NEDlab™

Reference Manual

This manual reflects *NEDlab* Version 1.27 software

This edition updated in October 2000 by AccelSoft Inc.

AccelSoft Inc.
a subsidiary of G. H. Gillespie Associates, Inc.
P.O. Box 2813
Del Mar, California, 92014
(858) 677-0133

Third Printing: October 2000

Printed in the United States of America

Abstract

The Numerical Electrodynamics Laboratory program, *NEDlab*™, is a 2.5 dimensional (cylindrical or Cartesian) particle and field simulation program featuring a (Graphical User Interface) GUI for highly interactive problem solving. Field solutions are Laplace, Poisson, Poisson-Boltzmann (for plasma sheath extraction problems), or full Maxwell equations, while particles are treated with multi species particle-in-cell, ray trace, or fluid modules. The ray trace and fluid algorithms may be iterated to self-consistency between applied fields and self fields of the particles. Complex antenna, conductor, and emitter geometry may be input by *NEDlab*'s graphical draw/paint tools, from scans, or other commercially available drawing packages. Integrated plotting capability and help screens are readily available from menus. *NEDlab* has been applied to electron guns, positive and negative ions sources, rf tubes, accelerator cavities, waveguides, and other problems.

Table of Contents

Preface

The ***NEDlab* Reference Manual** provides a complete description of *NEDlab*, the Macintosh software package for 2.5 D particle and field simulation.

NEDlab runs on the Macintosh and this reference assumes that you are familiar with this computer, or have access to the Macintosh Reference manual. That manual, available from Apple Computer, provides step-by-step instructions for all the basic tasks you can perform with a Macintosh.

This reference also assumes that the reader has an engineering or physics background and is familiar with the physics of intense charged particle beams interacting with electromagnetic fields, as well as some of the devices based on these principles.

How To Use This Manual

This Manual is divided into two parts: **Algorithms** (chapter 2) and the **User's Guide** (chapter 3). Unless you are already familiar with the numerical algorithms associated with the simulation of intense charged particle beams and electromagnetic fields, you should start your introduction to *NEDlab* by reading the **Algorithms** chapter. The **User's Guide** can then be consulted for the practical details of how to run *NEDlab* on the Macintosh computer.

Hardware Requirements

NEDlab requires a PowerPC Macintosh running System 8.x or later, or a 68020 or better Macintosh computer running system 6.04 - 7.x. For a 680x0 computer, a numeric coprocessor (floating point unit, or FPU) is recommended. Both FPU and non-FPU versions are available. (Note: PowerPC-based Macintosh users should use the non-FPU version.) Because of the emphasis on graphic output, a color monitor is recommended.

1 Introduction

NEDlab is a computer program written to aid the design of devices that utilize intense beams of charged particles interacting with electromagnetic fields, such as electron guns, ion sources, klystrons, linacs, etc. The program feaures a set of numerical solution modules that all operate on the same finite difference grid. The software, which features a graphical user interface (GUI), has been designed to emphasize and enhance the interaction of the user and the solution process. The Mesh Editor, in particular is a uniquely intuitive way to enter complex conductor geometry.

The *NEDlab* "Philosophy"

Anyone who has used one of the standard particle and field codes (e.g. PIC codes) on a mainframe computer to design an ion source, accelerator cavity, or microwave tube, knows that during the design process it is often possible to become detached from the lengthy cycle of input/output. It is easy to forget why you made changes in the input, by the time a problem has been run and re-run. These codes have their place, and in principle are more accurate for some jobs, but it comes at the penalty of simplicity, speed, and comprehension. In fact at the present time, PIC codes are impractical for most *design* jobs, and are most useful after several preliminary iterations with a quicker code.

NEDlab was written to emphasize user interactivity and quick graphical response in the solution process, with less emphasis on accuracy. Screen graphics are displayed repetitively at each time step, or iteration interval, as set by the user. This allows the user to watch the system simulation evolve in time while it is being calculated.

Another manifestation of this philosophy is the unique feature of *NEDlab* that allows the user to stop the code at any time-step and change the grid geometry, applied potentials, current density, or nearly any other physical quantity of interest, and continue the iterations *from where the calculations were left off* to observe the effect of the change, in "real time". This also allows the user to investigate transient responses of the system, as well. One example of this might be a plasma density fluctuation in a negative ion source which could cause the space charge of the extracted beam to change, with a subsequent increase in emittance. Other examples might be the turn-on of an electromagnetic antenna, or the sudden placement of a conducting object in a wave quide or resonant cavity. The goal of *NEDlab* is to allow the user to try things out almost as quickly as they are thought of.

A Set of Simulation and Diagnostic Tools

There are several modules in *NEDlab* that furnish the user with different algorithms to attack a problem with. The user has the freedom and responsibility to choose the algorithm most appropriate to the parameter regime of the physical system or device to be modeled. Inappropriate use of certain algorithms can lead to numerical instabilities. A classic example the case of a time-dependent treatment of a light mass species, with too large a time step selected, in violation of the Courant condition. To remedy this, the user must either make the time step smaller, or use an iterative-static-equilibrium (ISEQ) type solution for the low mass species at each time step (i.e. the raytrace option).

Stop Any Time... and Start Again

Clicking the mouse button at any time stops a calculation (there may be a slight delay until a convenient break is reached, e.g. ray tracing will not stop until the ray reaches a boundary). You can start again where you left off in the iteration or time stepping process (unless you zero an array by pulling down the Edit... Zero Arrays... menu item) .

Summary of *NEDLab* Features

A complete 2.5 D particle and field simulation featuring an intuitive, graphical user interface

Easy input of complex geometry using...

- most commercially available drawing programs
- scanned images
- *NEDlab's* own Mesh Editor mouse-driven tools (pencil, eraser, paint bucket, etc.)
- Cartesian or cylindrical symmetry with the click of a button

Multi species particle-in-cell or ray trace (static) modules...

- flexible particle emission from any irregularly shaped surfaces or volumes
- Maxwellian distribution, or a monoenergetic drift
- all phase space plots available immediately (transverse and longitudinal)

Time dependent (full Maxwell's equations), electrostatic or quasi-electrostatic (LaPlace, Poisson, Poisson-Boltzmann) solved...

- time-dependent (sinusoidal) applied voltages
- arbitrary antenna geometry
- vector, color map, and contour plots of all fields

Highly interactive problem solving - stop simulation at any time with mouse button, change parameters (e.g. conductor shapes, voltages) and continue. Export of fields and particle coordinates for processing by your favorite spreadsheet or custom program.

Online help screens... just push a button.

2 Algorithms

This chapter describes the physics and numerical algorithms used in *NEDlab*.[1]

2.1 Geometry

The solution modules operate on the same finite difference mesh. The modules can be categorized as either static (time-independent) solutions, and dynamic (time dependent solutions).

Since *NEDlab* is an integrated laboratory of numerical modules, they can be in cases where it is physically appropriate, static solutions and dynamic solutions can be used in the same overall solution. (An example of this is a system of ions and electrons: in a time dependent problem, the electrons almost immediately establish an equilibrium (static solution), while the ions will respond more slowly. A reasonable approximation (if short term fluctuations of the electrons is not an issue) is to use the iterative static equilibrium (ISEQ) solution for the electrons, while the ions would use the dynamic (PIC) algorithm. *NEDlab* allows the use of both modules in the overall solution of the problem.

> **Note on Units**: the equations in this chapter are mostly in the MKS system of units. Temperatures are in Volts.

The origin of the coordinate system used in *NEDlab* is in the upper left corner of the computer screen. The *x*-axis goes horizontally from left to right, while the *y*-axis (or *r*, in cyclindrically symmetric problems) goes vertically from top to bottom. The *z*-axis is into the plane of the screen (see Fig. 2.1). On the finite difference mesh, the integer pair (*i* ,*j*), where *i*=1..*iMax* and *j*=1..*jMax*, is associated with the position (*x*,*y*), where *x*=(*i*-1)*dx* and *y*=(*j*-1)*dy*. Some of the discussions in this chapter (and in particular the Poisson-Boltzmann equation solver of the algorithms) implicitly assume that the beam is directed along the *x*-axis (there is no paraxial approximation made in *NEDlab*, however).

[1] The reader may be interested in the references listed at the end of this chapter, which describe algorithms used in a representative list of some other codes.

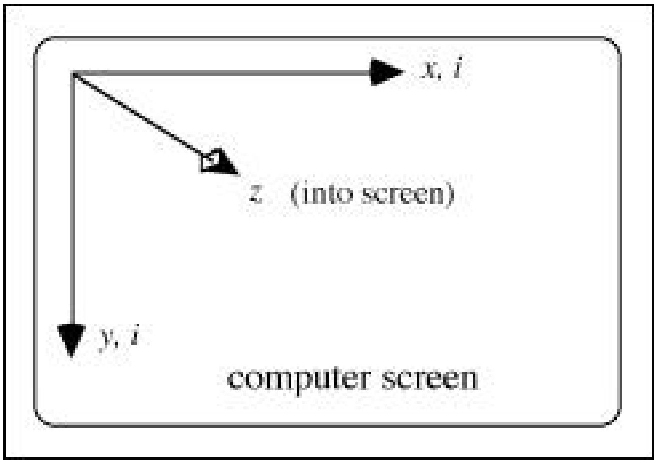


Fig. 2.1 The *NEDlab* coordinate system.

2.2 Static Solutions

2.2.1 Laplace Equation

In cases where there is negligible space charge, or where it is of interest to obtain the electtrostatic field from the applied potentials only, the Laplace equation may be solved. The potential φ is obtained by solving

$$^2\varphi = 0 \qquad [2.1]$$

by succesive-over-relaxation (SOR) on a the finite difference grid with tthe boundary conditions imposed at conductor surfaces. The following replacement

$$\alpha\left\{\frac{(\varphi_{i-1,j}+\varphi_{i+1,j})}{2(1+dy^2/dx^2)}+\frac{(f_-\varphi_{i,j-1}+f_+\varphi_{i,j+1})}{2(1+dx^2/dy^2)}\right\}+(1-\alpha)\varphi_{i,j}\rightarrow\varphi_{i,j} \qquad [2.2]$$

is iterated over the finite difference mesh a user-specified number of times, where *dx* and *dy* are the spatial mesh increments.

For Cartesian symmetry

$$f_+ = 1, \quad f_- = 1 \qquad [2.4]$$

and for cylindrical symmetry ($y = r$)

$$f_+ = 1 + \frac{2}{j}, \quad f_- = 1 - \frac{2}{j} \qquad [2.5]$$

The over-relaxation factor α is set by the user and usually satisfies (1<α<2).

2.2.2 Poisson Equation

The Poisson equation solver finds the potential for both the applied field and the particle self-fields from the particles. The finite difference form of

$${}^{2}\varphi = -\varepsilon_o^{-1} \sum_{speciesn} \rho_n . \qquad [2.6]$$

is solved by SOR on a the mesh with tthe boundary conditions imposed at conductor surfaces. Looping over the mesh *i,j* the replacement

$$\frac{\alpha}{2}\left\{ \frac{(\varphi_{i-1,j} + \varphi_{i+1,j})}{1 + dx^2 \,/\, dy^2} + \frac{(f_-\varphi_{i,j-1} + f_+\varphi_{i,j+1})}{1 + dy^2 \,/\, dx^2} + \frac{\rho_{i,j}}{\varepsilon_o} \right\} + (1-\alpha)\varphi_{i,j} \;\rightarrow\; \varphi_{i,j} \qquad [2.7]$$

is repeatedly performed for a specified number of iterations, where α, f_+, and f_- are defined in the section above on the Laplace equation.

Note on Indices: As noted previously, the integer pair *i ,j* are the indices of a given matrix, and is associated with the position (*x,y*) on the finite difference mesh. To indicate species, the integer *n* (and sometimes *m*) are generally used.

2.2.3 Poisson-Boltzmann Equation

When a plasma consisting of ions and a thermal distribution of electrons with temperature *T* is present, such as in the extraction region of an ion source, the potential may be determined by the Poisson-Boltzmann equation,

$${}^{2}\varphi = -\varepsilon_o^{-1}\left\{ \sum_{species} \rho_n + \rho_{eo} e^{e(\Phi_o - \varphi)/T_e} \right\} \qquad [2.8]$$

The first source term is the sum of all the ions and non-thermal electrons, while the second term is the nonlinear Boltzmann term for the thermal electron density.

This equation is solved, given the source terms, by the SOR method, with a nonlinear Newton-Raphson limit finder technique for the Boltzmann term. Looping over the mesh indices i and j, the following substitutions are made

$$\frac{1}{2}\left\{\frac{(\varphi_{i-1,j}+\varphi_{i+1,j})}{1+dx^2/dy^2}+\frac{(f_-\varphi_{i,j-1}+f_+\varphi_{i,j+1})}{1+dy^2/dx^2}+\frac{\rho_{i,j}}{\varepsilon_o}\right\} \rightarrow \varphi^o \qquad [2.9]$$

$$\frac{\left\{1+\frac{\varphi^o}{T}\right\}C_o e^{-\varphi^o/T}+\varphi^o}{1+C_o e^{-\varphi^o/T}} \rightarrow \varphi^1 \qquad [2.10]$$

$$\alpha\varphi^1+(1-\alpha)\varphi^1 \rightarrow \varphi_{i,j} \qquad [2.11]$$

where

$$C_o=\frac{-e^{e\Phi_0/T}\rho_o}{2(dx^2+dy^2)} \qquad [2.12]$$

and Φ_0 and ρ_o are the (plasma) potential and density in the "center of the plasma". The left side of the simulation (i=1; j=1 to jMax) is assumed to be the "center of the plasma" . As implemented in *NEDlab*, ρ_o is calculated as the average

$$\rho_o=\frac{1}{jMax}\sum_{species}\sum_{j=1}^{jMax}\rho_{1,j} \qquad [2.13]$$

and $e^{e\Phi_0/T}$ is set by the user.

The source terms (excluding the term for the thermal electrons) result from the particles, whose motion is tracked and charge density deposited (accumulated), in the overall charge density matrix.

Note that the debye length λ_D must be several times larger than the larger of dx or dy. where

$$\lambda_D=\sqrt{\frac{T\varepsilon_o}{e\rho_o}} \qquad [2.14]$$

to model the extraction sheath accurately. The Poisson-Boltzmann equation solver should only be used when

$$\lambda_D \gg dx,\ dy \qquad [2.15]$$

If λ_D is smaller than dx or dy then the approximate Poisson-Boltzmann technique should be used as described below.

2.2.3.1 Approximate Poisson-Boltzmann

In cold, high density plasmas, the screening may happen over an impractically small interval, and [2.12] cannot be satisfied. Then the gross effect of the plasma sheath can be modelled by using the Poisson equation solver step [2.7x] modified by the conditional statement

$$if\ \varphi_{i,j} < \varphi_{floor}\ then\ \varphi_{floor} \rightarrow \varphi_{i,j} \qquad [2.16]$$

(for negative ion extraction). This has the effect of "filling up with plasma" the regions where the space chage of the (negative) extracted beam is too intense (i.e. where it has not reached sufficient speed in the extraction/acceleration process). The floor potential φ_{floor} is set by the user and may be identified with the plasma potential Φ_0 described in the section on the Poisson-Boltzmann solution.

Important Note: the statement in [2.13] is always operative in all the electrostatic solvers (Laplace, Poisson, and Poisson-Boltzmann). To disable its effect, the floor potential φ_{floor} must be set to a very large negative number[2].

