Benchmarking the Merlin and CHEF simulation codes for ILC Low Emittance Transport

Paul Lebrun

February 15, 2007

Abstract

Relevant differences between the CHEF and Merlin simulation codes for Low Emittances transport (LET) of ILC positrons bunches are documented. Significant discrepancies on a previous LET benchmark are explained. The current systematic error on the projected vertical emittance in the Tesla style lattice is about one nano-Radiant. Finally, suggestions for code upgrades are made.

At the Darebury meeting, Jeff Smith presented the status of LET code benchmarking work[1]. A corresponding Fermilab/Cornell technical note has been drafted[2]. In his talk, Jeff included the CHEF results on Benchmark2, and to quote him and other participants, " CHEF really stands out" compared to the other 5 codes. In the note, no CHEF results were included, as CHEF and Lucretia[1] were being crosschecked. The work on Lucretia benchmarking was relatively easy and quickly done, in part because Lucretia inherited a lot of the from a previously adopted code, MatLIAR). For reasons explained below, the work on CHEF is a bit more involved.

The present note summarizes the salient differences between one the 5 adopted codes, Merlin, and CHEF. First, results shown at Daresbury are reported here, as well as the outcome if the (tedious!) work of bringing the Merlin bunch propagators into the CHEF applications. Once that was done,

[1] The new code for LET transport written at SLAC by Peter Tennenbaum

no significant difference in emittance growth were found. The bulk of this note consists of a list various approximations made in either Merlin or CHEF, along their impact on this LET benchmark. A list of suggestions for further improvements on CHEF is given at the end.

At the Daresbury, and documented in the Beam's Division database[3], the emittance growth on the Tesla Lattice, misaligned and steered, was found to be "different" from the other codes. The comparison of the projected emittance, Merlin/CHEF for the entire LINAC is shown on figure 1. While the emittance growth at $S \approx 800$ m. differs by about a factor 2, the averaged relative difference are much smaller, of the order of one or two nanometer Radiant. The normal emittances agree much better. A few month ago, at Fermilab, the judgment call on these curves was "good enough for now". Yet, the consensus at Daresbury was that it might be worth investigating a bit further the difference in the projected emittance.

In the previous note[3], a significant difference in the vertical emittance growth was observed when the intercept of the cavity rotation was moved along the beamline. In that calculation, the short range wakefield was applied at the end of the transport through the cavity. In other codes, Merlin in particular, the wakefield is applied at the middle of the cavity. However, as shown below, even in the absence of this short range wakefield, the emittance growth calculated using CHEF differs from the one estimated using Merlin. Thus, these difference are sorted out first, the issue of the wakefield will be treated later.

Until specified, we therefore run without wakefield. Since the most significant differences occurs in the upstream section of the Linac, most of the work has been done for the first 1500 meter of the Linac. The new comparison between the CHEF package, version 2006_12_19, with "duly modified" propagators for Drift, Quadrupole and R.F. cavities, and the unchanged Merlin code is shown on figure 2. Over 1.5 km, the difference in the projected, non-canonical vertical emittance growth between this version of CHEF and Merlin is less than a few picometerRadians. Over the entire length of the Tesla Linac, it is about 15 picometerRadians. The note is about how this last plot was obtained, and related consequence on accuracy of this type of calculations.

Details on this benchmark, as well Merlin and CHEF references, are available in the technical notes referred above. Different techniques have been used to study the differences between these two packages. As both packages

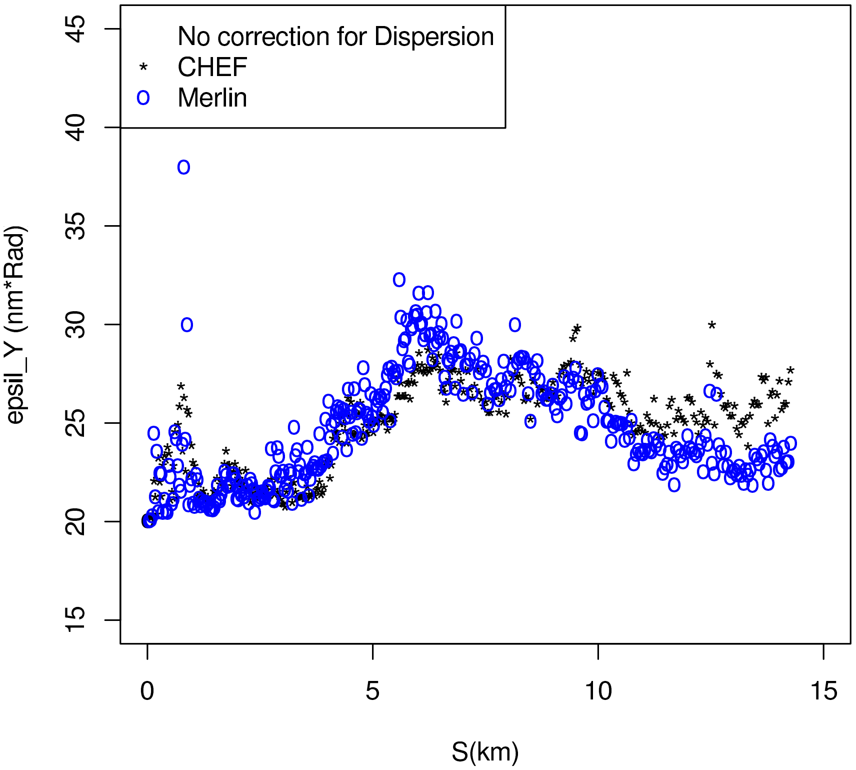

Figure 1: Invariant, Projected, transverse Emittances ε_y versus distance for the Benchmark2 problem, using the dipole corrector settings found in the file “nick23p4_misxy_ycor_1.txt”, distributed in February 2006.

