

A Monte Carlo Example

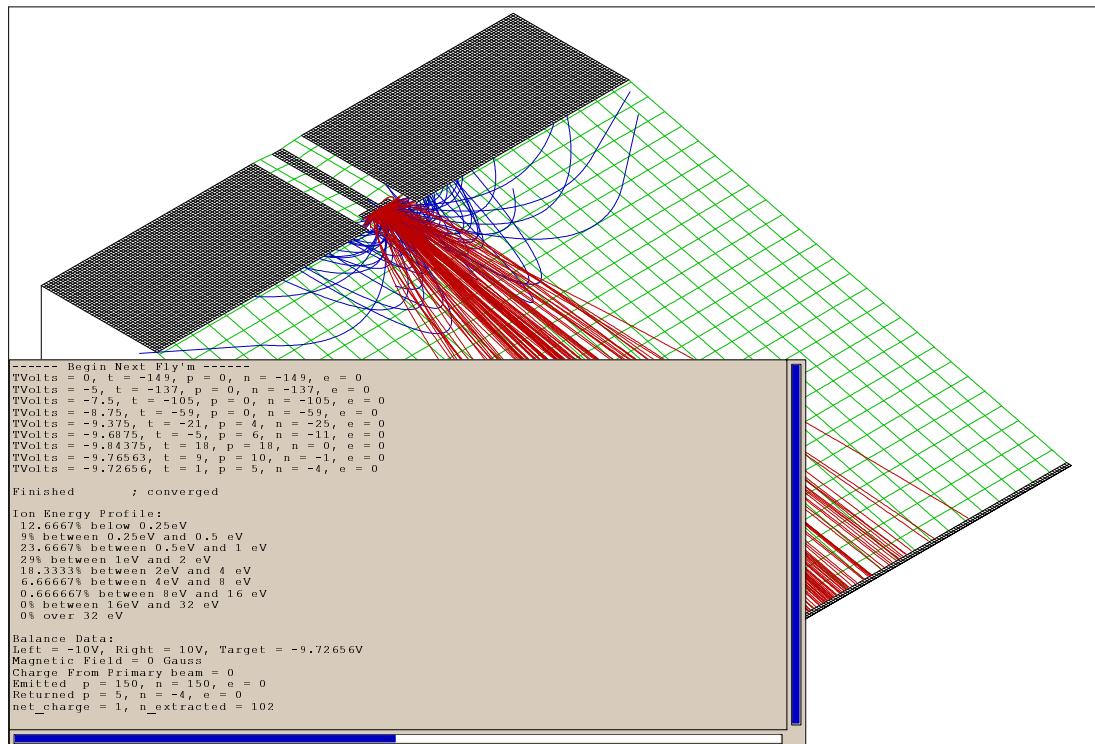
Introduction

The user program example in this directory is a simple model of self-charge stabilization [1]. Self-charge stabilization is a charge compensation method developed in our group for secondary ion mass spectrometry applications. In this the SIMS target is floated and the ion and electron trajectories are used to create an electrostatic servo that automatically controls sample charging. See the reference for more details.

What the simulation does

In this simulation a Monte Carlo approach is used to demonstrate self-charge stabilization. **Be sure to refine the RING.PA# file before trying to run the simulation.**

To run the simulation at its simplest level load the ion optics bench file **Charge Balance.iob** into SIMION and click Fly'm. The simulation flies 150 positive and 150 negative ions to simulate the random emission of secondary ions from the front surface of the target. A neutral primary beam and no secondary electron emission is assumed (a rather over simplified first assumption). The user programs keep track of each ion's fate. The target initially loses the charge from their departure (0 corresponds to equal numbers of positive and negative ions). If ions return to the target their charge is returned also.



Results of the first flight

At the end of flying all ions the net charging of the target is calculated. If the target charges more positive, its voltage is changed to the midpoint value between what it was and the current max

User Programs

Reference

Monte Carlo Example

voltage. The new min voltage is set to the old target voltage. A similar strategy is used if the target charges more negative.

The simulation is then rerun with the new target voltage. This binary approach is repeated each iteration until the charge convergence criteria is reached. At this point the results are summarized as shown in the figure above.

The kinetic energy distributions of emitted secondary ions and electrons (if any) are shown in the summary. The Initialize segment of the user program contains functions that mimic the observed energy distributions of secondary ions and electrons. Notice that most of the ions escape the target (very leaky by design). Only a few return to maintain charge balance (electrostatic servo control).

