FERMILAB-TM-252APC

A comparison and benchmark of two electron cloud packages: POSINST and VORPAL.

Paul L. G. Lebrun,[1] James F. Amundson,[1] Panagiotis G. Spentzouris,[1] and Seth A. Veitzer[2]

[1] *Fermi National Accelerator Laboratory, Illinois, USA*[*]

[2] *Tech-X Corp., Boulder, Colorado, USA*

(Dated: January 10, 2012)

Abstract

We present results from precision simulations of the electron cloud (EC) problem in the Fermilab Main Injector using two distinct codes. These two codes are (i)POSINST, a F90 2D+ code, and (ii)VORPAL, a 2D/3D electrostatic and electromagnetic code used for self-consistent simulations of plasma and particle beam problems. A specific benchmark has been designed to demonstrate the strengths of both codes that are relevant to the EC problem in the Main Injector. As differences between results obtained from these two codes were bigger than the anticipated model uncertainties, a set of changes to the POSINST code were implemented. These changes are documented in this note. This new version of POSINST now gives EC densities that agree with those predicted by VORPAL, within $\approx 20\%$, in the beam region. The root cause of remaining differences are most likely due to differences in the electrostatic Poisson solvers. From a software engineering perspective, these two codes are very different. We comment on the pros and cons of both approaches. The design(s) for a new EC package are briefly discussed.

PACS numbers: 29.27.-a

[*] lebrun@fnal.gov

Operated by Fermi Research Alliance, LLC under Contract No. De-AC02-07CH11359 with the United States Department of Energy.

I. MOTIVATION

The electron cloud (EC) phenomenon in high intensity proton storage rings and synchrotrons can limit the performance of such machines [1, 2]. This phenomenon is characterized by an exponential growth of the number of low energy (eV) electrons emitted at the surface of the beam pipe wall. Such electrons are then accelerated by the field induced by the passage of the proton beam, which itself causes more secondary emission of electrons at the beam pipe wall. Numerous simulation efforts have been conducted using various computational codes. In particular, the POSINST [2, 3] code has been used to model EC in the context of the Fermilab Main Injector. VORPAL [4] is a newcomer in this forum. VORPAL gives us the opportunity to perform fully 3D simulations to obtain electric field maps in non-trivial cases. There is however a recognized need for a benchmark with relative performance studies.

We would like to know if we will have a beam instability problem in the Main Injector attributed to an electron cloud when the bunch intensity increases from the current operating value of 0.7×10^{11} to $3. \times 10^{11}$ protons per bunch. Without prior knowledge of the secondary emission yield (SEY), simulations alone can not make definite predictions for the EC density. On the other hand, under the current operating conditions, there is no indication that EC induced beam instabilities do occur. Thus, either the EC beam instability has already been mitigated, or the SEY is small enough that EC buildup is sufficiently slow as to not be a problem. In the former case, the peak SEY is already above $\approx$ 1.4 secondaries/primary. As this is the most worrisome scenario we focused our attention on a relatively large maximum SEY for clean stainless steel surface of about 2.2 secondaries/primary. Under such conditions, in the MI dipoles, VORPAL predicts that the linear density of the EC space charge will be about 80% to 90% of the linear density of the proton beam.

A relatively recent update of the POSINST calculation [5] leads to a different conclusion regarding the dependence of the EC density on the bunch charge: Using POSINST, a sharp increase of the EC density occurs when the bunch charge increases, in contradiction with our VORPAL-based results. Also, POSINST produces deep fluctuations of the EC density due to a *virtual cathode* effect, that are not observed in VORPAL simulations.

Sorting out these discrepancies via a complete and thorough benchmark has been given high priority. Producing more results based solely on VORPAL will not be credible unless an understanding of such discrepancies is clearly spelled out. This note summarizes the results of this benchmark exercise and is organized as follows: First, a brief introduction, which includes a description of the benchmark, and the required improvements made to the POSINST code - and their justifications - are given. Next, we present new POSINST results and a comparison with VORPAL, with the associated beam physics implications. Some technical details about the scientific computing engineering used in this work are also discussed.

II. THE POSINST/VORPAL BENCHMARK

The goals of our simulation program are to guide the electron cloud experiments and interpret their results. Thus, the setup for such simulation must be realistic enough, without wasting computing resources on irrelevant details. This is an iterative process, i.e., such benchmark scope and assumption should never be cast in stone, as the realm of relevancy depends on the application. A simulation code may work fine in one specific case and completely fails in the next problem. In our case, some of the outputs of the simulations are a set of electric field maps and quantitative estimates of the electron cloud density in the Main Injector (MI). While details on the EC simulation setup for the MI are detailed in reference [6], the currently relevant specifics are:

- The static magnetic field that confines the EC is the one found in the Main Injector dipoles. While there is interest for other magnetic field configurations such as quadrupoles and drifts, the MI is mostly made of such dipoles. Also, easy mitigation strategies for the EC problem, based on the use of small and weak solenoids, exist for drifts. If the EC problem is found to be nonexistent in dipoles, we have a good chance that this problem would not be observed in the quadrupoles as well.

