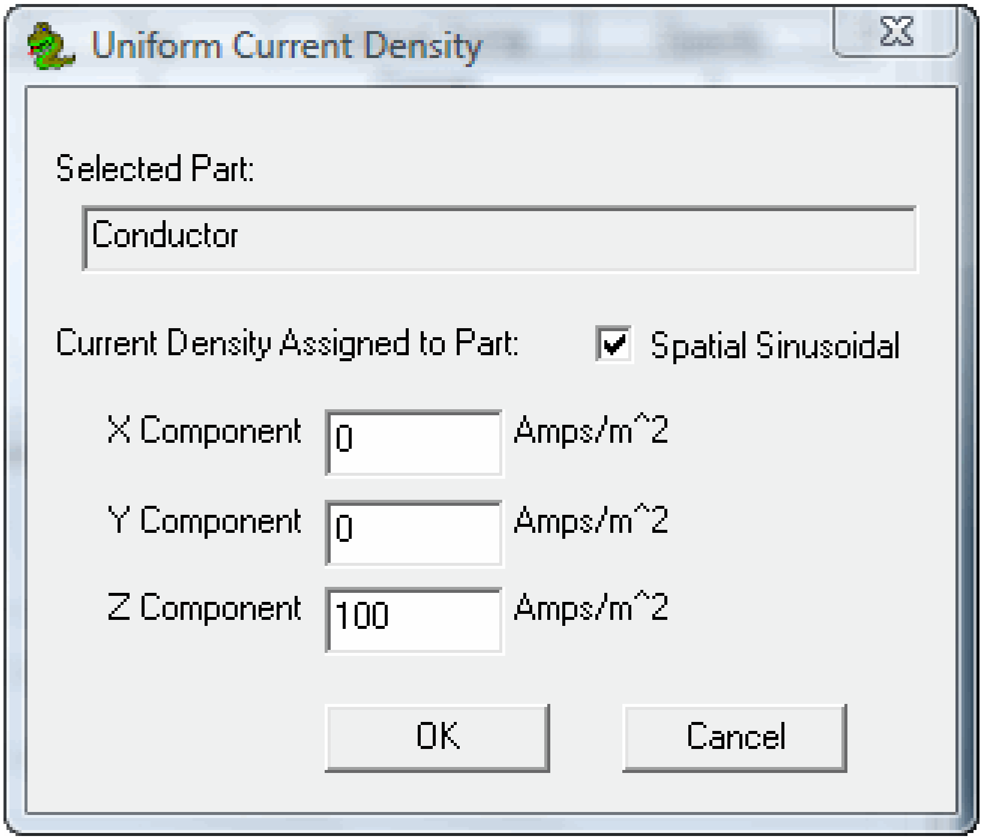
| Part Name | Type | Tag | Value | InletColor \| Axis | InletTag \| IR | OutletColoi |
|---|---|---|---|---|---|---|
| Conductor | UniformCurrentDensity | 1 | [0 0 100] | | | |

Select Current Vector Type: Uniform Current Density

Homogeneous Dirichlet BC (Ht = 0)

| Surface Name | Color | Tag | Value | |
|---|---|---|---|---|

**Figure 5** Helmholtz GUI dialog to enter operating frequency, material properties and boundary conditions.



Selected Part: Conductor

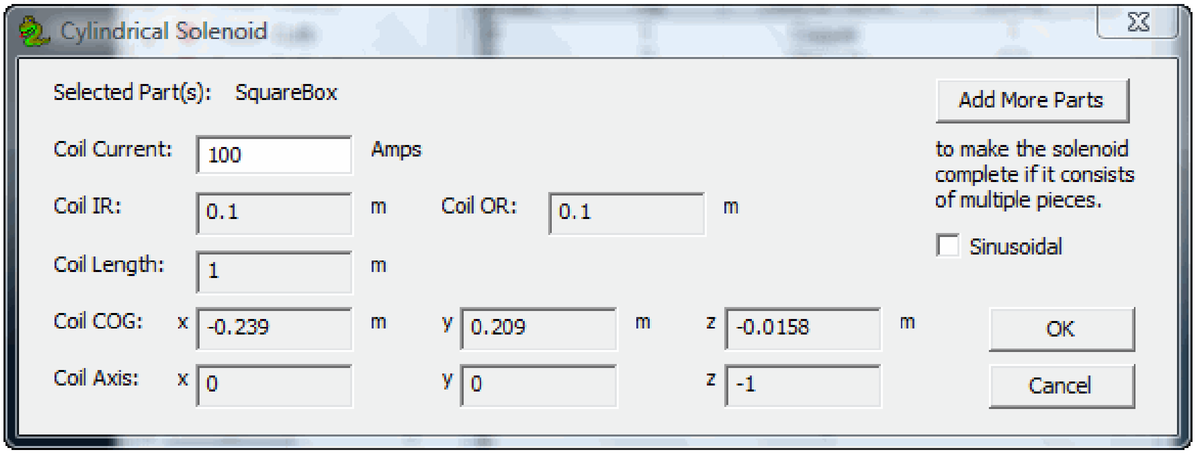
Current Density Assigned to Part: ☑ Spatial Sinusoidal

