WAKEFIELD MODELS FOR PARTICLE TRACKING CODES

A. Latina, G. Rumolo, D. Schulte (CERN, Geneva),
G. A. Blair, S. Malton (Royal Holloway, University of London, Surrey),
J.D.A. Smith (Lancaster University, Lancaster), R. J. Barlow, A. Bungau (UMAN, Manchester)

Abstract

Wakefields have a considerable effect on beam dynamics and they must not be neglected for emittance growth studies, background estimates and other problems. The codes used for these problems are normally not capable of self-consistent wakefield calculations. They should thus be extended with either analytical models or export the wakefields numerically evaluated with other codes (such as GdfidL[1]) when analytical models are not feasible. Both approaches are presented as well as their implementation in PLACET[2], MERLIN[3] and BDSIM[4]. The simulation results for the ILC and CLIC beam delivery systems are given as an example. Results produced with different codes are compared.

COLLIMATOR WAKEFIELDS IN PLACET

In PLACET, to account for the flat structure of a collimator, dipole and quadrupole components of the wake field are weighed by the Yokoya coefficients [5] at the moment when the wake field kick is applied to each bunch particle:

\[
W_{1x}(x, x_0; z) = W_1^\perp(z) \frac{\pi^2}{12} (x_0 + 0.5x)
\]

\[
W_{1y}(y, y_0; z) = W_1^\perp(z) \frac{\pi^2}{24} (y_0 - y)
\]

(1)

This is true only in the linear regime, that is for bunch particles that stay sufficiently close to the machine axis and far enough from the collimator walls. For halo particles close to the collimator jaws, the nonlinear components of the wake may become significant. The wakes scale as function of the coordinates of source and probe particles[6, 7] (at least in the plane where the gap is smaller) according to:

\[
W_{1x}(x, x_0, y, y_0; z) = W_1^\perp(z) \frac{\pi}{8g} \left[ x_0 \cos x_0 + \sin x_0 + \right.
\]

\[
\left. + \frac{\sin x_0 (x_0 \sin x_0 + y_0 \sin y_0)}{\cos y_0} \right] +
\]

\[
\frac{x_0 \cos x_0 + \sin x_0 - \sin x_0 (x_0 \sin x_0 + y_0 \sin y_0)}{(\cos y_0) \cos x_0}
\]

(2)

The first of the above equations reduces to:

\[
W_{1x}^\perp(x, x_0; z) =
\]

\[
= W_1^\perp(z) \frac{\pi}{8g} \left. \left( \frac{x_0 + \sin x_0}{1 + \cos x_0} - \frac{x_0 - \sin x_0}{1 - \cos x_0} \right) \right]
\]

(4)

if the beam lies vertically close to the plane \(y = 0\) such that \((y, y_0) \ll g\), or in general when \(y - y_0 \ll g\). The above formulæ are general and account for the near-wall effects. These formulae have been implemented in PLACET. A compromise between a requirement for low noise tracking and computational speed is certainly necessary for this type of calculations. For example, the non-oscillatory character of the wake allows for a coarse slicing of the bunch (even if the initial drop needs to be adequately resolved), which also translates into a lower number of macroparticles required to model the bunch. The difference between linear and nonlinear coefficients is shown in Fig. 1 for the “on axis” case \(y = y_0\), Eq. 4. At about half the gap size the wake becomes significantly nonlinear with the coordinates of the source and probe particles and increases more steeply going toward the chamber wall. Nevertheless, as \(y\) starts differing from \(y_0\) significantly in units of \(g\), the geometric cross terms in Eq. 2, which make the horizontal wake depend on the vertical coordinates and vice versa, cause a decrease of the effective wake values.

Figures 2 shows how the nonlinear wake field changes when going off axis: curves at \(y - y_0 = 0.5g\) (blue) and \(y - y_0 = 0.8g\) (red) are plotted, which clearly show how the effect of the wake, amplified on axis by the nonlinear contribution, becomes quickly weaker when moving away from the plane \(y = y_0\).

COLLIMATOR WAKEFIELDS IN MERLIN

The higher order wakefield modes are implemented into the MERLIN code as the they become important when the bunch passes through a collimator close to the collimator wedges. The collimator do not have an axial symmetry as the system is not characterized by resonances.

The formalism for incorporating intra-bunch wakefields into the particle tracking code MERLIN starts with several assumptions:

\[
D05 \text{ Code Developments and Simulation Techniques}
\]

D05 Code Developments and Simulation Techniques
The sums $C_m$ and $S_m$ depend only on the leading slice, $W_m$ depends only on the aperture geometry and $r$ and $\theta$ depend only on the trailing particle. Fig. 3 shows an example of high order mode wakefields calculated by MERLIN.