2.2.4 Electrostatic Boundary Conditions

The (Dirichlet) boundary condition is used for the potential φ on mesh points (i,j) in conductor

$$V_k \rightarrow \varphi_{i,j} \qquad [2.17]$$

where $V_k$is the voltage applied to the conductor. (In *NEDlab* the Mesh Edit Mode is used to "paint" the geometry of the conductors, and each voltage k has a color

[2] In the present version of the program there is no ceiling potential check, which would be appropriate for modelling positive ion extraction from a plasma.

associated with it from a list entered by the user).

On the boundary of the rectangular simulation box, the (Neumann) boundary condition is applied

$$\frac{\partial \varphi}{\partial n} = 0 \qquad [2.18]$$

(Unless the point has been painted as conductor, in which case the Dirichlet boundary condition is applied). In finite difference language, along the top boundary,

$$\varphi_{i,2} \rightarrow \varphi_{i,1} \qquad i = 1..iMax;\ (i,1)\ not\ \ conductor \qquad [2.19]$$

along the bottom boundary,

$$\varphi_{i,\,jMax-1} \rightarrow \varphi_{i,\,jMax} \qquad i = 1..iMax;\ (i, jMax)\ not\ \ conductor \qquad [2.20]$$

along the left boundary,

$$\varphi_{2,\,j} \rightarrow \varphi_{1,\,j} \qquad j = 1..jMax;\ (1, j)\ not\ \ conductor \qquad [2.21]$$

along the left boundary,

$$\varphi_{iMax-1,\,j} \rightarrow \varphi_{iMax,\,j} \qquad j = 1..jMax;\ (iMax, j)\ not\ \ conductor \qquad [2.22]$$

This is shown graphically in Fig. 2.2.

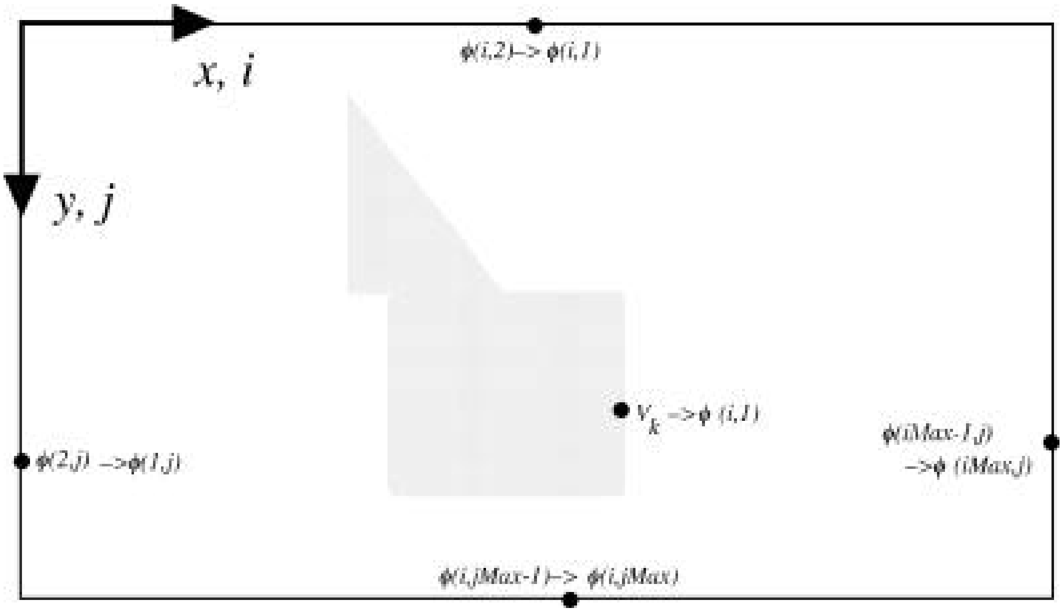


Fig. 2.2 Boundary conditions on potential.

2.2.5 Particle Trajectories

In static problems, the term "ray" refers to a particle trajectory in space, with an associated electric current.

There are two different trajectory solvers in *NEDlab* . One is appropriate for non-relativistic problems with cartesian symmetry and is a very quick and conserves energy accurately. The other trajectory algorithm, appropriate for cylindrical symmetry or relativistic cases, is slower, and itegrates 3D trajectories.

Quick Trajectory Algorithm	**Relativistic Trajectory Integrator**
2D Cartesian	3D Cartesian (projected onto 2D Cartesian or 2D cylindrical)
nonrelativistic only	relativistic

Rays terminate at the first conducting point encountered (within a distance of one half *dx* or *dy*).

2.2.5.1 Quick Trajectory Algorithm

In non-relativistic problems with Cartesian symmetry, a unique quick trajectory algorithm (or "cell hop algorithm") has been formulated for *NEDlab* to integrate the trajectories. The particles "hop" from mesh line to mesh line. A brief discussion of the method is presented here.

Instead of taking very small integration steps, the quick trajectory integrator solves exactly for the trajectory over a mesh rectangle (*dx* by *dy*). The potential at the four corners of the mesh rectangle, or "cell", defined by the mesh points (i, j) and (i+1, j+1) determine the parameters of a quadratic potential surface over the rectangle

$$\varphi(x,y) = \varphi_{i,j} - Hy - Gx - kxy \qquad [2.23]$$

The equations of motion are then

$$\frac{d^2x}{dt^2} = G + ky + \Omega\frac{dy}{dt} \qquad [2.24]$$

$$\frac{d^2y}{dt^2} = H + kx - \Omega\frac{dx}{dt} \qquad [2.25]$$

where the magnetic field B is assumed constant over the cell and

$$\Omega = qB_z / m. \qquad [2.26]$$

The solution for *x(t)* over the cell can be shown to be of the form[3]

$$x = ae^{\mu t} + be^{-\mu t} + ce^{i\nu t} + de^{-i\nu t} - \frac{H}{k} \qquad [2.27]$$

with a similar expression for *y(t)*

$$y = Ae^{\mu t} + Be^{-\mu t} + Ce^{i\nu t} + De^{-i\nu t} - \frac{G}{k}. \qquad [2.28]$$

The constants are determined by the initial conditions (where the particle enters the

[3] In the present version of the code, Ω is set to zero and the particle is rotated in the magnetic field at the end of each cell hop.

cell).

The next step is to advance the particle to a cell boundary. To do this, a procedure in *NEDlab* is called that estimates a time for the particle to hit each of the mesh lines. The time is determined by the Newton method for solution of a nonlinear equation is used. After an initial guess for the time t^{old}, the following expression is iterated

$$t^{old} - \frac{x - x_{goal}}{v_x} \;\rightarrow\; t^{new}; \quad t^{new} \;\rightarrow\; t^{old}; \quad etc.... \qquad [2.29]$$

where x is from [2.14] and v_x from its time deriviative[4], and x_{goal} is the coordinate of a mesh line perpendicular to the x-axis (see Fig. 2.1). A similar procedure is done with y to estimate times to the horizontal crossings. Four times are found (not all real and positive) from which the next place for the particle to "hop" is determined.

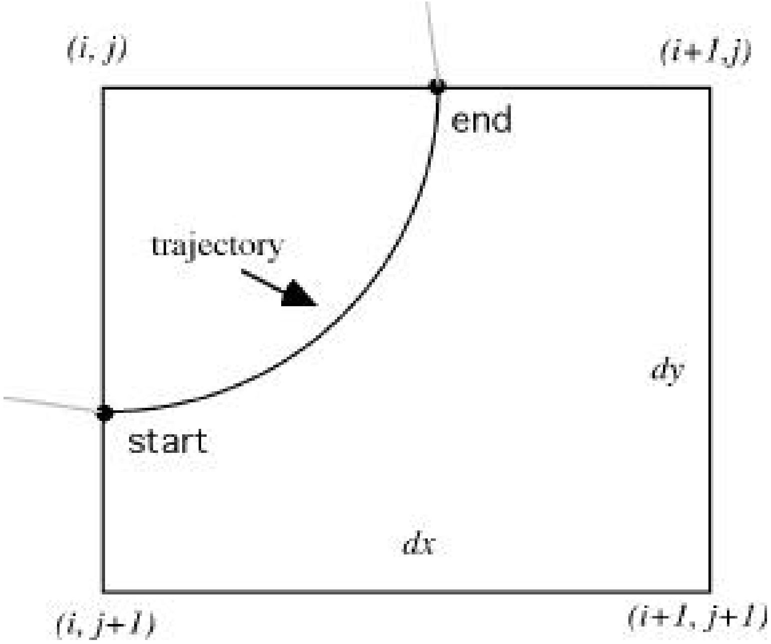


Fig. 2.3 Integration of a trajectory over a cell.

When reasonable convergence is obtained, the time is returned. If there is no real solution, then a flag is set. This is done for each of the possible edges of the cell

[4] The present coding uses a power series expansion to sixth order of eq. [2.14].

that the particle may hit. The shortest time is then used to determine what cell boundary will be hit next. In Fig. 2.2, the trajectory starts on a vertical cell boundary, and ends on a horizontal boundary. Particles can turn around, spiral, etc, (i.e. no paraxial approximation is used). Furthermore, when the particle hits the boundary, its kinetic energy is set by its position and the potential on the grid line (which is linear between points on horizontal or vertical lines, according to eq. [2.14]). This insures that energy is conserved.

2.2.5.2 Relativistic Trajectory Integrator

If the trajectories are relativistic, or if cylindrical symmetry is selected, then the relativistic trajectory integrator must be used.

The static part of the electric field **E** then obtained from the gradient of φ:

$$\mathbf{E} = - \quad \varphi \quad . \qquad [2.30]$$

When integrating the trajectory of a particle whose coordinates are between mesh point i..i+1, and j..j+1, i.e.

$$(i-1)dx < x < idx \qquad [2.31]$$

$$(j-1)dy < y < jdy \qquad [2.32]$$

finite difference form used for the electrostatic field is

$$E_x = (\varphi_{i,j} - \varphi_{i+1,j}) / dx \qquad [2.33]$$

$$E_y = (\varphi_{i,j} - \varphi_{i,j+1}) / dy \qquad [2.34]$$

for Cartesian coordinates.

If cylindrical coordinates have been selected, the trajectories are actually integrated in 3D cartesian coordinates using

$$E_y = (\varphi_{i,j} - \varphi_{i,j+1})\cos(\theta)/ dr \qquad [2.35]$$

$$E_z = (\varphi_{i,j} - \varphi_{i,j+1})\sin(\theta) / dr \qquad [2.36]$$

where $dr = dy$. (In 2D cartesian problems, z is always set to zero). The radial coordinate r is then obtained from y and z by

$$r = \sqrt{y^2 + z^2} \qquad [2.37]$$

The force on the particles from the total electric field the applied magnetic field is used in the calculation of the trajectory of species-*m*[5]

$$\frac{d}{dt}\gamma\frac{d\mathbf{x}_n}{dt} = \frac{q_n}{M_n}[-\quad \varphi + \mathbf{E}^{em} + \frac{d\mathbf{x}_n}{dt} \times (\mathbf{B}^{applied} + \mathbf{B}^{em})] \qquad [2.38]$$

in a manner similar to ref. [7].

Note: the electric force in eq. [2.38] is separated into two pieces, the static part and the dynamic part. The array that holds the potential φ (from the solution either of eq. [2.1], [2.6], or [2.8]) is separate from the $\mathbf{E}^{em}$ arrays which result from the solution of eq. [2.80] and [2.81]. Similar remarks apply to the magnetic field where $\mathbf{B}^{eapplied}$ is the static field applied, and $\mathbf{B}^{em}$ is the dynamic field (which can result from the particles themselves as well as time varying fields applied from antennae). In purely static problems, the user should make sure the $\mathbf{E}^{em}$ and $\mathbf{B}^{em}$ arrays are zero.

The method consists of first applying one half of the electric field impulse, rotating the particle around the magnetic field, and then applying the final half of the electric field impulse (see Fig. 2.2). This algorithm follows that of ref.[7].

The following steps are taken to advance the particle one time step δt

$$\mathbf{p}_a = \gamma\mathbf{v} + \frac{q\delta t}{2M}\mathbf{E} \qquad [2.39]$$

$$\frac{\alpha}{2} = \frac{q\delta t}{2M}\mathbf{E}\frac{c}{\sqrt{c^2 + \mathbf{p}_a{}^2}} \qquad [2.40]$$

$$\frac{\beta}{2} = \frac{\alpha}{2}|\mathbf{B}| \qquad [2.41]$$

[5] In cylindrical problems, the effect of the self-electric force partially cancelled by the self-magnetic force is taken care of by depositing the actual space charge density multiplied by $1/\gamma^2$. This is only valid for single species or the special case of multiple species with the same γ. Note that the PIC (dynamic) solution does not suffer this limitiation.

$$f_1 = \frac{\alpha}{2}\left[1 + 0.31755\left\{\frac{\beta}{2}\right\}^2 + 0.2033\left\{\frac{\beta}{2}\right\}^4\right] \qquad [2.42]$$

$$\mathbf{p}_b = \mathbf{p}_a + f_1 \mathbf{p}_a \times \mathbf{B} \qquad [2.43]$$

$$f_2 = \frac{2 f_1}{\sqrt{1 + f_1{}^2 |\mathbf{B}|^2}} \qquad [2.44]$$

$$\mathbf{p}_c = \mathbf{p}_a + f_2 \mathbf{p}_b \times \mathbf{B} \qquad [2.45]$$

$$\mathbf{p}_{new} = \mathbf{p}_c + \frac{q\delta t}{2M}\mathbf{E} \qquad [2.46]$$

$$\frac{1}{\gamma} = \frac{c}{\sqrt{c^2 + \mathbf{p}_{new}{}^2}} \qquad [2.47]$$

$$\mathbf{v}_{new} = \frac{\mathbf{p}_{new}}{\gamma} \qquad [2.48]$$

$$\mathbf{x}_{new} = \mathbf{x}_{old} + \mathbf{v}_{new}\delta t\,. \qquad [2.49]$$

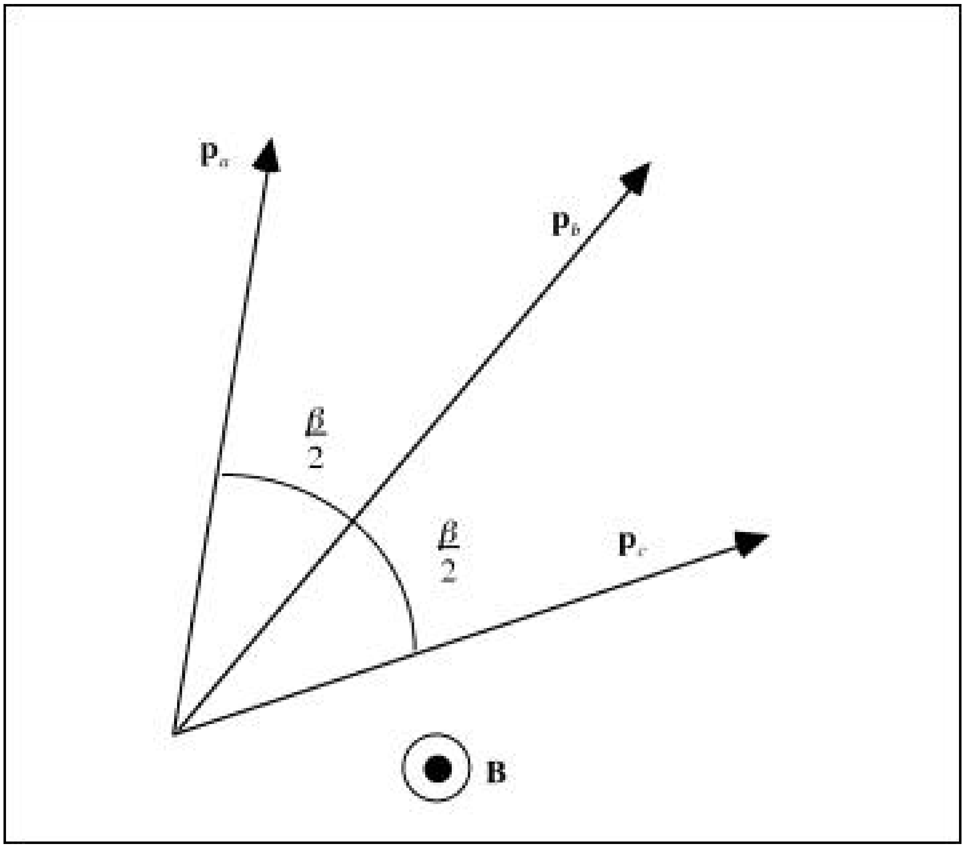


Fig. 2.4 Method for integrating relativistic trajectory through electromagnetic field.

