IMPACT-T User Document Version 2.2

Ji Qiang
Lawrence Berkeley National Laboratory

April 27, 2022

Contents

1 Introduction

IMPACT-T is a fully three-dimensional program to track relativistic particles taking into account space charge forces, short-range longitudinal and transverse wakefields and coherent synchrotron radiation (CSR) wakefield. IMPACT-T is one of the few codes used in the photoinjector community that has a parallel implementation, making it very useful for high statistics simulations of beam halos and beam diagnostics. It has a comprehensive set of beamline elements, and furthermore allows arbitrary overlap of their fields, which gives the IMPACT-T a capability to model both standing wave structures and traveling wave structures. It includes mean-field space-charge solvers based on an integrated Green function to efficiently and accurately treat beams with large aspect ratio, and a shifted Green function to efficiently treat image charge effects of a cathode [1]. It is also unique in its inclusion of energy binning in the space-charge calculation to model beams with large energy spread. It also has a direct N-body solver to calculate stochastic space-charge forces. IMPACT-T has a flexible data structure that allows particles to be stored in containers with common characteristics; for photoinjector simulations the containers represent multiple slices, but in other applications they could correspond, e.g., to particles of different species. Together, all these features make IMPACT-T a powerful and versatile tool for modeling beams in photoinjectors and other systems.

2 Recently Added Features

- three new distribution types.

- global random seed number (Dim) for multiple simulations with different random numbers.

- a beam rotation with respect to longitudinal z axis (-17).

- a beam longitudinal heating (-16).

- a dielectric wakefield model from Dianel Mihalcea (-13).

- 3D field data on Cartesian grid (111 EMfldCart) has to be in the format of **complex number** to represent both traveling wave and standing wave fields.

- an alpha magnet field model, traveling wave in meander plates model, and DC surface roughness field model in (113) EMfldAna.

- new initial particle distribution ijk, which contains a total of 24 types of initial distribution behind the cathode.

- Python script code, PhaseOpt.py, automatically finds the RF initial driven phases of cavities used in the input file, ImpactT.in, based on the user specified design phases in the ImpactT.in.

- switch (-12) for applying instant linear matrix kick to a beam at a given location.

- modified switch (-1) for steering so that steering means an instant kick.

- switch (-15) for mean-field grid based or point-to-point N-body space charge solver.

- quadrupole element: If v9 (rotation angle with respect to z) is nonzero, this is a skew quadrupole. IF v10 (rf frequency) and v11 (phase) are nonzero, this is an rf quadrupole.

- collimation function at given z location by using "-11".

- output slice-based information (current,uncorrelated energy spread, slice emittances, correlated energy spread) at given z location using "-9". The slice information for the initial distribution and the final output distribution is stored in file fort.60 and fort.70.

- modified the structure wakefield calculation so that the code can use both analytical expressions and read-in transverse and longitudinal wake function from external files.

- particle coordinates for initial read-in distribution are x(m), Px/mc, y(m), Py/mc, z(m), Pz/mc.

- particle coordinates for output phase distribution at given location are x(m), Px/mc, y(m), Py/mc, z(m), Pz/mc.

- switch flag for cathode (Nemission > 0, flagcathode = 1, cathode exists, otherwise, 0 no cathode model).

- 1D CSR wake module including the transient effects at the entrance and the exit of dipole bend magnet with integrated Green method.

3   Five Steps to Run the IMPACT-T Code

- Prepare input files (rfdatax or 1Tx.T7) for RF cavities, solenoids and bending magnets.

- Set up the IMPACT-T input file, ImpactT.in.

- Set up the RF driven phases inside the ImpactT.in with the design phase using a Python script code, PhaseOpt.py to run an on-axis single particle simulation of the IMPACT-T code.

- Run the multi-particle IMPACT-T tracking code.

- Check the simulation results.

4   Pre Processing

There are three preprocessing programs. The program **RFcoefext** takes discrete field data as provided on axis in the file rfdata.in, extends these data by symmetry to make a new set of data describing the fields on an extended domain, and generates the Fourier coefficients for these extended fields based on a Fourier series expansion. These are stored in the file rfdatax for IMPACT-T simulation use. (Here, "x" in rfdatax has to be replaced by the corresponding "number" in the ImpactT.in input line for that element.) This file is used when only half of the field in the injector is given. It contains both the real part and the imaginary part of the complex Fourier coefficients. *Note, the user should use another output file rfdata.out to check whether the input Fourier mode number is sufficient or not by comparing the plot of col. 2 vs. col. 1 with those from rfdata.in. Basically, rfdata.out contains the reconstructed and extended RF field and its derivatives using the Fourier coefficients.*

The program **RFcoeflcls** takes the discrete field data, generates the Fourier coefficients of these data directly and stores them in rfdataxxx. It also generates a shifted field data in rfdata.tmp based on these coefficients. This file will be used to regenerate Fourier coefficients for the shifted field data, which will be used to model a traveling wave field using the summation of two standing wave fields (one field is the shift of the other field). The file rfdataxx will be used in beam line elements such as SolRF. The output file rfdata.out is the field reconstructed using those Fourier coefficients. This will be used as a good check of the accuracy of the Fourier expansion approximation.

A Python scripting program, **ImpactTphase.py**, scans through the initial driven phase of a single RF cavity in the ImpactT.in file. It can also be used to scan through other parameters. The user needs to go into the code to specify the exact location of the parameter inside the file. Some other small changes (e.g. name of executable file) might also be needed. For the phase scan, the output of this scripting file is Engout with column one the driven phase, column five the final energy.

Another python scripting program, **PhaseOpt.py**, can be used to automatically find the driven phase for all RF cavities in the ImpactT.in file. Here, the user needs to specify the design phase of each cavity in the ImpactT.in file. Minor modification of the program is needed to specify the line range of the lattice inside the ImpactT.in file. The single on-axis particle can be specified using the waterbag distribution with 0 rms sizes and transverse offsets except that the longitudinal offset corresponds to $-v_0 t_{laser}/2$, where v_0 can be found from the initial beam kinetic energy "Bkenergy", t_{laser} is the total laser pulse length (in seconds).

There is a F90 program, **chicaneImpt.f90**, which will help generate four input files that will be used in the Impact-T simulation of a chicane. The user is encouraged to read this small program and to set the inputs appropriately.

5 Input Parameters Excluding Lattice

The following gives a line by line description of the input parameters appearing before the lattice layout used in the input file **ImpactT.in**. Note, *the comment line starting with* **!** *is not included in the line number.*

Line 1: Npcol, Nprow

> **Npcol** Number of columns of processors, used to decompose domain along Y dimension.
>
> **Nprow** Number of rows of processors, used to decompose domain along Z dimension.

Line 2: Dt, Ntstep, Nbunch

> **Dt** Time step size (secs).
>
> **Ntstep** Maximum number of time steps. IMPACT-T will stop after Ntstep time steps or when the center of the bunch goes beyond the end of the lattice, whichever is first.
>
> **Nbunch** During simulation of the rf photoinjector, when the laser pulse length is long and the rf acceleration gradient is high, the electron beam out of the cathode can have a significantly large energy spread. In this case, performing a single Lorentz transformation from the beam frame to the lab frame during the calculation of the space-charge forces is not sufficient. To model space-charge forces more accurately, the initial distribution of the bunch can be divided longitudinally into **Nbunch** slices. The Lorentz transformation can be done for each slice/bunch and the total space-charge forces are summed. More slices mean better accuracy, but at the cost of increased computation time. The computation time increases approximately linearly with Nbunch. When **Nbunch** > 1, more input files named **ImpactT2.in**, **ImpactT3.in**, ...**ImpactTNbunch.in** have to be provided. These input files can have independent definitions of the particle information. The total current is distributed among the different bunch files. However, the external lattice layout has to be the same. If each input file uses a read-in initial particle data file, **partcl2.in**, **partcl3.in**, ...**partclNbunch.in** have to be provided. This will be replaced as **partcl2.data**, **partcl3.data**, ...**partclNbunch.data** soon. Since each bin/bunch has independent parameters, the IMPACT-T code can be used to simulate a beam with multiple species.

Line 3: Dim, Np, Flagmap, Flagerr, Flagdiag, Flagimg, Zimage

Dim Random seed integer > 0.

Np Number of macroparticles to track.

Flagmap Type of integrator. Currently must be set to 1.

