

RECENT PROGRESS ON THE MARYLIE/IMPACT BEAM DYNAMICS CODE *

R. D. Ryne[†] J. Qiang, E. W. Bethel, I. Pogorelov, J. Shalf, C. Siegerist, M. Venturini, LBNL, Berkeley, CA, USA
A. J. Dragt, U. Maryland, College Park, MD, USA
A. Adelmann, PSI, Villigen, Switzerland
D. Abell, Tech-X Corporation, Boulder, CO, USA
J. Amundson, P. Spentzouris, FNAL, Batavia, IL, USA
F. Neri, P. Walstrom, LANL, Los Alamos, NM, USA
C. T. Mottershead, Lincoln City, OR, USA
R. Samulyak, BNL, Upton, NY, USA

Abstract

MARYLIE/IMPACT (ML/I) is a hybrid code that combines the beam optics capabilities of MARYLIE with the parallel Particle-In-Cell capabilities of IMPACT. In addition to combining the capabilities of these codes, ML/I has a number of powerful features, including a choice of Poisson solvers, a fifth-order rf cavity model, multiple reference particles for rf cavities, a library of soft-edge magnet models, representation of magnet systems in terms of coil stacks with possibly overlapping fields, and wakefield effects. The code allows for map production, map analysis, particle tracking, and 3D envelope tracking, all within a single, coherent user environment. ML/I has a front end that can read both MARYLIE input and MAD lattice descriptions. The code can model beams with or without acceleration, and with or without space charge. Developed under a US DOE Scientific Discovery through Advanced Computing (SciDAC) project, ML/I is well suited to large-scale modeling, simulations having been performed with up to 100M macroparticles. The code inherits the powerful fitting and optimizing capabilities of MARYLIE augmented for the new features of ML/I. The combination of soft-edge magnet models, high-order capability, space charge effects, and fitting/optimization capabilities, make ML/I a powerful code for a wide range of beam optics design problems. This paper provides a description of the code and its unique capabilities.

CODE DESCRIPTION

MARYLIE/IMPACT (ML/I) is a hybrid code that combines the beam optics capabilities of MARYLIE [1] with the parallel Particle-In-Cell (PIC) capabilities of IMPACT [2]. As such, it can be used to model beam dynamics in a wide range of rf accelerators, including linear and circular accelerators, with or without space charge, and including acceleration.

* Work supported in part by the US Department of Energy, Office of Science, under the Scientific Discovery through Advanced Computing program.

[†] rdyne@lbl.gov

ML/I was developed by taking certain routines (e.g. space-charge routines and rf cavity routines) from the IMPACT code and incorporating them in MARYLIE. MARYLIE itself was modified to automatically slice elements and, using split-operator methods, MARYLIE was then given the ability to perform space-charge kicks in between slices. This enables ML/I to track particles using a split-operator integration algorithm under the assumption that the single-particle Hamiltonian is given by $H = H_{ext} + H_{sc}$, where H_{ext} is the portion of the Hamiltonian corresponding to external fields, and where H_{sc} is the portion corresponding to space-charge fields (in the mean-field approximation).

The use of ML/I is backward compatible with MARYLIE. For example, all the MARYLIE beamline elements, commands, map analysis capabilities, and fitting/optimizing capabilities, are present in ML/I. However, the ML/I front end has the ability to read lattice descriptions in both the MARYLIE style and the Standard Input Format (SIF). This is an important new capability, given the very large number of files written in the SIF style. In some cases the input format involves an augmentation of SIF, since certain ML/I capabilities—notably the ability to treat space charge and the ability to automatically perform certain commands—requires additional parameters for some definitions. For example, every thick element can have an additional parameter denoting the number of slices into which the element should be cut.

SPECIFICATION OF UNITS IN ML/I

The issue of units has no doubt cost many accelerator designers hours of frustration and lost productivity. ML/I attempts to address this by providing a flexible and systematic treatment of units.

Essentially all particle-based beam-dynamics codes deal with six-vectors of coordinates and momenta (or coordinates and velocities, in codes that do not use canonical variables). Let (x, p_x, y, p_y, t, p_t) denote such a six-vector in *physical* units. In other words, x and y have dimensions of length; t (deviation in arrival time) has the dimensions of time; p_x and p_y have dimensions of momenta; and

p_t has dimensions of energy. Most beam dynamics codes (ML/I, MARYLIE, MAD8, MAD9, *etc.*) use dimensionless variables. We shall define these in terms of three scaling constants, l , δ , and ω , having, respectively, dimensions of length, momentum, and inverse time. The dimensionless variables are $(x/l, p_x/\delta, y/l, p_y/\delta, \omega t, p_t/(\omega l \delta))$. Note that each of the products $x p_x$, $y p_y$, and $t p_t$ has the same scale factor, $l \delta$, relating the new product to the old. As a result, the transformation between the old and new variables is canonical.

