

## Charged particle tracking through electrostatic wire meshes using the finite element method

L. J. Devlin, O. Karamyshev, and C. P. Welsch

Citation: *Physics of Plasmas* **23**, 063110 (2016); doi: 10.1063/1.4953801

View online: <http://dx.doi.org/10.1063/1.4953801>

View Table of Contents: <http://scitation.aip.org/content/aip/journal/pop/23/6?ver=pdfcov>

Published by the *AIP Publishing*

---

### Articles you may be interested in

[Thermal conduction in particle packs via finite elements](#)

*AIP Conf. Proc.* **1542**, 539 (2013); 10.1063/1.4811987

[Electrohydrodynamic modeling for manipulation of micro/nano particles in microfluidic systems](#)

*AIP Conf. Proc.* **1472**, 155 (2012); 10.1063/1.4748083

[An investigation into the constriction flow of a particle reinforced polystyrene melt using a combination of flow visualization and finite element simulations](#)

*J. Rheol.* **54**, 1097 (2010); 10.1122/1.3478307

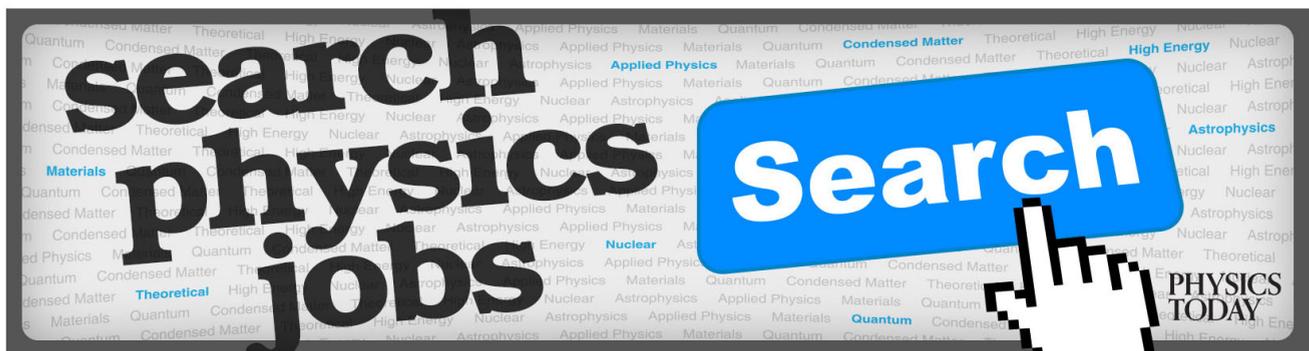
[Simulation of slowly dragging a cylinder through a confined pressurized bed of granular materials using the discrete element method](#)

*Phys. Fluids* **19**, 013301 (2007); 10.1063/1.2433738

[Advances in stabilized finite element methods for bulk and sheet metal forming processes](#)

*AIP Conf. Proc.* **712**, 14 (2004); 10.1063/1.1766494

---



# Charged particle tracking through electrostatic wire meshes using the finite element method

L. J. Devlin,<sup>1,2</sup> O. Karamyshev,<sup>1,2</sup> and C. P. Welsch<sup>1,2,a)</sup>

<sup>1</sup>The Cockcroft Institute, Daresbury Laboratory, Warrington, United Kingdom

<sup>2</sup>Department of Physics, University of Liverpool, Liverpool, United Kingdom

(Received 21 March 2016; accepted 26 May 2016; published online 15 June 2016)

Wire meshes are used across many disciplines to accelerate and focus charged particles, however, analytical solutions are non-exact and few codes exist which simulate the exact fields around a mesh with physical sizes. A tracking code based in Matlab-Simulink using field maps generated using finite element software has been developed which tracks electrons or ions through electrostatic wire meshes. The fields around such a geometry are presented as an analytical expression using several basic assumptions, however, it is apparent that computational calculations are required to obtain realistic values of electric potential and fields, particularly when multiple wire meshes are deployed. The tracking code is flexible in that any quantitatively describable particle distribution can be used for both electrons and ions as well as other benefits such as ease of export to other programs for analysis. The code is made freely available and physical examples are highlighted where this code could be beneficial for different applications. *Published by AIP Publishing.* [<http://dx.doi.org/10.1063/1.4953801>]