Flagerr Error study flag. 0 - no misalignment and rotation errors; 1 - misalignment and rotation errors are allowed for Quadrupole, Multipole (Sextupole, Octupole, Decapole) and SolRF elements. This function can also be used to simulate the beam transport through rotated beam line elements such as skew quadrupole etc.

Flagdiag Diagnostics flag: 1 - output the information at given time, 2 - output the information at the location of bunch centroid by drifting the particles to that location, 3 or more - no output.

Flagimg Image charge flag. If set to 1 then the image charge forces due to the cathode are included. The cathode is always assumed to be at $z = 0$. To not include the image charge forces set imchgF to 0.

Zimage z position beyond which image charge forces are neglected. Set z small to speed up the calculation but large enough so that the results are not affected.

Line 4: Nx, Ny, Nz, Flagbc, Xrad, Yrad, Perdlen

Nx Ny Nz Number of mesh points in x, y, and z. For the open boundary condition, currently, they have to be **a power of 2** due to the FFT algorithm used here. In the future, these will be generalized to arbitrary integer numbers.

Flagbc Field boundary condition flag: Currently must be set to 1 which corresponds to an open boundary condition.

Xrad, Yrad, Perdlen Size of computational domain. Xrad and Yrad define the transverse size, Perdlen defines the longitudinal size. Here, Perdlen has to be greater than the beam line lattice length.

Line 5: Flagdist, Rstartflg, Flagsbstp, Nemission, Temission

Flagdist Type of the initial distribution. This is a number between 1 and 27:

```
 1  Uniform - 6d uniform distribution
 2  Gauss3 - 6d Gaussian distribution
 3  Waterbag - 6d Waterbag distribution
 4  Semigauss - 3d Waterbag distribution in spatial and 3d Gaussian
                distribution in momentum space
 5  KV3d - transverse KV distribution and longitudinal uniform distribution
10 ParobGauss - transverse parabolic and longitudinal Gaussian distribution
15 SemicirGauss - transverse semi-circle and longitudinal Gaussian distribution
16  Read - read in an initial particle distribution from file Partcl.data
24  readParmela - read in Parmela particle format
25  readElegant - read in Elegant particle format
27  CylcoldZSob - uniform cylinder with longitudinal density modulation
                and Gaussian distribution in momentum space
ijk Combine - spatial transverse uniform or Gaussian (i=1,2), longitudinal
              flat-top with linear or Gaussian ramping or longitudinal
              Gaussian, or Tukey function (j=1,2,3,4), and 3D momentum distribution (k=1,2
```
The detailed input parameters for each type of initial distribution are defined in lines 6-8.
See the description of Line 6 - 8 for more information.

Rstartflag If $restartflag = 1$, restart the simulation from the previous check point. If $restartflag = 0$, start the simulation from the beginning.

Flagsbstp Not used.

Nemission There is a time period where the laser is shining on the cathode and electrons are being emitted. `Nemisson` gives the number of numerical emission steps. More steps gives more accurate modeling but the computation time varies linearly with the number of steps. If Nemission < 0, there will be no cathode model. The particles are assumed to start in a vacuum.

Temission Laser pulse emission time (sec.) *Note, this time needs to be somewhat greater than the real emission time in the initial longitudinal distribution so that the time step size is changed after the whole beam is a few time steps out of the cathode.*

Line 6 - 8 These three lines give the initial distribution parameters in x-px plane, y - py plane and z-pz plane. These lines, along with `distType` as given in line 5, is used to form the initial distribution.

There are 21 parameters for the initial distribution except the `read` option, which read in the particle data directly from the external file. The parameter names are:

```
sigx(m),  sigpx, muxpx, xscale, pxscale, xmu1(m), xmu2,
sigy(m),  sigpy, muypy, yscale, pyscale, xmu3(m), xmu4,
sigz(m),  sigpz, muzpz, zscale, pzscale, xmu5(m), xmu6
```

WARNING: 1)sigx, sigy, sigz cannot be zeros even for read-in distribution. 2)xmu1-6 will be added to the read-in distribution. 3)xmu6 is used to set beam energy. Here, `xscale` and `pxscale` are multipliers for sigx and sigpx. They are used when the `sigx` and `sigpx` numbers are derived from a match to the lattice and `xscale` and `pxscale` are then used to mismatch the initial distribution. If not used, `xscale` and `pxscale` should just be set to 1. `sigz` is related to the laser pulse length through $v_0 t_{laser}$ in some distributions, where v_0 is obtained from the kinetic energy Bkenergy. Note: for the ijk distribution, the zscale needs to be slightly greater than zero even for a hard edge cylinder beam. `xmu1-6` define the center offset of the initial distribution. The phase space density is generally written as

$$\rho(\widetilde{x}, \widetilde{p}_x, \widetilde{y}, \widetilde{p}_y, \widetilde{z}, \widetilde{p}_z) = \rho_x(\widetilde{x}, \widetilde{p}_x)\, \rho_y(\widetilde{y}, \widetilde{p}_y)\, \rho_z(\widetilde{z}, \widetilde{p}_z) \tag{1}$$

With the phase space coordinates being

$$\begin{aligned}
\widetilde{x} &= x - \text{xmu1} \\
\widetilde{p}_x &= p_x - \text{xmu2}
\end{aligned} \tag{2}$$

with similar equations for the other four coordinates.

1 Uniform

$$\rho_x(\widetilde{x}, \widetilde{p}_x) = \begin{cases} \text{const} & |\widetilde{x}| < \frac{\sqrt{3}\,\widetilde{\sigma}_x}{\sqrt{1-\text{muxpx}^2}} \text{ and } |\widetilde{p}_x| < \sqrt{3}\,\widetilde{\sigma}_{px}(1 - \frac{\text{muxpx}}{\sqrt{1-\text{muxpx}^2}}) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $\widetilde{\sigma}_x = $ sigx and $\widetilde{\sigma}_{px} = $ sigpx. For `muxpx = 0`, in $(\widetilde{x}, \widetilde{p}_x)$ phase space, the distribution is a rectangle of constant density.

2 Gauss3 The form of the density function is

$$\rho_x(\widetilde{x}, \widetilde{p}_x) \propto e^{-f_x/2} \tag{4}$$

where

$$f_x(\widetilde{x}, \widetilde{p}_x) \equiv \frac{\widetilde{x}^2}{\widetilde{\sigma}_x^2} + \frac{2\,\widetilde{x}\,\widetilde{p}_x\,\text{muxxpx}}{\widetilde{\sigma}_x\,\widetilde{\sigma}_{px}} + \frac{\widetilde{p}_x^2}{\widetilde{\sigma}_{px}^2} \tag{5}$$

with similar forms for ρ_y and ρ_z.

7

3 Waterbag The form of the density function is

$$\rho_x(\widetilde{x}, \widetilde{p}_x) = \begin{cases} \text{const} & f_x(\widetilde{x}, \widetilde{p}_x) < \sqrt{8} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where f_x is given in `Gauss3`. The density is a constant inside of an ellipse.

4 Semigauss Essentially a Waterbag distribution in $(\widetilde{x}, \widetilde{y}, \widetilde{z})$ space and a Gaussian distribution in $(\widetilde{p}_x, \widetilde{p}_y, \widetilde{p}_z)$ space.

5 KV3d $\delta(f_x, f_y)$–function transversely and uniform longitudinally. Here the transverse distribution is a self-consistent solution of the Poisson-Vlasov equation for a coasting beam. Not a realistic model but good for diagnostic checks.

16 Read Read distribution from an external file. called `partcl.data` Here the file `partcl.data` has the format:

nptot

x, px, y, py, z, pz

where, nptot is number of lines (i.e. particles) in the file, and $x(m)$, px/mc, $y(m)$, py/mc, $z(m)$, pz/mc are six phase space coordinates. Note: these coordinates will be shifted by the centroid parameters defined in lines 6-8.

