Space Charge Simulation

Christopher R. Prior

CCLRC Rutherford Appleton Laboratory,
Chilton, Didcot, Oxfordshire, United Kingdom

Abstract. Based on a recent ICFA mini-workshop held in Oxford, England [1], this paper surveys the computer codes available for simulating the behaviour of charged particle beams under space charge. Modelling tools for both linear and circular accelerating systems are covered. Lists of recent comparisons code v. code and code v. experiment are given and a set of experimental results that might be used to benchmark codes is identified. The Oxford workshop also drew up a detailed spreadsheet of the features of most of the simulation codes in current use, and this is available at [2].

INTRODUCTION

The revolution in computing technology and the ready availability of fast parallel processors have opened up the path for increasingly detailed and realistic simulations of the behaviour of beams in particle accelerators. The number of macro-particles used for modelling is now typically of the order of $10^6$, compared with only a few thousand ten years ago. Simulations that once were prohibitively slow now routinely run overnight.

Prompted by the need for reliable modelling to complement design work for the coming breed of high intensity proton accelerators, several new codes have recently been developed. With so much invested in the LHC, Tevatron, SNS and J-Parc projects, and ideas for proton drivers for neutrino factories, radioactive ion beams and nuclear waste transmutation under study, it is vital that the underlying theoretical work is comprehensive, accurate and realistic. Such machines reveal new aspects of beam behaviour and include novel techniques in their design. The codes correspondingly need to be able to handle the underlying physics, and advances in recent years now enable such features as halo formation, injection modelling, phase-space painting, resonances, impedances and instabilities to be treated. Optimisation using tracking codes is also now a realistic proposition.

However, in this world of opportunity and promise, caution is advised. Are we sure that our codes are correct? Have we included all the basic physics and do our computer predictions faithfully reproduce what actually happens in a machine? In the case of design work, how can we be sure until the machine is built and running? Are we being consistent in our definitions? Are we correctly interpreting the numbers and pictures that our processors supply?

It was against this background that calls began to be made a few years ago for code comparisons and the development of benchmarking tests. Suggestions were put forward at the Snowmass meeting in Colorado in July 2001, and, following the recent ICFA Beam Dynamics mini-Workshop in Oxford, a renewed attempt is planned in which a larger community will be involved. These tests will be described below, following a survey of the main simulation codes that expect to take part.

SIMULATION CODES

Simulation codes have traditionally split into two types: those that treat linear systems where a single pass is made through each accelerating element, and circular systems, where repeated passage through periodic elements may permit different modelling techniques.

Linear Accelerating Systems

Linac codes, such as PARMILA, have been available for a number of years. Quite basic in form in their initial versions with simplified (usually linear) space charge routines, they have recently been subject to some updating, though the underlying structures remain the same. The main example of a new code that grasps the opportunities for a fast parallel approach offered by linear systems is IMPACT [3], developed by Ryne and co-workers, originally at LANL and now at LBL. This code typically uses $10^6$-$10^7$ simulation particles and runs on the US NERSC computing system. It includes fast map generation capabilities and uses split operator methods so that the Hamiltonian is divided $\mathcal{H} = \mathcal{H}_{\text{eo}} + \mathcal{H}_{\text{sc}}$, where
$\mathcal{H}_{ext}$ corresponds to the magnetic optics and $\mathcal{H}_{sc}$ to the space charge forces. The map for a time step $t$ is then (Figure 1)

$$\mathcal{M} = \mathcal{M}_{ext}(\frac{1}{2}t)\mathcal{M}_{sc}(t)\mathcal{M}_{ext}(\frac{1}{2}t) + O(t^2).$$

Space charge is calculated using 3D parallel Poisson solvers and several types of boundary condition are available. The design philosophy is not to take tiny steps to push $10^7$ particles, but to put the detailed calculation into computing the maps, which are then used to push the particles forward in time.

Systems comprising quadrupoles, dipoles, RF cavities and solenoids can be treated, and 3D constant focusing is included mainly for comparison with analytical predictions. The types of accelerating structures that can be modelled are DTL, CCDTL, CCL and superconducting cavities, and there is also a facility for user-defined elements. Gradient, misalignment and rotation errors can also be taken into account. The code has been used to model the SNS linac, CERN’s SPL, LEDA and the J-Parc linac. Studies of the front end of the linac for the European Spallation Source have also been carried out on a system of 16 parallel processors at RAL, where 300 runs with randomly generated errors were easily completed within 3 hrs.