## I. INTRODUCTION

Wire meshes, an orthogonal array of equispaced crossing wires, are often used in charged particle transport in particle accelerators, such as in ion sources,<sup>1</sup> and are also used in mass spectrometers,<sup>2</sup> electrostatic precipitators for cleaning air,<sup>3</sup> and may be used as a retardation energy filter to study the distribution of low energy electrons.<sup>4</sup> In the first two cases, they are used to accelerate electrons and ions in a static field. In the case of a retardation filter, the exact potential at the center of the mesh cells needs to be known to achieve better resolution of the energy of particles that can pass through the filter. Electric fields around such a geometry are difficult to calculate accurately and analytically without making several crucial assumptions. The combination of multiple meshes placed at varying distances with different dimensions as well as other surrounding charged surfaces makes the solution even harder to calculate.

Finite element method (FEM) codes can be used to model complicated geometries in three dimensions and can calculate physical properties with high accuracy. Furthermore, using CAD models allows mechanical deformations to be introduced so their effects can be calculated and operational tolerances can be established. Finite element methods are limited in that an interpolation is required between nodes to obtain a value at any point in the geometry. However, with more advanced computers and solvers, it is now possible to achieve good resolution even in very narrow regions.

In Section II, a simple analytical model of the potential surrounding a wire mesh is presented and the assumptions

used to derive it are considered. Section III describes the methods in which a computational model is built to accurately model a wire mesh. Section IV describes a tracking code that uses electrostatic field data to simulate particle transport through a mesh. In Section V, a benchmark example is presented where the code has been tested with experiments.

## II. ANALYTICAL MODELING OF AN ELECTROSTATIC MESH

An analytical expression of the potential surrounding a mesh above a conducting plane can be derived by considering an infinitely long charged wire with constant charge density  $\lambda$  a distance  $z_0$  above the plane lying along the  $y$  axis.<sup>5</sup> If treated as a complex potential problem, the equation of the potential around an infinitely long charged wire can be written as

$$\phi(Y) = \frac{-\lambda}{2\pi\epsilon_0} \cdot \ln(Y), \quad (1)$$

where  $Y$  follows complex notation,  $Y = x + i \cdot z$ . Taking the real part of Eq. (1) recovers the standard equation of the potential around a wire as a function of radius:  $r = \sqrt{x^2 + z^2}$ . The imaginary term represents the force and the direction in which a particle will move in that field.

Following steps similar to Ref. 6 where a wire grid is considered instead and using the theory of image charges and potential superposition, the real potential for a wire mesh is found by summing two infinite sets of equispaced wires in two orthogonal directions,  $x$  and  $y$ , and is a function of  $\lambda$ ,  $x$ ,  $y$ ,  $z$ ,  $z_0$  and cell pitch  $s$ , given as

<sup>a)</sup>Electronic mail: carsten.welsch@cockcroft.ac.uk

$$V(x, y, z) = \frac{-\lambda}{4\pi\epsilon_0} \cdot \ln \left( \left[ \frac{\sin^2\left(\frac{\pi}{s}(x-x'_0)\right) + \sinh^2\left(\frac{\pi}{s}(z-z'_0)\right)}{\sin^2\left(\frac{\pi}{s}(x-x'_0)\right) + \sinh^2\left(\frac{\pi}{s}(z+z'_0)\right)} \right] \cdot \left[ \frac{\sin^2\left(\frac{\pi}{s}(y-y'_0)\right) + \sinh^2\left(\frac{\pi}{s}(z-z'_0)\right)}{\sin^2\left(\frac{\pi}{s}(y-y'_0)\right) + \sinh^2\left(\frac{\pi}{s}(z+z'_0)\right)} \right] \right), \quad (2)$$

