

'Special versions' of the Charged *Particle Optics* Programs

Five 'special' versions of the CPO programs are available, for 'user-defined cathodes', 'user-defined scattering', 'stochastic scattering', 'dielectrics' and 'neumann boundary conditions'.

The 'User-defined cathode' version

This version of the 2D and 3D space-charge programs allows the User to define the emission properties of new and unconventional cathodes.

The User can for example define a thermionic temperature that depends on position, or new forms of Schottky or cold-field emitters, such as liquid-metal ion sources.

As in the regular versions, the cathode is subdivided into segments and one ray starts from the centre of each segment. At the start of each ray the program sends information to the User-supplied routine on the positions of the cathode segments and on the strengths and directions of the electric fields at their surfaces. The routine then returns information on the emitted current densities. The rays are traced in the usual way, automatically taking care of any space-charge.

The User-defined cathodes fall into two types, depending on whether the space-charge in front of the cathode is significant or insignificant. In the first type the User can for example define a thermionic temperature that depends on position. The program will then automatically take care of the space-charge. In the second type the User can deal for example with new forms of field emitters. The choice of type is made in the databuilder. A User-supplied data file could be used to define parameters such as the maximum current density of a thermionic cathode or the work function of a field-emission cathode.

The CPO package includes 2D and 3D examples (in C++) that can provide a convenient starting point for further editing by the User.

The 'User-defined scattering' version

For more detailed information see the document 'Scatter'.

This version of the 2D and 3D programs allows scattering of the particles, using a User-supplied routine.

The scattering might be caused for example by:

- **a grid or mesh**
- **an artificial thin lens**
- **a background gas**
- **an electrode, to produce secondary electrons.**

At the end of each step in the ray tracing the program sends information (the coordinates, velocity components, potential, time, etc) to the User-defined routine and expects a return giving new values of the velocity components, charge and mass, or an instruction to terminate the ray.

The CPO package includes two sets of examples, for 2D and 3D programs respectively. Each set contains four examples of User-supplied routines. These routines (in C++) provide a convenient starting point for further editing by the User.

In brief:

(1) The first example routine in each set deals with scattering of rays at a grid or mesh. Here the grid is simulated by a scattering plane which is situated at $z=c$. When a ray reaches this plane its direction is changed by a random amount. The value of c and the statistical distributions of the changes in the x and y directions are controlled by numbers that are read from a User-supplied data file. These numbers would have been obtained beforehand from a more detailed simulation of the actual grid.

(2) The second example routine in each set deals with focusing by an artificial thin lens situated at $z=c$. This can be useful for example in optimising complicated systems of lenses in which one or more of the lenses have already been carefully designed. These lenses can then sometimes be effectively replaced using the scattering option. In the present example the value of c and the focal length and spherical and chromatic aberration coefficients are read from a User-supplied data file.

(3) The third example routine in each set deals with the production of secondary electrons at an electrode situated at $z=c$. When a ray (perhaps representing an ion) reaches this plane it is converted to an electron. The statistical distribution of the energies and directions of the secondary electrons are controlled by numbers that are read from a User-supplied data file.

(4) The final example routine in each set deals with the scattering or loss of rays due to collisions with background gas molecules. Again the details are controlled by numbers that are read from a User-supplied data file.

The 'User-defined stochastic scattering' version

This version of the 2D and 3D programs allows stochastic scattering (*ie* individual electron-electron or ion-ion Coulomb interactions) of the particles, using a User-supplied routine. It allows accurate calculations of effects such as Boersch energy spreading or spatial blurring, for any shape or size of beam and with no significant approximations or assumptions having to be made.

The version is essentially the same as the scattering version described above, except that additional data is sent to the User-supplied routine to enable that routine to simulate stochastic scattering. The additional data includes:

- (1) the local space-charge density in the vicinity of the mid-point of the present trajectory integration step,
- (2) the velocity components of the 10 trajectories that are nearest to the present trajectory,

An example User-supplied routine which shows how this information can be used to simulate the Coulomb scattering in individual electron-electron encounters is included in the package.

This is a powerful option. The technique used is fundamentally different from that used in other commercial programs and its accuracy has been verified by a benchmark test. This option has been used for several different types of simulations, including the most difficult simulation of stochastic scattering in a liquid metal ion source, which does not seem to be possible with other techniques.

'The 'dielectric' version of CPO3D.

This version treats non-space-charge systems that include dielectric materials that have a uniform dielectric constant (also called the relative permittivity).

Dielectric systems are very easy to set up. The interfaces that enclose each dielectric medium are represented by quasi-electrodes, called 'dielectric electrodes', that have the same choice of geometries as the normal 'conducting electrodes'. After defining the 'dielectric electrodes' all the user has to do is to specify the dielectric constant $K1$ of the medium and the dielectric constant $K2$ of the surrounding volume (usually 1), and also specify a 'reference point' and to say whether this reference point is inside or outside the medium. More than one dielectric medium can be included.

The 'dielectric electrodes' can be extremely close to 'conducting electrodes', although not touching.

Four benchmark tests of dielectric systems are included in the package:

- a spherical capacitor with dielectric in gap,
- a parallel plate capacitor with dielectric in gap and
- the field in a cavity inside a dielectric.

In all cases the error is typically 0.3%, using a fairly small number of segments. This error is larger than the extremely small errors achieved with the non-dielectric versions of the CPO programs, and the computing times tend to be longer, but the error is nevertheless acceptably small.

Full instructions are given in CPO Help on how to set up a dielectric system.

'The 'Neumann boundary condition' version of CPO3D.

In this version the electrostatic fields at selected boundaries can be specified instead of the potentials. This version is similar to the dielectric version.

Help and cost

Full instructions are given on how to produce a 'dll' ('dynamic linked library') file from a User-supplied routine. Examples of such routines are supplied. CPO Ltd will offer advice and help in constructing new User-defined routines.

Price

The special versions (perpetual licences only) for user-defined unconventional cathodes, scattering and stochastic scattering, each cost an additional 25%. The dielectric and Neumann-boundary-conditions versions (CPO3D only) each cost an additional 50%.