

A PARALLEL CODE FOR LIFETIME SIMULATIONS IN HADRON STORAGE RINGS IN THE PRESENCE OF PARASITIC BEAM-BEAM INTERACTIONS

A. Kabel*, Y. Cai; SLAC
B. Erdelyi, T. Sen, M. Xiao; FNAL

Abstract

The usual approach to predict particle loss in storage rings in the presence of nonlinearities consists in the determination of the dynamic aperture of the machine. This method, however, will not directly predict the lifetimes of beams. We have developed a code which can, by parallelization and careful speed optimization, predict lifetimes in the presence of 100 parasitic beam-beam crossings by tracking $> 10^{10}$ particles-turns. An application of this code to the anti-proton lifetime in the Tevatron at injection is discussed.

BEAM-BEAM EFFECTS IN THE TEVATRON

In its current setup, the Tevatron operates with 36×36 proton/anti-proton bunches, which share the beampipe in a helical scheme. During the injection stage, which takes place at 150GeV, all 72 interaction points are parasitic.

During injection, a relatively fast decay of the anti-proton current is observed; the lifetimes obtained by fitting to an exponential decay behavior are of the order of 1 hour. In the absence of the proton beam, the anti-proton lifetime is well beyond 20 hours. The proton beam itself is unaffected, having a vastly higher charge. Thus, it seems justified to assume that the lifetime behavior is due to weak-strong beam-beam effects.

LIFETIME CALCULATIONS

The beam lifetimes we would like to simulate are of the order of 1hour. For the Tevatron, this corresponds to a loss rate of $< 5.8 \cdot 10^{-9} N/\text{turn}$, where N is the number of particles in a weak bunch.

We choose a direct particle tracking approach to lifetime calculations, as opposed to the usual dynamic aperture calculations. As we are dealing with a proton machines, synchrotron radiation damping mechanisms are absent, so there is no natural limit on the number of turns one needs to track. Assuming that we need to observe 10^2 particles being lost during the tracking procedure to get decent statistics, we need to simulate $> 1.7 \cdot 10^{10}$ Particles · Turns. As we are interested in the injection stage, where we assume that all parasitic crossings will contribute to the dynamics of the beam, $> 1.2 \cdot 10^{12}$ weak-strong interactions need to be simulated. The computational effort necessary is clearly out of the range of single processor machines.

Parallelization clearly is needed to tackle the problem described above. But even when parallelized, the computational demands of a proton tracking code for lifetime calculations are pushing the limits of the computing resources available to us. Thus, a carefully designed, speed-optimized code is necessary; only the physics relevant to the problem should be included.

The code `plibB` (Parallel Lifetime Calculations with beam-Beam) was written with this goal in mind. It is not an optics code; rather it assumes a machine description prepared for it using MAD8, using a defined convention of marking the interaction points within the beamline.

Using the MAD input file, a PERL script then runs MAD several times to (1) calculate the closed orbit of both beams (2) calculate the linear transfer matrices for the anti-proton beam between interaction points with respect to the closed orbit and (3) calculate the one-turn matrix for the proton beam with respect to each interaction point. The Twiss parameters and transfer matrices are then used by the script to generate a C++ program file, containing an alternating sequence of beam-beam kick elements and linear transformations, as well as a single 'aperture', 'twiss', and 'one-turn chromaticity' element each (see below). This file is then compiled and linked into the main code.

In the code, we consider fully coupled motion, which is handled the following way: (Roman indices run over $1 \dots 2N$, Greek ones over $1 \dots N$.) Let M be the one-turn matrix with respect to a given interaction point, it is assumed to describe stable motion, i.e. all its eigenvalues are of unit modulus and pairwise complex conjugate, belonging to complex conjugate eigenvectors: $\vec{\lambda}_{(\mu)} = \lambda_{(\mu)}$, with normalization $\vec{\lambda}_{(\mu)}^+ J \vec{\lambda}_{(\mu)} = i$, where J is the symplectic pseudometric. We can now find correlation matrices of eigendistributions $C^{(\mu)} = M C^{(\mu)} M^\top$ where C is a real, symmetric, positive semidefinite matrix. Introducing $z_{(\mu)} = \vec{\lambda}_{(\mu)}^\top \vec{\lambda}_{(\mu)}$, it is easy to check that $C_{(\mu)} = \frac{\vec{\lambda}_{(\mu)} \vec{\lambda}_{(\mu)}^\top}{2} + c.c.$ forms a set on N such correlation matrices. The real normalized vectors $\vec{v}_{(\mu)}^\pm = \frac{\vec{\lambda}_{(\mu)} \pm \vec{\lambda}_{(\mu)}^*}{\sqrt{\pm 2}}$ are real vectors and span eigenplanes to which the motion associated with the tune ν_μ is restricted. We can construct matrices $D_{\mu,\pm}^l = v_{(\mu)}^\pm$ and $D^{-1} = J D J^\top$ which transform to a diagonal basis in which $D M D^{-1}$ is a decoupled rotation matrix and $D C D^\top$ are projectors onto the eigenplanes. Our code determines the matrices D, D^{-1}, C^μ from the one-turn matrix of protons and anti-protons. The μ indices are sorted according to the eigenvalue, so coordinates are labeled by the associated tune. Emittances are then chosen as to have the spatial diagonal elements of the total correlation matrix

* Work supported by Department of Energy contract DE-AC03-76SF00515.