Future developments for IMPACT envisage a new Poisson solver able to handle beams with a range of longitudinal:transverse aspect ratios, a model for high brightness electron bunches and an option for (simultaneous) multiple particle species.

Other codes written mainly for linacs include TRACE/WIN (CEA-Saclay) [4], which is a Windows development of PARMILA with improved space charge and graphics routines. This was used extensively in the CONCERT and ESS studies and works in harness with other codes such as PARTRAN, TRACE3D, TOUTATIS (for RF’s) and MONET.

### Circular Accelerating Systems

Given the increasing worldwide use of IMPACT, an extension to cover rings is natural and plans are in place to incorporate its 3D space charge capabilities and RF models into MARYLIE [5]. Based on Lie algebraic ideas using transfer maps, MARYLIE covers both linear beam transport systems and circular storage rings. The aim is for a code with nonlinear symplectic maps for beam line elements, symplectic and non-symplectic tracking, and facilities for optimisation and user-prescribed fitting.

The rings code that appears to be the most versatile is ORBIT [6], developed at ORNL, initially for theoretical studies of the SNS. Written in object oriented C++, this can now tackle a wide range of tasks including $H^{-}$ injection, foil heating, phase space painting, single particle transport through various types of magnets, effects of errors, closed orbit calculations and corrections, longitudinal and transverse impedances, collimation and feedback. Beam transport uses MAD/DIMAD matrices, the Fermilab MXYZPTLK library of differential algebra maps and symplectic “Teapot” style maps. RF cavities are modelled with longitudinal kicks and there is a facility for user-specified harmonics. However plans to develop the code in order to simulate complete cycles of synchrotrons have not yet been implemented.

One aim is to incorporate an electron cloud model to handle the possible e-p instability predicted for many high intensity proton machines under either construction or study. Since existing electron cloud codes tend not to use full lattices in their modelling, it would be an important development if predictions from a full particle tracking code could be achieved.

ORBIT appears to have spawned various offspring. The code has been adopted by Fermilab, where it is being used for Booster studies. Resources have been allocated for in-house development, resulting in incorporation of a Python shell along with other improvements for maintainability and usability. Similarly at Brookhaven, changes to ORBIT feature an interactive Poisson solver for space charge with a conducting or resistive wall boundary condition. One of the more interesting features of this solver is the projection of the beam localized in distance along the closed orbit to a time localized beam distributed over the lattice to perform the space charge calculation. To obtain new maps for acceleration, the parallel BNL-ORBIT can run MAD on a single processor and update the lattice. The space charge solver uses a 3D model, where the particles and longitudinal 2D grid slices are allocated to the processors using a generic
algorithm to optimize performance. In application to a 1 MW upgrade of AGS, space charge forces in the longitudinal bunch structure and dynamics have been found to influence the transverse beam dynamics.

At the PSR at Los Alamos, ORBIT has been used in conjunction with experiments on the ring. Good agreement has been found between simulation results and beam profile measurements. In particular, the observation that the beam profile does not depend on the specific painting technique above $3 \times 10^{13}$ protons is understood, and modelling results, showing intensity limitation as the tune is depressed by space charge and the vertical envelope tune approaches the integer 4, have led to a study to produce a compensation scheme.

The Fermilab version of ORBIT has been applied in an attempt to improve Booster performance by comparing simulation with experiment [7]. The motivation for studying the Booster Ring is to obtain the higher intensities necessary for MiniBooNE, NuMI and Tevatron Run 2. Although the linac and main injector can each transmit pulses of $3 \times 10^{13}$ particles, the Booster creates a bottleneck because its losses become unacceptable ($>1$ W/m) at intensities above $5 \times 10^{12}$ particles per pulse. The space charge experiment, designed to concentrate on the first 3 ms where losses are about 30%, involved the injection of 11 turns of beam with the RF cavities turned off. During the process, space charge tune shifts of approximately $\sim 0.3$ were reached. Transverse emittances were measured as functions of time and compared with the results from simulations. FNAL-ORBIT modelling used a 2.5D space charge routine with 10000 macroparticles per injection turn (110000 macroparticles total). Tracking was carried out for 2000 turns. Simultaneously, modelling was carried out with a locally written code, SYNERGIA, using a similar $64 \times 64 \times 32$ mesh. The results of the ORBIT calculations showed rapid emittance growth during the 11 turns of injection, followed by a slower growth. The SYNERGIA calculations showed much slower initial growth, followed by a gradual emittance growth to final values not significantly different from those obtained using ORBIT. The emittance measurements by a fluorescent technique were indeterminate, and need refinement. Clearly, at the time of writing, further work is needed on all fronts to resolve the discrepancies.