27 CylcoldZSob Uniform cylinder with longitudinal density modulation in spatial, Gaussian distribution in transverse momentum space and a semi-Gaussian distribution in longitudinal momentum space. Here, the relative modulation amplitude is "muzpz", the modulation wavelength is "zscale (m)".

ijk Combine Generating initial particle distribution based on the combination of transverse spatial distribution (2 types), longitudinal spatial distribution (3 types) and 3D momentum distribution (4 types). For example, flagdist = 111 denotes type 1 from the transverse spatial distribution, type 1 from the longitudinal spatial distribution, and type 1 from the 3D momentum distribution. In this case, it denotes a transverse uniform ellipse, longitudinal flat-top with linear ramping in spatial, and 3d full Gaussian distribution in momentum space. For flagdist = ijk, where i = 1 (transverse uniform ellipse) and 2 (Gaussian with cut-off set by xscale and yscale); j = 1 (flat-top set by sigz with linear ramping set by zscale), 2 (flat-top set by sigz with 2sigma Gaussain ramping set by zscale), and 3 (Gaussian with cut-off set by zscale); k = 1, (3D Gaussian momentum), 2 (transverse Gaussian momentum, longitudinal semi-Gaussian $p_z \exp(-\frac{p_z^2}{sigpz^2})$), 3 (x-ray emission model $f(E, \theta, \phi) \propto \frac{E}{(E+W_k)^4} \sin(2\theta)$, where $0 \leq \theta \leq \pi/2$ and $0 \leq \phi \leq 2\pi$, kinetic energy $0 \leq E \leq Emax$ (eV), and W_k is the work function in eV. The Emax is set by the input parameter sigpz and W_k is set by the input parameter pzscale, and 4 (3 step model given by Dowell et al. [2]). The input parameters for the 3 step model are photo energy (set by sigpz in eV), Fermi energy level (set by muzpz in eV) and effective work function (set by pzscale in eV).

Line 9: Bcurr, Bkenergy, Bmass, Bcharge, Bfreq, Tini

Bcurr Beam current in Amps.

Bkenergy Initial beam pseudo-kinetic energy in eV. WARNING: this one is used to calculate the drift velocity that is needed to convert the laser pulse length in seconds into the longitudinal bunch length in meters in the initial distribution and to drift the particle out of the wall. The real initial beam energy needs to be input from "xmu6" in the initial distribution or the particle data file for the readin distribution.

Bmass Mass of the particles in eV.

Bcharge Particle charge in units of proton charge.

Bfreq Reference frequency in Hz.

Tini Initial reference time in seconds.

6   Lattice Beam Line Elements

Line 10 and beyond, describe the lattice to track through. Each line of lattice input represents one element. Elements may overlap longitudinally. The general form of a lattice line is:

```
Blength, Bnseg, Bmpstp, Btype, V1  ...  V23
```

NOTE, even though the IMPACT-T code allows the overlap of element field, the maximum length of the computational domain is set by the starting location of the last element plus the length of the last element. The beam line element has to be arranged so that $\mathrm{zmin}(i+1) \geq \mathrm{zmin}(i)$, *where* $i = 1, \cdots, N$.

Blength The longitudinal length of the element.

Bnseg, Bmpstp Not used except for the Type less than 0, i.e. the BPM.

Btype Type of element. An integer specifying the type of element.See below for more details.

V1 ... V23 Element parameters. See below for more details.

The rf E field is parameterized by

$$\mathrm{field} = \mathrm{f}(\mathrm{x}, \mathrm{y}, \mathrm{z}) \, \cos(2 \, \pi \, \mathrm{f} \, \mathrm{t} + \theta_0) \tag{7}$$

Some of the element parameters are:

zedge Longitudinal position of the entrance face of the element.

scale This can be used to scale the field amplitude. Normally set to 1.

RF frequency Frequency of the RF field in Hz.

theta0 Initial phase in degree.

file ID ID number of the external file to open to read in field data.

radius Aperture radius. Not currently used in the calculation.

x misalignment error Used in Quadrupole, Multipole (sextupole, octupole, decapole) and SolRF beam line elements.

y misalignment error Used in Quadrupole, Multipole (sextupole, octupole, decapole) and SolRF beam line elements.

rotation error x Used in Quadrupole, Multipole (sextupole, octupole, decapole) and SolRF beam line elements.

rotation error y Used in Quadrupole, Multipole (sextupole, octupole, decapole) and SolRF beam line elements. Here, the notation of "+y" means clockwise y.

rotation error z Used in Quadrupole, Multipole (sextupole, octupole, decapole) and SolRF beam line elements.

The following are the specific parameters associated with each element type.

-1, -2, -3, -4, -5, -6, -7, -8, -9, -11, -12, -13, -15, -16, -17, -99: BPM Beam position monitor and etc. blength, Bnseg, Bmpstp, btype, V1, V2, V3, V4, ... V8. *NOTE: the location of the first "-1,-2,-4,-7,-9,-11,-12" line should be put after the initial beam distribution and aligned in sequence!!!*

If $btype = -1$, kick the transverse beam centroid at given location V2(m) by x offset V3(m), Px $(\gamma\beta_x)$ offset V4, y offset V5(m), Py $(\gamma\beta_y)$ offset V6, z offset V7(m), and Pz $(\gamma\beta_z)$ offset V8.

If $btype = -2$, output particle phase-space coordinate information at given location V3(m) into filename fort.Bmpstp with particle sample frequency Bnseg. Here, the maximum number of phase-space files which can be output is 100. Here, 40 and 50 should be avoided since these are used for initial and final phase space output.

If $btype = -3$, output particle phase-space and prepare restart at given location V3(m) into filename fort.(Bmpstp+myid). Here, myid is processor id. On single processor, it is 0. If there are multiple restart lines in the input file, only the last line matters.

If $btype = -4$, change the time step size from the initial Dt (secs) into V4 (secs) after location V3(m). The maximum number of time step change is 100.

If $btype = -5$, switch the simulation from azimuthal symmetry to fully 3d simulation after location V3(m). This location should be set as large negative number such as "-1000.0" in order to start the 3D simulation immediately after the electron emission. If there are multiple such lines in the input file, only the last line matters.

If $btype = -6$, turn on the wake field effects between location V3(m) and V4(m). If $Bnseg$ is greater than 0, the longitudinal and transverse wake function will be read in from file "fort.Bmpstp". If $Bnseg \leq 0$, the code will use analytical wake function described as follows. For analytical wake functions, the wake function parameters (iris radius) a = V5(m), (gap) g = V6(m), (period) L = V7(m). Here, the definition of these parameter can be found from SLAC-PUB-9663, "Short-Range Dipole Wakefields in Accelerating Structures for the NLC," by Karl L.F. Bane. This will be updated in the future since the parameters a, g, L might change from cell to cell within a single structure. For the backward traveling wave structure, the iris radius "a" has to be set greater than 100, gap "g" set to the initialization location of BTW. For backward traveling wave structures, the wakes are hardwired inside the code following the report: P. Craievich, T. Weiland, I. Zagorodnov, "The short-range wakefields in the BTW accelerating structure of the ELETTRA linac," ST/M-04/02. For $-10 < a < 0$, it uses the analytical equation from the 1.3 GHz Tesla standing wave structure. For $a < -10$, it assumes the 3.9 GHz structure longitudinal wake function. For external supplied wake function, The maximum number data point is 1000. The data points are assumed uniformly distributed between 0 and V7(m). The V6 has to less than 0. Each line of the fort.Bmpstp contains longitudinal wake function (V/m) and transverse wake function (V/m/m).

If $bytpe = -7$, merge the multiple bins into only one bin at given location V3(m) in order to save computing time. If there are multiple such lines in the input file, only the last line matters.

if $bytpe = -8$, switch on/off the space-charge calculation at given location V3(m) according to the sign of V2 (> 0 on, otherwise off).

If $bytpe = -9$, output slice-based information at given location V3(m) into file "fort.Bmpstp" using "Bnseg" slices.

If $bytpe = -11$, collimate particles at given location V2(m) with transverse aperture defined by V3(m) (xmin), V4(m) (xmax), V5(m) (ymin), V6(m) (ymax).

If $bytpe = -12$, apply instant 6x6 linear matrix (read in from linearmap.in) kick at given location V1(m).

If $bytpe = -13$, include dielectric wake field implemented by Dianel Mihalcea. Here, V1 = Z start for the dielectric structure. V2 = Z end for the dielectric structure. V3 = 0.0 if geometry is rectangular and 1.0 if cylindrical. V4 = inner radius or height. V5 = outer radius or height. V6 = dielectric relative permittivity (eps). V7 = Lx (length in x-direction). This parameter is ignored if cylindrical geometry (V3=1.0). V8 = number of modes associated with x-axis. For cylindrical symmetry only monopole modes (V8=0) are supported. V9 = number of modes associated with y-axis (ref: PRST-AB, 15, 081304, (2012)). A typical line looks like: 0.0 0 0 -13 0.02 0.12 0.0 2.5e-3 5.0e-3 4.0 10.0e-3 6.0 10.0 /

If $btype = -15$, switch on the direct point-to-point N-body calculation of the space-charge forces. The cut-off radius of particle is given by V3(m).