$C = \sum_{\mu} \varepsilon_{\mu} C^{(\mu)}$ coincide with the known beam dimensions. For the Tevatron at injection, $\varepsilon_0 \approx \varepsilon_y, \varepsilon_1 \approx \varepsilon_x, \varepsilon_2 \approx \varepsilon_s$ to within a few percent.

Chromaticity can be artificially inserted by reading off the eigenchromaticities by analyzing the eigenvalue dependence of M with energy (in the case of the Tevatron, this is in good agreement with the x,y chromaticity output by MAD). Then, the appropriate phase advance is applied each turn to each particle by going to the eigenbasis, rotating by a phase of $2\pi\xi\delta_i$, and going back to the usual basis.

ANALYTIC TREATMENT OF THE BEAM-BEAM KICK

The integrated transverse kick of a gaussian bunch can be expressed by the Bassetti-Erskine Formula[1] in terms of the complex error function $w(z)$.

The evaluation of w turns out to be the most time-critical component of the code. Thus, we surveyed a number of implementations, (for a comprehensive list, see [2]); the fastest we found are the Chiarella-Matta-Reichel algorithm ([3]) and brute force, namely, a pre-calculated 2-D lookup table and bilinear interpolation. This is good for 10^{-6} accuracy. The lookup method works very well on the IBM SP, which has a large data cache; on the PC platform, the first method turns out to be faster.

The implementations were checked with a reference implementation ([4]). Numerical experiments showed that a 10^{-5} relative accuracy or better everywhere is sufficient for typical simulation lengths; lifetime estimates (see below) are in excellent agreement with code using 10^{-16} accuracy implementations; test particles begin to showed visible deviation from the reference implementation not before $10^5 \dots 10^6$ turns.

Currently, the speed achieved is 10^6 (kicks,transforms)/second on Intel Xeon and $6.7 \cdot 10^5$ (kicks,transforms)/second on IBM SP.

Parallelization

The Weak-Strong is embarrassingly parallelizable; a parallel code merely assigns different chunks of particles of a common particle pool to each processor. Communications between processes is only necessary when collective quantities (lost particles, beam sizes, ...) are calculated.

Simplified Distributions

Usually, one would not expect particles in the core of the beam being lost. Therefore, one can use 'de-cored' distributions where particles near the center of the distribution are left out, thus increasing the effective speed of the simulation. Different strategies are possible. Clearly, a simple cutoff in each phasespace dimension is not sufficient, as it will not be a matched distribution. In action space, different cutoffs are possible, the limiting cases of which are (with an obvious notation referring to phasespace coordinates normalized to unity)

- An all-dimensional cutoff, i. e., a $2d$ -hypersphere of radius R cut out: $\rho(\vec{x}) \propto \Theta(|\vec{x}| - R)e^{-|\vec{x}|^2/2} d(|\vec{x}|^{2d}) d\Omega$; the ratio of particles within that distribution is

$$N/N_0 = e^{-R^2/2} \sum_{i=0}^{d-1} \frac{R^{2i}}{2^{2i}i!}$$

- pairwise radial cutoff, i. e., the direct product of distributions with a disc of radius R cut out: $\rho(\vec{x}) \propto \prod_i \Theta(x_i^2 + p_i^2 - R^2) e^{-(x_i^2 + p_i^2)/2} d^d x d^d p$, here

$$N/N_0 = (1 - \exp(-R^2/2))^n$$

For $d = 3$ (six-dimensional phasespace), the latter choice (which we use in our code) eliminates more particles for typical choices ($R \approx 1\sigma \dots 2\sigma$) of the cutoff. Typically, we use 2σ in both transverse direction, corresponding to a savings factor of ≈ 4.0 .

Optimized Transport in Beam-Beam Kicks

The eigendistributions obtained from the one-turn proton matrices in each interaction points are used to calculate the transverse proton beam profile ellipse. Depending on the quantities $\sigma_{long}/\sigma_{trans}, -2\langle xy\rangle/\langle x^2\rangle - \langle y^2\rangle$, and its derivative, the transport of a particle with non-zero longitudinal coordinate or the strong beam's ellipse into the actual interaction point may require to consider (1) the hourglass effect (2) the tilt of the beam ellipse (3) an s -dependent tilt of the beam ellipse. Consequently, 6 different routines were written to handle the possible cases in a speed-optimized fashion. The appropriate routine for a given interaction point is determined at the initialization stage of the code.