A rings code of fairly long standing is ACCSIM [8], developed by F. Jones at TRIUMF. This exploits the pre-existing codes MAD and DIMAD for lattice preparation and preliminary calculations. Longitudinal space charge forces are calculated from the bunch line density and transverse space charge kicks are calculated from the electric field of the entire ensemble with the local line density as a scaling factor for the force on each particle. The code is therefore effectively 2.5D, though there are plans for a phased upgrade to 3D. The wide range of ACCSIM’s utilities is summarised in Figure 2 and a flow chart explaining its integration with MAD/DIMAD is shown in Figure 3.

**FIGURE 2. ACCSIM Capabilities**

ACCSIM has been used to model rings such as CERN’s PS Booster, the Hitachi medical synchrotron, KEK-PS and the J-Parc 3 GeV ring, generally providing good results (in terms of predictions v. measurements) for rms matching, beam profiles, injection losses and coherent resonance losses. Future study will cover particle redistribution in rms matched beams, space charge resonances, halo formation and synchro-betatron effects from space charge and chromaticity.

**FIGURE 3. ACCSIM Data Management**

Although in principle very simple, longitudinal (1D) tracking codes can often prove very useful for initial system design. Into this category fall ESME (FNAL), LONG1D (TRIUMF) and Track1D (RAL). From knowledge of only a few global properties of the ring, these codes model bunches by sequentially updating values of phase and energy, with space charge calculated from the longitudinal line density. In recent years, phase space painting, RF manipulations, feedback loops, time-dependent external parameter variation and interactive graphics have been added. ESME has been used to model many machines, including the Fermilab Booster and
LANL-AHF; LONG1D has been used on ISIS, as a ba-
sis for the original longitudinal elements of ACCSIM
and in the development of tomography codes at CERN;
Track1D was at the heart of the ESS injection design, for
improvements to the ISIS cycle and as part of the mod-
elling carried out for the present ISIS 240 kW upgrade.

Also at RAL, Track2D [9] is a transverse tracking
code for either linear or circular machines with several
Poison solvers based on finite element techniques ap-
proach allows irregular shaped boundaries to be treated,
both perfectly conducting and lossy. The mesh “breathe”
with the beam to ensure good resolution and to handle
rapid field changes as particles near the wall. A triangu-
lar serendipity scheme is used (up to order 6) but a simple
adaptation to a rectangular grid converts the space charge
solver to r-z for use in an axisymmetric 2.5D code. The
transverse code was one of the first to model multturn in-
jection under non-linear space charge (in connection with
inertia confinement fusion studies) and has been used to
model ISIS, CERN-PS and PSB, ESS and the Fermilab
proton drivers in Studies I and II.

Extending the triangular mesh to one based on tetra-
hedra gives a 3D Poisson solver and is built into an-
other RAL code, Track3D. Many techniques developed
over the years for data management, efficient storage and
computational speed are incorporated into these codes.
Track3D is currently undergoing major re-development
to convert to a fast parallel processing system. Addi-
tional features of Track1D, Track2D and Track3D in-
clude the treatment of machine errors, variable charge
macro-particles and the ability to track simultaneously
particles of more than one species.

Another important code with similar facilities is
KEK’s SIMPSONS. This has both 2D and 3D options.
Although it deals only with circular perfectly conducting
boundaries and is applicable to long bunches only, it is
able to model acceleration, multturn injection, painting,
apertures, collimation and errors; it also has a variable
charge facility. Recently it has been tested at Femilab and
compared with FNAL-ORBIT.

Other Codes

An interesting new code is GPT (General Particle
Tracer) [10] developed by Pulsar Physics, partly under
contract from TESLA. Working in the time domain, this
package is very efficient, with cpu scaling as $N^{1.1}$ ($N =$
number of simulation particles) and is able to handle the
difficult problem of space charge calculations for bunch
aspect ratios from 0.01 to 100. A large effort has been
put into developing a suitable methodology for 3D space
charge routines, based on multgrid and pre-conditioned
conjugate gradient techniques. Integration is via a 5th
order Runge Kutta technique and there are interfaces
to other codes such as Superfish. Recent applications
of GPT include the design of electron beam optics and
radiation yield simulations for the FZR Free Electron
Laser, a study of a high-brightness photo-injector, and an
energy recovery system for a 12 A, 50-375 keV electron
beam. GPT can also be used for designing and optimising
collimation systems.