The choice of time step δt is made at every step in the trajectory integration as follows,

$$\delta t = MIN\left(dv_{step}\frac{|\mathbf{v}|}{|\mathbf{a}|},\ \frac{dx_{step}}{|\mathbf{v}|} \right) \qquad [2.50]$$

Where dv_{step} (a number < 1) and dx_{step} (in meters) are set by the user. Where particles have low kinetic energy and in high accelerating fields, care should be used in the selection of these parameters.

2.2.5.3 Charge Density Deposition

The space charge density accumulated on the mesh by the trajectory integrator, for each ray of each species, taking the current of a given ray the time spent in the cell δt.

Emitting mesh points are specified by the user. When a ray is started, its current I is calculated from the user input J, for the given species. I is conserved throughout the trajectory[6].

For Cartesian symmetry, the current I and the increment in charge density $\delta\rho$ is calculated from

$$I = \frac{Jdydz}{\{Rays\ /\ Mesh\}} \qquad [2.51]$$

$$\delta\rho = \frac{1}{\gamma^2}\frac{I\delta t}{dxdydz}\,. \qquad [2.51]$$

For cylindrical symmetry, the ray represents a ring of current that is sweeps along the trajectory, therefore the expressions used are

$$I = \frac{2\pi(j_{start} - 0.5)dydyJ}{\{Rays\ /\ Mesh\}} \qquad [2.52]$$

$$\delta\rho = \frac{1}{\gamma^2}\frac{I}{2\pi(r + 0.5dy)dydx}\frac{\delta t}{\{Rays\ /\ Mesh\}} \qquad [2.53]$$

where j_{start} is the starting radial mesh position of the ray, and $dy = dr$).

The charge is distributed proportionally to each of the four corners of the cell that the particle (ray integration point) is in, according to how close the particle is to the mesh point

$$\rho_{i,j} + g_x g_y \delta\rho \;\; \rightarrow \;\; \rho_{i,j} \qquad [2.54]$$

$$\rho_{i+1,j+1} + f_x f_y \delta\rho \;\; \rightarrow \;\; \rho_{i+1,j+1} \qquad [2.55]$$

[6] Unless the hybrid fluid model is used and there is particle creation or loss. Note also that the Hybrid Fluid solver is only for Cartesian, nonrelativistic problems.

$$\rho_{i+1,j} + f_x g_y \delta\rho \;\rightarrow\; \rho_{i+1,j} \tag{2.56}$$

$$\rho_{i,j+1} + g_x f_y \delta\rho \;\rightarrow\; \rho_{i,j+1} \tag{2.57}$$

where

$$f_x = (x - idx)/\ dx \tag{2.58}$$

$$f_y = (y - jdy)/\ dy \tag{2.59}$$

$$g_x = 1 - f_x \tag{2.60}$$

$$g_y = 1 - f_y\ \ . \tag{2.61}$$

2.2.6 Iterative Static Equilibrium (ISEQ) Method

For single and multiple species charged particle optics problems with no interactions other than with the smoothly varying externally applied and self fields, such as electron guns and positive ion sources, the raytrace/SOR method has been very successful. This method iterates the solution of potential (either eq. [2.1], [2.6], or [2.8]) with the trajectory integration (eq. [2.24] and [2.25], or eq. [2.38]) using at each step values from previous iteration steps (see Fig. 2.3 below). Usually convergence is achieved after a modest number of steps. Orbits are integrated from starting surfaces or regions that emit particles, and a current is assigned to the ray based on the current density of the emitting surface and the number of rays starting from the surface. This current is used to deposit space charge on the numerical mesh (using eqs. [2.54] - [2.57]) to be used as for the source term in the field calculation.

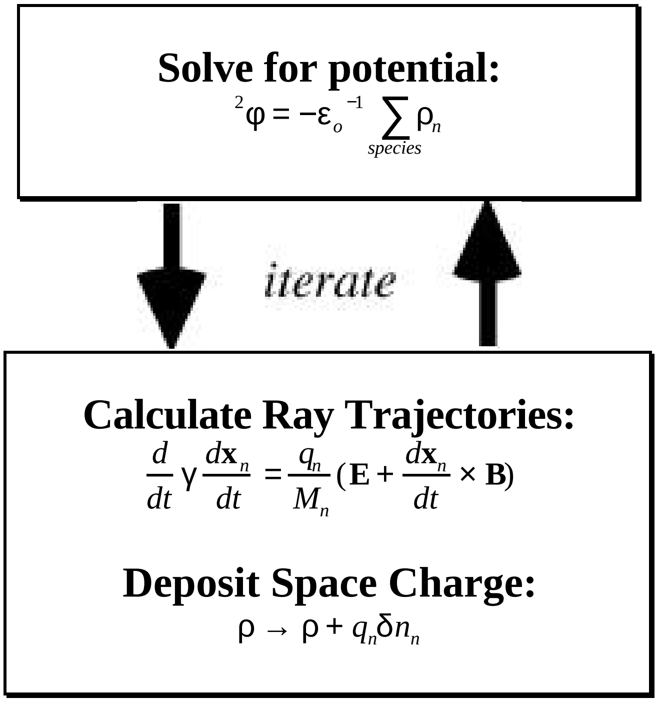

Solve for potential:
$^2\varphi = -\varepsilon_o^{-1}\sum_{species}\rho_n$
iterate
Calculate Ray Trajectories:
$\frac{d}{dt}\gamma\frac{d\mathbf{x}_n}{dt} = \frac{q_n}{M_n}(\mathbf{E} + \frac{d\mathbf{x}_n}{dt}\times\mathbf{B})$
Deposit Space Charge:
$\rho \to \rho + q_n\delta n_n$

Fig. 2.5 Typical flow schematic of the standard iteration scheme for the self-consistent solution of charged particle optics problems where the only inter-species interaction is through the electric field $\mathbf{E}=-\ \varphi$ from the total space charge (and possibly the magnetic field $\mathbf{B}$ from the total current). The fields φ (and $\mathbf{B}$) are computed in the first step, and the rays (trajectories) computed and the resulting space charge field tabulated in the second step. The process is iterated until convergence is reached.

2.2.7 Hybrid Fluid

The Hybrid Fluid model can be used to model nonrelativistic time-independent problems where species interact with each other. This algorithm is appropriate inside the plasma of an ion source.

The force on the ions is from the electric field and viscosity due to collisions with other species (including neutrals)

$$\frac{d^2\mathbf{x}_n}{dt^2} = \frac{q_n}{M_n}(-\ \ \varphi + \mathbf{v}_n \times \mathbf{B}) + \sum_m \nu_{n,m} n_m (\mathbf{v}_n - \mathbf{v}_m) \qquad [2.62]$$

The fluid description of the particles uses the coupled equations for number density n and the velocity $\mathbf{v}$. The fluid equations consist of the fluid continuity equation

$$\cdot\ (n_n \mathbf{v}_n) = \sum_{m,k} \chi_{n,m,k} n_m n_k \qquad [2.63]$$

and the fluid force equation,

$$\mathbf{v}_n \cdot \quad \mathbf{v}_n = -\frac{q_n}{m_n}(- \quad \varphi + \mathbf{v}_n \times \mathbf{B}) + \sum_m \nu_{n,m} n_m (\mathbf{v}_n - \mathbf{v}_m)\,. \qquad [2.64]$$

The solution for $\varphi(\mathbf{x})$ can be accomplished if the positions of all the particles is known, (or alternatively n and $\mathbf{v}$). On the other hand, to solve for the particle motion requires that $\varphi(\mathbf{x})$ be known.

With multiple interacting species, the numerical solution of the equations of motion for any one species requires knowledge of the solutions for the other species, in addition to the potential $\varphi(\mathbf{x})$. Therefore the system of particle equations must itself be iterated, in addition to the major iteration with the field solution.

A new algorithm, which is called the “hybrid fluid/ray trace” method is implemented in *NEDlab*. At each iteration step, an equation for $\varphi(\mathbf{x})$ is solved as usual (typically the Poisson-Boltzmann eq. [2.8]), and the particle orbits are then integrated (typically eq. 2.15-2.16]) using the electric field based on the deposited charge from the previous iteration. The viscosity forces from the other particles are calculated (the right-most term of [2.53]) using the n_n and $\mathbf{v}_n$ deposited from previous iteration steps, just as the total charge density was deposited in previous steps in the ISEQ technique. As the ray traverses the numerical grid, it may experience current gain or loss due to the right-most side of [2.55], which is calculated from the fluid fields n_n and $\mathbf{v}_n$ deposited in previous iterations. A schematic of this algorithm is shown in Fig. 2.6.

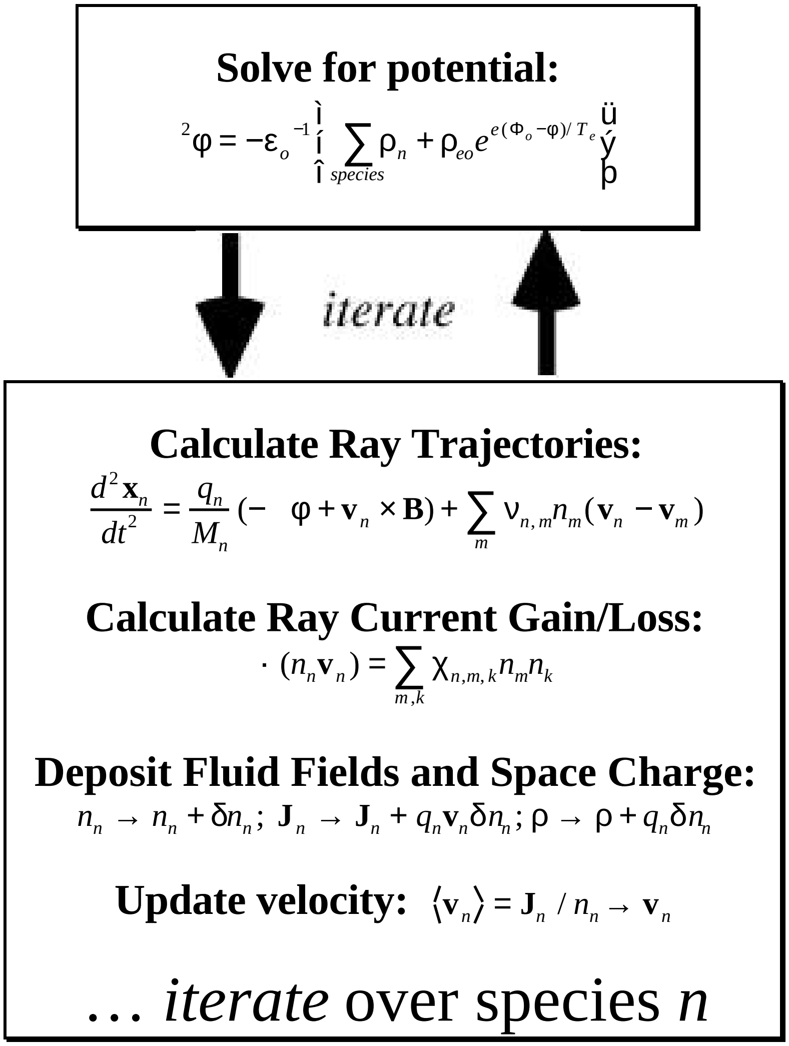
The field $\mathbf{v}_n$ is actually an average over all rays of species n , for a given iteration, obtained from

$$\langle \mathbf{v}_n \rangle = \mathbf{J}_n / n_n \qquad [2.64a]$$

where $\mathbf{J}_n$ is accumulated at each mesh point, each iteration, as follows

$$\mathbf{J}_n + q_n \delta n \mathbf{v}_n \rightarrow \mathbf{J}_n \qquad [2.64b]$$

A smoothing algorithm is applied to the $\mathbf{v}_n$ at each species iteration.



Solve for potential:

$$^2\varphi = -\varepsilon_o^{-1}\left\{\sum_{species}\rho_n + \rho_{eo}e^{e(\Phi_o-\varphi)/T_e}\right\}$$

Calculate Ray Trajectories:

$$\frac{d^2\mathbf{x}_n}{dt^2} = \frac{q_n}{M_n}(- \quad \varphi + \mathbf{v}_n \times \mathbf{B}) + \sum_m \nu_{n,m}n_m(\mathbf{v}_n - \mathbf{v}_m)$$

Calculate Ray Current Gain/Loss:

$$\cdot\ (n_n\mathbf{v}_n) = \sum_{m,k}\chi_{n,m,k}n_m n_k$$

Deposit Fluid Fields and Space Charge:

$$n_n \to n_n + \delta n_n;\ \mathbf{J}_n \to \mathbf{J}_n + q_n\mathbf{v}_n\delta n_n;\ \rho \to \rho + q_n\delta n_n$$

Update velocity: $\langle \mathbf{v}_n\rangle = \mathbf{J}_n / n_n \to \mathbf{v}_n$

… *iterate* over species *n*

Fig. 2.6 Hybrid Fluid/Raytrace algorithm flow diagram. Inter-species interaction is not only through the electric field from the total space charge, but through fluid fields (N_n and $\mathbf{v}_n$) coupling via the fluid coupling coefficients ν_{nm} and χ_{nmk} .

2.2.8 Particle Emission

Particles may be emitted from any mesh point, vacuum or conductor. If the emitting point is on a conductor, it will be emitted an energy, E_o, specified by the user along the normal to the local surface, (unless modified by thermal procedure or

Busch's theorem as described below). If the point is inside a conductor, the particle will be extinguished. The current density and the number of rays per mesh (both user-specified), and the mesh size of the transverse area of the emitting cell determine the current that the ray carries.

2.2.8.1 Thermal Emission

A thermal distribution with temperature *T* (specified by the user) can be imposed on emitted particles. The energy is set according to

$$-T\ln(|\mu_1|) \rightarrow E_o\,. \qquad [2.65]$$

The angle is found as follows

$$\theta = \mu_2\pi \qquad \text{(volume emission)} \qquad [2.66]$$

$$\theta + \mu_2\frac{\pi}{2} \rightarrow \theta \qquad \text{(wall emission)} \qquad [2.67]$$

$$\frac{2E_o}{m} \rightarrow v_o \qquad [2.68]$$

$$v_o\cos(\theta) \rightarrow v_x \qquad [2.69]$$

$$v_o\sin(\theta) \rightarrow v_y \qquad [2.70]$$

$$0 \rightarrow v_z \qquad [2.71]$$

where the μ_n are random numbers between -1 and 1.