X Component 0 Amps/m^2

Y Component 0 Amps/m^2

Z Component 100 Amps/m^2

**Figure 6** Popup dialog to input current density component for straight conductor.



Selected Part(s): SquareBox

Coil Current: 100 Amps

Coil IR: 0.1 m Coil OR: 0.1 m

Coil Length: 1 m

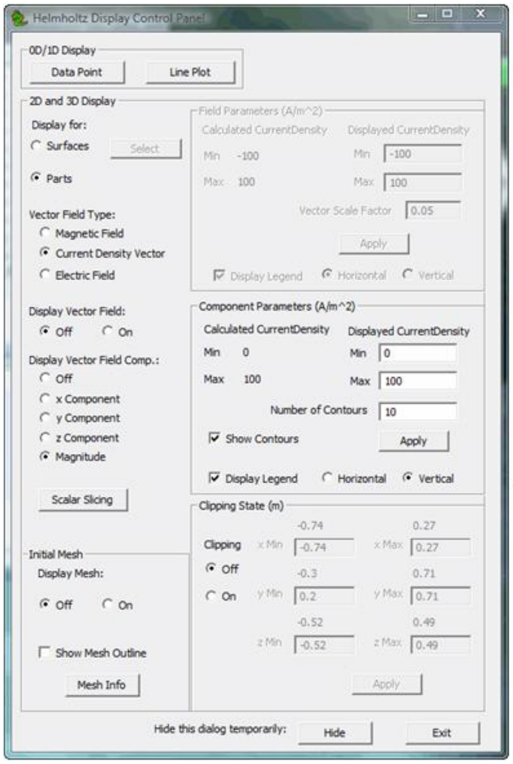
Coil COG: x -0.239 m y 0.209 m z -0.0158 m

Coil Axis: x 0 y 0 z -1

to make the solenoid complete if it consists of multiple pieces.

□ Sinusoidal

**Figure 7** Popup dialog for coils. Only current is required to enter, the rest will be filled in by the GUI.



Helmholtz Display Control Panel

0D/1D Display
Data Point | Line Plot

2D and 3D Display
Display for: ○ Surfaces [Select] ● Parts
Vector Field Type: ○ Magnetic Field ● Current Density Vector ○ Electric Field
Display Vector Field: ● Off ○ On
Display Vector Field Comp.: ○ Off ○ x Component ○ y Component ○ z Component ● Magnitude
[Scalar Slicing]

Field Parameters (A/m^2)
Calculated CurrentDensity: Min -100, Max 100
Displayed CurrentDensity: Min -100, Max 100
Vector Scale Factor 0.05
[Apply]
☑ Display Legend ● Horizontal ○ Vertical

Component Parameters (A/m^2)
Calculated CurrentDensity: Min 0, Max 100
Displayed CurrentDensity: Min 0, Max 100
Number of Contours 10
☑ Show Contours [Apply]
☑ Display Legend ○ Horizontal ● Vertical

Clipping State (m)
Clipping ● Off ○ On
x Min -0.74 [-0.74] x Max 0.27 [0.27]
y Min -0.3 [0.2] y Max 0.71 [0.71]
z Min -0.52 [-0.52] z Max 0.49 [0.49]
[Apply]

Initial Mesh
Display Mesh: ● Off ○ On
☐ Show Mesh Outline
[Mesh Info]

Hide this dialog temporarily: [Hide] [Exit]