If $btype = -16$, heat the beam $(\gamma\beta_z)$ at V1 location by rms size V2.

If $btype = -17$, rotate the beam with respect to z-axis at V1 location by V2 radians.

If $bytpe = -99$, stop the simulation at given location V3(m).

0: DriftTube Drift space

```
V1: zedge
V2: radius    Not used.
```

1: Quadrupole Quadrupole

```
V1: zedge
V2: quad gradient (T/m)
V3: file ID
    If > 0, then include fringe field (using Enge function) and
    V3 = effective length of quadrupole.
V4: radius (m)
V5: x misalignment error (m)
V6: y misalignment error (m)
V7: rotation error x     (rad)
V8: rotation error y     (rad)
V9: rotation error z     (rad)
If V9 != 0, skew quadrupole
V10: rf quadrupole frequency   (Hz)
V11: rf quadrupole phase    (degree)
```

2: ConstFoc 3D constant focusing beam line element.

```
V1: zedge
V2: x focusing gradient: kx0^2
V3: y focusing gradient: ky0^2
V4: z focusing gradient: kz0^2
V5: radius
```

3: Sol Solenoid.

```
V1: zedge
V2: Bz0 (T)                Not used.
V3: file ID
V4: radius
V5: x misalignment error  Not used.
```

```
     V6: y misalignment error  Not used.
     V7: rotation error x       Not used.
     V8: rotation error y       Not used.
     V9: rotation error z       Not used.
```

The discrete magnetic field data is stored in 1Tv3.T7 file. The read in format of 1Tv3.T7 is as following:

```
 ! the input range units are cm
          read(14,*,end=33)tmp1,tmp2,tmpint
          this%RminRft = tmp1/100.0
          this%RmaxRft = tmp2/100.0
          this%NrIntvRft = tmpint
 ! the input range units are cm
          read(14,*,end=33)tmp1,tmp2,tmpint
          this%ZminRft = tmp1/100.0
          this%ZmaxRft = tmp2/100.0
          this%NzIntvRft = tmpint
          n = 0
50        continue
             read(14,*,end=77)tmp1,tmp2
             nn = n+1
             j  = (nn-1)/(this%NrIntvRft+1) + 1
             i = mod((nn-1),this%NrIntvRft+1) + 1
             this%brdatat(i,j) = tmp1
             this%bzdatat(i,j) = tmp2
!             write(15,100)float(i-1),bzdatat(i,j)
             n = n + 1
          goto 50
77        continue
```

Here, ZminRft = 0.0, RminRft = 0.0.

4: Dipole Dipole bending magnet element. In the IMPACT-T code, the bending magnet is described by four linear functions that characterize the pole face and starting and ending face of the fringe field. Here, the coordinate system consists of Cartesian coordinates X-Z with origin located before the bending magnet. Here, X is the horizontal direction and Z is the longitudinal direction. The particles are tracked in this X-Z Cartesian coordinate system with external B field given in these coordinates. The space-charge forces are calculated by transforming the particle coordinates into rotated coordinates of the bunch centroid. The space-charge forces are calculated in these rotated coordinates, including relativistic effects. Then, the space-charge forces are transformed from the rotated coordinates to the X-Z coordinates to push the particles. At the exit of the bend, all the particle coordinates are transformed to the new X-Z coordinates of the next beam line element. The four functions which characterize the faces are: $z1 = k1*x1+b1$, $z2 = k2*x2+b2$, $z3 = k3*x3+b3$, $z4 = k4*x4+b4$. The region between z1 and z2 is the fringe field region for entrance, z2 and z3 is the region of constant By field, z3 and z4 is the region of fringe field for exit. Here, the parameters k1, b1, k2, b2, k3, b3, k4, b4 are stored in the input file rfdataV4 (from 3-10). "V4" is the input parameter as described below. The first two parameters in the file rfdataV4 are face-ID (not used) and gamma for the reference particle. So far, we have assumed a linear function on the middle plane (y=0) for the field in the fringe field region.

```
     V1:  zedge
     V2:  x field strength (T)
     V3:  y field strength (T)
```

```
V4:  file ID              file ID to contain the geometry information of bend.
V5:  half of gap width (m) .
V6:  x misalignment error  Not used.
V7:  y misalignment error  Not used.
V8:  rotation error x      Not used.
V9:  rotation error y      Not used.
V10: rotation error z      Not used.
```

Here, the input file **rfdatav4** contains 22 lines. The first line contains the switch flag for 1D CSR wakefield. The CSR wakefield will be included for a value greater than 0. The second line contains the γ of the beam. Lines 3 to 10 contain k1, b1(m), k2, b2(m), k3, b3(m), k4, b4(m), the geometric description of the pole faces at the entrance and the exit. Line 11 is twice the shift z0 (m) at the entrance fringe field region. Line 12 is twice the shift z0 (m) at the exit fringe field region. These two lines are used to determine the shift used in the Enge function fitting. The shifts are half of those values. Lines 13 to 20 contain 8 coefficients in the Enge function. Here, we have assumed that the entrance and the exit have the same Enge function coefficients. Line 21 is the effective starting location along the arc of the bend in meter. Line 22 is the effective ending location along the arc of the bend in meter. These two numbers are used in the calculation of transient effects of the CSR wakefield. Normally, these two numbers are the middle locations of the fringe region. However, if transient effects at the exit are important, the total length of the bend may include a section of drift.

Note, the length (Blength) in the dipole element input line should contain the arc length of the reference particle inside the fringe fields and the dipole field region.

Given the geometry of the bending magnetic field, the vertical field strength V3 should adjusted so that the final bending angle attains the design value. Here, the bending angle can be obtained from the output file fort.38 that contains the reference particle trajectory. (that is $arctan(\gamma\beta_x/\gamma\beta_z)$ in the last line of the fort.38)

5: Multipole Multipole (See Quadrupole for Quadrupole)

```
V1:  zedge
V2:  pole type: sextupole (2), octupole (3), decapole (4)
V3:  field strength (T/m^n)
V4:  file ID              If > 1e-5 then read in the fringe field
V5:  radius               Not used.
V6:  x misalignment error
V7:  y misalignment error
V8:  rotation error x
V9:  rotation error y
V10: rotation error z
```

101: DTL Drift tube linac

```
V1:  zedge
V2:  scale
V3:  RF frequency
V4:  theta0
V5:  file ID
V6:  radius
V7:  quad 1 length
V8:  quad 1 gradient
V9:  quad 2 length
V10: quad 2 gradient
```

```
V11: x misalignment error for Quad 1
V12: y misalignment error for Quad 1
V13: rotation error x for Quad 1
V14: rotation error y for Quad 1
V15: rotation error z for Quad 1
V16: x misalignment error for Quad 2
V17: x misalignment error for Quad 2
V18: rotation error x for Quad 2
V19: rotation error y for Quad 2
V20: rotation error z for Quad 2
V21: x misalignment error for RF cavity
V22: y misalignment error for RF cavity
V23: rotation error x for RF cavity
V24: rotation error y for RF cavity
V25: rotation error z for RF cavity
```

104: SC Superconducting cavity.

```
V1:  zedge, the real used field range in z is [zedge,zedge+Blength].
V2:  scale of RF field
V3:  RF frequency
V4:  theta0
V5:  file ID
V6:  radius
V7:  x misalignment error
V8:  y misalignment error
V9:  rotation error x
V10: rotation error y
V11: rotation error z
```

Here, the rfdataV5 file contains the Fourier coefficients for RF field. The following is an example of the rfdataV5 file:

```
1.26367829640838925 /Fourier coefficients of Ez on axis.
0.403913638991465318
-0.154151286798631891E-08
-0.148620139078017188
-0.460805472944262358E-0
0.863984014678981088E-01
0.369511645542406301E-09
```

105: SolRF Solenoid with embedded RF field.

```
V1:  zedge, the real used field range in z is [zedge,zedge+Blength].
V2:  scale of RF field
V3:  RF frequency
V4:  theta0
V5:  file ID
V6:  radius
V7:  x misalignment error
V8:  y misalignment error
V9:  rotation error x
V10: rotation error y
V11: rotation error z
V12: scale of solenoid B field. [Only used with SolRF element.]
```