- In these dipoles, magnetic confinement is also strong: the Larmor radius for ≈ 300 eV electrons is very small compared to the physical dimension of the beam pipe. In this restricting case, if the bunches are long compared to the Larmor radius and the transverse size of the beam, then the problem is “2D+”, in the sense that the longitudinal electric fields can be neglected, and the only relevant motion of the electrons is in the transverse plane, along the magnetic field lines. The Larmor phase is no longer relevant, as it has has no impact on what the space charge inside the EC is. This the realm of validity for POSINST and VORPAL 2D electrostatic models.

- The electrons in the EC are non-relativistic and do not radiate. The only relevant fields that can accelerate the electrons are those coming from the space charge in the EC and from the proton beam. Then, a relatively simple time-domain implementation of the problem can be devised, based on successive slices of the bunch, where static 2D (transverse) electric fields dictate the dynamics. Again, these are the realm of POSINST and VORPAL electrostatics.

- The seed electrons from which the EC builds upon are coming from a single process, namely, ionization out of the residual gas. This is thought to be the dominant mechanism in the MI, as synchrotron radiation emitted from the proton beam can be neglected. While beam losses could be a significant factor, such a process also difficult to predict and implement. To keep the benchmark relatively straightforward, this last source of seed electrons has been omitted from the VORPAL and POSINST simulation.

- The range for the secondary emission yield (SEY) must be large enough to incorporate the difficult scenarios where the EC density is high and beam instabilities become a concern. A peak SEY value above 2, ≈ 2.2, used in this benchmark, has not been entirely ruled out for "un-scrubbed" stainless steel found in the MI beam pipe. Other parameters for this type of surface have been picked out from reference [7].

- A relatively simple bunch fill pattern was used. In particular, we are interested in the EC density in a steady mode, with continuous trains of bunches. This is also the worst case, when the EC has no time to decay in absence of the proton beam.

- The beam is not necessarily centered on beam pipe axis, as misalignments of accelerator components is often unavoidable at the level of a few mm. These are corrected for by imposing corrective "bumps", or non-zero orbits. In our case, an vertical offset of 2.5 mm (or 0.83 of the beam's σ) has been arbitrarily imposed to the proton beam with respect to the beam pipe.

III. MODIFICATION TO THE POSINST CODE

A. Motivation

A fair question is to ask why did we feel compelled to improve the POSINST code, rather than questioning the new framework, VORPAL. Before answering this legitimate question, let us note that a partial benchmark between POSINST and VORPAL was actually successful, in the sense that we got the same answer for the following restricted circumstances: We took the EC phase space[1] generated by VORPAL, and used this as an initial condition in POSINST, switching off the proton beam. The lifetime of the EC in presence of its own space charge was found to be consistent between the two codes.

Both codes lead to a quasi equilibrium of the EC charge density, bunch after bunch, but the EC steady state charge density was consistently higher in VORPAL then in POSINST, by a factor three to four on average over the entire volume of the beam pipe, and by about a factor two around the beam. This gave us reasons to look at the implementation of the various production mechanism for secondary electrons in both codes. The seed electron density follows the proton beam profile. The ionization rate does not matter once the EC phenomena occurs, as the seed density is at least 4 to 5 orders of magnitude smaller than the quasi-equilibrated EC density. Only the geometry of the seed EC matters, and it simply matches the transverse distribution of protons in the bunch.

A production model for secondary electrons at the beam pipe walls has been agreed upon. While the surface physics is very complex and highly uncertain, both codes are based on a common description of the secondary emission phenomenas. Initially implemented in POSINST, a simple probabilistic model describes three distinct processes: elastic scattering, re-diffusion and genuine production of secondary electrons. The formalism and its associated equations [5], have been dutifully re-implemented in the TxPhysics numerical library, used by the VORPAL code.

However, some details of the implementation are different: Recent versions (v5p3) of POSINST were fine tuned for rather small SEY[8]. To save memory and CPU time, the number of true secondary electrons could not be higher than 10 per primary electron, no matter what the energy of that electron is. The VORPAL implementation has no such

[1] Position, velocities and electric charge of each simulation macroparticle

limitations. Aiming for more functionality in both codes, the VORPAL scripts were checked and found adequate. The TxPhysics library functions that drive the secondary emission of electrons were also tested separately from VORPAL, and found to conform to the equations in reference [5]. Meanwhile, a second look at the implementation of the true secondary emission for large SEY, in POSINST, was found to be warranted. As a result, POSINST functionality has been slightly modified, while minor changes to the VORPAL scripts were deemed necessary to match the requirements from newer versions of VORPAL.