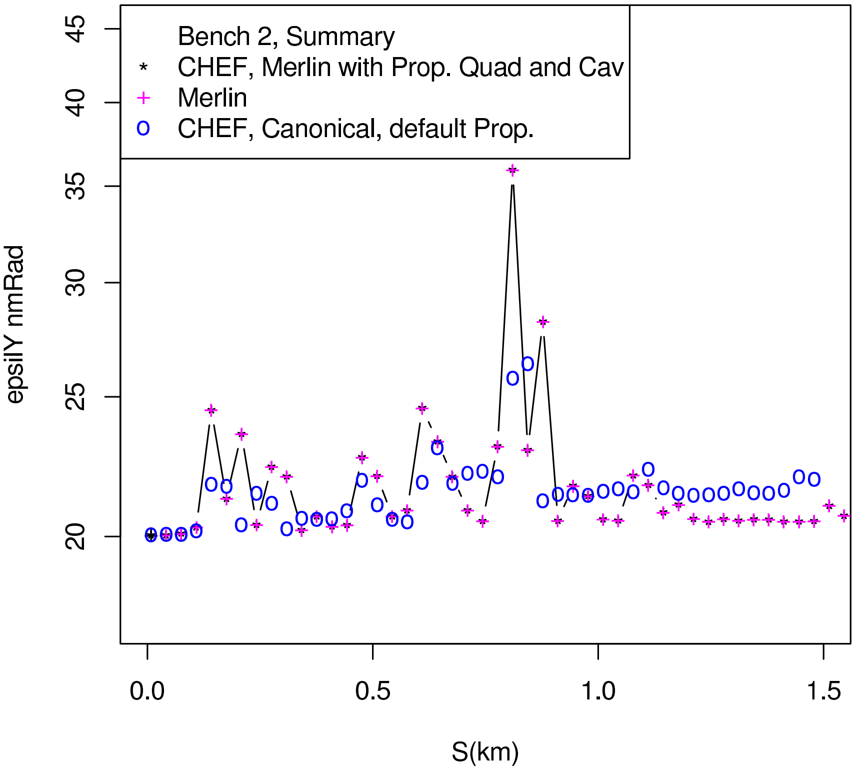

Figure 2: Projected Transverse Emittances using the current version of CHEF (version 2006_12_19), Merlin v3.1, and the duly modified CHEF package. These last two sets of data on top of each others on this graph, as they differ 2 picometer Radians, 10^{-4}, relative.

are written in C++, it is possible to exchange code fragments between these frameworks without translation, and compare results. However, this is easier said than done, because of several basic differences, or features:

1. Choice of the 6D variable representations: Merlin is an "optical" code, the basic 6D coordinates are $x, x', y, y', \delta z, \delta P/P_0$, where $x' = P_x/P_z$, $y' = P_y/P_z$ and P_x, P_y, P_z are the 3 component of the particle's momenta P, and are canonical with respect to the spatial coordinate x, y, z. By contrast, CHEF can be considered a "canonical" code, as it propagate in the 6D space-time position and momenta, $x, y, \delta z, P_y/P_0, P_y/P_0, \delta P/P_0$, where P_0 is the reference momenta, $P = |\vec{P}|$. Only the last coordinate, the momentum difference of the particle with respect to the reference particle, are identical. While the re-ordering is trivial, the change of transverse momentum coordinate impacts the results:

   $$y' = P_y/P_z \approx P_y/P_0/\sqrt{(1.0 + \delta P/P_0)^2 - y'^2 - x'^2}$$

   While y'^2 are usually small $(y' \approx 10^{-6})^2$ to be ignored (but included in our calculation!), the $(1.0 + \delta P/P_0$ factor can not be neglected. Preservation of the emittance at Main Linac injection (or the RTML) is mostly about Dispersion and/or chromatic effects. Codes deal with this factor differently in each propagators, given the choice of the variables. Code to go back and forth between between the two representation has to be written.

2. Frames: Both codes express the particle 6D coordinates with respect to a nominal reference particle. However, frame transformations must occur while propagating the bunch in a misaligned machine. The CHEF and Merlin authors took different approaches. In Merlin, the frame transformation occurs entering and leaving a given misaligned element. In CHEF, the transformation must occur in a "slot" located between a perfectly aligned drift and the misaligned element in which the particle leaves or enter. Such slots are created from drifts by the CHEF

[2] Careful! After a cavity rotation of one miliRadiant, $y'^2 \approx 10^{-6}$ once projected on the electrical field. While global angles are indeed small local angles can be - unfortunately some three orders of magnitude higher.

system automatically upon commissioning (or "registering") the beamline. In our case, each drift separating the quadrupole package from the accelerator sections, and the the drifts between cavities, have been decomposed in three separate drifts, de-facto simulating the presence of bellows between misaligned elements. Thus, CHEF handles misalignment as true 3D displacement rotation, whitout approximation.