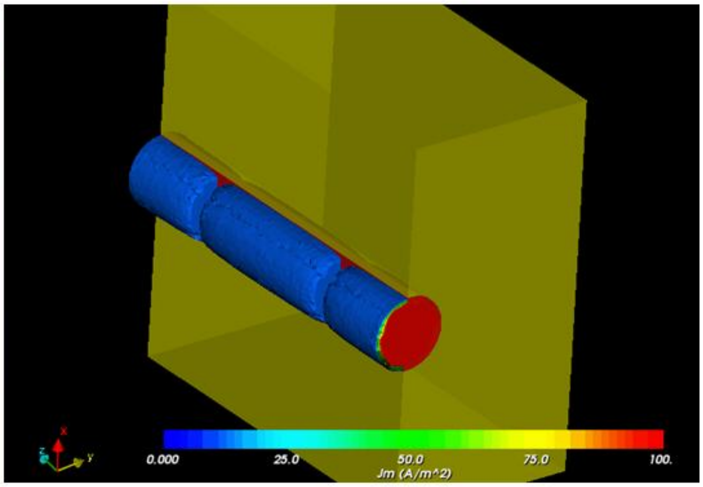


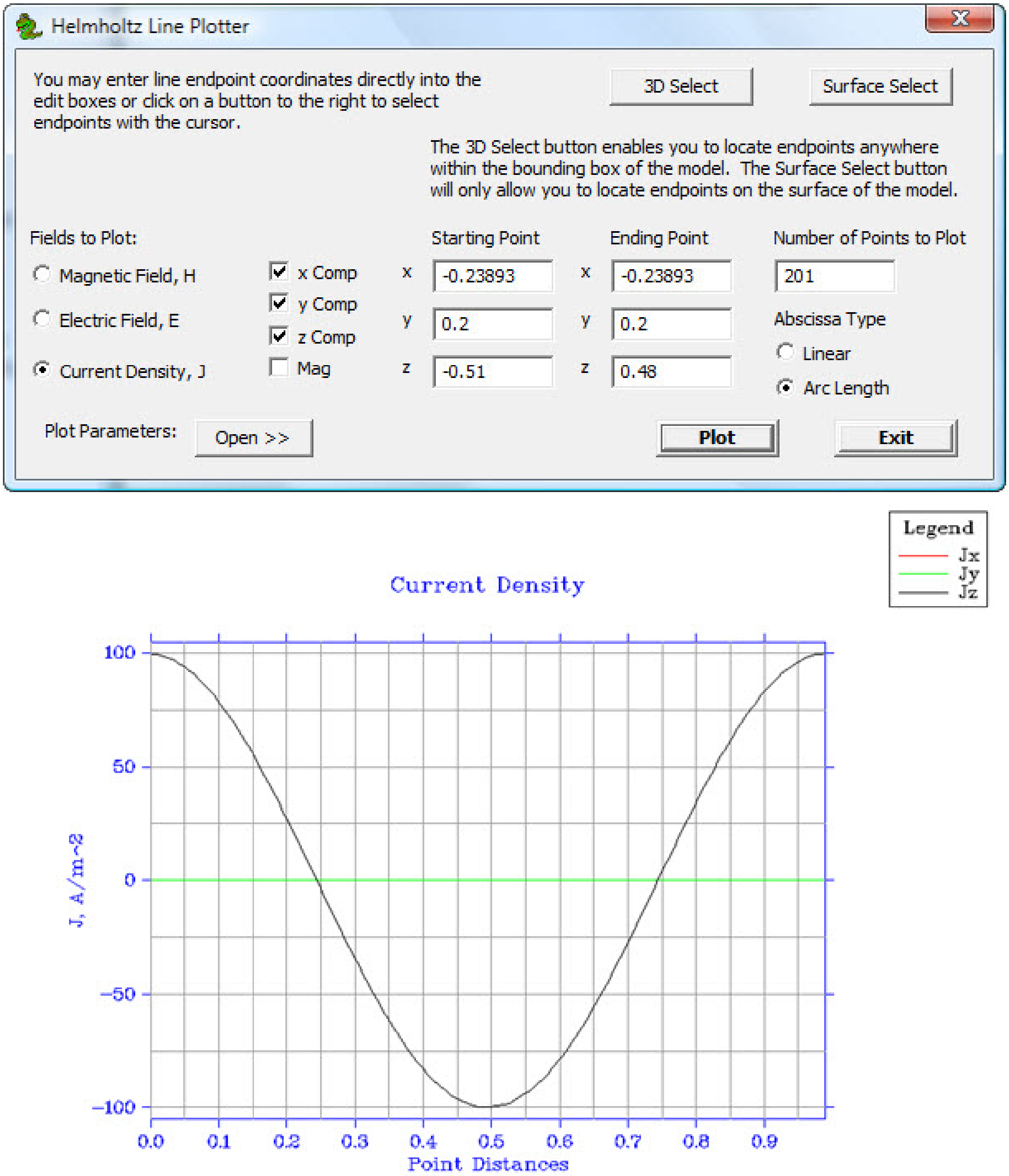
**Figure 8** Helmholtz Postprocessor Control Panel to plot fields on point, line, entity surface, arbitrary face, as gradient or contour.