B. Changes to the POSINST code

In terms of added lines, many of them are simple debugging statements or I/O facilities to control the information flow. As no formal flow charts nor detailed technical notes on such programs are published, one has to rely on such old fashioned, straightforward hacks to find out what's going on. Besides those, the important changes and upgrades are:

1. Extension of the memory for the EC phase space data. The stability of the EC density against parametric changes (seed density, culling factors, etc..) and checks on the accuracy in the space charge calculation leads to always increasing the number of macro-electrons in the simulation. The original version of POSINST was targeted to run on rather small computers, where memory (heap size) was a concern. Even with 2 million macro-electrons, we now typically use only a few percent (or less) of the memory available on our workstations or on the Accelerator Simulation Cluster [9] at Fermilab. The parameter *nemax* has been changed from 100 thousands to 2 millions. Also, to prevent accidental overwrite of the *phasespace* F90 module (a.k.a., a good old fashioned, fixed size, FORTRAN COMMON block), a check on the availability of memory prior to the copy of the transient secondaries in this *phasespace* module has been implemented.

2. Uploading of EC *phasespace* information from VORPAL electron dump files. A specialized HDF5 to ASCII stand-alone translator has been written and a simple I/O function has been implemented in POSINST. An optional pre-scaling filter has also been provided, to avoid an excessive number of macro-electrons.

3. The POSINST *secelec* module has been rewritten, such that it can be tested stand-

alone against VORPAL as well as figures provided in reference [5]. The Poisson fluctuations of the true secondary rate are now computed based on an algorithm from *Numerical Recipes* [10]. The arrays containing the temporary values for kinematical variables associated to true secondary electrons have also been extended from a maximum of 10 to a maximum of 100 electrons per single incident primary electron.

This maximum size was determined by noting that the peak flux for secondary electrons, including elastic scattering and re-diffusions, was set to 2.2. Since the probability for each of these two production processes was set to 7 and 74% respectively, the average number of true secondaries for this third process, which occurs with a probability of 19%, must be about 7.3 to reach an average flux of 2.2 secondaries electrons per primary electron. Thus, reserving a maximum of 10 secondary electrons per incident electrons was clearly inadequate.

This arithmetic was verified via a stand-alone program, whose results were compared against the TxPhysics library secondary emission results.

All changes to the original v5p3 version have been maintained via the "git" tool [11], for tracking and, should our EC community wish, for dissemination purposes. The package ran under *valgrind* to check for eventual memory problems. After all the above modifications were implemented, none were found.

IV. BENCHMARK RESULTS

The new POSINST results were found to be numerically stable against increases of the number of time slices and macro-electrons. Also, a "flat culling" algorithm was revived and found to give stable average densities, at any time during the growth period or the quasi-steady state of the EC. This "flat culling" algorithm is straightforward: once the number of macro-electrons doubles, half of the macro-electrons are removed from the simulation and the charge associated to the remaining macro-electrons is doubled. The probability to remove an electron is completely independent of the position of the macro-electron in 6D phase space. This works because the number of macro-electrons per cell is sufficiently high at any time during the simulation. The MI setup and beam conditions are those stipulated above, for a beam intensity of 7.0×10^{10} protons per bunch. The grid spacing is 64 by 64, and covers the entire cross-section of the beam pipe, for both codes.

As a result, the electron density is a smooth function of time within a few % as shown in figure 1. The mixed-fill pattern for the bunches from the Booster batches is realistic and taken from reference [5]. The rise of the EC density at the beginning is fictitious, as the fill pattern wraps around the 13 μ sec one-turn cycle of the MI. However, the decay followed by the re-birth of the EC after the first Booster batch is realistic, and is a non-trivial result of our simulation.

Even at lower SEY ($\approx$ 1.4), there are no indications of sudden and seemingly chaotic change in density, i.e., no "virtual cathode" effects mentioned in reference [5].