where  $x'_0$  and  $y'_0$  are the positions of the central wires lying on the  $y$  and  $x$  plane, respectively. To arrive at Eq. (2) three assumptions are used: (1) The mesh is infinite in the transverse plane ( $x, y$ ). (2) The wires are one dimensional, i.e., they have no thickness. (3) A constant linear charge density is applied. Whilst the first assumption is reasonable as a mesh with many cells will converge onto an infinite mesh solution, see Section III, the second and third assumptions lead to an incorrect evaluation of the potential and field close to the wire when a constant potential is applied to the mesh rather than a constant charge to each individual wire. This often be the case when the mesh is not composed of wires but etched from a metallic sheet but still retains the overall mesh shape.

Figure 1 shows a plot of Eq. (2) calculated from the conducting plane through a cell of the mesh for two different cases, (1) through the center of a cell, i.e.,  $x = y = \frac{s}{2}$  and (2) on line with a wire in the mesh, i.e.,  $x_0 = y_0 = 0$  where in each case  $x'_0 = y'_0 = 0$ . Far from the mesh region, between the conducting plane and wire mesh the solution approaches  $V = \frac{2\lambda z_0}{s\epsilon_0}$  which is the value of the peak potential between a conducting plane and a sheet charge with surface charge density  $\sigma = \frac{2\lambda}{s}$ . The structure of the mesh is not apparent at distances greater than the cell pitch, beyond this distance the solutions converge.

A wire mesh, even though from a distance looks like a sheet, has some transparency in which field lines may permeate. Reversing the situation what has been discussed so far, by turning the conducting plane into a potential source and grounding the wire mesh, the situation is reminiscent of a Faraday cage. The amount of transparency of a Faraday cage

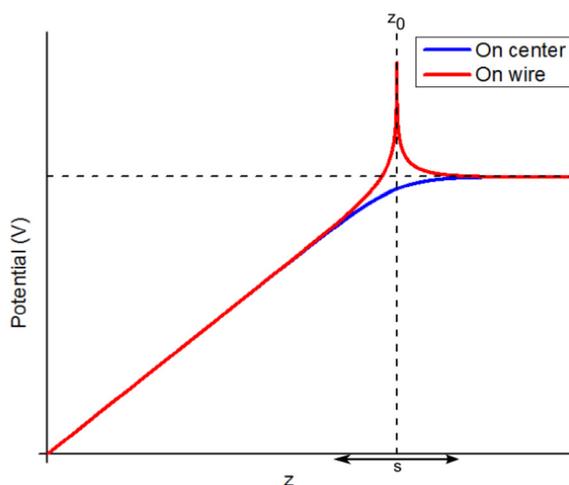


FIG. 1. Plot of the potential along a straight line between a grounded conducting plane located at the origin and a wire mesh with constant line charge at  $z_0$  for two cases, blue, through the center of the wire and red, on line with a wire in the mesh.

has been shown<sup>7</sup> mathematically to depend on the thickness of the wires and spacing between the wires in the grid/mesh. Assuming infinitesimal thickness means that the re-orientation of charges due to an external field is not accounted for. Such considerations show for more complex systems that simple analytical models are not sufficient.

### III. COMPUTER MODEL OF AN ELECTROSTATIC MESH

Finite element codes allow wire meshes to be modeled accurately and the fields around them to be calculated with good resolution.

Comsol Multiphysics<sup>8</sup> is primarily the particular solver which is presented here, however, comparative studies are also performed with the TOSCA package, the electrostatics package, of Opera.<sup>9</sup>

Figure 2 shows a wire mesh built in Comsol. The mesh is defined by three parameters: the cell pitch, the wire thickness, and the wire bend radius. In this section, a mesh with sizes given in Ref. 4 are used as a representation of a physical mesh with cell pitch, wire thickness, bend radii, and depth of 0.5 mm, 0.05 mm, 0.05 mm, and 0.035 mm, respectively.