2.2.8.2 Busch's Theorem

When charged particles are emitted from a surface emersed in an axial magnetic field in cylindrically symmetric problems, the conservation of cannonical angular momentum has ramifications for the dynamics of the emitted particles.

When space charge limited flow is modeled, the Busch's Theorem option should be used to insure that particles started at the point marked #1 in Fig. 2.5 will have the proper physical angular momentum so that canonical angular momentum is conserved.

Busch's Theorem is also useful for modeling the transport of beams in solenoidal transport channels, without starting the particles from rest, where the proper azimuthal velocity must be set for focusing to occur.

If originally the emitted particle has a velocity (v_X, v_Y,0), and the change in axial magnetic field from particle birth and the start of the trajectory calculation is $B_X(x_{start})$-$B_X(x_O)$, then its azimuthal velocity (=v_Z, since z=0 to start) must be adjusted according to

$$-\frac{qy}{2\gamma M}[B_x(x_{start}) - B_x(x_o)] \rightarrow v_z. \quad . \qquad [2.72]$$

To conserve energy, the other components are adjusted as follows,

$$\eta = \frac{\sqrt{{v_x}^2 + {v_y}^2 - {v_z}^2}}{v_o} \qquad [2.73]$$

$$v_x \rightarrow \eta v_x \qquad [2.74]$$

$$v_y \rightarrow \eta v_y. \qquad [2.75]$$

2.2.8.3 Space Charge Limited Emission

In problems such as the design of electron guns, space charge limited emission from the cathode may occur. The algorithm used in *NEDlab* is described below. It must be iterated with the Poisson solution (eq. [2.33]) for self-consistency. The algorithm assumes a local planar emission model (Child Law).

The space charge limited current density associated with rays started at the equipotential surface running through mesh point #1 is

$$J_{start} = \left\{ \begin{matrix} Child \\ factor \end{matrix} \right\} \times \frac{4\sqrt{2}\varepsilon_o}{9\sqrt{m}} \frac{(q\Delta\varphi)^{\frac{3}{2}}}{d^2} \qquad [2.76]$$

where $\Delta\varphi$ is the difference in potential between the conductor surface (labeled #4) and the starting surface, q is the species charge, m the species mass, d is the shortest distance between the equipotentials, and Child factor is set by the user (for stability to be ≤1). The {Child factor} is used to help achieve stability in high perveance situations. It can be set to a number < 1 for early iterations, and brought up to 1.0 as convergence is achieved.

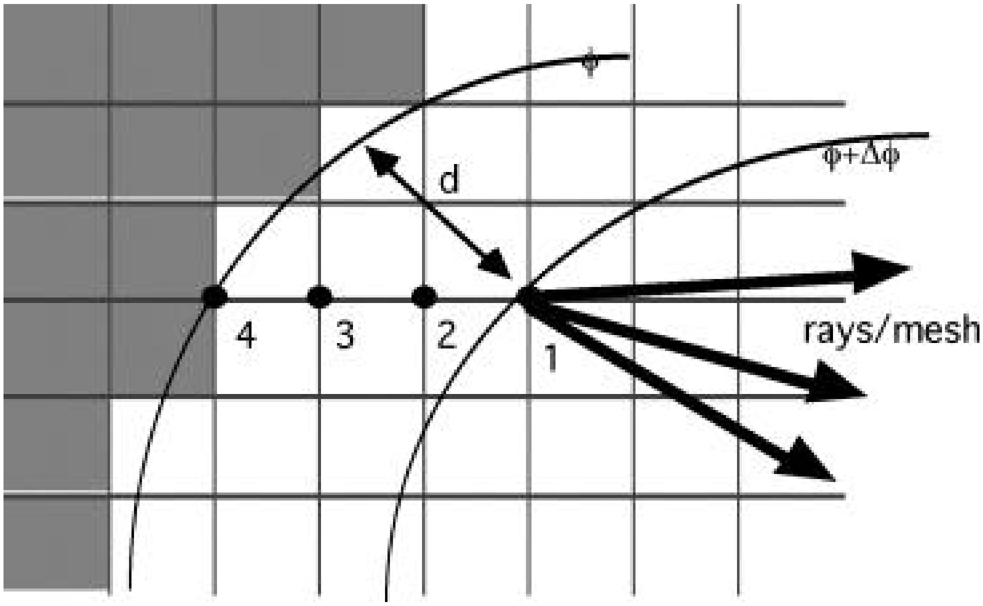


Fig. 2.7 Space charge limited flow algorithm. Particles are emitted at position 1, and space charge is deposited at positions 4, 3, 2 and 1 from an analytical expression eq. [2.69] - [2.73]). If there is no thermal spread, then the particles are directed along the normal to the equipotential surface. In the situation depicted above, a thermal distribution has been imposed, and three raays (rays/mesh=3) have been emitted with a random thermal spread about the normal.

The charge density must be deposited on the points behind the starting surface. Defining

$$\rho_{start} = \left\{\frac{1}{RaysPerMesh}\right\}\frac{J_{start}}{v_o\varepsilon_o} \qquad [2.77]$$

where v_O is the speed of the particles, the charge density is accumulated between the cunductor surface and the starting equipotential surface as follows

$$\rho_1 + \left\{\frac{4}{1}\right\}^{2/3}\rho_{start} \quad \to \quad \rho_1 \qquad [2.78]$$

$$\rho_2 + \left\{\frac{4}{2}\right\}^{2/3}\rho_{start} \quad \to \quad \rho_2 \qquad [2.79]$$

$$\rho_3 + \left\{ \frac{4}{3} \right\}^{2/3} \rho_{start} \quad \rightarrow \quad \rho_3 \qquad [2.80]$$

$$\rho_4 + \left\{ \frac{4}{4} \right\}^{2/3} \rho_{start} \quad \rightarrow \quad \rho_4 \qquad [2.81]$$

2.3 Dynamic Solutions

The time-dependent algorithms in *NEDlab* operate on the same finite difference mesh as the static solvers. In certain physical situations it may be desirable to use both, and this can be done[7]. The particle emission algorithms described previously work in the time-dependent mode as well.

2.3.1 Maxwell Equations

The Maxwell equations

$$\times \mathbf{E}^{em}(\mathbf{x},t) = -\frac{\partial \mathbf{B}^{em}(\mathbf{x},t)}{\partial t} \qquad [2.82]$$

$$c^2 \quad \times \mathbf{B}^{em}(\mathbf{x},t) = \frac{\partial \mathbf{E}^{em}(\mathbf{x},t)}{\partial t} + \varepsilon_o^{-1} \sum_{species} \mathbf{J}_n(\mathbf{x},t) \qquad [2.83]$$

are solved in 2D cartesian or cylindrical coordinates by a time stepping technique. In cartesian coordinates the field variables are E_x, E_y, B_z, J_x, and J_y. In cylindrical coordinates the variables are E_x, E_r, B_θ, J_x and J_r .

In the finite difference solution of differential equations, quantities that are related by first order spatial derivatives (e.g. curl) of other quantities are best defined on grids displaced one half grid step. The defining locations of magnetic field $[B_z]_{i,j}$, and electric field components $[E_x]_{i,j}$ and $[E_y]_{i,j}$, are shown in Fig. 2.8.

This principal holds for the time dimension as well, so that **E** and **B** are defined one half-step in time (dt/2) apart.

[7] An example might be a beam containing electrons and negative ions. The electrons respond essentially instantaneously and can be treated as rays with the ISEQ method, while the time-dependent motion of the ions could be treated with the PIC method.

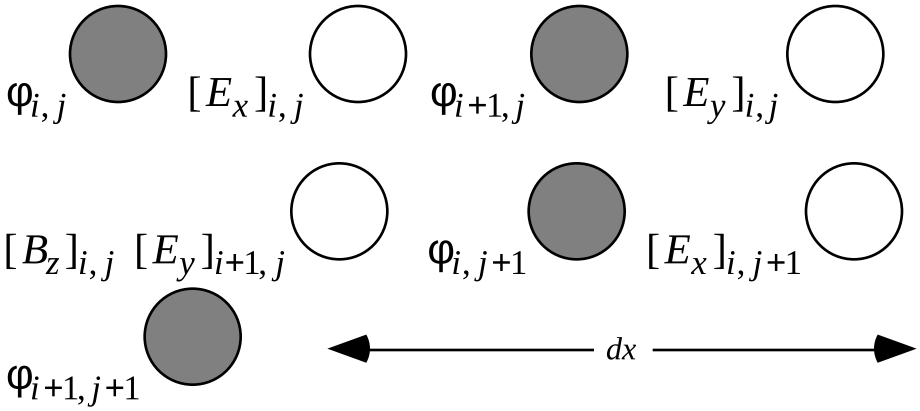


Fig. 2.8 Dual grid and the definition of electric and magnetic fields in the Maxwell equation solver algorithm. The magnetic grid is offset by half mesh intervals from the electric field.

The "leap-frog" method is used to advance the Maxwell equations in time. For Cartesian coordinates, the first step is to loop over all *i* and *j* making the following replacements,

$$[\mathbf{E}_x]_{i,j} + \frac{c^2\delta t}{dy}([B_z]_{i,j} - [B_z]_{i,j-1}) - \frac{[J_x]_{i,j}}{\varepsilon_o} \rightarrow [\mathbf{E}_x]_{i,j} \qquad [2.84]$$

$$[\mathbf{E}_y]_{i,j} + \frac{c^2\delta t}{dx}(f_+[B_z]_{i-1,j} - f_-[B_z]_{i,j}) - \frac{[J_y]_{i,j}}{\varepsilon_o} \rightarrow [\mathbf{E}_y]_{i,j} \qquad [2.85]$$

where f_+ and f_- are given by eq. [2.4] for Cartesian symmetry, and eq. [2.5] for cylindrical symmetry (r=y, z=θ). Following this, the boundary conditions are applied to **E**.

The conductor-vacuum interface boundary conditions applied to the electric field at the first step in the algorithm are shown in Fig. 2.9. There are eight types of boundaries, the first four are horizontal or vertical, and the second four are diagonal types[8].

[8] In the present version, the edges of the simulation box are treated as conductors. Note also that the octagonal boundary conditions of Fig. 2.9 could be simplified to square (as in most other codes of this type). The octagonal boundary conditions are a result of early versions of the code that used time dependent fluid equations for the particles.

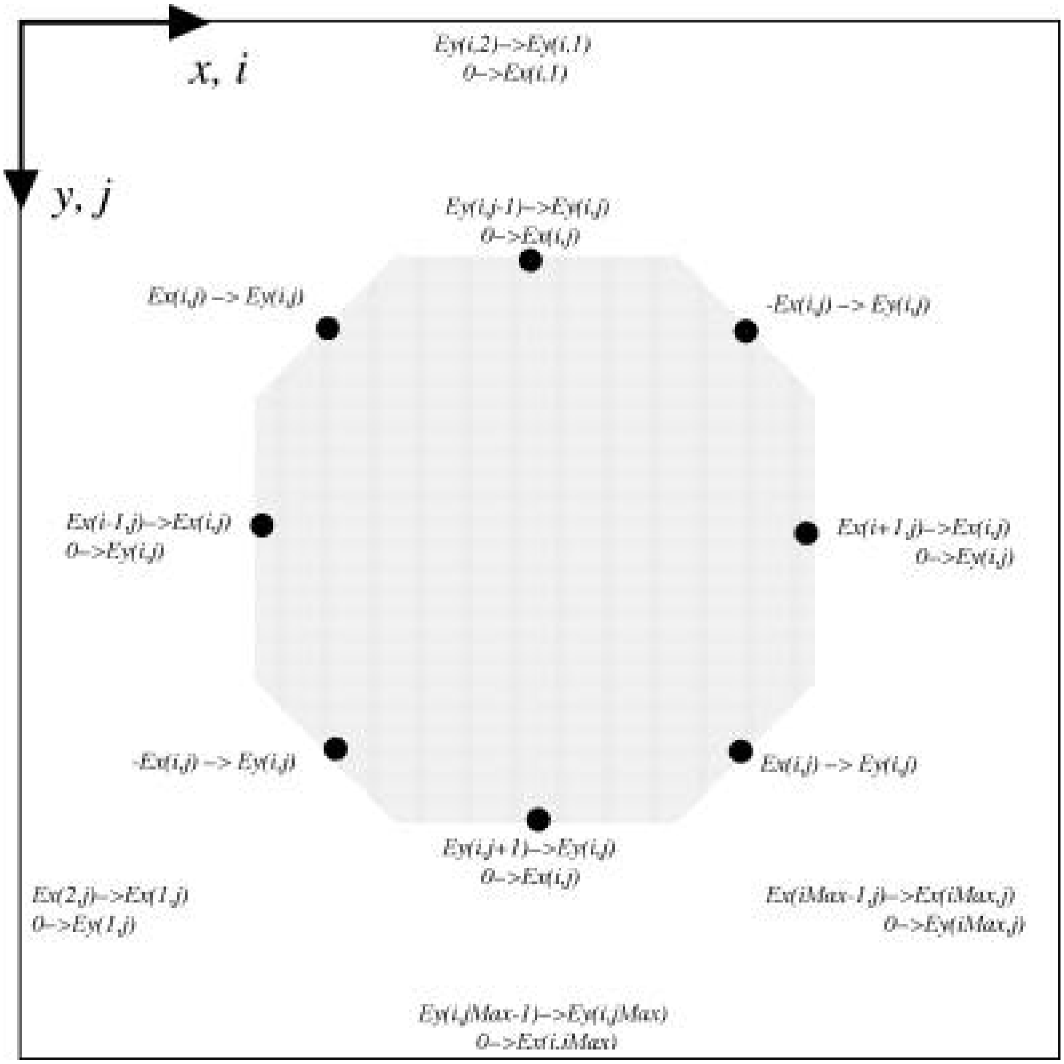


Fig. 2.9 Boundary conditions on the electric field at conducting surfaces.

The other class of boundary conditions on the electric field are from antenna regions. These are rectangular shapes where the field is set to a given antenna value multiplied by a sinusoidally oscillatiing factor

$$E_x^{\,n} \sin(2\pi f^{\,n} t + \psi^{\,n}) \;\; \rightarrow \;\; [E_x]_{i,\,j} \qquad [2.86]$$

$$E_y^{\,n} \sin(2\pi f^{\,n} t + \psi^{\,n}) \;\; \rightarrow \;\; [E_y]_{i,\,j}\,. \qquad [2.87]$$

where the strength, frequency and phase (E_x^n, E_y^n , f^n, ψ^n,), and the size and location of the driving rectangles are set by the user[9].

The second step is to again loop over all i and j , updating B_z

$$[B_z]_{i,j} - c^2\delta t\left\{\frac{f_+[E_x]_{i,\,j+1} - f_-[E_x]_{i,j}}{dy} - \frac{[E_y]_{i+1,\,j} - [E_y]_{i,j}}{dx}\right\} \;\rightarrow\; [B_z]_{i,j} \qquad [2.88]$$

and then apply boundary conditions to the magnetic field, as shown in Fig. 2.10.

The antenna regions also set the magnetic field, as follows

$$B_z^n \cos(2\pi f^n t + \psi^n) \;\rightarrow\; [B_z]_{i,j} \qquad [2.89]$$

If cylindrical symmetry is used, the boundary conditions E_r are the same as for E_y. The magnetic boundary conditions on B_θ are the same as for B_z except that along the axis B_θ is set to zero.

[9] Up to five different areas are allowed in the current version of the code.