Furthermore, CHEF, when implementing these slots, has to change the incremental path length S through the beam line. Note that S now depends on the geometry of the slots with respect to the unperturbed drifts. This means that beam line of the same type/name can no longer be keyed unambiguously based on their location along the beam line. The MAD parser also does not unambiguously generate a list of such elements. This makes debugging the system a bit inconvenient.

3. Although both packages are "frameworks" for beam physics simulation, none of them really fully support a user-defined beam element in a consistent and well documented way. When it came to re-implement the Merlin propagator in the CHEF context, a kludge version of the relevant CHEF propagators had to be re-written, starting from a set of beam element virtual classes, with formal interface made "user-public" would have been perhaps better.[3] Note also that, for "easy" comparison of the performance with the tilted/displaced r.f. cavities, using the Merlin propagator, the coordinate transform had to be applied exiting the slots, instead of entering the cavity.

4. Linear Algebra. Both packages evidently are using vectors and matrices. None of them are based on the C++ STL classes, mainly for historical reasons: these low-level utilities were implemented before STL/C++ became a standard. Thus, a *Vector* in CHEF is sufficiently different from a Merlin *PSVector* to force a code translation. The basic Linear Algebra algorithm are nevertheless very similar, as the TLAS package in Merlin is nothing but a wrapper arounf the well known Numerical Recipes, also used in CHEF.

Thus, these different features interfere with each other, making the debugging of the "duly modified" CHEF code a bit tedious. To avoid any

[3] Or may be this reflect of the ignorance of the present author of what the true capabilities of CHEF are...

confusion on the initial state of the bunch (6D Gaussian, matched to Twiss parameter of the beam line), Merlin write a flat list of 6D vectors, that are translated into the CHEF coordinate system, in the CHEF framework. Facilities to compare individual coordinates, in the Merlin coordinates system had to be written.

We now list the relevant differences between CHEF and Merlin are, in terms of Beam Physics. That is, the term "duly modified" used above is explained in detail.

1. On Invariant, Projected, Canonical, Vertical emittances.

   While the last adjective is non-controversial, care must be applied on all the other terms. The first term refers to Lorentz invariance: as $< y.P_y >$ is the invariant, not $< y.P_y/P_0 >$, one must multiply by a gamma factor the surface area on the vertical phase space $y, P_y/P_0$. In both codes, the gamma factor is computed based on the reference momentum of the bunch at the location where the emittance is quoted. The second term also has the same meaning in both codes: the 6D phase space is simply projected onto the vertical phase space, without any rotation along the other 4 eigenvector. In this case the large emittance growth moving from one quadrupole package to the next seen on figure 2 is due to uncorrected Dispersion, leading to large correlation between $\delta P/P$ and y or y'. Since the Liouville theorem only applies to the 6D phase space, this growth is mostly "apparent.

   The third term, "canonical" refers in this context to the fact that the emittance quoted in our document is based on the approximate transverse phase space, y, y', not the "canonical" [4] one, based on $y, P_y/P_0$. Since the apparent growth is strongly related to correlation with $\delta P/P$, based the above kinematic equation, this make a difference. The non-canonical and canonical emittances for the un-modified CHEF package applied to this benchmark is shown on figure 3. While the non-canonical is closer to the experiment, angles being more directly measured than transverse momentum, the canonical one is a bit more, well, canonical, i.e "correct". But this is indeed academical and irrelevant, provided quantities are clearly defined.