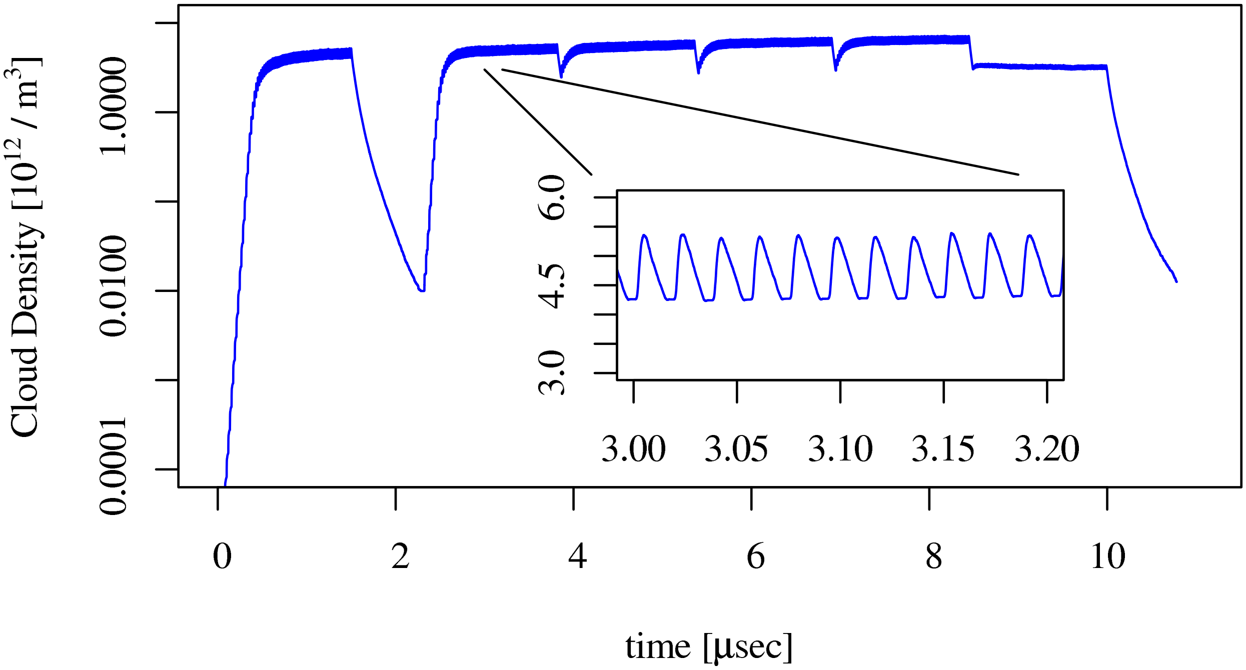

FIG. 1. The electron density, in units of 10^{12} electrons per cubic meter, is shown for a complete MI turn. Also shown in the inset, on a linear scale, the same density for $\approx$ 10 18.8 ns long bunches, during the quasi steady state, in the middle of the second continuous bunch train.

A comparison of EC density between VORPAL and POSINST is shown in more detail in figure 2, in the quasi steady state situation. We still have a discrepancy of about 20% in EC density between the two codes. However, near the beam, where it counts regarding EC induced beam instabilities, the agreement is much better. The discrepancy for the average (over the entire pipe) density has been traced back to a sharp increase of the EC near the wall and is attributed to a difference in the Poisson solver.

Using a facility by which a VORPAL EC can be dumped into the POSINST run, one can compare the electric field produced by the EC in both codes, for a given density. This is shown in figure 3. This has been done for beams that are centered on the beam pipe axis, for sake of further simplifications. A rescaling of $\approx 20\%$ of the EC induced electric field in POSINST leads to commensurate changes in EC density.

The accuracy of the calculation of this EC's space charge induced electric field is a bit unwieldy in POSINST: this is a code that has not yet been parallelized and for which only selected grid spacing is allowed. VORPAL has a bit more flexibility in choosing the grid spacing and runs on large multi-core, multi-node computers. Thus, our comparison ends here.

V. MORE RESULTS FROM POSINST

A study of the EC density versus peak SEY and bunch intensity has been performed using the fixed POSINST code. Shown on figure 4 is the is the EC density vs time, at a beam intensity of 0.7×10^{11} protons per bunch and for a peak SEY ranging from 1.1 to 2.2. As previously found, there is sharp threshold at peak SEY ≈ 1.3, below which the EC is evanescent and above which the nearly saturated linear charge density of the proton beam and of the EC balance each other. The growth time of the EC is also much slower at lower SEY. Once we are above the critical threshold, the EC density becomes rather insenstivie to the peak SEY.

Our aim is to determine what the EC density under Project-X conditions will be, where the bunch intensity will reach 3.0×10^{11} protons per bunch. The result of our new POSINST simulation is shown in figure 5. Scaled back to the current bunch intensity conditions, the EC density will be in fact slightly lower than it is now. This paradoxical result comes from two facts: First of all, at a peak SEY at or above the critical value, the EC saturates, i.e., its linear density is close to the linear density of the proton beam. The dynamics of the EC is dominated by self-space charge close to the wall. Secondly, the average kinetic energy of the electrons reaching the wall in time to repopulate the cloud is a bit too high for optimum production of secondary electrons, at a bunch intensity of 3×10^{11}. This result is consistent with previously obtained VORPAL calculations, presented at ECloud10 [12].