The demo simulation is extracting negative ions with the electrode surrounding the target set to -10 V and the extraction electrode (to the right) set to $+10$ V. Note that charge balance occurs at around -9.72 V, and that about 100 of the 150 negative secondary ions strike the extraction electrode.

Effect of secondary electrons

There is a .FLY file called **150 pos neg elect.fly** that also contains 150 secondary electrons. If you load and run this file a striking change occurs. The charge balance shifts dramatically. The balance potential is now around -7.3 V and only 14 of 150 negative ions actually reach the extraction electrode. The higher energy secondary electrons preferentially escape forcing the lower energy secondary negative ions to return to achieve charge balance.

Using a magnetic field to shift the balance

The user programs allow a magnetic field to be defined (via mag.pa and its user program file) across the face of the target. If the magnetic field is set to 500 gauss the electrons are preferentially scavenged by the target. This restores the charge balance to about -9.69 V and around 100 of 150 negative secondary ions reach the extraction electrode.

Other games

You have the option of specifying charging from the primary beam (for non-neutral beams). Moreover, when you click fly'm the adjustable variable option screen has an OK button. Depress the button and ALL the adjustable variable will be shown. Look at the user program segments to determine what the variables do and perhaps what you might want to try. Have fun.

About the user programs

The following information provides insight into the user programs and how they work. It is suggested that you print a copy of **ring.prg** and **mag.prg** to follow along in the discussion.

Iteration method

The net charge is updated according to the fate of each secondary ion and electron. After all ions have flown the net charge should be zero if we have converged or are within the convergence criteria (default of 2 charges).

If the convergence criteria has been achieved the terminate program is exited with the rerun flag not set. The energy spectra for ions and electrons are displayed as well as the balance conditions achieved.

If the convergence criteria has not been achieved the terminate program calculates the next target potential to use (based on a binary approach scheme) and sets the rerun flag so that the simulation will be rerun with a re-randomized collection of ions.

The initialize program segment

The initialize program segment has two tasks. The first is to initialize key variables before the first fly'm. The second is to initialize key variables before each fly'm. Adjustable variables serve to flag first flight and first ion to speed up the process.

Two fitted functions are used to create secondary ions and electrons with representative initial energies. These functions are selectively used depending on whether the ion is an ion or an electron.

Counters are maintained of the numbers of positive and negative ions encountered as well as electrons. The initial energies of ions and electrons are tracked separately so that separate energy spectra can be displayed.

Emission directions and positions are calculated based on user specified cone angles, offset displacements, and technique flag.

The voltage adjustment program segment

The Init_P_Values program segment was used instead of the Fast_Adjust program segment to adjust the electrostatic potentials. The reason for this is that the potentials need only be set at the beginning of the flight (they remain constant). Thus this approach should be faster (perhaps 30%) than using Fast_Adjust. You can switch comments in the user program file to swap between these two approaches to verify that they both work with the same lines of code.

The fastest runs are obtained with trajectory quality at zero (really doesn't change balance potential much) and trajectory viewing and recording turned off.

The other_actions program segment

The other actions program segment is used for several tasks. When magnetic fields are active electrons can occasionally spiral for long periods of time in front of the target. This slows down the simulation. Thus a time step counter is included for electrons that automatically kills an electron when it has been integrated more than a fixed number of time steps (1,000 is the default).

When an ion is killed (splattered) its location is checked to see if it has hit the extraction electrode (been extracted – electrons don't count) or has returned to the target. If it has returned to the target its charge is added back to the net charge of the target for charge balance purposes.

The terminate program segment

The terminate program segment is used to test for convergence to charge balance (zero change in net charge). If the convergence criteria is reached (default = 2 charges) the simulation is exited after displaying the energy distributions for the ions and electrons (if any) along with the offset potential and other parameter values at termination.

If convergence has not been obtained, a binary approach algorithm is employed (divide by two) to adjust the range of estimation and obtain the next trial value for the target potential. The rerun flag is set and a new group of re-randomized ions are flown.

The Mag.prg file

The mag.prg user program file is used to provide an optional transverse magnetic field in front of the target to suppress electron emission. A uniform magnetic field is assumed. The mag.pa is a dummy magnetic array with no pole points. A mfield_adjust program segment is used to specify a magnetic field in the x direction (0 by default). Magnetic fields in the y and z direction are always zero. The projected instance of mag.pa is oriented so that the magnetic field traverses the target's front surface.

¹ D.A. Dahl, A.D. Appelhans, Int. J. of Mass Spectrom. 178 (1998) 187-204