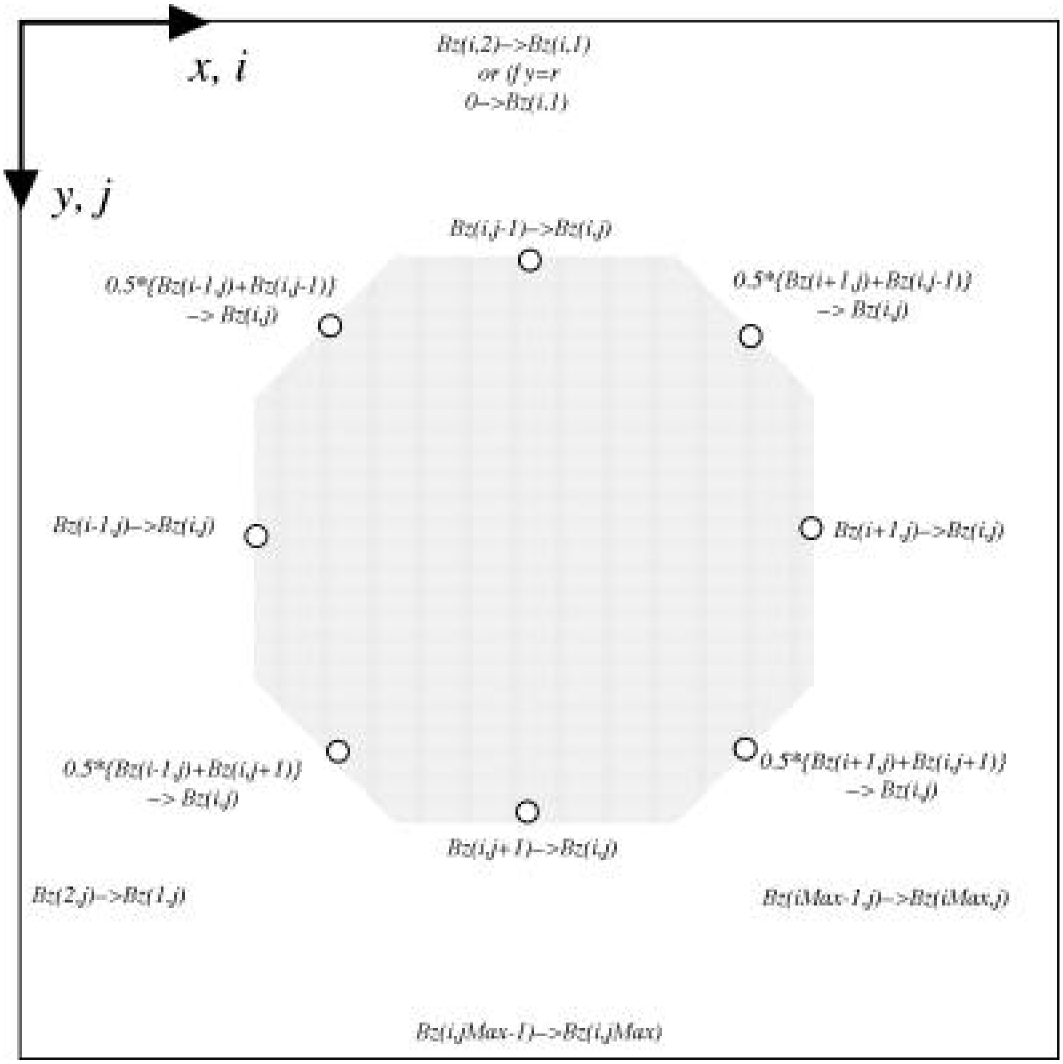


Fig. 2.10 Boundary conditions on the magnetic field at conducting surfaces.

2.3.2 Particle Trajectories

For time-dependent problems there is only one trajectory integrator in *NEDlab* . Eq. [2.38] is advanced as described in the discussion of eq. [2.39] - [2.49], with the only difference being that in the time-dependent solution, the time step δt is fixed (by the user) and is the same time step used to advance the Maxwell equations.

2.3.3 Current Density Deposition

In dynamic simulations, the PIC analog of the ISEQ ray is the macroparticle. The macroparticle follows the particle trajectory and has associated with it a charge q and mass m (set by the user for its species). It also carries a macroparticle weight w which determines how much current density will be deposited on the mesh.

The current density from the particles is accumulated from all macroparticles, over one time step. The trajectory integrator procedure also does the current density accumulation, taking the particles charge, weight, and the and the time spent in the cell δt.

> **Note**: the particles should not be going so fast as to cross more than one cell per time step: if this occurs, the time step should be decreased. A warning dialog box will alert the user if the Courant condition has been violated.

For Cartesian symmetry,

$$w = \frac{Jdydz\delta t}{q} \times SkipSteps \qquad [2.90]$$

$$Q^{macro} = wq \qquad [2.91]$$

$$\delta\rho = \frac{Q^{macro}}{dxdydz} \qquad [2.92]$$

where *SkipSteps* is the number of time steps between macroparticle emission

For cylindrical symmetry, the macro particle is actually a ring of charge, and the following expressions are used

$$w = \frac{2\pi(j_{start} - 0.5)dydzdtJ}{q} SkipSteps \qquad [2.93]$$

$$\delta\rho = \frac{Q^{macro}}{2\pi(j - 0.5)dydxdz} \qquad [2.94]$$

where j_{start} is the y-axis mesh coordinate at the position that the macroparticle was initially emitted.

The increment in the current density $\delta\mathbf{J}$ from particles with charge density $\delta\rho$ traveling at velocity $\mathbf{v}$ is then

$$\delta\mathbf{J} = \mathbf{v}\delta\rho \ . \qquad [2.95]$$

The current density contribution is distributed proportionally to each of the four corners of the cell that the particle is in, according to how close the particle is to the mesh point

$$\mathbf{J}_{i,j} + g_x g_y \delta\mathbf{J} \ \rightarrow \ \mathbf{J}_{i,j} \qquad [2.96]$$

$$\mathbf{J}_{i+1,j+1} + f_x f_y \delta\mathbf{J} \ \rightarrow \ \mathbf{J}_{i+1,j+1} \qquad [2.97]$$

$$\mathbf{J}_{i+1,j} + f_x g_y \delta\mathbf{J} \ \rightarrow \ \mathbf{J}_{i+1,j} \qquad [2.98]$$

$$\mathbf{J}_{i,j+1} + g_x f_y \delta\mathbf{J} \ \rightarrow \ \mathbf{J}_{i,j+1} \qquad [2.99]$$

where f_x, f_y, g_x and g_y are given by eq. [2.58] - [2.61].

Just as in the static soulutions, the emitted macroparticles can be made to obey Busch's Theorem, (see the discussion for ray trajectories).

2.4 References

[1] S. A. Self; Exact Solution of the Collisionless Plasma-Sheath Equations; Phys. Fluids 6, 1762 (1963).

[2] W. B. Hermannsfeldt; SLAC Electron Trajectory Program; SLAC 266, (1979), original SLAC 166, (1973).

[3] W. S. Cooper, K. Halbach, S. B. Magyary; Computer Aided Extractor Design; Proceedings of the Second Symposium on Ion Sources and Formation of Ion Beams, Berkeley, LBL-3399, (1974).

[4] See for example J. H. Whealton, and J. C. Whitson; Space Charge Ion Optics Including Extraction from a Plasma; Particle Accelerators 10, p. 235-251 (1979).

[5] J. E. Boers; SNOW a Digital Computer Program for the Simulation of Ion Beam Devices; Sandia National Laboratory Report SAND 79-1127, (1979).

[6] P. Spadtke, S. Wipf; KOBRA3, A Code for the Calculation of Space-Charge-Influenced Trajectories in 3-Dimensions, GSI-89-09 report, ISSN 0171-4546, June 1989.

[7] User's Manual For SOS, Mission Research Corporation publication MRC/WDC-R-065.

3 User's Guide

This chapter tells how to use the *NEDlab* software

Note: The outline of the sections in this chapter follow closely the outline in chapter 2 (Algorithms), which contain more technically complete explanations.

program icon

document icon

The general course of tasks in solving a problem is to first define the mesh dimensions. The conductor geometry is then entered, and possibly antennae (electromagnetic sources), after which the details of the type of field solution and particle emission and tracking are determined. The problem is then iterated (static solutions) or time-stepped (dynamic solutions) The last step is to display the results in graphical form. After analysis of the results by the user, theprocess may be repeated with new geometry, sources, etc., until the desired results are obtained.

To begin *NEDlab*, double click on the program icon , or double click on one of the program's documents . When the introductory screen appears, click the mouse button to begin the program.

HELP SCREENS are available from most dialog boxes, and also from the main menu.

3.1 Geometry

SetSetting the Problem Symmetry and Size

To set the problem symmetry and mesh size, go to the **Edit Main...** menu item as shown in Fig. 3.1.

The **Main** dialog box is then displayed, as shown in Fig. 3.2.

The finite difference mesh runs from $i = 1..iMax$ in the horizontal direction, and j

= *1..jMax* in the vertical (down) direction.

The *x*-axis range is 0*..xMax*, and the *y*-axis range is 0*..yMax*.

The symmetry of the problem, either Cartesian or cylindrical, is also set in the **Main** screen. (The radio button for Cartesian symmetry has been selected in Fig. 3.2).

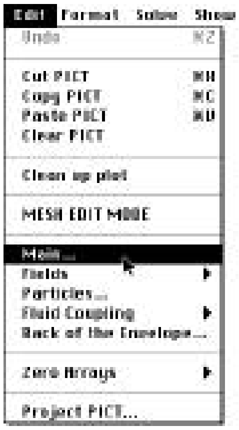
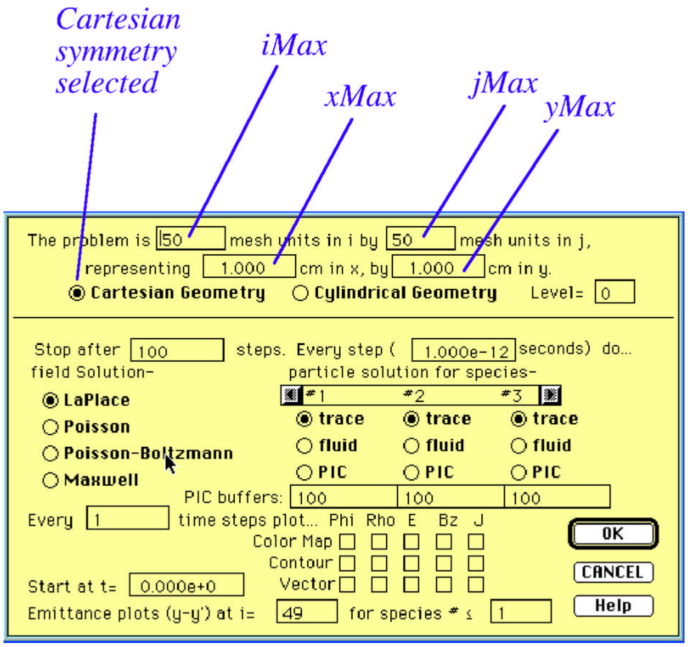


Fig. 3.1 Selecting the Main Edit screen.



The problem is 50 mesh units in i by 50 mesh units in j,
representing 1.000 cm in x, by 1.000 cm in y.
● Cartesian Geometry ○ Cylindrical Geometry Level= 0

Stop after 100 steps. Every step (1.000e-12 seconds) do...
field Solution-
● LaPlace
○ Poisson
○ Poisson-Boltzmann
○ Maxwell

particle solution for species-

#1	#2	#3
● trace	● trace	● trace
○ fluid	○ fluid	○ fluid
○ PIC	○ PIC	○ PIC

PIC buffers: 100 100 100

Every 1 time steps plot...

	Phi	Rho	E	Bz	J
Color Map	□	□	□	□	□
Contour	□	□	□	□	□
Vector	□	□	□	□	□

Start at t= 0.000e+0
Emittance plots (y-y') at i= 49 for species # ≤ 1

Fig. 3.2 The Main dialog box

Entering the Conductor Geometry

There are two ways of entering conductor geometry. The first and simplest is to use the mesh editor. Pull down the **Edit** menu and select **MESH EDIT MODE.**

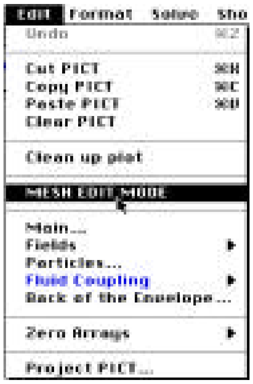


Fig. 3.3 Selecting Mesh Edit Mode.

The mesh editor has drawing tools to enter and alter the geometry.

The drawing tools set the mesh points as conductor or vacuum. When drawing conductor, the color used determines the voltage that will be set.

The voltage of the conductor to be set is selected by clicking the second row. (The association of colors with list of numbered voltages will be described later).

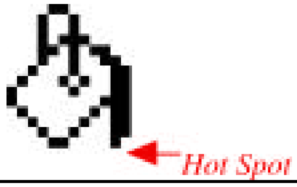
The pencil tool can be used to draw in fine details, by setting individual mesh points.

The eraser is used to replace conducting points with vacuum. This can also be done with any drawing tool if the is vacuum set.

The brush is similar to the pencil, but wider in the horizontal direction.

The paint bucket tool will reset any contiguous region it is clicked in. The new voltage (or vacuum) is whatever has been selected on the second level of the pallet.

Note: The paint bucket tool has a "hot spot" that determines the exact pixel where the paint will begin pouring. Failure to realize this can cause large regions to be filled with the wrong "paint".



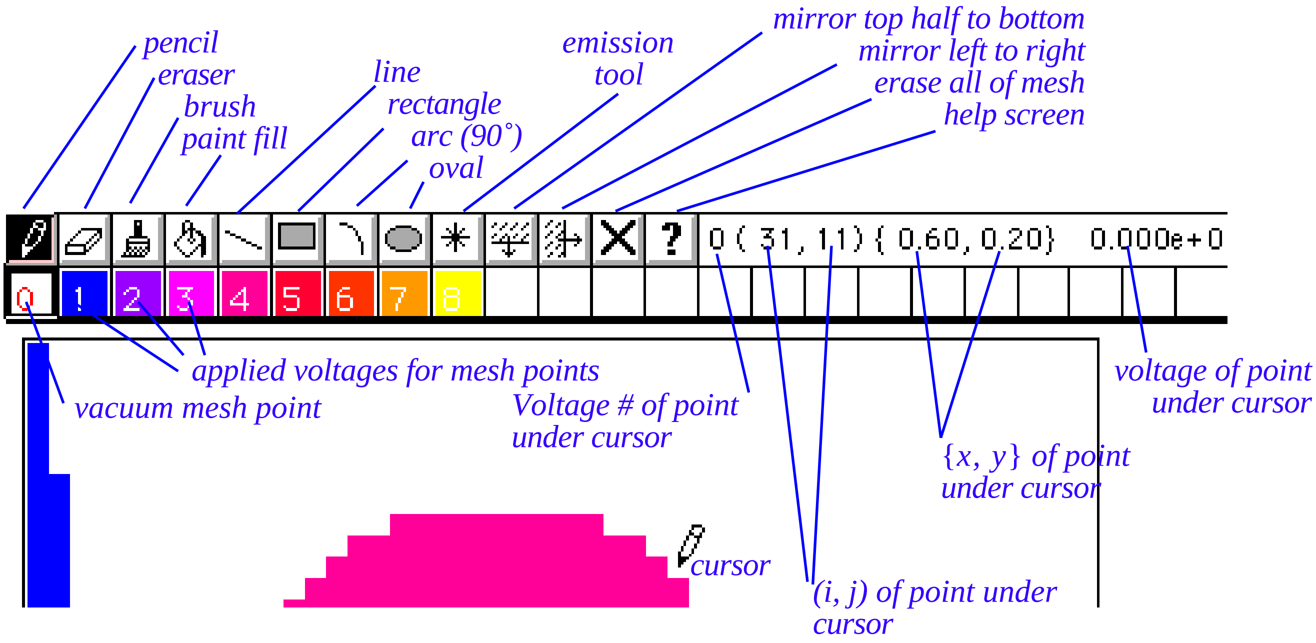
The line tool draws straight lines. The rectangle, arc, and oval tools can be used to quickly draw in other shapes.