Here, the rfdataV5 file contains the Fourier coefficients for both E fields and B fields. The first half contains E fields, and the second half contains B fields. The following is an example of the rfdataV5 file:

```
7.0 /# of Fourier coef. of Ez on axis
-0.0616710037 /distance before the zedge.
0.0616710037 /distance after the zedge.
0.1233420074 /length of the Fourier expanded field.
 1.26367829640838925 /Fourier coefficients of Ez on axis.
 0.403913638991465318
 -0.154151286798631891E-08
 -0.148620139078017188
 -0.460805472944262358E-0
 0.863984014678981088E-01
 0.369511645542406301E-09
7.0 /# of Fourier coef. of Bz on axis.
-1.04665 /distance before the zedge.
0.95335 /distance after the zedge.
2.0 /length of total Fourier expanded field
 0.980347705847995596E-04 /Fourier coefficients of the Bz on axis.
 0.723873881539517666E-04
 0.297047022969352459E-10
 0.400457811524145360E-04
 0.905081032754064534E-10
 0.202293442430655808E-04
 0.933729888398092589E-10
```

Even though there is only one type of field, RF Ez or Bz, one still needs to keep five lines for the 0 field. For example,

```
7.0 /# of Fourier coef. of Ez on axis
-0.0616710037 /distance before the zedge.
0.0616710037 /distance after the zedge.
0.1233420074 /length of the Fourier expanded field.
 1.26367829640838925 /Fourier coefficients of Ez on axis.
 0.403913638991465318
 -0.154151286798631891E-08
 -0.148620139078017188
 -0.460805472944262358E-0
 0.863984014678981088E-01
 0.369511645542406301E-09
1.0 /# of Fourier coef. of Bz on axis.
0.0
0.0
0.0
0.0  /Fourier coefficients of the Bz on axis.
```

110: **EMfld** Read in discrete EM field data as a function of (x,y,z) or (r,z) or analytical representation of EM field data. This element is not used in the IMPACT-T code but used in the z based IMPACT code.

111: **EMfldCart** Read in discrete EM field data, E_x, E_y, E_z, B_x, B_y and B_z, as a function of (x,y,z). Here, each field data point is a **complex number** to represent both traveling wave and standing wave fields. For standing wave field, the electric field is the real part of the complex number (E,0),

the magnetic field is the imaginary part of the complex number (0,B). For a static electric field, it can be written as (E,0) with 0 frequency and phase. For a static magnetic field, it can be writte as (0,B) with 0 frequency and -90 degree phase.

```
V1:  zedge (meter)
  V2:  scale of RF field
  V3:  RF frequency
  V4:  theta0
  V5:  file ID
  V6:  radius                    not used yet
  V7:  x misalignment error      not used yet
  V8:  y misalignment error      not used yet
  V9:  rotation error x          not used yet
  V10: rotation error y          not used yet
  V11: rotation error z          not used yet
```

The discrete field data are stored in the file 1Tv3.T7. Some read-in FORTRAN codes are as following:

```
          ! the input range units are m
          read(14,*,end=33)tmp1,tmp2,tmpint
          this%XminRfgt = tmp1
          this%XmaxRfgt = tmp2
          this%NxIntvRfgt = tmpint
          read(14,*,end=33)tmp1,tmp2,tmpint
          this%YminRfgt = tmp1
          this%YmaxRfgt = tmp2
          this%NyIntvRfgt = tmpint
          read(14,*,end=33)tmp1,tmp2,tmpint
          this%ZminRfgt = tmp1
          this%ZmaxRfgt = tmp2
          this%NzIntvRfgt = tmpint
          n = 0
50        continue
              read(14,*,end=77)tmp1,tmp2,tmp3,tmp4,tmp5,tmp6
              n = n+1
              k = (n-1)/((this%NxIntvRfgt+1)*(this%NyIntvRfgt+1))+1
              j = (n-(k-1)*(this%NxIntvRfgt+1)*(this%NyIntvRfgt+1)-1)/(this%NxIntvRfgt+1) + 1
              i = n - (k-1)*(this%NxIntvRfgt+1)*(this%NyIntvRfgt+1) - &
                  (j-1)*(this%NxIntvRfgt+1)
              this%Exgridt(i,j,k) = tmp1
              this%Eygridt(i,j,k) = tmp2
              this%Ezgridt(i,j,k) = tmp3
              this%Bxgridt(i,j,k) = tmp4
              this%Bygridt(i,j,k) = tmp5
              this%Bzgridt(i,j,k) = tmp6
          goto 50
77        continue
```

112: EMfldCyl Read in discrete EM field data $E_z(MV/m)$, $E_r(MV/m)$, and $H_\theta(A/m)$ as a function of (r,z) of EM field data (from SUPERFISH output).

```
  V1:  zedge
```

```
   V2:  radius
   V3:  RF frequency
   V4:  theta0
   V5:  file ID
   V6:  radius                        not used yet
   V7:  x misalignment error          not used yet
   V8:  y misalignment error          not used yet
   V9:  rotation error x              not used yet
   V10: rotation error y              not used yet
   V11: rotation error z              not used yet
```

The discrete field data is stored in 1Tv3.T7 file. Some read-in FORTRAN codes are as following:

```
          ! the input range units are cm
          read(14,*,end=33)tmp1,tmp2,tmpint
          this%ZminRft = tmp1/100.0
          this%ZmaxRft = tmp2/100.0
          this%NzIntvRft = tmpint
          ! the input range units are cm
          read(14,*,end=33)tmp1
          read(14,*,end=33)tmp1,tmp2,tmpint
          this%RminRft = tmp1/100.0
          this%RmaxRft = tmp2/100.0
          this%NrIntvRft = tmpint
          n = 1
50        continue
       read(14,*,end=77)tmp1,tmp2,tmp3,tmp4 !Ez, Er, |E| (MV/m), H (A/m)
             j  = (n-1)/(this%NzIntvRft+1) + 1
             i = mod((n-1),this%NzIntvRft+1) + 1
             this%ezdatat(i,j) = tmp1*1.0e6
             this%erdatat(i,j) = tmp2*1.0e6
             !convert from H (A/m) to Tesla
             this%btdatat(i,j) = tmp4*mu0
             n = n + 1
             write(15,100)float(i-1),this%ezdatat(i,j),this%erdatat(i,j)
          goto 50
77        continue
```

113: EMfldAna EM field data as a analytical function. Note: the following three analytical functions are test functions.

```
   V1:  zedge
   V2:  escale
   V3:  RF frequency
   V4:  theta0
   V5:  file ID
        -1: alpha magnet field:
            bx = 0.0d0
            by = escale*pos(3)
            bz = escale*pos(2)
        -2: traveling wave field in meander plates:
            t0 = V3 !wave starting time
            tt = pos(4) - t0
```

```
                    tt1 = V4 - t0 !wave ending time
                    z0 = V6 !wave starting location
                    zslope = V7 !wave rising length
                    vzspeed = V8 !wave z speed
                    if(tt>=0.0 .and. tt<= tt1 ) then
                      z0t = z0 + vzspeed*tt
                      z1 = z0t - zslope
                      z2 = z0t + zslope
                      if(pos(3)>=z1 .and. pos(3)<=z2) then
                        ey = escale/zslope*(pos(3)-z0t)
                      else if(pos(3)>z2) then
                        ey = escale
                      else
                        ey = -escale
                      endif
                    else if(tt<0.0) then
                      ey = escale
                    else
                      ey = -escale
                    endif
            -3: DC surface roughness field:
                    ww = V3*2*pi
                    an = V7
                    wkn = V8
                    ex = escale*an*wkn*exp(-wkn*(pos(3)-zedge))*&
                         sin(wkn*pos(1))
                    ey = 0.0d0
                    ez = escale+escale*an*wkn*exp(-wkn*(pos(3)-zedge))*&
                         cos(wkn*pos(1))
     V6:
     V7:
     V8:
     V9:
     V10:
     V11:
```