To model a full mesh, potentially with hundreds of cells, with finite mesh elements multiple times smaller than the wire mesh it would be difficult to model each individual cell, especially for a modest desktop computer. To make the solution more approachable the electric field of the mesh for one cell is found and repeated for all cells. The mesh is first modeled as a flat potential sheet, and a single cell is removed from the center of this sheet. To attain the correct fields for the single cell, a convergence study is performed where more and more cells are included around the central single cell until the solution converges. In Fig. 2, the  $1 \times 1$  shaded

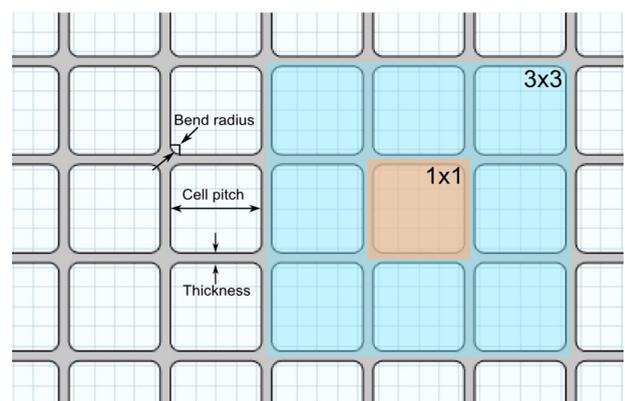


FIG. 2. Wire mesh drawn in Comsol Multiphysics. The mesh is defined by the cell pitch, wire thickness, and the bend radius. The colored boxes represent  $n \times n$  cells which are used in cell convergence studies to approximate a full mesh.

box is the one cell scenario with a higher  $3 \times 3$  configuration surrounding it. How fast the solution converges largely depends on other meshes and surrounding potential surfaces, the cell dimensions, and the voltage configurations of the system, as well as the resolution required for the application.

In the simplest case of a single mesh, a convergence study reveals only a small change when further cells are included into the model. In this simplest case, the percentage change in the potential calculated at the center of the mesh between a  $3 \times 3$  and  $5 \times 5$  configuration was 0.3% whilst from a  $5 \times 5$  to  $7 \times 7$  configuration it is 0.07%. For more complex layouts, such as two meshes held at different potentials, a greater configuration is needed with a  $9 \times 9$  structure required to reach the same improvement from a  $7 \times 7$  configuration compared to the single mesh case.

Figure 3 shows the convergence study performed for a three mesh system used for an energy filter design with applied voltages of  $-60$  V to a sheet at  $z = 0$  mm,  $0$  V to two meshes at  $41$  mm and  $43.7$  mm and  $-60$  V to a mesh at  $42.3$  mm. The inset shows the potential solution along a central cut line through the central mesh. The main plot shows the difference in potential from the preceding configuration, i.e., the difference between the  $n \times n$  solution and the  $n-2 \times n-2$  solution along the same cut line. The dashed purple lines show the longitudinal position of the three meshes. The inset shows the general solution of the potential. For every increase in the number of cells included, there is a general improvement with the change approaching ever smaller values. The largest changes occur around each mesh. The change from a  $9 \times 9$  to a  $7 \times 7$  configuration only improved the value by  $11$  mV at the center of the second mesh which was sufficient for the application and so a  $7 \times 7$  structure was used in calculations.

