IMPROVEMENTS IN MODELING OF COLLECTIVE EFFECTS IN ELEGANT*

M. Borland, R. Lindberg, A. Xiao, ANL, Argonne, IL 60439, USA

Abstract

elegant has long had the ability to model collective effects, including beam-driven cavity modes, short-range wakes, coherent synchrotron radiation, and intrabeam scattering. Recently, we made improvements specifically targeting simulations that require multiple bunches in storage rings. The ability to simulate long-range, non-resonant wakes was added, which can be used, for example, to study the effect of the resistive wall wake on multibunch instabilities. We also improved the implementation of short-range and resonant wakes to make them more efficient for multibunch simulations. Finally, improvements in the parallel efficiency were made that allow taking advantage of larger parallel resources.

INTRODUCTION

elegant [1] has long had the ability to simulate collective effects, starting with the addition of impedances and rf cavity modes in the early 1990s, in support of the design of the APS Positron Accumulator Ring [2]. In the late 1990s, wakefields and coherent synchrotron radiation were added, primarily in response to the needs of free-electron laser projects, e.g., [3] These features were subsequently implemented in the parallel version, Pelegant, [4, 5]. More recently, intrabeam scattering [6, 7] and Touschek scattering [8, 9] simulation were added. To confidently design a next-generation accelerator, such as the multi-bend achromat (MBA) upgrade for the Advanced Photon Source (APS) [10, 11], further improvements were needed.

The primary factor driving the recent improvements was the need for a bunch-lengthening higher-harmonic cavity (HHC) in the new ring [12] and the need for high-fidelity simulations including a multi-bunch beam and the impedance of other structures in the ring [13, 14]. Although this could be done with earlier versions of elegant and Pelegant, improved performance was needed for simulations with many bunches. Part of the issue is that inclusion of the short-range impedance necessitates the use of a large number of simulation particles in each bunch, which coupled with a desire to simulate as many as 324 bunches, makes for challenging computations.

In addition, because of the narrow beam pipe in the upgraded machine, we need to simulate multi-turn, non-resonant wakes, e.g., resistive wall wakes. This capability is new to elegant and Pelegant, which previously could only simulate single-turn wakes. Multi-turn collective effects could previously only be simulated using resonant modes.

With these improvements came the need for enhanced diagnostics, specifically the need for more convenient bunch-by-bunch diagnostics.

WAKE AND RF CAVITY SIMULATION

Bunch Mode Simulations

In order to minimize changes to the structure of the code, the bunched-mode feature was added by making use of the existing particleID property of each particle. In particular, the user may declare that different blocks of particle ID values correspond to distinct bunches. This requires use of the sdds_beam command and is documented in the manual. In brief, the user may either provide the entire multi-bunch beam as an SDDS file, or else provide a file specifying the phase space for a single bunch and request that it be duplicated *n* times. In the latter case, the spacing of the duplicates should also be specified. In the former case, the beam can be generated by a previous elegant run using the bunched_beam to specify the bunch properties and the bunch_frequency parameter of the run_control command to specify the interval between bunches. For more complex bunch patterns, the manual provides detailed instructions on preparing a beam file.

Given a properly prepared beam file, bunched-mode simulation is automatic for FRFMODE, FTRFMODE, IBSCATTER, LRWAKE, RFMODE, WAKE, TRFMODE, TRWAKE, ZLONGIT, ZTRANSVERSE elements. It may be turned off for any element by setting the BUNCHED_BEAM_MODE parameter to 0. Bunched mode simulation is presently not implemented for CSRCSBEND, CSRDRIFT, and LSCDRIFT.

For the diagnostic elements WATCH and HISTOGRAM, bunch-by-bunch output is obtained using the START_PID and END_PID parameters to specify the range of particle ID values to include in the output. For large numbers of bunches, one may typically select only a few bunches for output, so the setup is not too tedious. Creation of a parameter file for use with load_parameters is a convenient way to configure large numbers of diagnostic elements.