The emission tool allows the definition of emitting mesh points. When pressed, it causes the presently defined emitting points to sparkle. Draw a rectangle around a surface that is to be defined as an emitter. A dialog box will appear and ask which species to emit.

The two mirror buttons can be used to symmetrize the mesh layout.

The large × button will completely erase the mesh (replace conductor with vaccum).

The voltage number, (*i,j*), { *x*, *y* } and voltage of the mesh point under the cursor (pencil tool in Fig. 3.4) are displayed in the upper right hand box of the pallet.



0 (31, 11) { 0.60, 0.20}   0.000e+0

pencil
eraser
brush
paint fill
line
rectangle
arc (90˚)
oval
emission tool
mirror top half to bottom
mirror left to right
erase all of mesh
help screen
0
1
2
3
4
5
6
7
8
?
applied voltages for mesh points
vacuum mesh point
Voltage # of point under cursor
voltage of point under cursor
{x, y} of point under cursor
cursor
(i, j) of point under cursor

Fig. 3.4 The tool pallet in the mesh editor.

Using Drawings from Other Programs

The other way to enter geometry is to import a drawing or scan from a commercial graphics program. A PICT (graphics file) from another drawing program can be attached to the main display using **File, Import PICT...** or by using the clipboard. The drawing can then be projected (**Edit, Project PICT...)** onto the mesh. Use the paint bucket (or other tools) to set the voltages of the imported conductor mesh geometry.

To summarize (for the clipboard method of transfer):

1) In the drawing program, select allof the drawing (or just the part of the drawing you want). This works in either object (draw) mode, or paint (bitmap) mode.

2) Copy the selected part of the drawing into the clipboard.

3) Now go into the *NEDlab* program using MultiFinder. Paste the clipboard into the *NEDlab* PICT buffer (**Edit, Paste PICT...**).

4) Project the PICT onto the numerical mesh (**Edit, Project PICT...**).

5) Use the paint bucket and other tools in Mesh Edit Mode to perfect your mesh, (remember the PICT from the drawing program is projected in monochrome).

For the disk method of transfer, replace 1) through 3) with:

1) While in the drawing program, save your picture in PICT format.

2) In *NEDlab*, load the PICT using **File, Import PICT...**

Entering the Applied Voltage List

The voltage list can be edited by pulling down the **Edit, Fields ▶ Electrostatic** menu item.

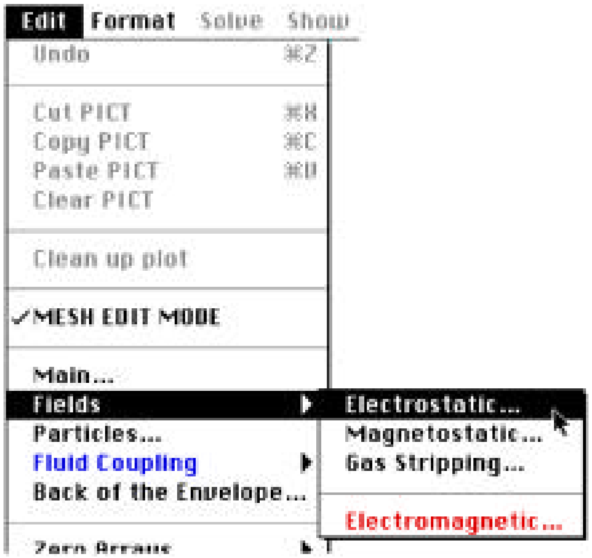
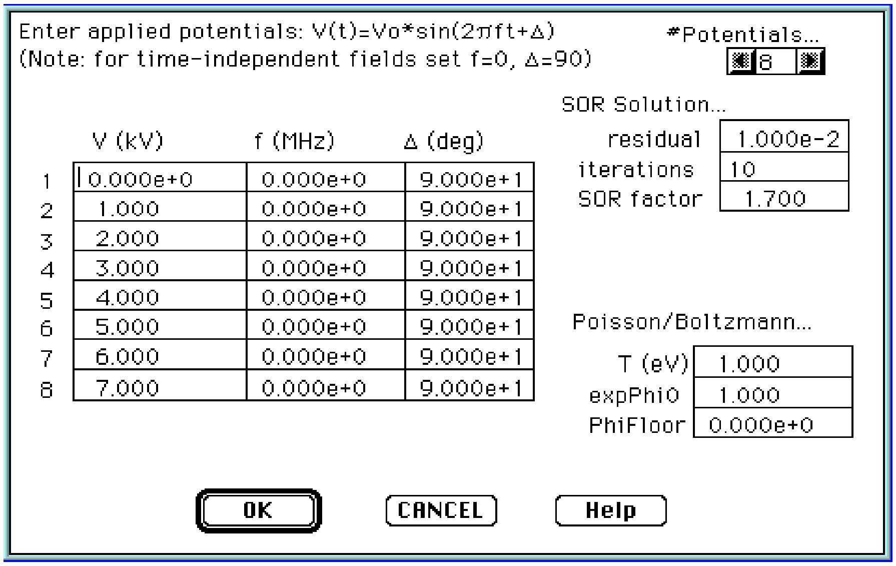


Fig. 3.5 Initiating the editing of electrostatic fields.



Enter applied potentials: V(t)=Vo*sin(2πft+Δ)
(Note: for time-independent fields set f=0, Δ=90)

#Potentials... 8

	V (kV)	f (MHz)	Δ (deg)
1	0.000e+0	0.000e+0	9.000e+1
2	1.000	0.000e+0	9.000e+1
3	2.000	0.000e+0	9.000e+1
4	3.000	0.000e+0	9.000e+1
5	4.000	0.000e+0	9.000e+1
6	5.000	0.000e+0	9.000e+1
7	6.000	0.000e+0	9.000e+1
8	7.000	0.000e+0	9.000e+1

SOR Solution...

residual	1.000e-2
iterations	10
SOR factor	1.700

Poisson/Boltzmann...

T (eV)	1.000
expPhi0	1.000
PhiFloor	0.000e+0

Fig. 3.6 **Electrostatic Field** dialog box.

The voltages in the first column are associated with the colors painted on the mesh by the mesh editor. For quasi electrostatic solutions (time varying but not truly electromagnetic) the frequency and phase of the applied voltages are also set in this screen.

Note! For purely electrostatic problems, be sure that the voltage driving frequencies are zero and the phases are 90˚ .

Note!! Unless you are simulating screening by a plasma, (e.g. extraction of negative ions or electrons from a plasma sheath), be sure that PhiFloor is less than the lowest voltage you expect in the problem. Similarly, PhiCeiling must also be larger than the largest potential you expect in your solution.

Leaving the Mesh Editor

To leave mesh edit mode, pull down the menu item **Edit- MESH EDIT MODE** and uncheck the item. You will then return to the simulation mode.

3.2 Static Solutions

The electrostatic field solvers (Laplace, Poisson, Poisson-Boltzmann) and the two ray trace solvers are described here.

Static Field Solutions

3.2.1 Laplace Equation

The Laplace solver will find the potential distribution that results from the applied potentials and the conductor geometry only (i.e. excluding the particles space charge).

To set up the parameters of the Laplace solution, pull down the **Edit, Fields ▶ Electrostatic...** as shown in Fig. 3.5. The **Electrostatic Field** dialog box is then displayed as shown in Fig. 3.6. The voltages applied to conductors may be set in this dialog box, as well as the number of iterations, and the SOR factor. For quasi electrostatic (time-varying but not truly electromagnetic) the frequency and phase of the applied voltages are also set in this screen. Be sure that the frequencies are zero and the phases are 90˚ for time-independent solutions.

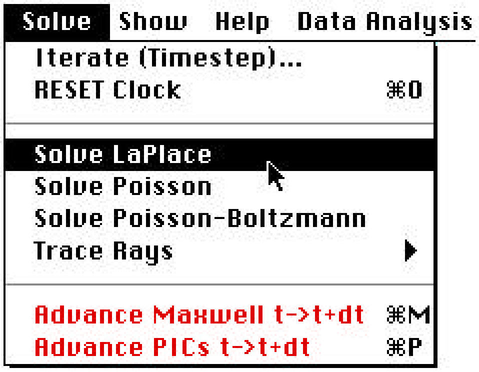


Fig. 3.7 Launching the LaPlace solver.

Next pull down the Solve menu and select Solve LaPlace (Fig. 3.7). The Laplace solver will then iterate the SOR solution for the number of times set in the **Electrostatic Field** dialog box (Fig. 3.6).

.3.2.2 Poisson Equation

The Poisson solver will find the potential distribution that results from the applied potentials and the conductor geometry and the space charge of the particles.

To set up the parameters of the Poisson solution, pull down the **Edit**, **Fields... Electrostatic...** as shown in Fig. 3.5. The **Electrostatic Field** dialog box is then displayed as shown in Fig. 3.6. The applied voltages may be set in this dialog box, as well as the number of iterations, and the SOR factor. For quasi electrostatic (time varying but not truly electromagnetic) the frequency and phase of the applied voltages are also set in this screen.

Next, select **Solve, Poisson** (Fig. 3.7). The Poisson solver will then iterate the SOR solution for the number of times set in the **Electrostatic Field** dialog box (Fig. 3.6).

3.2.3 Poisson-Boltzmann Equation

The Poisson-Boltzmann solver will find the potential distribution that results from the applied potentials and the conductor geometry and the space charge of the particles, with screening by a thermal distribution of plasma electrons.

To set up the parameters of the Poisson solution, pull down the **Edit, Fields... Electrostatic...** as shown in Fig. 3.5. The **Electrostatic Field** dialog box is then displayed as shown in Fig. 3.6. The applied voltages may be set in this dialog box, as well as the number of iterations, and the SOR factor. For quasi electrostatic (time varying but not truly electromagnetic) the frequency and phase of the applied voltages are also set in this screen. Be sure that the frequencies are zero and the phases are 90˚ for purely electrostatic problems.

The temperature of the screening electrons, *T*, and the quantity $e^{\Phi_o/T}$, are also set in the **Electrostatic Field** dialog box. (See the chapter 2 on Algorithms for definitions of these terms).

Next pull down the **Solve** menu and select **Poisson-Boltzmann** (Fig. 3.7). The Poisson-Boltzmann solver will then iterate the SOR solution for the number of times set in the **Electrostatic Field** dialog box (Fig. 3.6).

3.2.3.1 Approximate Poisson-Boltzmann

The procedure for running the Approximate Poissson-Boltzmann equation is identical to the procedure for solving the Poisson equation as described above, with the exception that the quantity φ_{floor} is set to the plasma potential.

3.2.4 Electrostatic Boundary Conditions

The boundary conditions on the edges of the simulation are Neumann ($\partial\varphi/\partial n = 0$) unless conductor is placed there, in which the boundary condition is $\varphi = V_{\text{conductor}}$.

3.2.5 Particle Trajectories

The type of trajectory integrator for the particles is set in the Particles dialog box. To get to this dialog box, pull down the **Edit** menu and select **Particles...** as shown in Fig. 3.8. The Particles dialog box will then be displayed as shown in Fig. 3.9. The type and amount of particle emission is set in this box.

3.2.5.1 Quick Trajectory Algorithm

The Quick Trajectory Algorithm is much faster than the Relativistic Trajectory integrator (discussed below).

The Quick Trajectory integrator will be used if the radio button in the upper right corner is not pushed in (the symmetry must be Cartesian also-- symmetry is set in **Edit, Main...**). If symmetry is cylindrical, the Relativistic Trajectory Integrator will be automatically selected.

3.2.5.2 Relativistic Trajectory Integrator

The Relativistic Trajectory integrator will be used if the radio button in the upper right corner is set (the symmetry can be Cartesian or cylindrical).

The quantity dx Error is the spatial tolerance (in cm) and dv/v Error is the relative velocity tolerance for the Relativistic Trajectory Algorithm integrator.

For more details on the **Particles** dialog box, see Fig. 3.9 below.

Note: The term *emission*, which ordinarily implies low energy emission from conductor surfaces, is used in the throughout this text to mean the general starting conditions of the particle trajectories. Particles may be started on conductor-vacuum boundaries, on simulation box boundaries, or even from rectangular patches in space. Thus the term is applied equally to, for instance, a high energy beam that enters the simulation box on the left, as well as to low energy, thermal emission from the cathode of an electron gun.

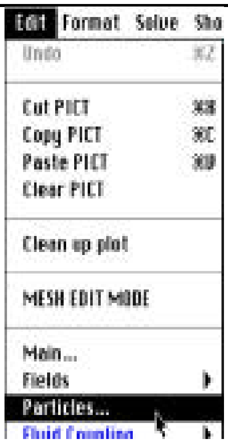
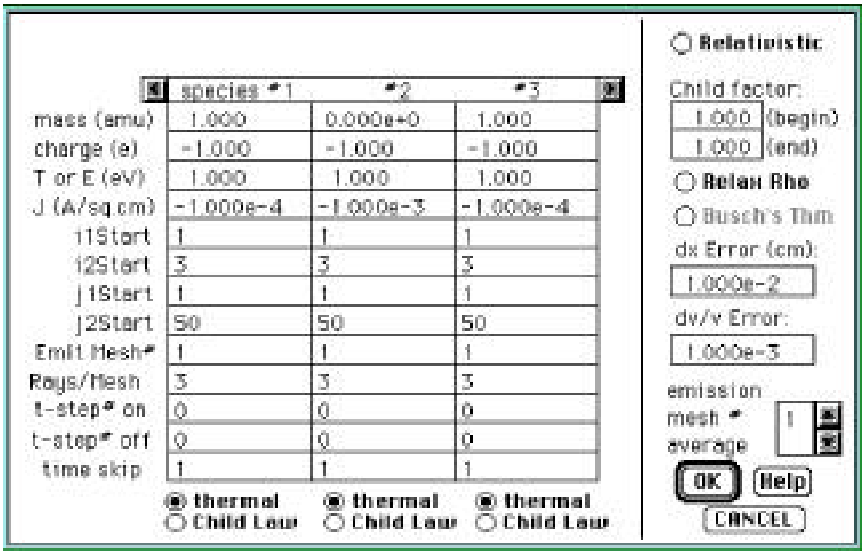


Fig 3.8 Selecting the **Particles** dialog box.



	species #1	#2	#3
mass (amu)	1.000	0.000e+0	1.000
charge (e)	-1.000	-1.000	-1.000
T or E (eV)	1.000	1.000	1.000
J (A/sq.cm)	-1.000e-4	-1.000e-3	-1.000e-4
i1Start	1	1	1
i2Start	3	3	3
j1Start	1	1	1
j2Start	50	50	50
Emit Mesh#	1	1	1
Rays/Mesh	3	3	3
t-step# on	0	0	0
t-step# off	0	0	0
time skip	1	1	1
	● thermal ○ Child Law	● thermal ○ Child Law	● thermal ○ Child Law

○ Relativistic

Child factor: 1.000 (begin) 1.000 (end)

○ Relax Rho

○ Busch's Thm

dx Error (cm): 1.000e-2

dv/v Error: 1.000e-3

emission mesh # average 1

Fig. 3.9 The **Particles** dialog box.