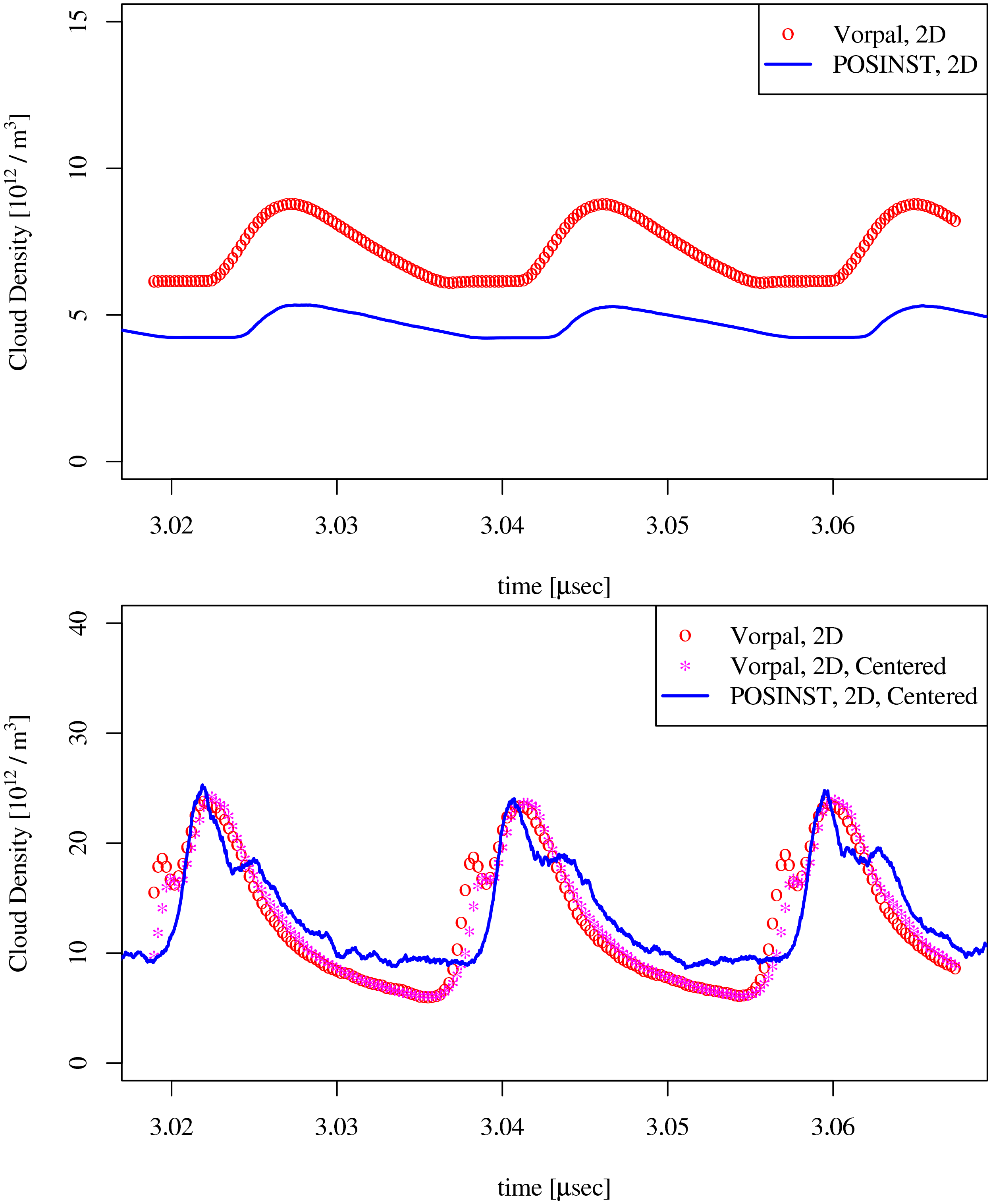

FIG. 2. The electron density, in units of 10^{12} electrons per cubic meter, is shown during a quasi steady-state period of a few 18.8 ns bunches, for both VORPAL and POSINST. On the top graph, the EC density averaged over the entire cross-section of the beam pipe. Bottom plot: The EC density over a one beam σ cylinder centered on the beam pipe axis, in the case of POSINST, and either centered on the displaced beam, or (red open circles) on the beam pipe center (magenta stars), respectively.