7   Output Data

fort.18:

```
    1st col: time (secs)
    2nd col: distance (m)
    3rd col: gamma
    4th col: kinetic energy (MeV)
    5th col: beta
    6th col: Rmax (m) R is measured from the axis of pipe
    7th col: rms energy deviation normalized by MC^2
```

fort.24, fort.25: X and Y RMS size information

```
    1st col: time (secs)
```

```
2nd col: z distance (m)
3rd col: centroid location (m)
4th col: RMS size (m)
5th col: Centroid momentum normalized by MC
6th col: RMS momentum normalized by MC
7th col: Twiss parameter
8th col: normalized RMS emittance (m-rad)
```

fort.26: Z RMS size information

```
1st col: time (secs)
2nd col: centroid location (m)
3rd col: RMS size (m)
4th col: Centroid momentum normalized by MC
5th col: RMS momentum normalized by MC
6th col: Twiss parameter
7th col: normalized RMS emittance (m-rad)
```

fort.27: maximum amplitude information

```
1st col: time (secs)
2nd col: z distance (m)
3rd col: Max. X (m)
4th col: Max. Px (MC)
5th col: Max. Y (m)
6th col: Max. Py (MC)
7th col: Max. Z (m) (with respect to centroid)
8th col: Max. Pz (MC)
```

fort.28: load balance and loss diagnostic

```
1st col: time (secs)
2nd col: z distance (m)
3rd col: min # of particles on a PE
4th col: max # of particles on a PE
5th col: total # of particles in the bunch
```

fort.29: cubic root of 3rd moments of the beam distribution

```
1st col: time (secs)
2nd col: z distance (m)
3rd col: X (m)
4th col: Px (MC)
5th col: Y (m)
6th col: Py  (MC)
7th col: Z (m)
8th col: Pz (MC)
```

fort.30: square root, square root of 4th moments of the beam distribution

```
1st col: time (secs)
2nd col: z distance (m)
3rd col: X (m)
4th col: Px (MC)
5th col: Y (m)
6th col: Py (MC)
7th col: Z (m)
8th col: Pz (MC)
```

fort.34, fort.35: X and Y output information in dipole reference coordinate system (inside dipole ONLY)

```
1st col: time (secs)
2nd col: z (m)
3rd col: x or y (m)
4th col: xrms or yrms (m)
5th col: Centroid momentum normalized by MC
6th col: RMS momentum normalized by MC
7th col: correlation parameter
8th col: normalized RMS emittance (m-rad)
```

fort.36: Z output information in dipole reference coordinate system (inside dipole ONLY)

```
1st col: time (secs)
2nd col: z (m)
3th col: rms (m)
4th col: Centroid momentum normalized by MC
5th col: RMS momentum normalized by MC
6th col: correlation parameter
7th col: normalized RMS emittance (m-rad)
```

fort.37: maximum amplitude information in dipole reference coordinate system (inside dipole ONLY)

```
1st col: time (secs)
2nd col: z distance (m)
3rd col: Max. X (m)
4th col: Max. Px (MC)
5th col: Max. Y (m)
6th col: Max. Py (MC)
7th col: Max. Z (m) (with respect to centroid)
8th col: Max. Pz (MC)
```

fort.38: reference particle information in dipole reference coordinate system (inside dipole ONLY)

```
1st col: time (secs)
2nd col: x distance (m)
3rd col: Px/MC
4th col: y (m)
5th col: Py/MC
6th col: z (m)
7th col: Pz/MC
```

fort.40: initial particle distribution at t = 0. Particle coordinates are dimensionless as defined before.

fort.50: final particle distribution projected to the centroid location of the bunch.

fort.60 output file for slice information of the initial distribution: 1st col: bunch length (m) 2nd col: number of macroparticles per cell 3rd col: current profile 4th col: x slice emittance (m-rad) 5th col: y slice emittance (m-rad) 6th col: energy spread per cell without taking out correlation (eV) 7th col: uncorrelated energy spread per cell (eV)

fort.70 output file for slice information of the final distribution:

8 Physical Models

The general equations of motion used in the IMPACT-T code are:

$$\dot{\mathbf{r}} = \frac{\mathbf{p}}{m\gamma} \tag{8}$$

$$\dot{\mathbf{p}} = q(\mathbf{E} + \frac{\mathbf{p}}{m\gamma} \times \mathbf{B}) \tag{9}$$

where $\gamma = 1/\sqrt{1 - \beta^2}$, $\beta_i = v_i/c$ with $i = x, y, z$, c is the speed of light, m is the rest mass of particle, q is the charge of particle. The electric field, $\mathbf{E}$, and the magnetic field, $\mathbf{B}$, include the contributions from the external focusing and accelerating fields and the space-charge fields of intra-particle Coulomb interactions.

Given electric and magnetic fields, the equations of motion are solved using a second-order leap-frog algorithm: the particles are drifted half time step; the particles are collected and deposited onto a three-dimensional grid; the Poisson equation is solved in the beam frame; the electric and magnetic fields are obtained in the laboratory frame through the Lorentz transformation; the particle momenta are updated using both the space-charge fields and external fields for one time step according to Eq. 2; the particles are drifted another half time step. This procedure is repeated for many time steps until the beam is out of the computational domain of beam line elements.

8.1 Internal Coordinates of Particle

The coordinates of particle used inside the IMPACT-T are dimensionless variables:

$$x/dz,\ p_x/mc,\ y/dz,\ p_y/mc,\ z/dz,\ p_z/mc \tag{10}$$

where the scale length $dz = c\,dt$ with dt being the initial time step size, the momentum $p_k = \gamma\beta_k$ for $k = x,\ y,\ z$, $\beta_k = v_k/c$, v_k is the velocity in each direction and c is the speed of light.

8.2 Particle Emission from Cathode

The emission process of electrons from the photocathode is simulated by generating a 3D particle distribution behind the cathode. The longitudinal bunch length is determined by the laser pulse length Δt and longitudinal velocity associated with the initial averaged kinetic energy of the beam. This particle distribution is moved out of the cathode using $N_{emission}$ steps within Δt seconds. Here, $\Delta t \beta C$ gives the full length of the beam. During the emission, both the external fields and the space-charge effects are taken into account in the simulation. The image charge effects from the cathode can be included depending on the material property of the photocathode. If $N_{emission} < 0$, the emission model is turned off. The simulation will start as if there is no cathode.

8.3 Space-Charge Effects

For a beam with an input current > 0, space-charge effects will be included in the simulation. (0 current means no space-charge effects.) By default, the code assumes that the beam has an azimuthal symmetry. This allows the use of a smaller number of macroparticles in the simulation, which reduces the computation time significantly. For a fully 3d beam, an input line with type a code "-5" should be inserted in the input file, ImpactT.in (see more explanation in the beam line element section). This

line starts the 3d simulation after a given location in the input file. To turn on the 3d space-charge effects right at the beginning, that location needs to be set < 0. The electron beam out of the cathode can have a large energy spread. In this case, the quasi-static approximation using one reference frame might not be sufficient. Multiple reference frames (also called energy bins) can be used so that within each reference frame the energy spread is small. The total space-charge forces are the summation of contributions from all reference frames. In the current version of IMPACT-T, the multiple reference frames are modeled as multiple bunches. For modeling the beam dynamics in photoinjector, the initial distribution can be divided into multiple slice longitudinally so that each slice after the emission will have a small energy spread. These slices can be merged together after some distance when the total relative energy spread becomes small (see BPM -7 for usage).

The space-charge forces are calculated by solving the 3D Poisson equation with open boundary conditions using an integrated Green function method. The image charge effects of the conducting cathode are also included using a shifted Green function method. A detailed description of the space-charge solver can be found in reference [1].

8.4   Short Range Longitudinal and Transverse Wakefields

Besides the space-charge forces, the short range longitudinal wake field (monopole) and transverse wake field (dipole) are also included. To include the wakefield effects, the element type "-6" needs to be used (see the BPM explanation). (Note: the wakefield module works correctly only for one bin/slice model. To turn on the wakefields, the multiple bins have to be merged into one bin by using BPM "-7" function.)

The effective forces from the wake fields are given as:

$$F_x(s) = q \int_s^\infty W_T(s' - s) x(s') \lambda(s') ds' \tag{11}$$

$$F_y(s) = q \int_s^\infty W_T(s' - s) y(s') \lambda(s') ds' \tag{12}$$

$$F_z(s) = q \int_s^\infty W_L(s' - s) \lambda(s') ds' \tag{13}$$

where W_T is the transverse wake function, W_L is the longitudinal wake field function, and λ is the line density function of the beam. To compute the effective forces from wake fields more efficiently, we extend the above integrals into a full domain convolution as:

$$F(s) = \int_{-\infty}^{+\infty} G(s' - s) \rho(s') ds' \tag{14}$$

where

$$G(s) = \begin{cases} W(s) & for \ s \geq 0 \\ 0 & for \ s < 0 \end{cases} \tag{15}$$

$$\rho(s) = \begin{cases} x(s)\lambda(s) & for \ transverse \ wake \\ y(s)\lambda(s) & for \ transverse \ wake \\ \lambda(s) & for \ longitudinal \ wake \end{cases} \tag{16}$$

The above convolution is calculated using an FFT-based method on a doubled computational domain.