[4] as in Classical Mechanics

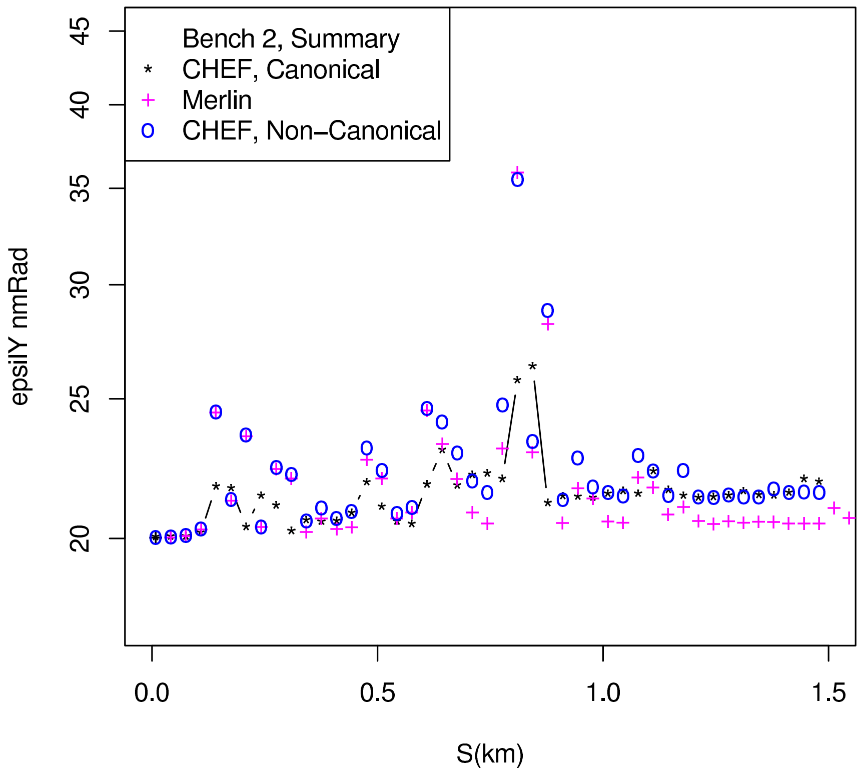

Figure 3: Projected Transverse Emittances using the current version of CHEF (version 2006_12_19). Canonical and non-canonical.

While this explains most of the discrepancy, a one to two nmRadians difference at ≈ 1.5 km. Note that most of the parasitic Dispersion introduced at $S \approx 800.$ meters has been compensated, and Merlin obtains a near perfect emittance preservation. While in absolute term the emittance growth is small, the relative emittance growth between the two packages differs by almost ≈ 100%. Since we do not a-priori know the scaling properties of this difference, in regard to quad displacements or cavity tilts, it is prudent to keep studying other difference.

2. Drifts and Particle velocities

Drifts are either explicitly defined in the lattice file, to simulate region where no active accelerator components are implemented, (bellows or pipes), or implicitly to model magnets or cavities, as a way to do implement efficiently the particle propagator via Gaussian Quadrature integration. As such, they are ominous. The optical transport or rays through them is obvious. However, the longitudinal part of the propagator differ between the two packages. Merlin assumes the particle are ultra-relativistic, $\beta = v/c = 1$, while CHEF takes into account the proper *beta* factor. For a relativistic particle, $\beta - 1 \approx 0.5/\gamma^2$. This is a negligible correction for the problem at hand: as we start at 5 GeV/c, $\beta - 1 \approx 10^{-8}$, and quickly accelerate. To first order, the registration process set the synchronous phase for the klystron, and for moderate distance and momentum spread, the ultra-relativistic approximation is valid.

To illustrate this, in a variant of our benchmark, we have kept the machine perfectly aligned, turned off the dipole corrector, and turned all the klystrons for the first 13 km. We then accelerate the beam at nominal gradient up to the end of the Tesla Linac. In one case, the CHEF application runs with real particle velocities. In the other case, all particles are moving at the speed of light, i.e., the $c\,\tau$ distribution remains unchanged. The $\sigma c\ \tau$ changes by less than 0.02 μ meter, just a few mRad at 1.3 GHz, and the $\delta P/P$ by less than 1.0^{-5}.

However, if we plan to simulate the ILC positron or electron source, the dark current in the Linac, this approximation will evidently be no longer valid.

3. The quadrupole Propagator

The numerical integration of the equation of motion in a magnet, or an r.f. cavity is simply too CPU intensive to be considered. Both packages use a non-linear model to propagate the particles through a quadrupole. In the case of Merlin, a non-linear transfer map is used. In the case of CHEF, the TPOT algorithm [5] is used. A series of successive thin quad inter spaced with judiciously placed drifts replaces each quadrupole of finite length. By default, CHEF runs with 4 thin quads per 66.6 cm long quadrupole. The Merlin quadrupole transfer map has been installed in CHEF, and the non-canonical, projected,

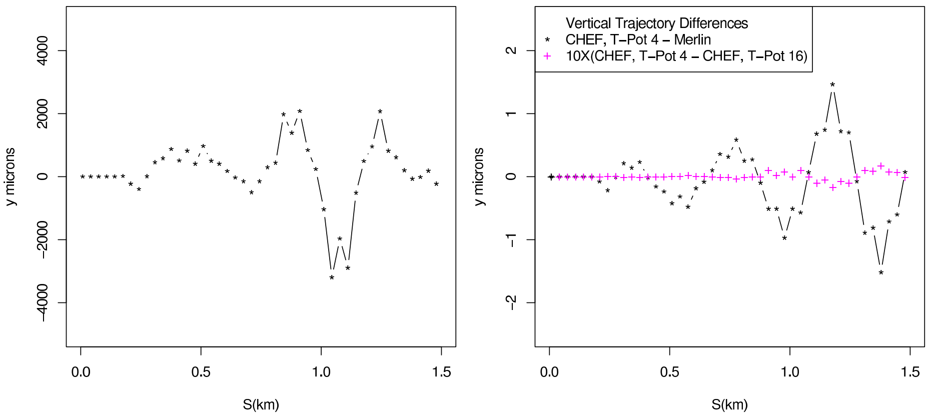

Figure 4: Left: Vertical position of the bunch with respect to the straight, centered beamline for the misaligned Tesla Linac, using the CHEF package. Right: Difference in these positions running with the Merlin Quadrupole propagator in CHEF, versus the default TPOT quadrupole propagator.

vertical emittance changed by less 0.01 nm. Radiant after 1.5 km.