#### IV. TRACKING CODE

The tracking code is built using Matlab-Simulink<sup>10</sup> which is a data flow programming tool. Using this package

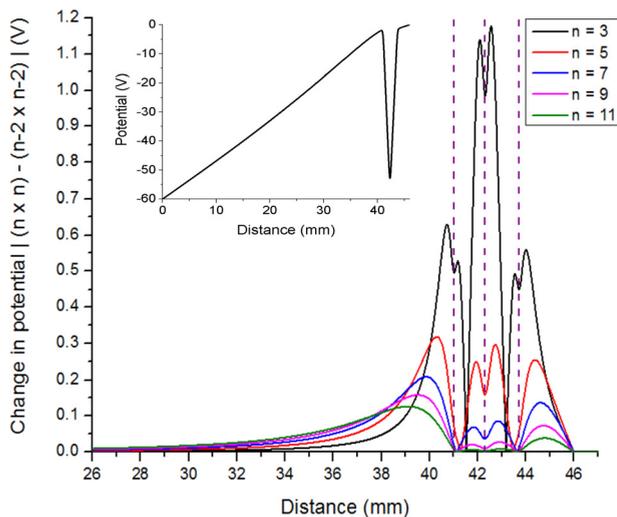


FIG. 3. Convergence study of the number of cells included into the CAD model. The inset shows the potential along a cut line through the centre of the model. The main plot shows the difference in the potential for the point calculated between that configuration and the preceding one.

has several benefits such as: the ability to simply attach or disconnect aspects of the code befitting to the situation, an expansive range of premade Matlab functions are available such as a wide range of integration methods and look up tables, functionality with traditional script algorithms and any initial profile or velocity profile may be implemented by the user. Figure 4 shows a minimalist layout of the tracking code in block diagram form.

Before the code is ran the initial positions and velocities of the particles are set and the field data are loaded and sorted into a three dimensional look up table with position breakpoints so the field data can be found during simulation. The end time is set before the simulation is started, once the integration time has reached this value the program will exit. The end time is set to be enough time to allow particles to pass through the mesh. Also before simulation is started, the integration method and integration step size is set. A fixed step Runge-Kutta (ode4) is sufficient in all applications with which the codes have been used. Higher order solvers such as Dormand-Prince (ode5) as well as variable step solvers are available; however, they offer little benefit and only increase computation time.

Particle tracking works by calculating the acceleration  $\vec{a}$  due to a force on a particle with mass  $m$  and charge  $q$  whilst in an electric field  $\vec{E}$ . This is written in the non-relativistic case as

$$\vec{a}(x, y, z) = \frac{q}{m} \cdot \vec{E}(x, y, z). \quad (3)$$

Defining the charge and mass in suitable units allows the velocities to be used in more accessible units.

When the simulation is initiated, the code looks up the electric field at that point. As the electric field map is defined by an array of finite points exported from Comsol, interpolation methods are required to evaluate the field values between these points. Two interpolation methods are available within the Simulink framework, linear and cubic spline. Linear interpolation fits a straight line between points defined in the look up table to determine the electric field at that point. A cubic spline method fits a continuous piecewise function with continuous derivatives to then determine the field at that location. The choice between linear and cubic spline is a balance between time and accuracy. To linear interpolate between a

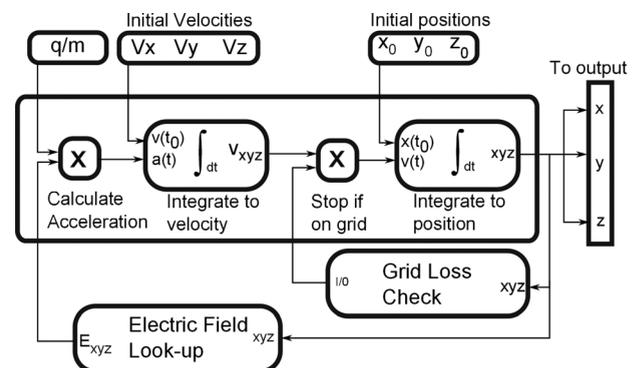


FIG. 4. Layout of Matlab Simulink tracking code.

non-linear field such as the field surrounding the meshes requires field maps with small spacings. Any small noises between points in the field map introduce errors into the calculation. Many finite element solvers, however, smooth and interpolate data to be exported at points defined by the user. Cubic spline interpolation is a more accurate method and requires less points due to higher order fitting but computation time can be much longer.