The choice of the three scaling quantities l , δ , and ω determines the choice of dimensionless variables (*i.e.* the choice of units). In MARYLIE, for example, l is chosen by the user, δ is set equal to p_0 (the reference momentum), and ω is automatically selected so that $\omega l/c = 1$. This is a natural choice for beam transport systems which have no acceleration; we call these *magnetostatic* or *static* units. When acceleration is present, p_0 changes, and hence a different (constant) scale factor is needed for momentum. One very natural choice is $\delta = mc$; we call this choice *dynamic* units. Note that IMPACT uses dynamic units with the added restriction that $\omega l/c = 1$ (where ω is specified by the user).

ML/I allows all three quantities l , δ , and ω to be set by the user. This is accomplished via the `units` command. Valid examples include

```
myunits: units, type=static
myunits: units, type=dynamic
myunits: units, l=5.0, p=2.5, w=3.6
```

The default scale length for ML/I is $l = 1$ m; the default scale angular frequency is $\omega = c/l = 299792458$ rad/sec; and the default units for ML/I are magnetic units. These values are the same as for MARYLIE, except that in MARYLIE the user must specify the scale length.

SUMMARY OF NEW CODE FEATURES

We have already mentioned four key features in the ML/I code:

- ability to model beams with space charge;
- ability to model accelerating beams;
- ability to read lattices in the Standard Input Format;
- ability to perform very large-scale simulations on parallel computers.

Other key features include:

- fifth order rf cavity model: The original version of ML/I treated rf cavities in the linear approximation. The ability to model rf cavities through 5th order has now been developed and incorporated [3]. This capability was developed by D. Abell (Tech-X Corporation).

- soft-edge magnets: ML/I has a comprehensive set of soft-edge magnet models. In addition, the code has the ability to treat magnets as coil stacks, which allows for the treatment of overlapping fringe fields. These capabilities were developed by P. Walstrom and F. Neri (LANL), respectively.
- high aspect ratio solvers: Poisson solvers based on an Integrated Green Function (IGF) approach provide an efficient means for accurately modeling space-charge effects in beams with high aspect ratio. ML/I currently contains optional 2D IGF solvers developed by R. Ryne (LBNL). A 3D capability developed by J. Qiang currently exists in IMPACT [4], and additional 3D capabilities are under development.
- wakefield effects: ML/I has the ability to treat wakefield effects using several models. The wakefield capability was developed by R. Samulyak (BNL).
- 3D rms envelope calculations: ML/I has the ability to track rms envelopes and to find matched envelopes in periodic transport systems [5]. This capability was developed by R. Ryne (LBNL).

Lastly, the synthesis of a beam optics code (MARYLIE) with a particle-in-cell code (IMPACT) is enhanced by the addition of certain functionality that is useful for tracking and other purposes. This functionality is embodied in the following “automatic” commands:

- Automatic slicing: thick elements now have an additional parameter, `slices`, for which the default value is one. This can have a number of uses. One important use is related to the inclusion of space-charge. Namely, there can be a space-charge kick in the middle of every slice. Other uses involve autoslicing combined with automatic application of commands. This may be used, for example, to print lattice functions at points within thick elements.
- Automatic application of a command or string of commands: Using the command `autoapply`, the user can specify a menu element or the name of a line (a sequence of commands and/or elements) that is to be automatically applied before and after every slice. For example, if a user wanted to tabulate the rms moments as a function of distance along the beamline, a user could slice the elements and then automatically apply the commands to compute and print the moments.

Two more automatic commands are available to the user:

- Automatic tracking: Normally, ML/I concatenates maps whenever a beamline element is encountered. This behavior can be changed by using the `autotrack` command, which causes ML/I to track particles, with or without space charge, whenever an element is encountered.

- Automatic concatenation: The `autoconcat` command causes ML/I to switch from `autotrack` mode back to the default mode of operation, where beam-line elements are automatically concatenated. There is also an option for `autoconcat` that causes the code to “sandwich” while concatenating, i.e. whenever a map is computed it is used to perform a similarity transformation on the total transfer map; this option is useful for computing lattice functions.