Another code, GenTrackE, is under development by
Adelmann (LBL/PSI). Aiming to model large scale ma-
chines with complicated 3D geometries, this code has
been making full use of the opportunities offered by the
US NERSC Seaborg parallel system, showing good per-
formance and results. There are plans to incorporate a
full model for electron cloud studies.

At Princeton, a group has been working on a code
called BEST (Beam Equilibrium Stability and Trans-
port) [11], which models both linacs and rings in 3D
using nonlinear Vlasov-Maxwell equations and the $\delta f$
method. The advantage of this approach is that simulati-
on noise is considerably reduced. The code’s main use
has been for two-stream instability and beam echo stud-
ies, and progress has been made in understanding the
beam loss, which is possibly electron cloud related, seen
in the LANL-PSR.

Of longer standing is the WARP suite of codes [12]
developed at Livermore National Laboratory specifically
for the study of space charge dominated ion beams with
applications to fusion driver concepts. The package con-
tains 2D ($x \gamma$) and ($r_\rho z$) and 3D options and uses
electrostatic particle-in-cell techniques. There is a wide
range of possibilities for specifying the lattice of exter-
nal fields, including bent beam pipes, which are treated
using the “warped” coordinate system from which is de-

drived the code’s name. In a parallelised form, it has been
used for high resolution simulations of the LBL elec-
tric field accelerator injector and multi-lap simulations
of emittance growth in a small recirculator experiment.

In a separate category fall codes such as VADOR [13],
based on Vlasov techniques for studying the evolution of
particle density in phase space. The approach has some
advantages, such as effective use in regions of low phase
space density. However, while their provision as a tool
for cross-checking PIC co it is clear that extension to
cover even 2D (transverse) space with reasonable cpu
times remains a development for the future.

CODE COMPARISON

This list of codes, though far from comprehensive, covers
the main space charge accelerator modelling tools iden-
tified in the Oxford workshop [1]. Questions then arise
to their accuracy and their uses in modelling real ma-
### TABLE 1. Code v. Code Comparison

<table>
<thead>
<tr>
<th>Codes</th>
<th>Test</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCSIM, ORBIT, SIMPSONS</td>
<td>KV beam, rms emittance, PSR model</td>
<td>Good</td>
</tr>
<tr>
<td>ORBIT, ESME</td>
<td>1D longitudinal</td>
<td>Good</td>
</tr>
<tr>
<td>ORBIT, SYNERGIA</td>
<td>FNAL booster, multiturn injection, emittance blow-up</td>
<td>Discrepancy</td>
</tr>
<tr>
<td>Track1D, LONG1D, ACCSIM</td>
<td>1D long., ISIS, SNS, ESS models</td>
<td>Good</td>
</tr>
<tr>
<td>Track2D, SIMPSONS, ORBIT</td>
<td>SNS ring modelling; FNAL proton driver injection</td>
<td>Good</td>
</tr>
<tr>
<td>Micromap, IMPACT</td>
<td>Octupole resonance with space charge</td>
<td>Good</td>
</tr>
</tbody>
</table>

### TABLE 2. Code v. Experiment Comparison

<table>
<thead>
<tr>
<th>Code v. Machine</th>
<th>Measurement</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORBIT v. PSR</td>
<td>Profile measurements</td>
<td>Good</td>
</tr>
<tr>
<td>ESME v. FNAL</td>
<td>1D longitudinal</td>
<td>Good/fair</td>
</tr>
<tr>
<td>ESME v. CERN-PS</td>
<td>1D longitudinal</td>
<td>Good/fair</td>
</tr>
<tr>
<td>Micromap v. CERN-PS</td>
<td>Montague resonance</td>
<td>In progress</td>
</tr>
<tr>
<td>ACCSIM v. CERN-PSB</td>
<td>1D profile</td>
<td>Fair</td>
</tr>
<tr>
<td>ACCSIM v. KEK-PS</td>
<td>1D profile</td>
<td>Good</td>
</tr>
<tr>
<td>IMPACT v. LANL-LEDA</td>
<td>Halo studies</td>
<td>Matched agreement; mismatched discrepancy</td>
</tr>
<tr>
<td>ORBIT v. FNAL-Booster</td>
<td>Emittance growth from multiturn injection</td>
<td>Inconclusive</td>
</tr>
<tr>
<td>GPT v. Felix</td>
<td>Emittance, radiation, profiles</td>
<td>Good</td>
</tr>
<tr>
<td>BEST v. PSR</td>
<td>Electron cloud effect</td>
<td>Fair</td>
</tr>
<tr>
<td>Track1D v ISIS</td>
<td>Long. profiles, beam loss, injection studies</td>
<td>Good</td>
</tr>
<tr>
<td>Track2D v. CERN-PSB</td>
<td>Instability, emittance effect</td>
<td>Good</td>
</tr>
</tbody>
</table>

