In this paper we give an overview over our algorithms developed for space charge calculations, the recently implemented tracking routines and our plans for electron cloud simulations.

Since 1999 fast algorithms for 3D space charge simulations, more precisely fast Poisson solvers, have been developed by G. Pöplau. Based on the multigrid technique these Poisson solvers have optimal performance, i.e. the numerical effort depends linearly on the number of macro particles. Another goal of these algorithms is that they are constructed for non-equidistant tensor product meshes which allows for instance a discretization of the bunch adapted to the distribution of the macro particles. The algorithms are available as software package MOEVE 2.0 (MOEVE: Multigrid for non-equidistant grids to solve Poisson’s equation) [7]. More precisely, MOEVE includes different iterative solvers: geometric multigrid (MG) and multigrid pre-conditioned conjugate gradients (MG-PCG), a pre-conditioned conjugate gradient method (PCG) with Jacobi pre-conditioner, (mainly for comparison reasons) the successive over relaxation (SOR), Biconjugated gradients (BiCG) and BiCGSTAB as stabilized version of BiCG. The algorithms of our software package we call MOEVE Poisson solvers in the following. The software package MOEVE contains three different types of boundary conditions: open boundary conditions, a perfectly conducting boundary conditions on the space charge fields is not trivial especially if one observes the fields far from the beam axis and near the boundary. This certainly has a big importance for the simulation of the e-cloud build-up and its dynamics because different space charge fields could mean different energies and number of electrons in the cloud. Motivated by [1] the results of the solver on elliptical shaped domains are compared with the usually chosen open or Dirichlet boundaries on a rectangular box [5, 6].

Furthermore, in order to advance the charged particles in the presence of an electromagnetic (EM) field we have implemented the algorithm known as the Boris pusher [2]. It has been successfully implemented in many particle-in-cell (PIC) codes and the penalty for its parallelization is relative low. The EM fields are grid quantities while the quantities related to the particles (position and momentum) are stored with the particles which can be anywhere inside the computational domain. In order to update particle position and momentum, the EM fields on the grid are interpolated at the position of each particle. For the computation of the space charge fields the charge of the particles should be deposited on the grid points. When a particle hits the wall, the position, the incidence angle and the energy of the particle are known and can be used as an input to an e-cloud modelling routine. Although the electrons and the beam particles are defined in the same lab frame the space charge computation should be done separately for the two kinds of particles. For the relativistic beam particles the computational grid (in the rest frame) is stretched in longitudinal direction by the relativistic factor γ. The potentials computed from both kinds of particles are superposed on the grid in the lab frame.

A detailed description of the Poisson model for a beam pipe with elliptical cross section is given in [4]. As illustrated in [5, 6] the influence of different boundaries and boundary conditions on the space charge fields is not trivial especially if one observes the fields far from the beam axis and near the boundary. This certainly has a big importance for the simulation of the e-cloud build-up and its dynamics because different space charge fields could mean different energies and number of electrons in the cloud. Motivated by [1] the results of the solver on elliptical shaped domains are compared with the usually chosen open or Dirichlet boundaries on a rectangular box [5, 6].

Furthermore, in order to advance the charged particles in the presence of an electromagnetic (EM) field we have implemented the algorithm known as the Boris pusher [2]. It has been successfully implemented in many particle-in-cell (PIC) codes and the penalty for its parallelization is relative low. The EM fields are grid quantities while the quantities related to the particles (position and momentum) are stored with the particles which can be anywhere inside the computational domain. In order to update particle position and momentum, the EM fields on the grid are interpolated at the position of each particle. For the computation of the space charge fields the charge of the particles should be deposited on the grid points. When a particle hits the wall, the position, the incidence angle and the energy of the particle are known and can be used as an input to an e-cloud modelling routine. Although the electrons and the beam particles are defined in the same lab frame the space charge computation should be done separately for the two kinds of particles. For the relativistic beam particles the computational grid (in the rest frame) is stretched in longitudinal direction by the relativistic factor γ. The potentials computed from both kinds of particles are superposed on the grid in the lab frame.

Our plan for self-consistent beam-e-cloud simulations with focus on a single bunch instability is to have a full 3D computational domain around the bunch moving through the e-cloud in the lattice element of interest. Through the remaining part of the lattice the bunch should be mapped and transported to the next lattice element where it interacts with an e-cloud. After the first passage of the bunch through a certain lattice element the decay of the e-cloud in the element will be simulated and stored. The decay will be simulated over the time equal to the time distance between two neighboring bunches in the train. The presumption of a certain periodicity in the e-cloud decay allows to load the
stored e-cloud distribution after the decay for each further bunch passage.

REFERENCES