## 5. *Nemesis Validation*

This task was deleted after it was determines modification of Nemesis was impractical to upgrade as planned. The emphasis was shifted to generating infrastructures for integrating a 3D large signal code with C++ and modern programming into BOA.

## Future Plans

CCR completed the GUI for specifying the temporal electrode potentials via symbolic expressions and is currently working on the LCR equivalent circuit algorithm. The latter will permit simulations of a charge-loaded, modulating grid in an IOT electron gun. The particle pusher was updated to permit simulation of beam bunching in the cathode-grid region. We plan to seek support for a fully 3D, GUI driven, large signal code.



Helmholtz Line Plotter

You may enter line endpoint coordinates directly into the edit boxes or click on a button to the right to select endpoints with the cursor.

3D Select | Surface Select

The 3D Select button enables you to locate endpoints anywhere within the bounding box of the model. The Surface Select button will only allow you to locate endpoints on the surface of the model.

Fields to Plot:

○ Magnetic Field, H

○ Electric Field, E

● Current Density, J

☑ x Comp

☑ y Comp

☑ z Comp

□ Mag

Starting Point: x -0.23893, y 0.2, z -0.51

Ending Point: x -0.23893, y 0.2, z 0.48

Number of Points to Plot: 201

Abscissa Type

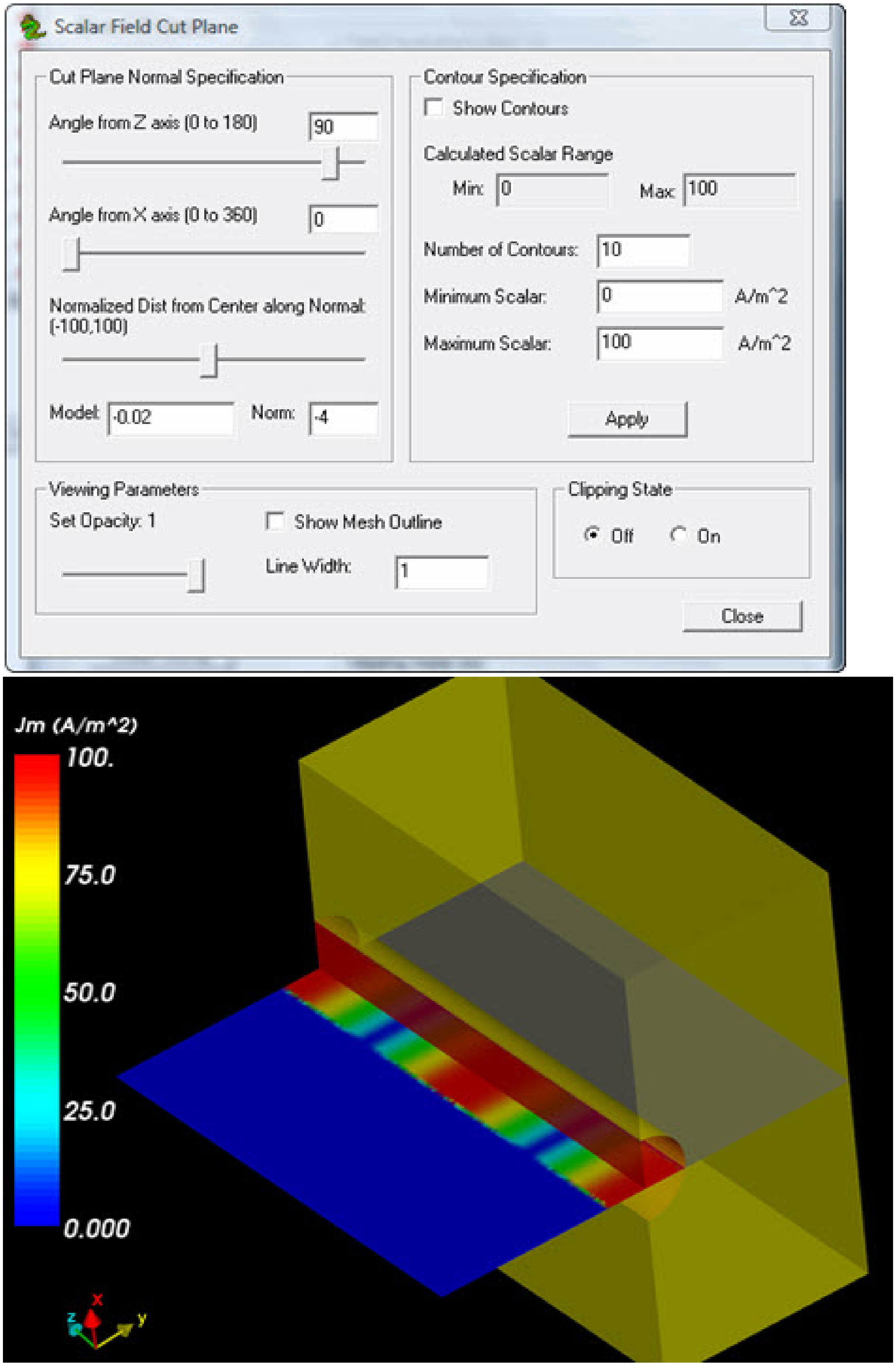
○ Linear

● Arc Length

Plot Parameters: Open >>

Plot | Exit

**Figure 9 (a)** Line plot popup dialog for the Helmholtz solver. Changing plot parameters by clicking on the Open button. **(b)** Current density along the z axis.



Scalar Field Cut Plane

Cut Plane Normal Specification

Angle from Z axis (0 to 180) 90

Angle from X axis (0 to 360) 0

Normalized Dist from Center along Normal: (-100,100)

Model: -0.02 Norm: -4

Contour Specification

□ Show Contours

Calculated Scalar Range

Min: 0 Max: 100

Number of Contours: 10

Minimum Scalar: 0 A/m^2

Maximum Scalar: 100 A/m^2

Viewing Parameters

Set Opacity: 1

□ Show Mesh Outline

Line Width: 1

Clipping State

● Off ○ On

**Figure 10** **(a)** Helmholtz popup dialog for plotting field on an arbitrary plane. **(b)** Magnitude of the current density on a cut plane.