CHINES. However, as a prelude to a formal benchmarking programme, it is useful to identify first those comparisons that have been made to date. Table 1 shows instances of simulations where codes have been compared with each other and Table 2 provides examples where codes have been tested against results from experiments.

Apart from the FNAL-ORBIT/Synergia discrepancy mentioned above, there is reasonably good agreement in code v. code comparisons using the same piece of analytical data. However, the agreement is more variable when codes are tested against experiments and this is to be expected as it is almost impossible, without taking very special measures, to cover every aspect of particular machines.

### BENCHMARKING

Two benchmarking tests have been specifically identified based on experiments recently carried out at CERN. The aim of the first exercise was to measure transverse emittance increase due to space charge in the PS as a function of such parameters as tune, bunching factor and bunch intensity. The variations considered were sufficient to cause the beam to cross integer and half-integer resonances under space charge tune depression. Emittance increases up to a factor 3 on a time scale of 10-100 ms were recorded. The results, including all the necessary machine parameters, are published at [14] with the intention that interested parties should attempt to simulate the experiments as a means of testing their modelling codes.

A second benchmarking exercise involves a comparison of simulations and measurements on the CERN PS of emittance exchange in crossing the Montague resonance $2Q_h - 2Q_v = 0$. In tests using the Micromap code (GSI) [15], agreement has been achieved on the level of emittance exchange but not on the width of the stop-band, which experimentally is much wider than predicted. Though theoretical work is needed to resolve the discrepancy, this study also provides scope for benchmarking and details are also included at [14].

It is expected that the following codes will take part (names of participants in brackets):

- Micromap (I. Hofmann, G. Franchetti)
- FNAL-ORBIT (W. Chou, J-F. Ostiguy, P. Lucas)
- BEST (H. Qin)
- ACCSIM (F.W. Jones)
- BNL-ORBIT (A. Luccio)
- ORBIT (J. Holmes, S. Cousineau)
- GenTrackE (A. Adelmann)
- IMPACT (R. Ryne, J. Qiang)
- SIMPSONS (D. Johnson, F. Neri)

Based on work using BEST, Hong Qin has also suggested a number of theoretical predictions for bench-
marking tests: 1D thermal equilibrium beam profiles, stable beam propagation, and eigenmodes in a space charge dominated beam. Modelling the two-stream instability and beam echo are other possibilities.

Plans for an experimental study of space charge using the ISIS synchrotron at RAL are also under way and the results could provide further material for code comparison. With a tune shift of about -0.4, several resonant lines are crossed during ISIS injection, including the half-integer line (3.5) in the vertical plane and the integer line (4.0) in the horizontal plane, and space charge plays a significant role in the total beam loss. Measurements have been made using a residual gas beam profile monitor and it is expected that parallel simulations with Track2D and Track3D will start soon.

CODES SPREADSHEET

The Space Charge Simulation workshop [1] concluded by drawing up a spreadsheet containing information about the main codes described above. Available at [2], each code is categorised under language, platform, GUI, whether parallelised or not, the type of transport systems that can be modelled and whether in 1D, 2D or 3D. Details of space charge solvers and their associated boundaries are included, along with the nature of the tracking and whether impedances and field maps are options. Nonlinear modelling relies heavily on graphical output and, particularly if included as an integral part of the package, this can often have a bearing on a code’s portability. Such aspects are included in the spreadsheet, in addition to availability of user manuals and standard test cases. The final columns cover special features, limitations and, most importantly, contact details of each code’s owner.

REFERENCES

2. http://www.isis.rl.ac.uk/ AcceleratorTheory/