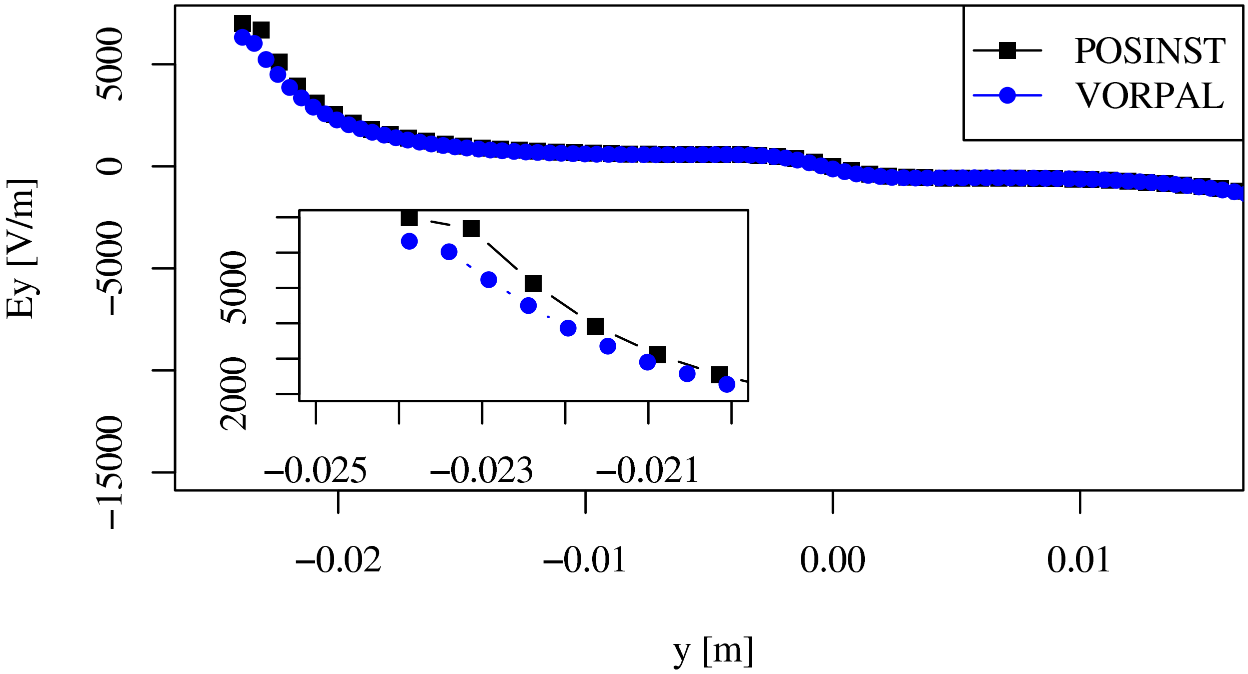

FIG. 3. The EC induced electric field for POSINST and VORPAL.

VI. STATUS AND PLANS

Regarding software engineering, POSINST and VORPAL are quite distinct from each others. They not only differ on the programming language (respectively, FORTRAN (F77 and some recent constructs from F95) and a mixture of C++, Python, with scripts written in a domain specific language resembling Python), but also in scope and methodology. POSINST was written by a few authors with a limited scope (EC studies only), while VORPAL is a generic beam and plasma simulation tool. POSINST is distributed informally among interested parties, while VORPAL is proprietary software with tested releases and aggressive regression testing.

Thus, the method for debugging such codes are fairly different. On the POSINST side, the casual user has near immediate access to all the code (about 10,000 lines). Inserting F77 style *PRINT* statements is straightforward. The limited functionality and compactness of POSINST becomes an advantage. On the VORPAL side, one tends to rely on the correctness of the core functionality of this well engineered package, and must look for mistakes in VORPAL input files. However, because of the complexity of some of these processes, no immediate knowledge of what VORPAL actually does is a available. Expertise in reading and interpreting the VORPAL documentation is often required. This work could not have