Although the quadrupole propagator is not the source of the remaining discrepancy shown on figure 2, let us document the difference in these propagators a bit more quantitatively. In this benchmark, the maximum deviation in y is less than 500 microns, and the angles are typically less than 10 μ Radians. If the same machine is mis-aligned but not "steered", i.e., all dipole corrector currents are set to zero, the trajectory deviations from the straight line are larger, up to 3 mm. In this case, the trajectory of the bunch centroid recorded in CHEF using the Merlin Quadrupole map differ from the one computed using the TPOT approximation in CHEF by about one micron. See figure 4. The emittance growth through this machine is shown on figure 5, as well as the relative differences. The differences in trajectory or emittance between the TPOT approximation running with 4 thin quads or 16 thin quads are even smaller.

4. The Thin dipole Propagator

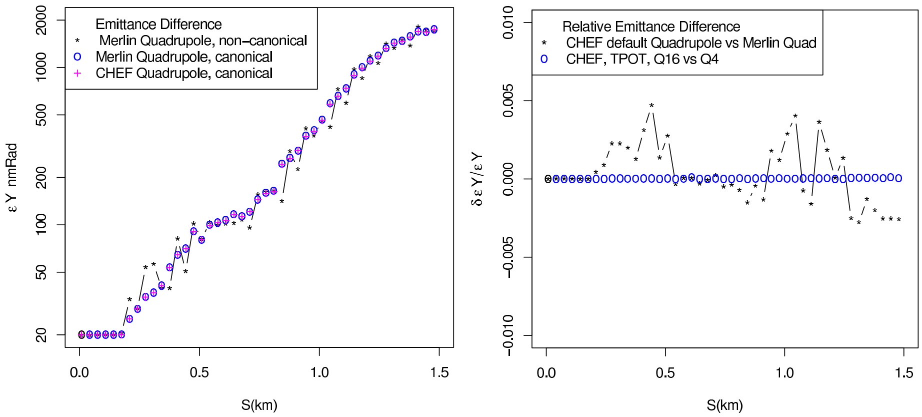

Figure 5: Left: The projected, vertical emittance Right: Relative Difference in the projected, non-canonical, vertical emittance running with the Merlin Quadrupole propagator in CHEF, versus the default, TPOT- quadrupole propagator with 4 and 16 thin quads.

In both package, the dipole corrector are approximated by thin dipole, just a simple vertical kick, based on the actual longitudinal momentum of the particle. No difference were found.

5. The Radio Frequency Traveling Wave Cavity Propagator

It has been verified that the intercept of axis of rotation of the cavities with the beam line is located at the center of the cavity in the Merlin framework. Hard-typed. IN CHEF, it is up to the user to locate this axis. For consistency, the axis of rotation is now located at the center of the cavity. Both CHEF and Merlin support traveling wave r.f. cavities with edge focusing. The deflection upon entering or leaving the cavity is given by:

$$\delta y' = \pm 0.5 * y * E_z * / E$$

where $\delta y'$ is the vertical deflection, y the position entering or leaving the cavity, E_z the electric field strength and E is the energy of the particle. The deflection is negative entering the cavity, positive leaving the cavity. There is a net deflection, because the the particle gains energy in the cavity. In Merlin, the correction is applied as

$$y' = P_y/P_z \rightarrow y' = P_y/P_z + \delta y' * 1./(1.\delta P/P)$$

In CHEF, we have:

$$y' = P_y/P_0 \rightarrow y' = P_y/P_0 + \delta y'$$

That is, $P_z \approx |\vec{P}|$. There is an other subtlety: in CHEF, E_z is computed based on the nominal accelerating gradient and the synchronous phase ϕ_s[5]. In Merlin, this field is correctly computed based on the actual kinematics of the particle:

$$E_z = gcos(\phi_s \; - \; k * c\tau)$$

where g is the accelerating gradient, k the wave number $c\,\tau$ is the position of the particle with respect to the reference particle of the bunch. That is, in CHEF, the edge focusing kick is correct only if $c\tau = 0.$. This is clearly an oversight in the CHEF code. This correction does make a difference. For a particle at 1 mm. from the center of the bunch along the Linac axis, the relative difference on this field is only 2.10^-3, for a cavity running 27 degree of crest. The difference in kick is of the order of 2 nano Radiant, for a gradient of 23 MeV/m, at 7 GeV and an offset position of $600\,\mu$ m. Since the y', in the reference frame of the nominal beam line is of the order of $\approx 0.25\mu$ Radiant, this make a fraction of one percent, or smaller, difference, per misaligned cavity. Yet, this adds up. A scatter plot of the $y' = P_y/P_z$ difference, with vs without this correction, versus $c\,\tau$ is shown on figure 6. The corresponding difference in the non-canonical, projected emittances is shown on figure 7.