**PLACET—BDSIM INTERFACE**

Halo particles which are close to the walls of the beam pipe may be kicked by the collimator wakefields and interact with the beam pipe material, producing secondary particles. A single simulation code that implements both tracking and secondary particles generation, does not exist. On the one hand, codes designed to track single particles and their secondaries deriving from the interactions with the materials, not including intra-bunch interactions, exist (BDSIM); on the other hand, codes such as PLACET and MERLIN take into account collective effects but do not simulate the interactions of the particles with the walls of the beamline.

Combining the abilities of BDSIM and PLACET allows to achieve an accurate simulation of the secondary particles generation and their tracking in such delicate components as the collimators, taking into account the most relevant collective effects. One method to achieve this is under development. It consists in interfacing the PLACET tracking code and BDSIM, based on the following idea: the tracking is performed in parallel by both codes. BDSIM tracks core and halo particles, while PLACET tracks only the core particles. When BDSIM reaches an element where it is desired to include the effects of wakefields, BDSIM sends to PLACET the halo particles to perform the calculations of the wakefield kick; then, PLACET reports the resulting kicks to BDSIM.
kick angles for each particle back to BDSIM, which applies
the kicks and continues the tracking.

Although the exchange of particle information has been
successfully implemented, difficulties are currently being
experienced in the post-exchange tracking. The work is
still in progress.

IMPORTING NUMERICAL WAKEFIELDS

When analytical models are not feasible, the wakepoten-
tials should be calculated using numerical solvers, such
as GdfidL. The wakepotentials calculated with such codes
should thus be imported into the tracking codes, in order to
be used during the tracking.

Even exploiting the intrinsic symmetries of the wakepo-
tentials, the wakefunctions calculated using these solvers
may require a considerable amount of disk space if the full
information of a computation in a fine mesh is requested.
Depending on the collimator’s geometry and on the beam
size, they can even require several gigabytes of disk stor-
age per collimator. Such big files are not very practical and
they need to be compressed in order to be actually used into
tracking codes.

In order to compress the wakepotential files, a code was
written to read the output files produced by GdfidL and
perform the multipole expansion of the wakepotentials in
there, taking into account the symmetries of the collimator
to further reduce the number of coefficients that need to be
stored. In order to reproduce a typical longitudinal wake-
potential generated in a rectangular collimator, only few
modes are usually necessary and therefore the final output
file requires only a few kilobytes on disk. Fig. 4 show a
longitudinal wake potential calculated using GdfidL and its
multipole expansion up to the 5-th (decapole) mode, in the
region of convergence. For such expansion only 5 complex
numbers need to be stored on disk, instead of a grid of num-
bers mapping the potential in the whole collimator space as
the direct output of the solver is. To cover a whole three-
dimensional volume, the collimator is divided in slices and
the multipole expansion is performed slice by slice, storing
the results into a single output file.

Another code, that soon will be included into PLACET,
is already capable of to reading the multipole expansion
of a longitudinal wakepotential and restoring the informa-
tion in the whole three-dimensional volume of the collima-
tor. This code also calculates the transverse components of
the wakepotential using a numerical implementation of the
Panofsky-Wenzel theorem and finally returns the wakefield
class at any position in the volume.

Figure 4: Longitudinal component of the wake-potential gener-
ated by a test charge in a simple test geometry, as calculated using
GdfidL (green) and reconstructed via a multipole expansion to the
5th order (red). In this case, the files produced using GdfidL re-
quired 40 Mb of disk space, whereas the multipole coefficients
used less than 10 Kb.

CONCLUSIONS

The collimator wakefields might have a strong impact
on the luminosity of future linear colliders. Realistic sim-
ulations of the collimators in the ILC and CLIC BDS are
being pursued. Analytic calculations of the wakefields are
already implemented both in PLACET and in MERLIN.
Where analytical methods are not sufficiently accurate, nu-
merical calculations of the wakefields must be carried out
using solvers, such as GdfidL, and plugged into the track-
ing codes. A software package that expands in multipoles
and calculates the wakefield kick directly from the solver
output has been developed. Files for realistic ILC collima-
tors are already available[8] and it is anticipated that these
will shortly be used in tracking calculations of the BDS.

REFERENCES

http://www.desy.de/~merlin
EUROTeVReport-2006-014
[7] K. Bane, I. Irwin, T. Raubenheimer, in NLC ZDR, SLAC
the ILC Collimators”, this conference

D05 Code Developments and Simulation Techniques

Proceedings of PAC07, Albuquerque, New Mexico, USA
3380
1-4244-0917-9/07/$25.00 ©2007 IEEE