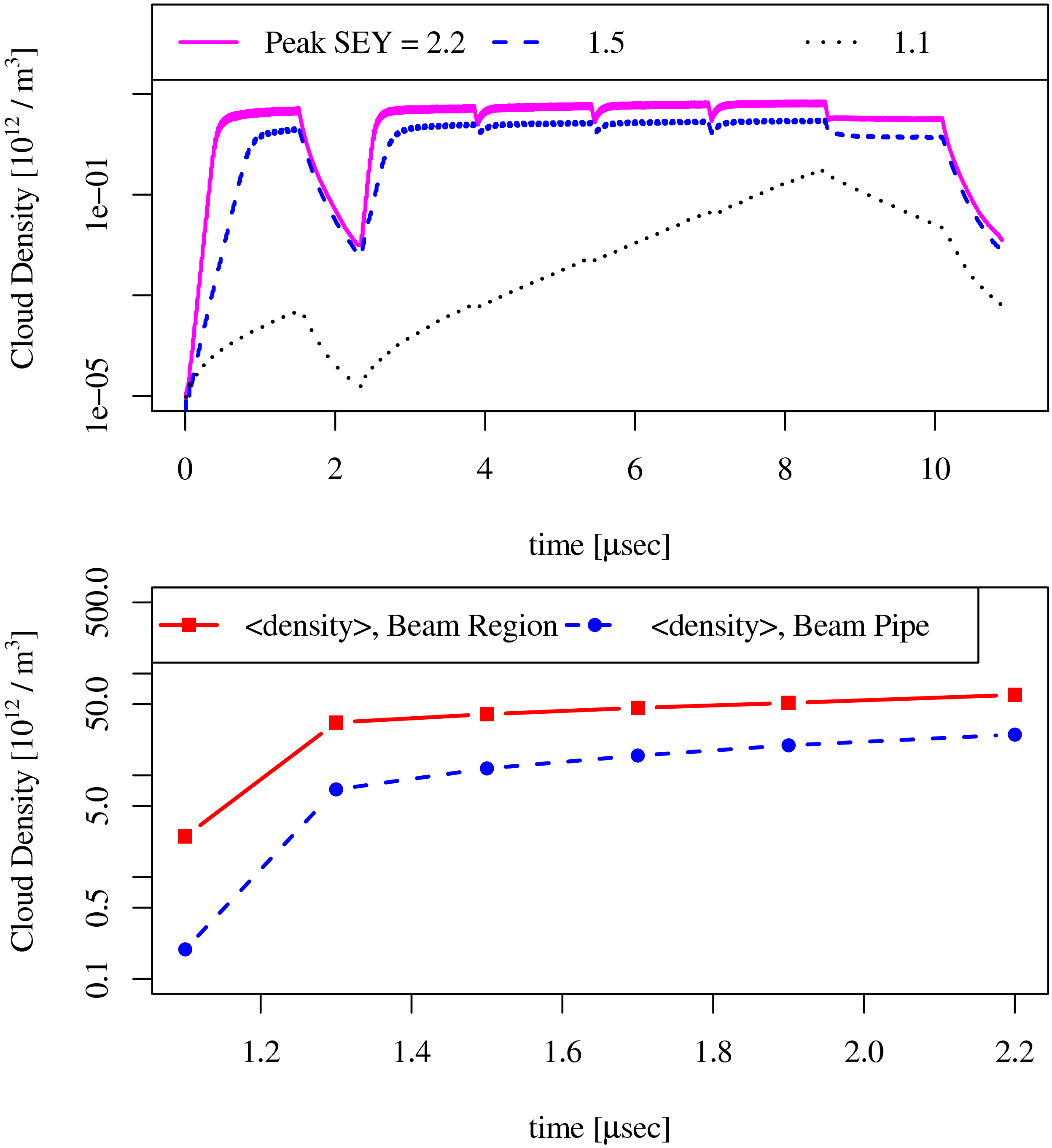

FIG. 4. Top: The electron density, in units of 10^{12} electrons per cubic meter averaged over the entire beam pipe volume and for the entire MI cycle is shown versus time, for various SEY coefficients. The bunch intensity is 0.7×10^{11} protons per bunch. Bottom: These densities have been averaged over the most intense booster batches, from $3.75 \times \mu$ sec to $8.5 \times \mu$ sec into the MI cycle and are shown versus the peak SEY.