The new particle position is attained by the double time integral over a small time step using the integration method and step size selected at the start of the simulation. On the first step, the initial positions and velocities are used as initial conditions for the integration. A check is then performed to test if the particle is in the position of the mesh, a Boolean signal is then sent to adjust the velocity to zero if it is the position of the mesh before the next position step is calculated keeping it at the same position until the simulation ends. On each step of calculating the new position the output, and indeed any other calculated parameter, velocity for instance, can be sent to the Matlab workspace or saved to file or both.

Since a full mesh is not calculated in the FEM solver, several field maps are required which the code will switch between depending on the position co-ordinate. Far from the mesh, a field map which covers the whole system is used in which the mesh is mostly modeled as a flat plate with several cells included. Closer to the mesh, the code uses the single cell solution field map which needs to be used for all cells of the mesh. To do this, an algorithm is implemented that scales the transverse position of the particle as if it were at the same point in the single cell mesh solution. The sudden shift between field maps can have a detrimental effect if the electric field is not flat at the switching point, as particles will experience a discontinuity in force. This effect is more pronounced on the transverse velocity component so the overall effect is small if there is a high acceleration in the longitudinal frame, however, to reduce the effect as much as possible the switching point needs to be carefully selected.

The code was developed for an application where space charge effects were not required to be taken into account, however, a space charge algorithm may be implemented in the here presented code with a standard particle to particle technique, calculating the force due to every other particle, being the most suitable choice.

## V. BENCHMARK AND APPLICATIONS

The code was developed to study a multi-mesh system which filters electrons based on their initial energy when emitted from a photocathode.<sup>4,5,11</sup> In this system, two meshes are placed between a photocathode and detector with the first mesh held at a potential that will accelerate particles from the cathode source towards the mesh and the second mesh at a potential that will stop particles passing through which have insufficient energy to do so. The general potential layout of this system was shown in the inset of Fig. 3. The potential of the second (filter) mesh is then scanned to measure the detected electron intensity as a function of filter energy until no signal is recorded.

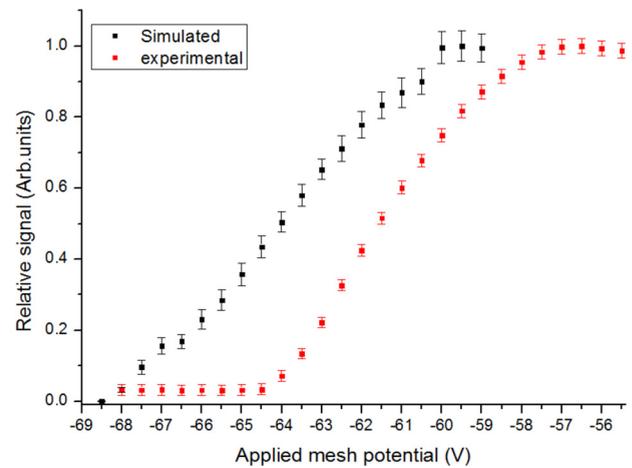


FIG. 5. Benchmark of simulated data with experimental output of a low energy electron filter designed to measure the initial energy spread of electrons emitted from a photocathode.

Figure 5 shows a comparison between results obtained from experimentation and also from using the code. Whilst there is a clear shift of a couple of volts between the two, this is due to the contact potential difference between the materials of the photocathode and meshes which were connected to the same power supply unit. This was not taken into account in the code, however, the shift is constant and may be subtracted. It is clear that the general trends are in agreement with differences in shape attributed to instrumental responses and the initial velocity distributions assumed in the simulation. By using this code, the importance on various different parameters can be found such as misalignment in placement of the meshes and the initial launch point of electrons from the cathode which are the main source of uncertainty in simulation results.