In a parallel code, a natural choice for domain decomposition is that each bunch is handled by a single core. However, this is suboptimal as it forces processors to wait in line to process bunches through elements that have effects on following bunches (e.g., rf modes or long-range wakes). In addition, this choice limits the size of computational resources that can be used. In Pelegant, these issues are avoided because bunches are shared among processors, with all processors working together on each bunch in turn. One minor

* Work supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Contract No. DE-AC02-06CH11357.

ISBN 978-3-95450-168-7

pitfall is that the number of particles per bunch must be at least $C-1$, where C is the number of processor cores.

Short- and Long-range Wakes

The basic algorithms for short-range wakes (WAKE, TRWAKE) and impedances (ZLONGIT, ZTRANSVERSE) have not changed with the implementation of bunched mode. The only change is to identify the particles in each successive bunch and process them independently of the others. Hence, we will not describe it further here.

Long-range non-resonant wakes were implemented using a new element called LRWAKE. For each LRWAKE element, the user provides an SDDS file tabulating one or more of the wake functions $W_{x,y,z}(t)$. These wakes may extend over an arbitrary number of turns, with the user declaring how many turns to actually use as part of the element definition. The long-range wake is assumed to be constant within any single bunch, with the fine variation of the wake assumed to be computed using WAKE, ZLONGIT, etc.

For each bunch b, sums are performed over all prior bunches 0, 1, ..., b and passes $p : [0, P]$ to compute the voltage or kick. For the longitudinal wake, we have

$$V_z(b) = \sum_{i=0}^{b} \sum_{p=0}^{P} q_{i,p} W_z(\langle t_b \rangle - \langle t_{i,p} \rangle), \qquad (1)$$

where i is the index of the bunch and angle brackets indicate averages over particles in a bunch. The self-wake is included here for maximum flexibility, but in general should be excluded by setting the appropriate values in the wake tables to zero; this will avoid double-counting with the true short-range wake. For the horizontal wake we have

$$V_x(b) = \sum_{i=0}^{b} \sum_{p=0}^{P} q_{i,p} \langle x_{i,p} \rangle W_x(\langle t_b \rangle - \langle t_{i,p} \rangle) \frac{\beta_{x,S}}{\beta_{x,E}}, \qquad (2)$$

with the vertical wake being similar. The ratio $(\beta_{x,S}/\beta_{x,E})$ reflects the fact that the transverse wakefield kick scales with the local beta function, so that W_x should be multiplied by the ratio of the beta function at the wakefield source $\beta_{x,S}$ to that at the location of the WAKE element $\beta_{x,E}$.

Rf Cavity Modes

The algorithm for simulation of rf cavity modes using the RFMODE, TRFMODE, FRFMODE, and FTRFMODE elements is again unchanged for the bunched mode simulation, with the only change being the grouping of particles into successive bunches. The primary benefit in terms of performance is that the density histograms need not cover the empty region between the bunches, which was the primary pitfall of the prior version.

Performance Tests

A combined simulation for the APS multi-bend achromat (MBA) upgrade lattice [11] was used to test the overall parallel efficiency of many of the features just described. This simulation included the short-range longitudinal wake [14,15], the long-range longitudinal resistive wall wake, the main mode for the accelerating cavities, feedback for the accelerating cavities, and a passive harmonic cavity [13]. A 48-bunch beam was tracked with 100,000 particles per bunch, somewhat more than the number (about 30,000) needed to get accurate modeling of the longitudinal microwave instability. The number of turns was adjusted in proportion to the number of cores, in order to reduce the effect of start-up overhead on the runs with many cores and reduce the wall time for runs with few cores. The run statistics are analyzed in terms of time per turn.

Figure 1 shows the parallel efficiency for the simulations as a function of the number of cores. We see that the efficiency is greater than 1 for small numbers of cores, a reflection of the fact that the single-core simulation is resource-limited (presumably by memory). The efficiency is good up to 256 cores, but falls below 50% with 512 cores. If the number of particles is increased to 200,000, the efficiency is higher for larger numbers of cores, indicating that in some sense a 100k particle run, having a mere 196 particles per working core, is not large enough for 512 cores. Figure 2 shows another view of the same data, this time in terms of the wall time used per turn tracked. Again, for the larger problem (200k particles), a significant decrease in wall clock time is obtained even with 512 cores.