The request to implement this correction in CHEF has been made. Yet, we are not quite done, are $\approx 0.25nm$ Radians to account for, at S = 1.5 km. This time, we suspect that the CHEF propagator is slightly

[5] To make thing just a bit more fun, there is 90 degree difference in the definition of this quantity.

more accurate than the Merlin one. In Merlin the actual integration is one step calculation. In CHEF, the physical cavity is split into three section: a drift of 1/2 the length of the physical cavity, followed by a thin cavity, where the change in $\vec{P}$ occurs, followed by a second drift of 1/2 length. Applying the kick at the center of the beamline element is know to reduce an error that scales for y (y') like $1/4 * \delta_{acc}E/E * y' * l$ ($1/8. * \delta_{acc}E/E * y'^2$)[6], respectively; where $\delta_{acc}E$ is the energy gain in the cavity, $y' = P_y/P_Z$ in the reference frame of the cavity and l is the length of the cavity. Figure 8

A complete implementation of the Merlin "LCAV" traveling wave integrator in CHEF has been done, and leads to the result shown on figure 2. However, neither the default CHEF cavity propagator nor the Merlin one was found adequate if a precision of a fraction of nm. Radians is expected. The plan is to run with the Merlin style integrator, with edge focusing kick corrected. In addition, a thorough study of a genuine fine grain step numerical integration of the equation of motion in these rotated, and/or imperfect cavities is worthwhile, as non-uniformity of the field are currently not taken into account.

6. The Short Range Wakefield

   The algorithms used to simulate the effect of the short range wakefield are identical in CHEF and Merlin. The implementation differs a bit, because of the differences listed above. In previous work[3], this wake was applied on the bunch after this bunch had been propagated through the cavity, in CHEF, while is applied in the middle of the propagation in Merlin. The CHEF "rfcavity" beamline element has recently been modified to allow the application of the wake in the middle of the propagation. This is just prototype code, a better version by CHEF authors is in the works. Meanwhile, the non-canonical, projected, vertical emittance with wakefield in CHEF and Merlin are shown on figures 9 and 10

   While the agreement is acceptable around S=1.5 km, significant differences remains at the end of the Linac. Note that the current uncertainty on the rf cavity integrator, documented on figure 8 in absence

[6] The numerical coefficient have been empirically determined, comparing CHEF with Merlin. These relation are error estimates, not the result of an exact analytical calculation

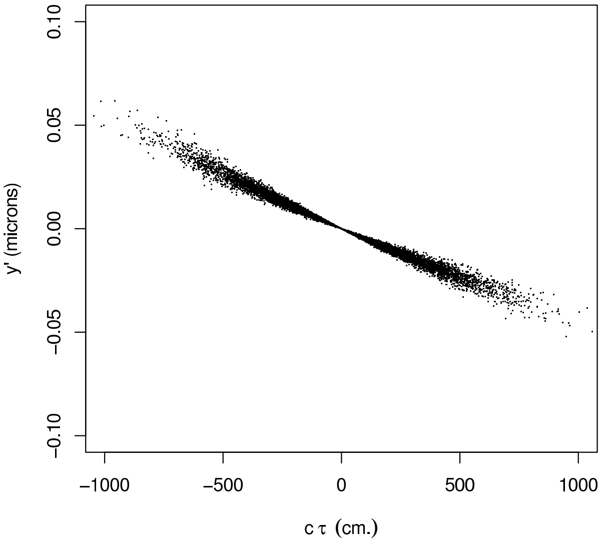

Figure 6: Illustration of the effect of correcting the edge focusing kick in CHEF. Shown here is the correlation between $\Delta y' = (P_y/P_z)_{corrected} - (P_y/P_z)_{default}$ and $c\,\tau$, at $S = 944m.$.

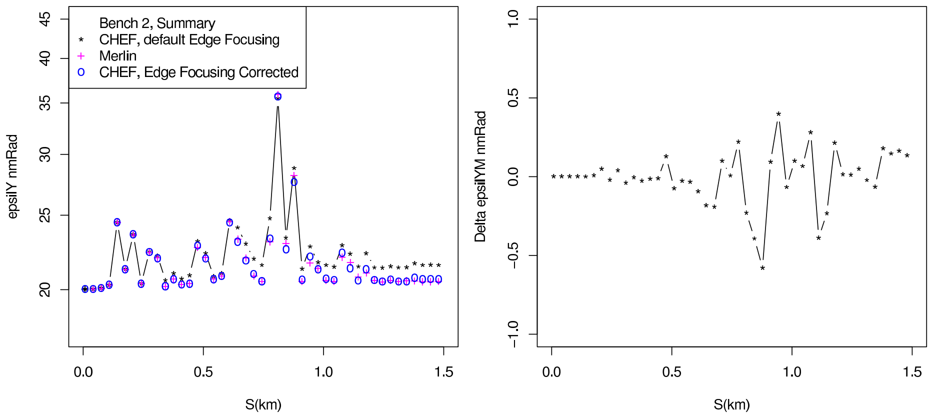

Figure 7: Left: The non-canonical, projected, vertical emittance with and with the correction on edge focusing kicks. Right: The same data sets, showing the difference in this emittance, running CHEF with and without this correction.

of wake fields, plays a significant role over long distance, with wakes. Thus, this calls for a more accurate description of the transfer maps, or numerical integration, through tilted and displaced r.f. cavities. Furthermore, the wake potential used in this calculation is computed for one or 3 consecutive, perfectly aligned, cryo-modules, and a bunch displaced with respect to the center of these cavities. Instead, we have to consider multiple cavities displaced and rotated with respect to each others, with a displaced bunch going through them. The wake potential function might be different than the simpler case where all cavities are on axis.