## Summary

A major accomplishment of the Phase I program was completion of a 3D vector finite element Helmholtz solver providing time-harmonic fields for an IOT output cavity. The solver supports CAD, automeshing and adaptivity. A fully functional GUI with both preprocessor for inputting geometry, material properties and boundary conditions, and postprocessor to process and display field results was also completed. Other 3D infrastructures, such as 3D Poisson solver to model self electric field effects, 3D magnetostatic solver to analyze self magnetic fields and symbolic expression capability to permit input temporal boundary conditions, were all achieved. Although we did not follow the exact proposed plan to investigate methods and techniques to make Nemesis into a 3D large signal code, we succeeded in providing building blocks for a future 3D large signal code built in C++  using modern program techniques. The new code would be easy-to-maintain, extensible and efficient. This final product would be a single design tool for the IOT electron gun, cavities and collector.

## References

1. H.P. Freund, W.H. Miner, Jr., J. Verboncoeur, Yanxia Li, E. Wright, “Time-Domain Simulation of Inductive Output Tubes,” IEEE Trans. Plasma Sci., Vol. 35, No. 4, August 2007, pp 1081-1088.
2. R. Ives, J. Neilson, and W. Vogler, “Cascade- An advanced computation tool for waveguide system and circuit design,” IEEE Intern. Conf. on Plasma Sci., 1998.
3. B. Lewis, H. Tran, M. Read, L. Ives, “Design of an electron gun using computer optimization,” IEEE Trans. Plasma Sci., Vol. 32, No. 3, PP 1242-1250, June 2004.
4. J. Neilson, “Optimal synthesis of quasi-optical launchers for high power gyrotrons,” IEEE Trans. Plasma Sci., Vol. 34, No. 3, June 2006.
5. J.A. David, C. Kory, H. Tran, R.L. Ives, D. Chernin, “Enhanced Features for Design of Traveling Wave Tubes Using CHRISTINE-1D,” IEEE Trans. Plasma Sci., Vol. 35 No. 4, part 3, pp 1056-1064, August 2007.
6. J. David, R.L. Ives, H.T. Tran, T. Bui, and M.E. Read, "Computer Optimized Design on Electron Gun," IEEE Trans. on Plasma Science, Vol. 36, No. 1, pp. 156-168, February 2008.
7. L. Ives, T. Bui, A. Attarian, H. Tran, M. Read, M. Posht, S. Davis, S. Gadson, W. Tallis, “Implementation of Computer Optimization for Design of Electron Guns,” 33rd Intern. Conf. Infrared, Millimeter, and THz Waves, Pasadena CA, Sept. 2008.
8. L. Ives, T. Bui, A. Attarian, M. Read, J. David, H. Tran, W. Tallis, S. Davis, S. Gadson, N. Blach, D. Brown, E. Kiley, “Computational Design of Asymmetric Electron Beam Devices,” , manuscript submitted to IEEE trans. Electron Device, Special issue on Vacuum Electronics, May 2009.
9. H.G. Kosmahl, G.M. Branch, “Generalized representation of the electric fields in interaction gaps of klystron and traveling wave tubes,” IEEE Trans. Electron Devices, Vol. ED-20, No. 7, pp. 621-629, July 1973.