Setting Species Mass, Charge

A description of the parameters defined in the **Particles** dialog box follows:

mass: Enter mass in multiples of proton mass units (e.g. 1=proton, note: mass=0 causes the electron mass to be entered.

charge: 1 = proton, -1 = electron.

T (eV): Emission temperature of specie (if the **thermal** radio button is set; otherwise T is the drift energy at emission).

J (A/sq.cm): Current density of specie at starting surface (Note on sign: J is automatically set negative if charge < 0). Note also that if the **Child Law** button is set on (see below), the program will automatically adjust the current density at each iteration, since it is an unknown variable to be found by iteration to convergence.

i1start, i2start, j1start, j2start: Define a rectangle in i,j space on the numerical mesh to emit species from mesh type- Emit Mesh. This may also be set graphically in **MESH EDIT MODE** using the emission tool (∗).

Emit Mesh: The mesh type (conductor potential number) that can emit species (only along conductor-vacuum interface). For open volume emission, set to 0. It may also be set graphically in **MESH EDIT MODE** using the emission tool.

Rays/Mesh: Allows for multiple emission from a mesh point.

t-step on: Determines when emission begins (integer).

t-step off: Determines when emission stops (integer).

time skip: Skips emission (regularly) for specified step interval (integer).

Relativistic: If set on, use the Relativistic Trajectory Algorithm integrator. If off <u>and</u> Cartesian symmetry is selected (in **Edit, Main...**) the Quick Trajectory Algorithm integrator is used.

dx Error: Absolute spatial tolerance (in cm) for the Relativistic Trajectory Algorithm integrator.

dv/v Error: Relative velocity tolerance for Relativistic Trajectory Algorithm integrator. (Note: the Quick Trajectory integrator ignores dx and dv/v).

thermal: If set on, emit Maxwellian distribution with temperature T (eV) (set above). If off, then all particles are given a drift energy T (eV), with the direction of the drift velocity set by the local electric field.

Child Law: Use local planar Child law for space charge limited emission (for arbitrary surface shape).

Child factor: When using **Child Law** the space charge deposited is multiplied by this factor. It will ramp linearly from the beginning value to the end value over the number of iteration steps until stop set in **Edit, Main...** (see Fig. 3.10). Use this feature to insure convergence of high perveance ISEQ problems.

Set **Relax Rho** for high perveance space charge limited flow problems. This causes the space charge to be averaged with the previous iteration (iterative static equilibrium problems).

Busch's Thm (cylindrical symmetry only) : When set on, this enforces conservation of cannonical momentum of emitted particles. When a high energy beam is input through one of the simulation box boundaries (left or right), the setting of this button will give particles the appropriate angular momentum under the assumption that they were emitted at low energy (long before entering the simulation) in a magnetic field-free region. (If the field is too high, then the program assumes the particle was mirrored, and no particle is introduced).

emission mesh# average: For emission surfaces with complex geometry, this parameter effects the "bumpiness" of the distribution of particle emission directions. An averaging of the local electric field direction over a square patch of this number of mesh points is used to determine the emission direction (there is no effect on the emission energy, which is always consistent with conservation of energy).

3.2.5.3 Charge Density Deposition

The current density set from the **Particles** dialog box determines the current carried by the rays (ISEQ solutions) or the charge carried by the macroparticles (PIC solutions).

Note that when **Relax Rho** is set (usually in space charge limited flow ISEQ problems), the space charge from the present iteration is averaged with the previous iteration (see previous section).

When running a PIC calculation, the PIC buffers should be set for each species to the maximum number of macroparticles (plus a prudent margin) expected in the simulation at any given time during the simulation. This is done in the **Edit, Main…** dialog box (see Fig. 3.10 below).

Self- Consistent Static Solutions

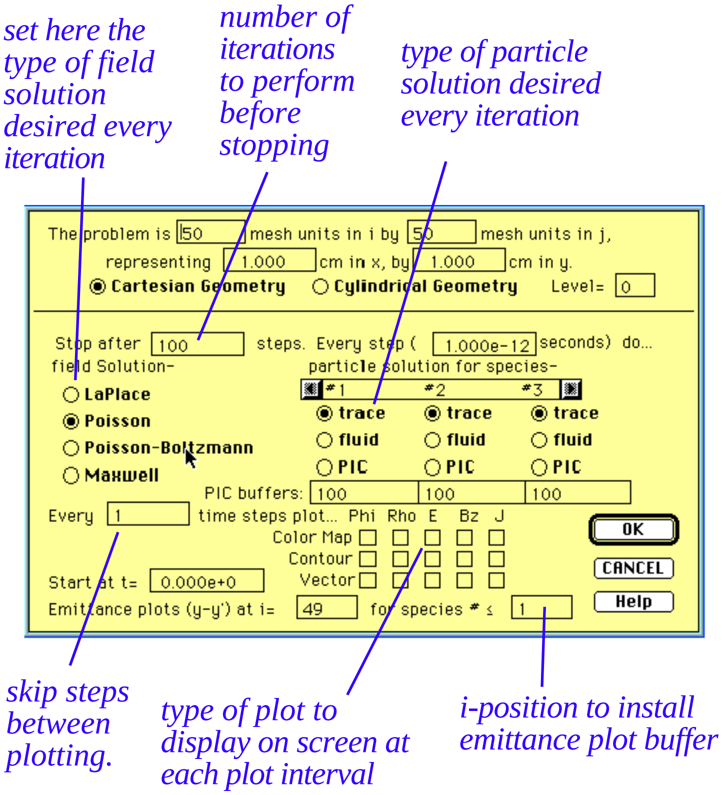
3.2.6 Iterative Static Equilibrium (ISEQ) Method

ISEQ solutions are accomplished by setting the type of field and particle solutions, plots desired, as well as the the total number of iterations to perform, in the Main dialog box. Pull down **Edit, Main…** to get the Main dialog box, as shown in Fig. 3.10 below.

After the ISEQ iteration parameters have been set, launch the iteration process by pulling down **Solve, Iterate (timestep)….** The code will then iterate the specified number of times, solving the specified equations at each step, and plotting on the screen each plotting interval.

Note: plotting takes time, and from that point of view should be minimized. On the other hand, plotting can also give insight and enhance the interactivity of the solution process. Depending on your needs, set the plotting interval accordingly.

To stop sooner than the limit set in the **Main** dialog box, click the mouse button at any time. The program will finish whatever solution it is doing (or in some cases, such as plotting, immediately jump out) and return to idle mode.



set here the type of field solution desired every iteration

number of iterations to perform before stopping

type of particle solution desired every iteration

The problem is 50 mesh units in i by 50 mesh units in j,
representing 1.000 cm in x, by 1.000 cm in y.
● Cartesian Geometry ○ Cylindrical Geometry Level= 0

Stop after 100 steps. Every step (1.000e-12 seconds) do...
field Solution-
○ LaPlace
● Poisson
○ Poisson-Boltzmann
○ Maxwell

particle solution for species-

#1	#2	#3
● trace	● trace	● trace
○ fluid	○ fluid	○ fluid
○ PIC	○ PIC	○ PIC

PIC buffers: 100 100 100

Every 1 time steps plot...

	Phi	Rho	E	Bz	J
Color Map	□	□	□	□	□
Contour	□	□	□	□	□
Vector	□	□	□	□	□

Start at t= 0.000e+0
Emittance plots (y-y') at i= 49 for species # ≤ 1

skip steps between plotting.

type of plot to display on screen at each plot interval

i-position to install emittance plot buffer

Fig. 3.10 The **Main** dialog box and items relevant to ISEQ solutions.

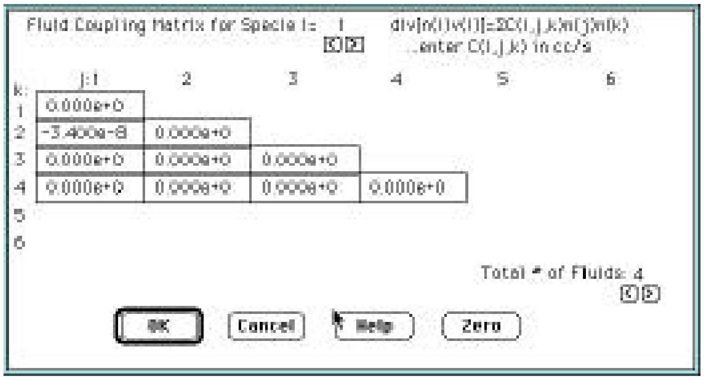
Note: The value of the time step set in the **Main** dialog box is irrelevant in ISEQ solutions .

Note: After iteration has ceased, either by limit set or by clicking the mouse button, all field array data is ready to be manipulated for whatever reason (e.g. geometry changes, plots, zero charge density). Iteration can then be resumed, if desired.

3.2.7 Hybrid Fluid

The Hybrid Fluid particle algorithm is selected in the main dialog box (**Edit, Main...**) by pushing the fluid radio button for each species that a fluid solution is desired.

The fluid continuity coupling parameters are entered by pulling down the **Edit, Fluid Coupling ▶ Continuity...** The coupling matrix dialog box is then displayed as shown if Fig. 3.11 below. The matrices for each species can be paged through by pushing the "<" or ">" buttons at the top of the dialog box. The number of fluids coupled can be increased or decreased by pushing the "<" or ">" buttons at the lower left.



Fluid Coupling Matrix for Specie i: 1 div[n(i)v(i)]=ΣC(i,j,k)n(j)n(k)

...enter C(i,j,k) in cc/s

k:	j: 1	2	3	4	5	6
1	0.000e+0					
2	-3.400e-8	0.000e+0				
3	0.000e+0	0.000e+0	0.000e+0			
4	0.000e+0	0.000e+0	0.000e+0	0.000e+0		
5						
6						

Total # of Fluids: 4

Fig. 3.11 Fluid Continuity matrix dialog box.

To edit the force couplings between species fluids, select **Edit, Fluid Coupling ▶ Force...** The fluid force coupling matrix dialog box displayed is similar to Fig. 3.11.

The procedure for a Hybrid Fluid-ISEQ solution is identical to the Ray Trace-ISEQ procedure described previously.

Emitting Particles from Complex Geometry

3.2.8 Particle Emission

Each species can be emitted from a rectangular region as defined by i1start, i2start, j1start, j2start: in (*i,j*) space on the numerical mesh. The species are emitted from mesh type-Emitting Mesh, as also entered in the **Particles** dialog box shown in Fig. 3.9 (menu item **Edit, Particles...**). If the Emitting Mesh number (integer) is a conducting mesh type embedded in a vacuum, emission occurs only at the conductor-vacuum interface.

If the Emit Mesh number (integer) is a vacuum, then the emission will occur throughout the rectangular shaped region.

3.2.8.1 Thermal Emission

To get thermal emission of a species from a surface or a volume, pull down the menu item **Edit, Particles…** and set the **thermal** radio button. Set the **Child Law** button to use local (planar) Child law for space charge limited emission. If the button is not set, the emission will be with zero temperature in the direction normal to the emitting surface at the energy set by T (eV).

3.2.8.2 Busch's Theorem

To get conservation of cannonical momentum in emission of a species from a surface or a volume, pull down the menu item **Edit, Particles…** and set the **Busch's Thm** radio button.

3.2.8.3 Space Charge Limited Emission

For space charge limited emission of a species from a surface or a volume, pull down the menu item **Edit, Particles…** and set the **Child Law** radio button. For extremely high perveance situations, set the **RelaxRho** button - this causes the space charge density to be layed down as an average with previous iterations, and suppresses instabilities.

3.3 Dynamic Solutions

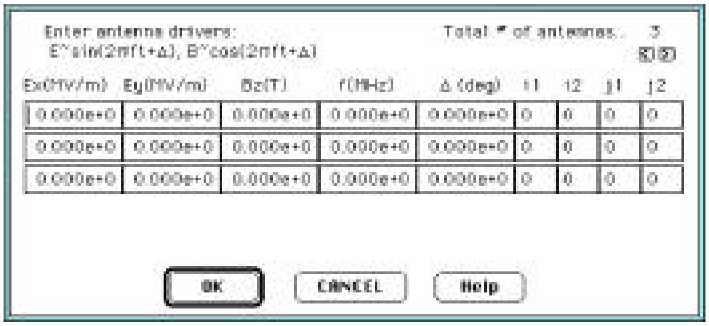
Each of the dynamic solvers can be advanced a single time step, or any number of steps automatically. A negative time step will cause the solutions to be advanced backwards in time.

3.3.1 Maxwell Equations

Sources for the Maxwell equations are set up as rectangular patches that force E_x, E_y, and B_z (or the corresponding cylindrical fields) to time-sinusoidal values. Impulses can also be delivered by turning an antenna on for an instant and turning them off the next (manually).

To set up antennae, use the menu command sequence **Edit, Fields ▶ Electromagnetic…** The antenna dialog box is then displayed (Fig. 3.12).

To advance the Maxwell equations one time step, select **Solve, Advance Maxwell t->t+dt**.



Enter antenna drivers:
E*sin(2πft+Δ), B*cos(2πft+Δ)

Total # of antennas: 3

Ex(MV/m)	Ey(MV/m)	Bz(T)	f(MHz)	Δ (deg)	i1	i2	j1	j2
0.000e+0	0.000e+0	0.000e+0	0.000e+0	0.000e+0	0	0	0	0
0.000e+0	0.000e+0	0.000e+0	0.000e+0	0.000e+0	0	0	0	0
0.000e+0	0.000e+0	0.000e+0	0.000e+0	0.000e+0	0	0	0	0

Fig. 3.12 **Antenna** dialog box.

3.3.2 Particle Trajectories

PIC solutions for particles are selected in the **Main** dialog box (**Edit, Main...**, see Fig. 3.10). The particle properties (mass, charge, etc) are set in the Particles dialog box (**Edit, Particles...**, see Fig. 3.9).

3.3.3 Current Density Deposition

Current density (and charge density) deposition is automatic in the PIC solution. The current density acts as a source to drive the Maxwell equations (if they are being solved).

3.4 Output

3.4.1 Screen Plots

Plots of current values for fields or particle quantities are available at any time in idle mode below the **Show** menu item.

When in iteration mode, the plots that were selected in the **Main** dialog box (**Edit, Main...** see Fig. 3.10) will be displayed at the interval specified in the **Main** dialog box.

Scalar fields such as φ, or ρ, and also B_z, can be plotted as color maps, contour plots, or gradient arrow plots. Vector fields (**E**, **J**) are plotted as arrows (the maximum value is written in the lower left corner of the plot).

Some of the display parameters of the color map, contour, and vector plots (e.g. color map scheme, number of contours, arrow size and placement) can be modified in the **Format, Plot Parameters…** dialog box.

3.4.2 Hard Copy

Hard copies of plots can be obtained by

1) Making a screenshot using the key sequence -shift-3 which creates PICT file on the main directory called "screen1", which can subsequently be printed from a drawing program, or cut and pasted from the drawing program into a word processing document, and then printed.

2) Trajectory plots can be printed using the **File, Print Screen…** (it is usually desirable to use **File, Page Setup…** to change from portrait to landscape printing). **Note**: this only works with Macintosh QuickDraw compatible printers, i.e. PostScript is not supported).

3) Exporting field or particle arrays to other spreadsheet programs by writing disk files that are subsequently imported by the spreadsheet. The user must write his own analysis program and plot the results in using the spreadsheet language.