Let us now list the requests for modification or upgrade to be made to the CHEF package. We list them in order of priority.

- The edge focusing kick needs to be corrected. This is expected to have a serious impact on the RTML, where the synchronous phase is off-crest[7]

[7] Done, as of Feb 2, in CVS repository

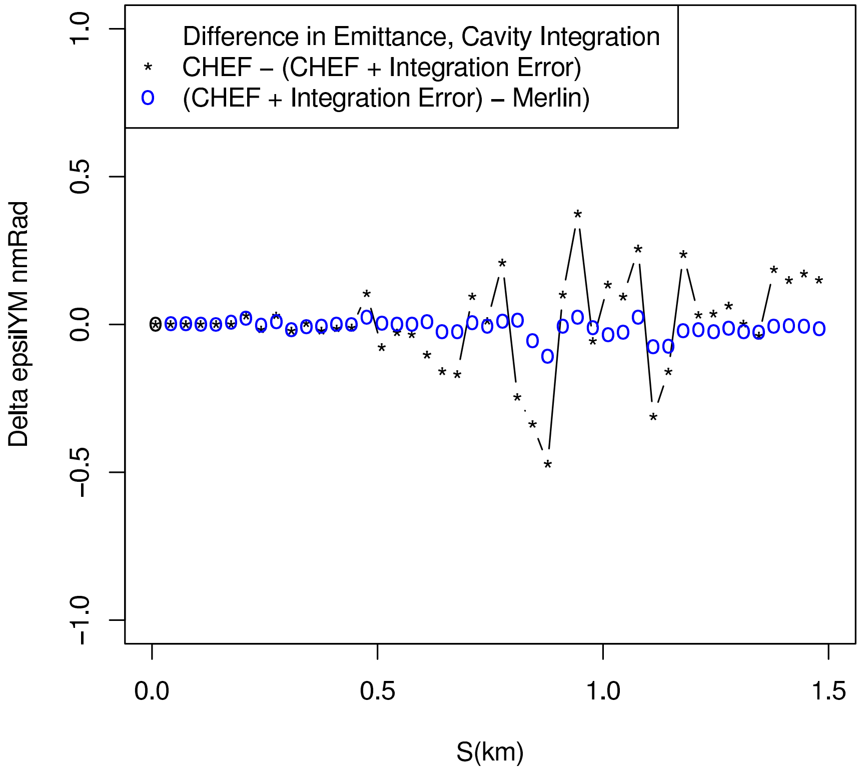

Figure 8: Quantitative illustration of the single step integration approximation . Shown here is the difference between the non-canonical, projected vertical emittance in CHEF running with vs without an additional integration error, and between this calculation and the default Merlin result.

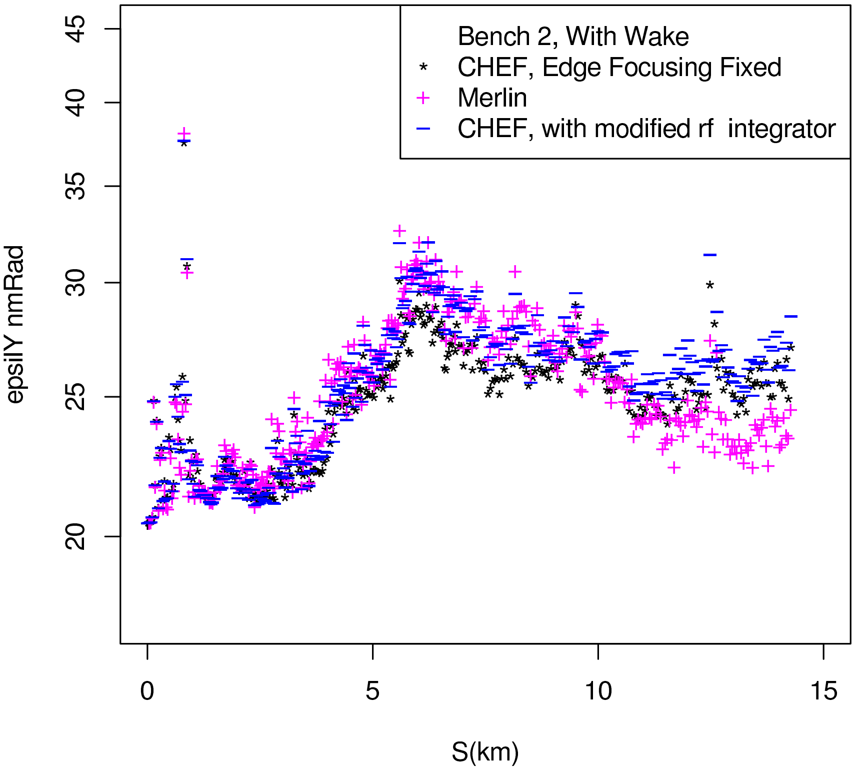

Figure 9: Non-Canonical, projected vertical emittance versus S, for CHEF with the r.f. edge focusing correction applied. Also shown is the Merlin v3.1 performance. The third set is CHEF, with edge focusing corrected an additional cavity integration "error" term included.