The transverse and longitudinal wake functions can be calculated following some analytical expressions or be read in from external files. For analytical representation, the transverse and longitudinal wake

functions for the SLAC $2\pi/3$ DDS structure is given by [3]

$$W_T(s) = \frac{4Z_0 c s_0}{\pi a^4} \phi(s)(1 - (1 + \sqrt{s/s_0} \exp(-\sqrt{s/s_0}))) \tag{17}$$

$$W_L(s) = \frac{Z_0 c}{\pi a^2} \phi(s) \exp(-\sqrt{s/s_{00}}) \tag{18}$$

with

$$s_0 = 0.169 \frac{a^{1.79} g^{0.38}}{L^{1.17}} \tag{19}$$

$$s_{00} = \frac{g}{8} \left(\frac{a}{\alpha(g/L)L} \right)^2 \tag{20}$$

$$\alpha(s) = 1 - \alpha_1 \sqrt{s} - (1 - 2\alpha_1)s \tag{21}$$

with $\alpha_1 = 0.4648$. Here, the structure parameters are iris radius a, gap g, period L, and $Z_0 = 120\pi$, $\phi(s)$ is a step function of s ($\phi(s) = 1$ for $s > 0$, 0 for $s < 0$). For the BTW accelerating structure at the ELETTRA linac, the transverse and longitudinal wake functions are [4]

$$W_T(s) = 2.8 \times 10^{16} \phi(s)((1 - (1 + \sqrt{s/1.2 \times 10^{-4}} \exp(-\sqrt{s/1.2 \times 10^{-4}}))) + 0.5\sqrt{s}) \tag{22}$$

$$W_L(s) = 1.0 \times 10^{12} \phi(s)(1226 \exp(-\sqrt{s/3 \times 10^{-4}}) + \frac{0.494}{\sqrt{s}} + 494\sqrt{s}) \tag{23}$$

For read-in external wake function option (see "-6" of BPM), the data has to be uniformly distributed between 0 and L with maximum data number 1000.

8.5   Longitudinal CSR Wakefield

The longitudinal CSR wake can be included in the simulation together with the 3D space-charge calculation if the first line input parameter in the input rfdata file is greater than 0. (*Note: the following CSR model is no-longer used in the IMPACT-T code. The current version uses an integrated Green function CSR model described in the references [5, 6].*)

Under the ultra-relativistic approximation, the rate of change of energy in the bunch is given by [7, 8]

$$\frac{dE(s,\phi)}{cdt} = -\frac{2e^2}{4\pi\epsilon_0 3^{1/3} R^{2/3}} \left(\int_{s-s_L}^{s} \frac{1}{(s-s')^{1/3}} \frac{\partial \lambda(s')}{\partial s'} ds' + \frac{\lambda(s-s_L) - \lambda(s-4s_L)}{s_L^{1/3}} \right) \tag{24}$$

where R is the bend radius, ϕ is the angle into the bend, s is the relative position within the bunch, t is time, $s_L = R\phi^3/24$ is the slippage length. The first term is normally called the steady-state wake when the slippage length becomes longer than the bunch length. The second term denotes the entrance transient wake.

For a bunch moving out of the bend, the CSR wake function under ultra-relativistic approximation are given by [9]

$$W(s) = \begin{cases} -\frac{4}{R} \frac{1}{(\phi_m + 2x)} \lambda(s - \frac{R}{6}\phi_m^2(\phi_m + 3x)) & for\ source\ in\ front\ of\ the\ bend \\ \frac{4}{R} \left(\frac{\lambda(s - \Delta s_{max})}{(\phi_m + 2x)} + \int_{s-\Delta s_{max}}^{s} \frac{1}{\psi + 2x} \frac{\partial \lambda}{\partial s'} ds' \right) & for\ source\ inside\ the\ bend \end{cases} \tag{25}$$

where dimensionless x ($x/R -> x$) is the distance from the observation point to the exit of the bend magnet, ϕ_m is the total bend angle of the magnet, ψ is related to $s - s'$ through the equation

$$s - s' = \frac{R\psi^3}{24} \frac{\psi + 4x}{\psi + x} \tag{26}$$

23

and the Δs_{max} is defined by

$$s - s' = \frac{R\phi_m^3}{24}\frac{\phi_m + 4x}{\phi_m + x} \tag{27}$$

In the IMPACT-T code, the line density function λ and its derivatives are smoothed using a first-order Savitzky-Golay filter [10]. The first derivatives of the density function is further smoothed using a customer designed local filter to calculate the second derivative [11].

8.6 RF Fields in Standing Wave Structures

For an RF linac, under proper gauge ($\phi = 0$), the external electromagnetic fields in a cylindrically symmetric accelerating structure can be obtained from [12]:

$$\mathbf{E} = -\frac{\partial \mathbf{A}}{\partial t} \tag{28}$$

$$\mathbf{B} = \nabla \times \mathbf{A} \tag{29}$$

where the vector potential $\mathbf{A}$ is given by

$$A_x = \frac{1}{\omega}x\sum_{n=0}^{\infty}\frac{1}{2(n+1)}e_n'(z)r^{2n}\sin(\omega t + \theta) \tag{30}$$

$$A_y = \frac{1}{\omega}y\sum_{n=0}^{\infty}\frac{1}{2(n+1)}e_n'(z)r^{2n}\sin(\omega t + \theta) \tag{31}$$

$$A_z = -\frac{1}{\omega}\sum_{n=0}^{\infty}e_n(z)r^{2n}\sin(\omega t + \theta) \tag{32}$$

with $r^2 = x^2 + y^2$ and

$$e_{n+1}(z) = -\frac{1}{4(n+1)^2}(e_n''(z) + \frac{\omega^2}{c^2}e_n(z)) \tag{33}$$

where $e_0(z)$ is the electric field on the axis, ω is the RF angular frequency, θ is the initial phase of the RF field with respect to global time zero, and the superscript prime denotes the derivative with respect to z. In the current version of the IMPACT-T code, n is up to 1 in the above equations' summation so that only second order and third order nonlinearity are included for off-axis fields.

If RF cavity does not have a cylindrical symmetry, the fully three-dimensional electromagnetic fields are read in from external data files.

8.7 Traveling Wave Structures

The electromagnetic fields in a traveling wave structure can be viewed as having the form of standing wave around the entrance and the exit of the structure, and a traveling wave in most cells between. In the IMPACT-T code traveling wave are modeled using a superposition of two standing waves. Here, the forwarding traveling wave field on the axis is given by [13]:

$$\begin{aligned}E_t(z,t) = \ &\frac{1}{\sin(\beta_0 d)}(E_s(z)\cos(\omega t + \beta_0 d - \pi/2 + \theta)\\ &+ E_s(z + d)\cos(\omega t + \pi/2 + \theta))\end{aligned} \tag{34}$$

where, $E_s(z)$ is the electric field of a standing wave, d is the length of a single cell, $\beta_0 = \omega/c$, and θ is the initial driven phase of the wave, which is the same as that in the entrance and exit standing wave fields. For backward traveling wave, the field on the axis can be written as:

$$
\begin{aligned}
E_t(z,t) &= \frac{1}{\sin(\beta_0 d)}(E_s(z)\cos(\omega t + \beta_0 d - \pi/2 + \theta) \\
&\quad + E_s(z-d)\cos(\omega t + \pi/2 + \theta))
\end{aligned}
\tag{35}
$$

8.8   Solenoid

In the IMPACT-T code, by using type code "3", the magnetic fields of solenoid B_r and B_z are read in from an external file containing discrete data on $r-z$ plane. By using type code "105", the magnetic fields of the solenoid can be obtained from the on-axis field data described by Fourier coefficients as

$$
\begin{aligned}
B_x &= -B_0'x/2 + B_0'''x(x^2+y^2)/16 \\
B_y &= -B_0'y/2 + B_0'''y(x^2+y^2)/16 \\
B_z &= B_0 - B_0''(x^2+y^2)/4
\end{aligned}
$$
$$\tag{36}$$
$$\tag{37}$$
$$\tag{38}$$

where B_0', B_0'', B_0''' are the first, the second and the third derivatives of field on axis.