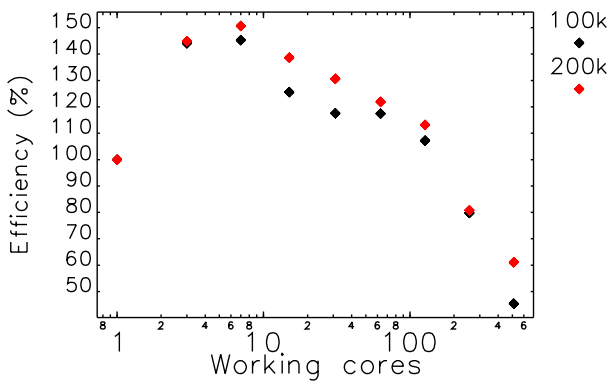

Figure 1: Parallel efficiency vs the number of working cores for collective effects simulation with 48 bunches, each having 100,000 or 200,000 particles.

INTRABEAM SCATTERING

Improvements were also made to the parallel implementation of a different type of collective effect, namely, intrabeam scattering (IBS). A tracking-based calculation of IBS is useful as a check of other calcuations and as part of a larger simulation including, say, impedances and rf modes. Previously, elegant could simulate IBS during tracking, but the code for storage rings assumed that the beam had a gaussian longitudinal shape [7]. Neither the transport-line or the ring code was parallelized. However, in both cases we may have a non-gaussian longitudinal density. For example, in rings, as illustrated in [13], when using a higher-

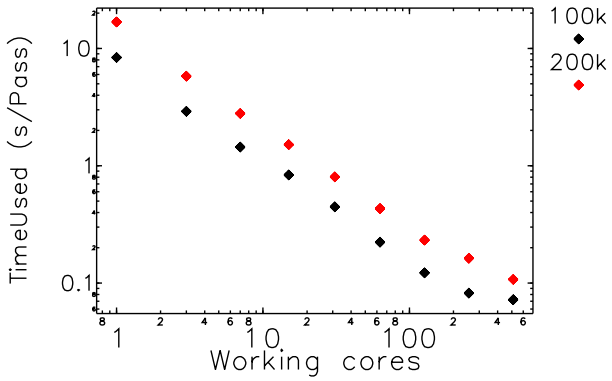

Figure 2: Wall clock time used per turn tracked vs the number of working cores for collective effects simulation with 48 bunches, each having 100,000 or 200,000 particles.

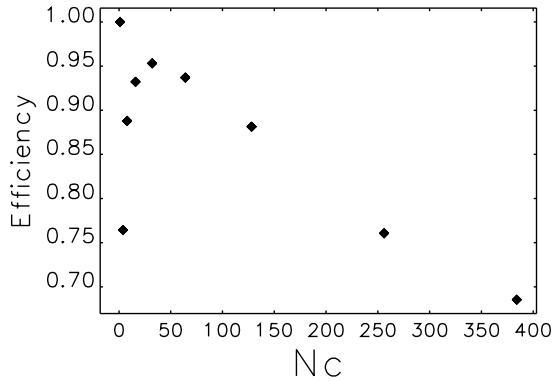

Figure 3: Parallel efficiency for tracking a single 10,000-particle bunch with IBS.

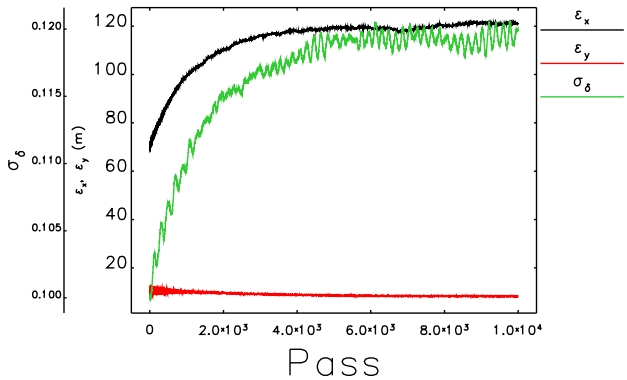

Figure 4: Time evolution of emittance and energy spread under the influence of IBS and synchrotron radiation, after injection into an APS MBA lattice.

harmonic cavity to lengthen the bunch, the bunch shapes may be considerably different from a gaussian. The latest version supports longitudinal slicing for IBS calculations in transport lines and storage rings, including multi-bunch beams in rings.