The code presented here may be used to simulate many applications in which wire meshes are used. A system of a single or multiples wire meshes are typically used to accelerate particles towards an area and to also focus them in applications such as electron sources, time of flight mass spectrometers<sup>12</sup> but they can also be used in spacecraft propulsion.<sup>13</sup> In all of these cases, the beam profile once it has passed through the mesh can be modeled to optimize these devices.

The code is made available<sup>14</sup> so the code may be inspected and used. Furthermore, an example is prepared in which electrons may be tracked through a mesh field.

## VI. CONCLUSIONS

This article presents an easy-to-implement tracking code of charged particles through a wire mesh structure. The electric field and associated potential around a wire mesh is impossible to calculate accurately when trying to account for physical sizes of a mesh. With desktop computers becoming more powerful, it is now possible to model the intricate details of thin wire meshes using the finite element method. A particle tracking code written in Matlab-Simulink has been presented which can be used to track charged particles with any initial position and velocity distribution through

wire meshes. Since a full mesh cannot be modeled on a desktop computer, an algorithm is implemented which uses one mesh cell for all cells of the mesh. The code is made available online<sup>14</sup> with field data included as an example.

## ACKNOWLEDGMENTS

Work supported by STFC Cockcroft Core Grant No. ST/G008248/1.

<sup>1</sup>F. Heymarch, M. Galonska, R. Hollinger, K. D. Leible, P. Spadtke, M. Stork, and E. Oks, "Technical design of the mevva ion source at gsi and results of a long uranium beam time period," in *Emerging Applications of Vacuum-Arc-Produced Plasma, Ion and Electron Beams*, NATO Science Series Vol. 88 (Springer, Netherlands, 2002), pp. 59–65.

<sup>2</sup>C. Herbert and R. Johnstone, *Mass Spectrometry Basics* (CRC Press, 2002).

<sup>3</sup>J. Vincent, "Note on the grid-type electrostatic precipitator," *J. Air Pollut. Control Assoc.* **22**, 200–201 (1972).

<sup>4</sup>L. J. Devlin, L. Jones, O. Karamyshev, B. L. Milityn, T. Noakes, and C. P. Welsch, "Measurements of the longitudinal energy distribution of low energy electrons," in Proceedings of IPAC2014 (2014).

<sup>5</sup>L. J. Devlin, "Studies towards an enhanced understanding of electron beams and their diagnostics," Ph.D. thesis (The University of Liverpool, 2016).

<sup>6</sup>W. Blum, W. Riegler, and L. Rolandi, *Particle Detection with Drift Chambers* (Springer, 2008).

<sup>7</sup>S. J. Chapman, D. P. Hewett, and L. N. Trefethen, "Mathematics of the Faraday Cage," *SIAM Review* **57**(3), 398–417 (2015).

<sup>8</sup>See [www.comsol.com/](http://www.comsol.com/) for Comsol Multiphysics.

<sup>9</sup>See <http://operafea.com> for Cobham.

<sup>10</sup>See <http://mathworks.com/> for Mathworks.

<sup>11</sup>L. B. Jones, R. J. Cash, B. D. Fell, J. W. McKenzie, K. J. Middleman, B. L. Milityn, H. Scheibler, D. Gorshkov, and A. Terekhov, "The commissioning of tess: An experimental facility for measuring the electron energy distribution from photocathodes," in Proceedings of FEL2013 (2013).

<sup>12</sup>M. Guilhaus, "Principles and instrumentation in time-of-flight mass spectrometry," *J. Mass Spectrom.* **30**, 1519–1532 (1995).

<sup>13</sup>E. Choueiri, "New dawn for electric rockets," *Sci. Am.* **300**, 58–65 (2009).

<sup>14</sup>See <https://goo.gl/ym8hO7> for see (url) for access to the code and prepared example.