10. T. Bui, L. Ives, M. Read, H. Freund, "A Vector Finite Element Helmholtz Solver for Nemesis", IVEC, Monterey CA, May 2010.

11. T. Bui, L. Ives, M. Read, H. Freund, "A Vector Finite Element Helmholtz Solver", to be appeared at IRMMW-THz, Rome, September 2010.

12. S.R. Ramo, J.R. Whinnery, T.V. Duzer, "Fields and Waves in Communication Electronics," 3rd Ed., John Wiley and Sons, New York, 1994.

13. Development of an Advanced 3D Charged Particle Code with Adaptive Finite Element Meshing, U.S. Department of Energy Small Business Innovative Research Grant Number DE-FG03-00ER82966.

14. T. Bui, R. Ives, D. Datta and M. Shephard, “Beam Optics Analysis – A 3D Finite Element Charged Particle Code with Adaptive Meshing,” Paper 9.3, IVEC, Monterey, CA, USA, 2002.

15. T. Bui, R. Ives, D. Datta and M. Shephard, “Initial Operation of Beam Optics Analysis, a 3D, Charged Particle Code with Adaptive Meshing,” Paper 2A02, ICOPS, Banff, Alberta, Canada, 2002.

16. T. Bui and R. Ives, “Code Development of a 3D Finite Element Particle-In-Cell Code with Adaptive Meshing,” IVEC, Noordwijk, The Netherlands, 2005.

17. T. Bui and R. Ives, “Nonlinear 3D Magnetostatic Solver in Beam Optics Analyzer,” Paper 23.5, IVEC, Monterey, CA. USA, 2008.

18. Modeling and Simulation of a Multiple Beam Induction Output Tube, U.S. Department of Defense Small Business Innovative Research Phase I Grant Number N65538-08-M-0085.

19. R. Jackson, “Ballistic Model of Cathode-Grid Electron Transit in IOT Guns,” CCR Internal Report, November 2008.

20. MAGIC by Mission Research Corp.

21. O.C. Zienkiewicz and J.Z. Zhu, "The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique". International Journal of Numerical Methods in Engineering, 33:1331-1364, 1992.

22. O.C. Zienkiewicz and J.Z. Zhu, "The superconvergent patch recovery and a posteriori error estimates. Part 2: The recovery technique". International Journal of Numerical Methods in Engineering, 33:1365-1382, 1992.

23. N. Yan and A. Zhou, "Gradient recovery type of a posteriori error estimates for finite element approximation on irregular meshes", Computer Methods in Applied Mechanics and Engineering, Vol. 190, 2001.

24. M. Ainsworth and J.T. Oden, "A Posteriori Error Estimation in Finite Element Analysis", John Wiley & Sons (New York), 2000.

25. T. Bui, L. Ives, A. Attarian, J. David, H. Tran, M. Posth, *An Optimizer for Beam Optics Analyzer,* IVEC , Rome, 2009.

26. T. Bui, L. Ives, A. Attarian, J. David, H. Tran, M. Posth, *An Optimizer for Beam Optics Analyzer,* ICOPS-SOFE, San Diego, 2009.

27. J.A. Nelder and R. Mead, *A simplex method for function minimization*, Comput. J., 7, pp. 308-313, 1965.

28. C.T. Kelley, *Iterative Methods for Optimization*, SIAM, Philadelphia, 1999.