The theory is described in [16]. The parallel algorithm, executed in turn for each bunch, is as follows: First, we globally compute the minimum and maximum particle arrival time at the IBSCATTER element. This defines the time limits of each slice, allowing binning of particles by slice. For the particles in each slice, we compute emittances, energy spread, number of particles, and bunch duration by sharing partial sums across processors. Using these values, we compute the IBS growth rates for the just-completed turn. The required integrals involving lattice functions are done in parallel. Finally, we apply kicks or smooth beam parameter changes to the particles within the slice. For this work, we used the latter method, which simply inflates (or deflates) the bunch volume uniformly rather than adding random kicks; this method produces less noisy results.

We tested the algorithm using an earlier version of the APS MBA upgrade lattice [11]. IBS effects are significant if the bunch is not lengthened and the emittance ratio is small. To test the parallel efficiency, we tracked with five bunches of 10,000 particles for 20 turns, using 10 slices. In addition to a single IBS element per turn, the full lattice was tracked including element-by-element synchrotron radiation. As Fig. 3 shows, the parallel efficiency is good for up to 640 cores, the largest available on the system used. The efficiency for few cores is greater than 1 because the parallelization of the lattice function integrals is very effective and involves very little interprocess communication.

Figure 4 gives an example of the evolution of beam parameters as computed by the updated code. The horizontal emittance and energy spread growth are large owing to the small vertical emittance.

CONCLUSIONS AND PLANS

We have described some recent improvements to simulation of collective effects with `elegant` and `Pelegant`. The improvements allow for faster simulation of bunch-by-bunch short-range wakes and rf cavity effects. Simuilation of long-range non-resonant wakes is also possible. In addition, a new parallel algorithm for intrabeam scattering was implemented that allows simulation with multi-bunch beams with non-gaussian beam distributions. The parallel performance of these algorithms is good for 256 cores and beyond, depending on problem size.

Future efforts will include extending transverse feedback modeling to include multiple bunches and adding longitudinal feedback.

ACKNOWLEDGMENTS

Many of these simulations made use of the Blues cluster at Argonne's Laboratory Computing Resources Center.

ISBN 978-3-95450-168-7

REFERENCES

[1] M. Borland. ANL/APS LS-287, Advanced Photon Source (2000).

[2] M. Borland. *PAC 1995*, 287 (1996).

[3] M. Borland et al. *PAC 2001*, 2707 (2001).

[4] Y. Wang et al. *AIP Conf Proc*, 877:241 (2006).

[5] Y. Wang et al. *Proc. PAC07*, 3444–3446 (2007).

[6] A. Xiao. *Proc. of Linac 2008*, 296–298 (2008).

[7] A. Xiao et al. *Proc. of PAC 2009*, 3281–3283 (2009).

[8] A. Xiao et al. *Proc. of PAC 2007*, 3453–3455 (2007).

[9] A. Xiao et al. *Phys Rev ST Accel Beams*, 13:074201 (2010).

[10] G. Decker. *SRN*, 27 (2014).

[11] M. Borland et al. *TUPJE063, IPAC 2015, these proc.*

[12] M. Kelly et al. *WEPTY008, IPAC 2015, these proc.*

[13] M. Borland et al. *MOPMA007, IPAC 2015, these proc.*

[14] R. R. Lindberg et al. *TUPJE078, IPAC 2015, these proc.*

[15] R. R. Lindberg et al. *TUPJE077, IPAC 2015, these proc.*

[16] A. Xiao et al. *MOPMA012, IPAC 2015, these proc.*