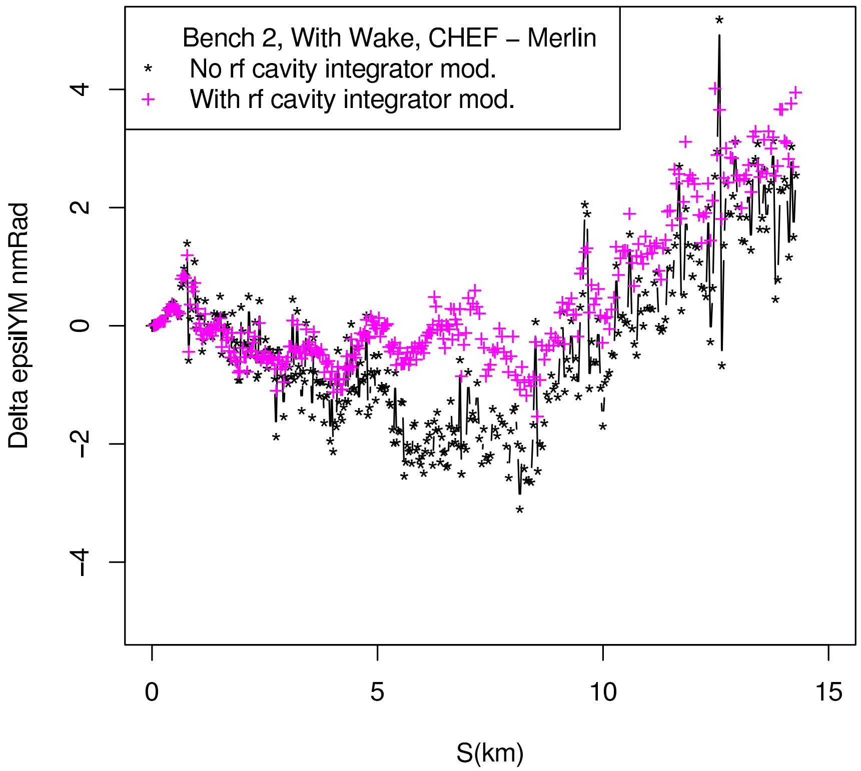

Figure 10: Difference between the Non-Canonical, projected vertical emittance estimated by Merlin and CHEF. Data sets are identical to the one shown on the previous figure.

- Implement the Particle::setTag and ParticlegetTag methods.

- The Short Range Wake field needs to be incorporate as a CHEF beamline element that can be inserted in rfcavity element, at an arbitrary location.

- The rfcavity element should allow for more successive drift followed by "thinrfcavity", to improve the accuracy of the integration of equation of motion. If the Wakefield has to be applied in the middle, it is reasonable to have at least two such thinrfcavities per real cavity. However, it would be good to very that LET is not overly sensitive the number of steps in this propagator.

- A well documented example of user-defined beamline element, with implementation of all bmlnElmnt virtual methods. Clearly, advanced user will need to implement their own thing, soon or later, like BPM cavities that have their own sets of wakefield, i.e., not really a cavity, and not really a monitor. These extensions were always part of the intentions of the CHEF authors.

- The beamline element names are poorly used in the context of the MAD8Factory: the names refer generically to an ensemble, or a class of elements. For debugging purposes, it would be really nice to be able to address a particular element via an un-ambiguous name.

- A Merlin style Quadrupole propagator, which is possibly faster than the TPOT propagator. With user-accessible method to easily switch from one model to the other (i.e., without bringing the CHEF core code in the user working directory (May be this can already be done, if so, apologies!)

- A method to return an "optical" representation of the state of a particle (i.e., $y' = P_y/P_z$ instead of $y' = P_y/P_0$

As one will have to investigate more accurate integration of motion through r.f. cavities, with wake field, short and/or long ranges, user-defined beamline elements will become critical. This includes the capability of insertion into an existing beamline and specialized "localPropagate" methods.

None of this work would have been possible without the dedicated support of the CHEF team (Leo Michellotti, Francois Ostiguy and Lynn Garren). Having access to the Merlin code was also extremely useful. Moving on!....

References

[1] Jeffrey C. Smith *http://ilcagenda.linearcollider.org/contributionDisplay.py?contribId=92&sessionId=25&confId=1265*

[2] Jeffrey C. Smith *et al* FERMILAB-TM-2373-AD

[3] Paul L. G. Lebrun *Studies of the robustness of the DFS algorithms*, Accelerator Division Document Database # 2589-v1, at *http://beamdocs.fnal.gov/AD-public/DocDB/ShowDocument?docid=2589&version=2*

[4] CHEF Documentation. Write-up in progress. Leo Michelotti, private communication.

[5] R.Talman *Representation of Thick Quadrupoles by Thin Lenses*, SSC-N-33, 1985.