4) Similar to 2) but a custom program is written in a programming language.

> **Note**: The field quantity that is displayed under the magnifying glass cursor is selected under the **Format** menu item (see Figure 3.13).

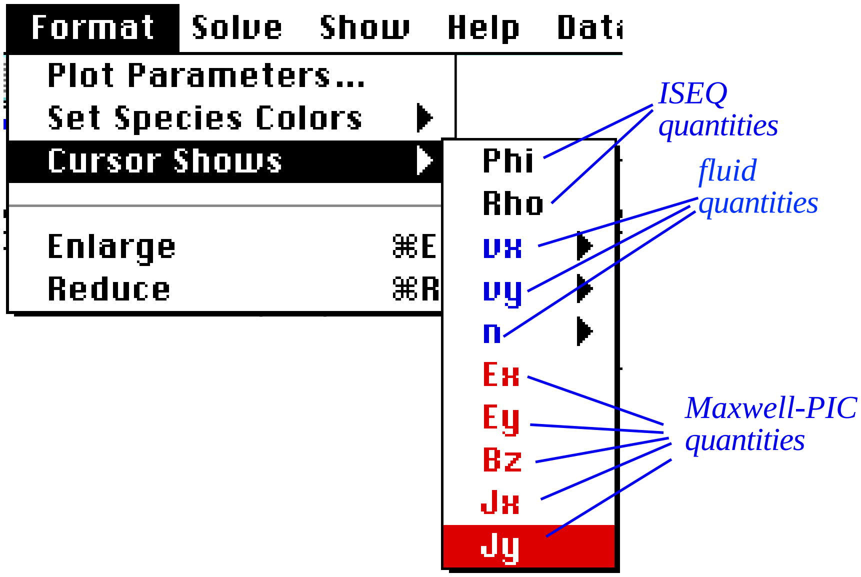


Fig. 3.13 **Format** -> **Cursor Shows** menu selections.

For static particle solutions (rays), the emittance, energy, transverse energy, and transverse (*y* or *r*) position are plotted from a buffer whose *i*-position is specified in the **Main** dialog box (Fig. 3.10).

For dynamic particle solutions (PIC), the phase space plots are snapshots in time.

Exporting Data

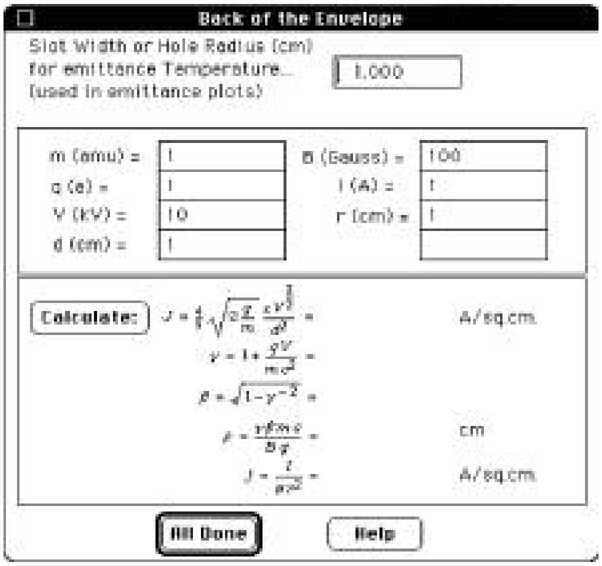
3.4.2 Disk Output of Arrays

The **File, Export** menu item selection sequence will write to disk the various arrays selected, in ASCII format. This function is provided to allow external analysis and post processing by spreadsheet or graphics packages.

3.5 Miscellaneous

Back-of-the-Envelope calculations

Some simple calculational aids are provided under **Edit, Back of the Envelope…** and are intended to be useful in designing certain devices (Note: the aperture width entered here is used in the temperature that is calculated from the emittance and displayed in emittance plots).



Back of the Envelope

Slot Width or Hole Radius (cm) for emittance Temperature... (used in emittance plots) 1.000

m (amu) =	1	B (Gauss) =	100
q (e) =	1	I (A) =	1
V (kV) =	10	r (cm) =	1
d (cm) =	1		

$J = \frac{4}{9}\sqrt{2\frac{q}{m}}\frac{\epsilon V^{\frac{3}{2}}}{d^2} =$ A/sq.cm.

$\gamma = 1 + \frac{qV}{mc^2} =$

$\beta = \sqrt{1-\gamma^{-2}} =$

$\rho = \frac{\gamma\beta mc}{Bq} =$ cm

$J = \frac{I}{\pi r^2} =$ A/sq.cm.

Fig. 3.14 Edit -> Back of the Envelope calculator.

Displaying Experimental Emittance Data

Typically code predicted emittance diagrams are presented in scatter plot format, while experimental data is usually shown as contour plots. This is because a code generates x-x' pairs and it is natural to plot them individually in a scatter plot. The experimental apparatus, on the other hand, is effectively a two dimensional matrix of receptors or bins that integrate the particle current into each bin, so it is natural to plot the matrix as a contour plot. This does not allow for easy comparison between theory and experiment because dense scatter plots can have point overlay, making it difficult for the eye to visually ascertain weight.

NEDlab emittance diagrams may be plotted in a combination color map/contour plot format (**Show, Emittance Diagram ▸ bin plots**). *NEDlab* also allows importing experimental emittance data for comparison with code results (**Data Analysis, Emittance ▸ Read Disk File**). Code generated emittance distributions may also be exported in the same format (**Data Analysis, Emittance ▸ Wead Disk File**).

The ASCII input file format for measured data is shown in Figure 3.15 below. The first line is a string containing file name, date and time. It is used only to identify the data set. The next line contains the values: dx=increment in x (cm); dxp=increment in angle (milliradians) -- note that the small angle approximation sin(θ)=θ is used here; Nx=number of x measurements; Nxp=number of angle measurements; beta=beam velocity divided by the speed of light; and Current=beam current (mA). The measurement matrix F (Nx rows of Nxp columns) follows.

FileName Date Time

dx dxp Nx Nxp beta Current

$F_{1,1}$ $F_{1,2}$ ... $F_{1,Nxp}$

$F_{2,1}$ $F_{2,2}$ ... $F_{2,Nxp}$

...

...

...

$F_{Nx,1}$ $F_{Nx,2}$ ... $F_{Nx,Nxp}$

Figure 3.15 Emittance Scan Input File Format.

4 Examples

4.1 Electron Gun

The example "ElectronGun.Big" illustrates the solution of a typical iterative static equilibrium problem on a large mesh[10].

The steps taken to solve ElectronGun.Big problem were:

1) A scale drawing of the grid geometry was made in the commercial drawing program DeskDraw™ (you can use SuperPaint®, MacDraw® or equivalent) and saved in PICT format. (The absolute dimensions in the PICT are arbitrary as long as they the same scale in both dimensions). The actual dimensions of the grid area represented by the PICT were entered in *NEDlab* under **Edit, Main...** The PICT was read into *NEDlab* and then projected on the numerical mesh. The Mesh Editor was used to clean it up and tag the potentials (using the paint bucket).

2) The applied voltages were set in Edit...Applied Fields... Electrostatic, and the trial axial B field was set in **Edit, Applied Fields ▶ Magnetostatic**. Space charge limited flow electron emission from the cathode (mesh type 1) was set up in **Edit, Species...** .

[10] It is twice the size of the previous example file ElectronGun (v1.10 release). When changing a problem size, be sure to change any other items that are specified in terms of the integer mesh coordinates, e.g. the emission i1Start, i2Start, j1Start, and j2Start, as well as the applied B-field rectangles, emittance buffer, etc.

3) The iteration and plotting parameters were specified in Edit...Main... and the program was set to iteratively solve the Poisson equation and trace rays. After leaving **Mesh Edit Mode**, the **Solve, Iterate...** menu item set the iteration process going.

4) The Child factor for space charge limited flow was set (in **Edit, Particles**) to ramp automatically from 0.2 to 0.9 in the first 10 iterations. If the geometry were planar, then one would expect to be able to go to 1.0, however since the emission surface is concave the final value is expected to be somewhat less than 1.0 (for convex emitting surfaces the Child factor could be greater than 1.0). The final step in the solution of this problem is to search for the maximum Child factor for this geometry- it will probably be close to the theoretical value for spherical emission. To insure stability in this high perveance problem, the **RelaxRho** button set on.

4.2 Wave Guide

The file "Waveguide Structure" illustrates what happens when a wave in a guide hits a notch and turns a corner. Load the file and use **Solve, Iterate (time step)...** . Fill in the notch with the mesh editor and iterate again to see how much smoother the fields are (no reflections off the change in impedence). Antennas are set up in **Edit, Applied Fields ▶ Electromagnetic**.

4.3 Particle-in-Cell (PIC) Problem

There is also an example of the time-dependent (PIC+Maxwell) mode on the program disk called "SlowWave". Load the file and use **Solve, Iterate(time step)...** . The B-field and current are not optimized. Run it for a while and look at **Show, PIC Phase Ppace** to see how the particle distribution evolves in phase space.

Appendix A Notes on Negative Ion Source Physics

Many applications require a source capable of producing copious amounts of low temperature negative ions and as few electrons as possible. Most source designs have a transverse magnetic field to suppress the electrons in the plasma before they can reach the extraction area. The extraction grids must apply an electric field distribution that not only accelerates but also compensates for the diverging tendency of the negative ion (and electron) beam space charge. The extraction region has a transition from plasma to vacuum whose shape can have either a converging or diverging effect, which is also very important to the optics of the beam. Usually, outside this sheath a bending magnet applies a perpendicular magnetic field that deflects the electrons to a dump before they have a chance to enter the accelerating grid region. A schematic of a generic negative ion source is shown in Fig. A.1 below.

The necessity of having magnetic fields inside the plasma and on the sheath is a significant complication over positive ion sources. Furthermore, in almost any configuration imaginable, the magnetic field will make the problem three-dimensional. The electrons inside the plasma may have a non-isotropic diffusion tensor, and the temperature distribution may be non-Maxwellian because the applied magnetic field probably discriminates on the basis of electron momentum. At the sheath surface, there may be an **E×B** drift that affects the density in a way that destroys any attempt at 2D symmetry in the design (making it 3D), and the electrons extracted may also be bent in the **E×B** direction, further contributing to an asymmetric buildup of charge on the extraction aperture.

The negative ion beam may also undergo partial stripping after extraction as it traverses the stream of neutrals that are exiting the extraction aperture. These neutrals must be removed by pumping in order to avoid excessive stripping. If the pumping is asymmetric, in principle this can also impart a 3D asymmetry to the space charge forces. Charge exchange and other processes can impart an energy and angular spread to the beam and therefore cause emittance growth.

There is evidence that the plasma-vacuum interface in the extraction region of negative ion sources may not be abrupt as in positive ion sources, particularly where electromagnetic electron supressors are used. There is also evidence that significant negative ion production may take place in this region, so that if negative ions are born at different potentials and different magnetic fields, again emittance growth can occur.

The source physics is complex, not only is there the previously described interplay of three dimensional electric and magnetic fields, along with sheath curvature, grid geometry boundary conditions, and space charge, but also some rich, complex, and incompletely understood atomic and molecular physics. The formation of H^- in many sources is thought to be some variation of the following two step process: vibrational/rotational excitation of a hydrogen molecule

$$H_2 \rightarrow H_2^*$$

(either by collision with a fast electron or with the wall, see ref.[1] for a good discussion of the current state of knowlege), followed by dissociative attachment of an electron

$$e^- + H_2^* \rightarrow H + H^- .$$

Most measurements inside negative ion sources indicate a relatively low temperature (indeed the dissociative attachment process requires this). *Therefore it is probable that most of the observed emittance growth in negative ion injectors takes place during and after extraction.*

To summarize, negative ion source physics is significantly complicated by:

- the transverse B-field needed to filter out electrons
- the problem tends to be inherently 3D
- Boltzmann distribution approximations are suspect
- 3 species (+ and - ions, e^-) may need to be orbit-integrated
- the electron-ion mass and time scale difference can cause numerical instabilies in the system of equations
- electron gyroradii are smaller than the numerical mesh in many places of interest
- the sheath is probably thick and complicated by an **E×B** drift, cross-field diffusion, nonuniform plasma density, charge exchange, ion production in varying fields, etc.

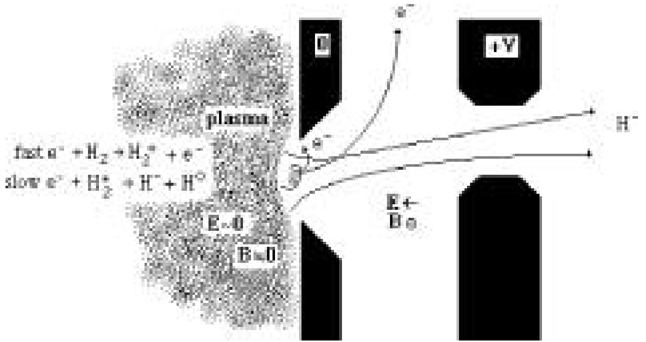


Fig. A.1

Schematic of a negative ion source. The electron and negative ion species should be orbit integrated. Because there is a nonzero magnetic field in the plasma and sheath, a Boltzmann approximation for the electron density is not valid.

Glossary

Busch's Theorem: As a consequence of conservation of canonical angular momentum, particles emitted in a low solenoidal field are spun up as the axial field increases.

Canonical Angular Momentum: Total angular momentum (physical plus magnetic field component)

Emittance: Measure of the quality of an accelerated beam (see **RMS Emittance**).

Finite Difference: The solution of differential equations by approximation of continous variables as descrete.

Gas Neutralization: A beam of charged particles traveling through a neutral background gas may partially ionize the gas, which may in turn cause partial screening of the beam's own space charge.

Iterative Static Equilibrium ("ISEQ") Method: Self consistent solution of Poisson equation and particle motion by iteration to convergence.

Hybrid Fluid Method: An ISEQ method with a special particle motion algorithm that includes fluid interaction (e.g. viscosity).

Laplace Equation: Equation for the electrostatic potential in a vacuum with applied potentials.

Maxwell's Equations: Equations for the electromagnetic field driven by charged particles in motion, as well as antennae.

Normalized Emittance: Emittance multiplied by the relativistic factors $\gamma\beta$.

Perveance: A relative measure of the influence of space charge effects in ion extraction and electron guns.

Phase Space: A mathematical space occupied by particles with axes being velocity and position.

Poisson Equation: Equation for the electrostatic potential in the presence of space charge from particles, as well as applied potentials.

Poisson-Boltzmann Equation: Equation for the electrostatic potential in the presence of space charge from ions, including screeninng effects of a background thermal distribution of electrons, as well as applied potentials.

Particle-in-Cell (PIC) Method: Self consistent solution of Maxwell's equations and particle motion by time stepping (integration in time).

RMS Emittance: Emittance of a distribution of particles in *x*-*x*' phase space calculated from

$$\varepsilon_{RMS} = \sqrt{\left\langle [x - \langle x \rangle]^2 \right\rangle \left\langle [x' - \langle x' \rangle]^2 \right\rangle - \left\langle [x - \langle x \rangle][x' - \langle x' \rangle] \right\rangle^2}$$

Space Charge Limited Flow: The maximum limit on current, due to space charge repulsion, that can be extracted from a surface (in time-independent situations).

Successive-Over-Relaxation: Numerical method of solving the Poisson equation in finite difference form.

Thermal Emission: Particles emitted from a Thermal Emission surface have a drifting Maxwellian distribution characterized by a temperature T.

Index