8.9   Bending Magnet

The bending magnet in the IMPACT-T code is modeled as a constant vertical field region and two fringe field regions on both ends of the magnet. At the entrance of the bending magnet, a Cartesian coordinate is set. A schematic plot of the layout of a bending magnet in this local coordinate system is shown in Fig. 1. The pole faces of the bending magnet at the entrance and the exit are described by four linear equations:

$$
z = k_i x + b_i
\tag{39}
$$

where i is from 1 to 4 corresponding to the four edge lines defining the fringe field and center field region in the figure. For a bending magnet defined with entrance angle ϕ_1, bending angle θ, and exit angle ϕ_2, the above k_i and b_i can be obtained by:

$$
\begin{aligned}
b_1 &= 0 \\
b_2 &= \frac{\Delta}{\cos(\phi_1)} \\
b_3 &= \frac{\Delta}{2\cos(\phi_1)} + R\sin(\theta) + (R - R\cos(\theta))\tan(\theta - \phi_2) - \frac{\Delta}{2\cos(\theta - \phi_2)} \\
b_4 &= b_3 + \frac{\Delta}{\cos(\theta - \phi_2)} \\
k_1 &= -\tan(\phi_1) \\
k_2 &= k_1 \\
k_3 &= -\tan(\theta - \phi_2) \\
k_4 &= k_3
\end{aligned}
$$
$$\tag{40}$$
$$\tag{41}$$
$$\tag{42}$$
$$\tag{43}$$
$$\tag{44}$$
$$\tag{45}$$
$$\tag{46}$$
$$\tag{47}$$

where Δ is the width of the fringe field region and R is bending radius of the flat field region. Note: the sign of the entrance or exit angle needs to be checked.
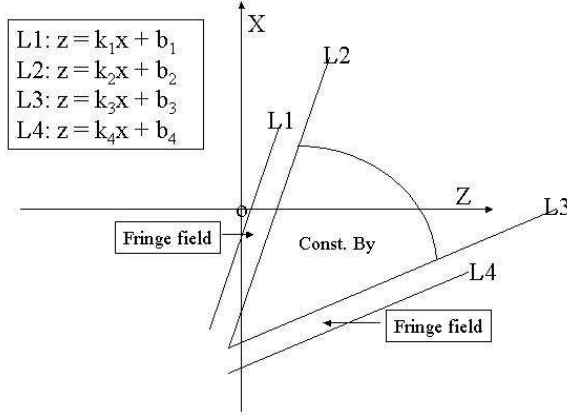
Figure 1: A schematic plot of the layout of a bending magnet in a local Cartesian coordinate system.

The mid-plane field inside the fringe region of exit is assumed to be an Enge function [14]:

$$B_y(0,0,z)/B_{y0} \quad = \quad \frac{1}{1 + exp(c_1 + c_2 s + c_3 s^2 + c_4 s^3 + c_5 s^4 + c_6 s^5 + c_7 s^6 + c_8 s^7)} \tag{48}$$

where $s = (z - z0)/g$, g is the full gap width, z is the distance measured perpendicular to the pole face L_3. Here, the shift in z_0 is the half of the input parameters 11 and 12 inside the rfdata file. The coefficients (absolute values) of the Enge function for the fringe field at the entrance are assumed to be the same as the exit ones except that the shift is controlled by the input parameter 11. The coefficients c_1 to c_8 are obtained from the least square fitting of the measured data profile.

A reference particle is defined with only initial longitudinal components from the centroid of the beam in the local coordinate system. Inside the bend, both simulated macroparticles and the reference particle are advanced using the dipole magnetic field. To calculate the space-charge forces or CSR forces, we transform the macroparticles coordinates into a coordinate system originating at the reference particle through rotation. A momentum kick is done inside the reference particle coordinate system and transformed back to the local coordinate system. When the beam is out of the bending magnet, all macroparticles are rotated again into the reference particle coordinate system. This gives the macroparticles local coordinates in the next beam line element. Here, we have assumed that the reference particle will move through the axis of the bending magnet and into the axis of the next element.

26

8.10 Quadrupole

The magnetic fields in the IMPACT-T code for constant quadrupole are:

$$B_x = B_g y \tag{49}$$
$$B_y = B_g x \tag{50}$$
$$B_z = 0 \tag{51}$$

where B_g is the constant field gradient. If the fringe field is included (see the input lattice description), the magnetic fields are

$$B_x = B_g y - B_g''(y^3 + 3x^2 y)/12 \tag{52}$$
$$B_y = B_g x - B_g''(x^3 + 3xy^2)/12 \tag{53}$$
$$B_z = B_g' xy \tag{54}$$

where B_g' and B_g'' are first and second derivatives of field gradient with respect to z. In the current version of the IMPACT-T code, the fringe field function is assumed to be

$$B_g(0,0,z)/B_{g0} = \frac{1}{1 + exp(c_1 + c_2 s)} \tag{55}$$

where $s = (z - z0)/g$, g is the diameter of aperture, the shift $z0 = (total\ length - effective\ length)/2$, and $c_1 = -0.00004$, $c_2 = 4.518219$ in current version. The values of c_1 and c_2 are hardwired into the code but can be modified according to the real quadrupole fringe field profile. For the current values of c_1 and c_2, the fringe field region is about $3g$. The effective length will be the total length subtracting the fringe field region width.

The skew quadrupole can be modeled with a rotation angle error of above quadrupole magnet.

8.11 Multipole

In the IMPACT-T code, the multipole element includes sextupole, octupole and decapole. For sextupole,

$$B_x = B_0 xy \tag{56}$$
$$B_y = B_0(x^2 - y^2)/2 \tag{57}$$

For octupole,

$$B_x = B_1(3x^2 y - y^3)/6 \tag{58}$$
$$B_y = B_1(x^3 - 3xy^2)/6 \tag{59}$$

For decapole,

$$B_x = B_2(x^3 y - xy^3)/6 \tag{60}$$
$$B_y = B_2(x^4 - 6x^2 y^2 + y^4)/24 \tag{61}$$

where B_0, B_1, and B_3 are input parameters.

9 Others

IMPACT-T has a sister program called IMPACT which also does tracking with space charge forces. The difference between the two is that the independent variable in IMPACT is the longitudinal position

z while IMPACT-T uses the time t as the independent variable. The advantage of IMPACT-T over IMPACT is that space charge forces need to be evaluated at constant t. This allows IMPACT-T to more accurately model the space charge forces. This difference between IMPACT-T and IMPACT is most noticeable at low energy. That is, with particles near the gun. When there are external fields (solenoid field, RF field, etc.) then potentially IMPACT-T has to do more work to find the external fields for each individual particle location. This makes it slower than the z-based IMPACT code. It should be noted that at the end of the calculation IMPACT-T stops at some time t so that the particles will have different z.

Acknowledgments

References

[1] J. Qiang, S. Lidia, R. D. Ryne, C. Limborg-Deprey, Phys. Rev. Special Topics - Accel. Beams 9, 044204, (2006).

[2] D. H. Dowell and J. F. Schmerge, Phys. Rev. Special Topics - Accel. Beams 12, 074201, (2009).

[3] K. L.F. Bane, "Short-Range Dipole Wakefields in Accelerating Structures for the NLC," SLAC-PUB-9663, 2003.

[4] P. Craievich, T. Weiland, I. Zagorodnov, "The short-range wakefields in the BTW accelerating structure of the ELETTRA linac," ST/M-04/02.

[5] J. Qiang, C. E. Mitchell, R. D. Ryne, NIM-A 682, 49, (2012).

[6] C. E. Mitchell, J. Qiang, R. D. Ryne, NIM-A 715, 119 (2013).

[7] E. L. Saldin, E. A. Schneidmiller, and M. V. Yurkov, Nucl. Instrum. Methods Phys. Res., Sect. A398, 373 (1997).

[8] M. Borland, Phys. Rev. Sepecial Topics - Accel. Beams 4, 070701 (2001).

[9] G. Stupakov and P. Emma, "CSR Wake for a Short Magnet in Ultrarelativistic Limit," SLAC-PUB-9242, 2002.

[10] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in FORTRAN*, Cambridge University Press, New York, 1992.

[11] I. Pogorelov, J. Qiang, R. D. Ryne, M. Venturini, A. Zholents, R. Warnock, "Recent Developments in IMPACT and Application to Future Light Sources," in Proceedings of 9th Intern. Comp. Accel. Physics Conf., p. 182, 2006.

[12] P. L. Morton, Particle Dynamics in Linear Accelerators, Ph. D. Thesis, Midwestern Universities Research Association, The Ohio State University, 1963.

[13] G. A. Loew, R. H. Miller, R. A. Early and K. L. Bane, SLAC-PUB-2295 (1979).

[14] H. A. Enge, Rev. of Sci. Instr. 35, 278, (1964).