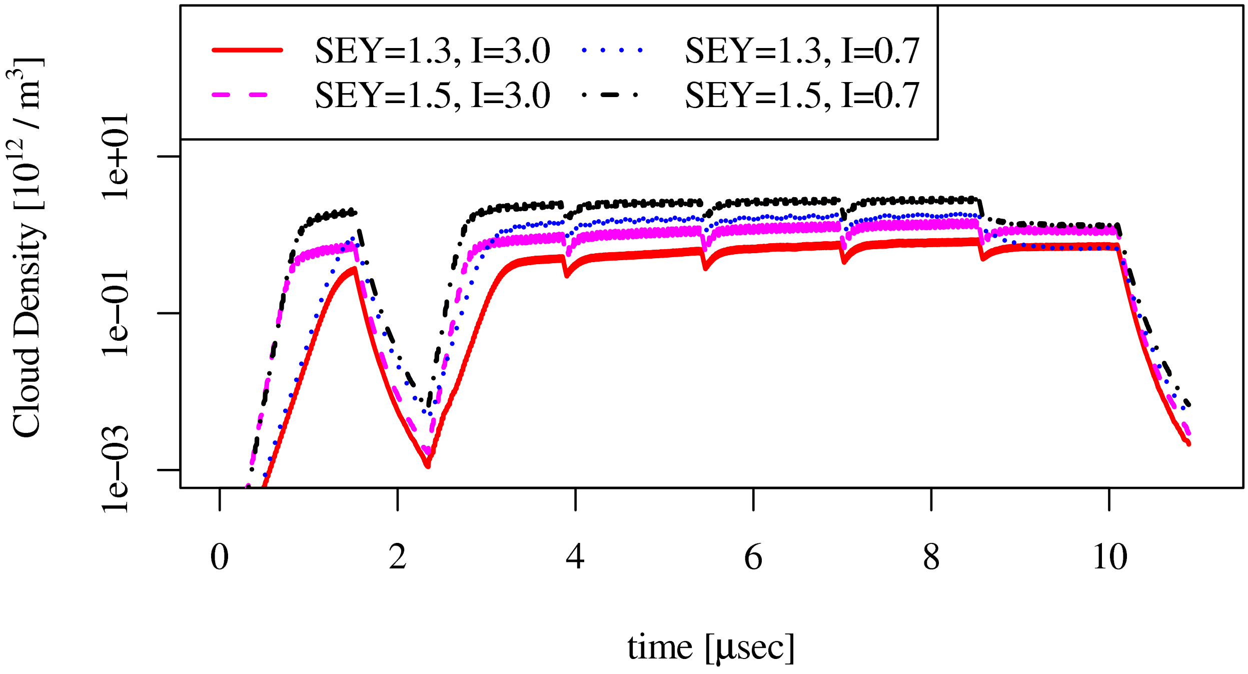

FIG. 5. The EC density over the enire pipe, for two bunch intensities and two peak SEYs, close the critical value. The EC density for the Project-X era have been scaled down by a factor 3.0/0.7. Normalized to the proton current, the EC density will actually be lower than under the present running conditions.

been successful without the close collaboration of Tech-X experts.

Synergia[13] is yet an other development environment: Open code, C++ based, with Python access layer. An electron cloud module for that toolkit is being discussed. For realistic high energy synchrotrons, such as the Fermilab Main Injector, a 3-dimensional, self-consistent plasma-like simulation of the EC and the beam is prohibitively expensive, CPU wise, for any HEP accelerators. However, for long bunches and for strong magnetic confinement (i.e., the magnetic field created by the beam current can be neglected), the physics implemented in POSINST is good enough. However, the current POSINST code has not been designed to run in parallel. The current version of POSINST has limited "test-units", as the critical algorithms are all bundled in a tight FORTRAN package. This makes the certification process more difficult for potential collaborators. A more object oriented approach will lead to a more robust package. Implementing parallelism is a must, as this simulation task is still CPU demanding.

[1] F. Zimmermann, Phys. Rev. ST Accel. Beams **7** (2004).

[2] M. T. Pivi and M. A. Furman, Phys. Rev. ST Accel. Beams **6** (2003).

[3] X. Zhang *et al.*, "Electron cloud studies at tevatron and main injector," `http://accelconf.web.cern.ch/AccelConf/p07/PAPERS/THPAN117.PDF` (2007).

[4] C. Nieter and J. R. Cary, J. Comp. Phys. **196** (2004).

[5] M. Furman, "Ecloud build-up simulations for the fnal mi for a mixed fill pattern: Dependence on peak sey and pulse intensity during the ramp," `http:mafurman.lbl.gov/LBNL-4215E.pdf` (2011).

[6] P. Lebrun *et al.*, "Simulation of the electron cloud in the fermilab main injector," `http://inspirebeta.net/record/913965` (2011).

[7] M. A. Furman and M. T. Pivi, Phys. Rev. ST Accel. Beams **5** (2002).

[8] M. A. Furman, Private communication, Phone meeting, April 7 2011 (2011).

[9] A. Singh, `http://tev.fnal.gov/index.shtml` (2011).

[10] W. T. V. William H. Press, Saul A. Teukolsky and B. P. Flannery, *Numerical Recipes in FORTRAN, the Art of Scientific Computing, second edition* (Cambridge University Press, 1992).

[11] `http://git-scm.com/`.

[12] P. Lebrun *et al.*, "Simulation of the electron cloud in the fermilab main injector using vorpal," `http://edms.classe.cornell.edu/agenda/getFile.py/access?contribId=10\&sessionId=6\&resId=1&materialId=slides\&confId=10` (2010).

[13] J. Amundson *et al.*, "Recent advances in the synergia accelerator simulation framework," `accelconf.web.cern.ch/accelconf/PAC2009/papers/th5pfp018.pdf` (2009).