Finally, there are certain ML/I commands that are similar to MARYLIE commands but provide added functionality. For example, the ML/I command `raytrace` is similar to the MARYLIE command `rt`, but allows the user to specify file names, control the precision of output, control which subset of particles is printed, *etc.*

Tracking in RF cavities with multiple reference trajectories

As mentioned previously, ML/I contains a new capability for computing fifth order transfer maps for rf cavities [3]. Combined with its previous functionality, this gives ML/I full fifth order capability including s-dependent magnet fringe fields and realistic accelerating fields.

Another new feature related to rf cavities is the ability to track particles through cavities using multiple reference trajectories. This allows one, for example, to model an initially DC beam as it undergoes bunching. The new capability works as follows: when an rf cavity is encountered while the code is in “autotracking” mode, the code determines a bounding box in longitudinal phase space that contains all the particles being tracked. The phase space in this box is then sampled with a regular grid of reference particles, and the rf cavity map around each reference particle is computed. Next, the reference trajectory whose initial value is closest to each particle at the entrance to the rf cavity is determined, and the corresponding map is used to track the particle. After this has been done for all particles in the beam bunch, the final coordinates are then recomputed around the original reference trajectory. This approach is useful but has limitations. First, it only samples the longitudinal phase space for determining reference trajectories; thus it can handle extreme longitudinal nonlinearities, but the transverse nonlinearities are the same as those of the map-based representation (i.e. first order using the old ML/I capability, fifth order using the new capability). Second, there may be a lack of smoothness between particle trajectories that are nearby in phase space but are associated with different reference particles.

Figure 1 shows an example of this capability. The red line denotes an initial beam of zero energy spread and 360 degree phase spread as it enters an rf cavity. The green curve (straight line) shows the final conditions of the particles after passing through the cavity using a single reference particle and using the linear map for an rf gap. The blue (sin-like) curve shows the result of using a 20x100 grid and the linear map for an rf gap. As is clear from the

figure, using multiple reference particles with linear maps captured the nonlinear behavior of the rf cavity.

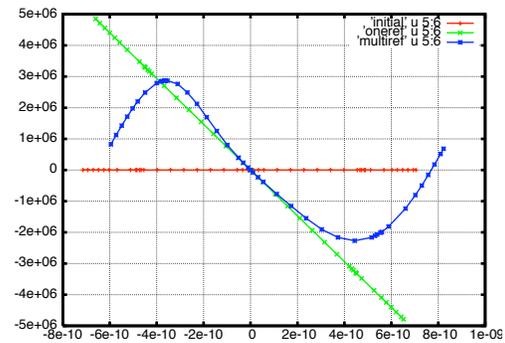


Figure 1: Longitudinal phase space (horizontal axis: arrival time; vertical axis: energy deviation) showing the result of tracking an initial set of particles (red points) using a single reference particle (green line) and multiple reference particles (blue curve).

Future Plans

Under the SciDAC Accelerator Science and Technology project, standards have been developed for reading, writing, and providing information in headers of particle data files, and these standards have been incorporated into ML/IWork is underway to adopt standards for parallel I/O (see, for example, reference [6]). Work is also underway to develop standards for transfer map I/O. Other planned enhancements to ML/I include an updated version of the wakefield module; an enhanced 3D Integrated Green Function package with linear basis functions; improved modeling of wiggler magnets; general treatment of multiple reference particles (motivated by the need to model long bunches with space charge), and inclusion of time-dependent beam-cavity interactions in rf cavities.

REFERENCES

- [1] A. Dragt et al., MARYLIE 3.0 User’s Manual, Department of Physics and Astronomy, University of Maryland, College Park, MD 20742, USA.
- [2] J. Qiang, R. Ryne, S. Habib, and V. Decyk, *J. Comp. Phys.* 163, 434 (2000).
- [3] D. Abell, *Phys. Rev. ST Accel. Beams* 9, 052001 (2006).
- [4] J. Qiang, S. Lidia, R. D. Ryne, C. Limborg-Deprey, *Phys. Rev. ST Accel. Beams* 9, 044204 (2006).
- [5] R. Ryne, “Finding Matched Envelopes in RF Linacs: A Hamiltonian Approach,” arxiv.org/abs/acc-phys/9502001 (1995).
- [6] A. Adelmann et al., “H5Part: A Portable High Performance Parallel Data Interface for Electromagnetics Simulations,” proceedings of this conference. Also see <http://amas.web.psi.ch/docs/H5Part-doc/h5part.html>