LIFETIME ESTIMATES

During the tracking procedur, the 3 action variables of the particle are determined (in the "Aperture" element). The action is recorded in the particle's data structure if it is greater than actions recorded before. Thus, after completion of each turn, one has a record of the maximum action for each particle.

Scanning through I_x, I_y -action space, one can count the number of particles beyond a certain I_x, I_y . Assuming that this action pair represents the physical aperture of the machine, one gets a number $N_{Lost}(I_x, I_y; t)$ (where t represents the number of turns), which is a monotonously falling function with respect to I_x, I_y for fixed t and a monotonously rising function with respect to t for fixed I_x, I_y . In order not to slow down the code, this scan is typically run after each $10^3 \dots 10^4$ turns. The instable particles are identified and its tag number printed out; in a subsequent run, the code can be run with only the instable particles in the initial distribution; phase space dumps then are used to help; identify the instability mechanism.

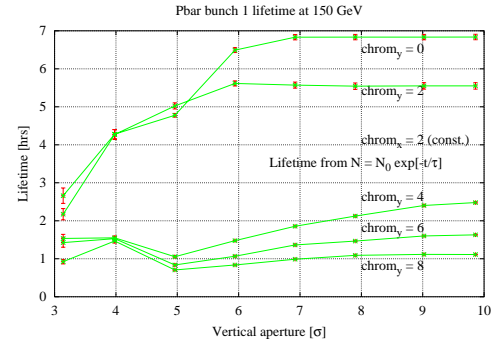
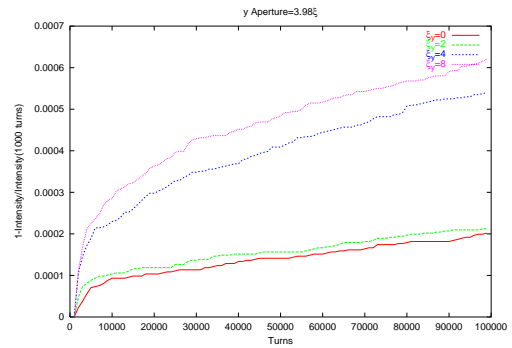
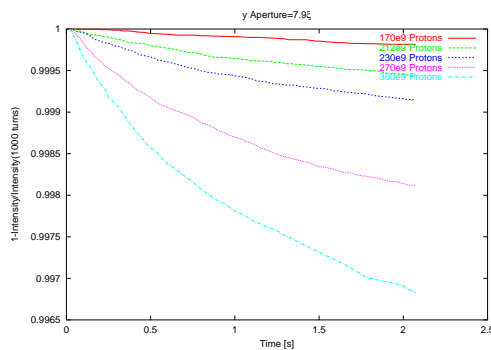
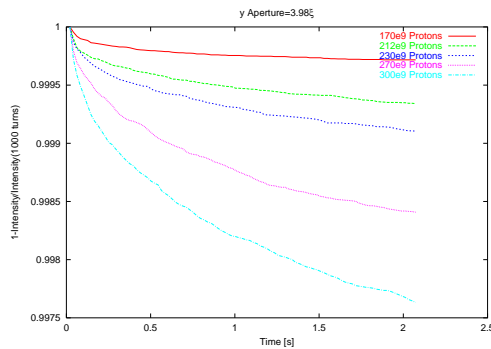
Looking at N_{Lost} for fixed I_x, I_y , one can observe typical lifetime patterns: a quick loss of particles (due to fast resonances or mismatching effects), going over into a linear

behavior for large turn numbers. Extrapolating the linear behavior, one can estimate the lifetime, assuming a certain aperture, by looking at the slope of the linear part of N_{Lost} . The advantage of this procedure is, of course, the fact that it allows to check the connection between physical aperture and lifetime for a number of apertures with a single run.

PARAMETER SCANS

We run a series of studies, varying proton current and chromaticity. The intensities were normalized to unity at 1000 turns (disregarding, in effect, fast losses due to mismatching). Typical results are given in figures The x aperture was fixed at $> 9\sigma$. Fig. (plot courtesy T. Sen) shows the results of an exponential fit for the intensity results for the chromaticity scan. The apparently decreasing lifetimes for increasing apertures near 3σ can be explained by the choice of the fit, only for large apertures can the intensity be approximated by an exponential function; for smaller apertures, $I(t) \approx \exp(-\sqrt{t/\tau})$. Clearly, a better model for the $I(t)$ is needed to accurately predict actual lifetimes.

An interesting observation is the increase in lifetime with decreasing y chromaticity. This signature is indeed observed experimentally¹. However, the region of the predicted jump in lifetime has not been accessed experimentally so far.



CONCLUSION AND FURTHER DIRECTIONS

Through parallelization and optimized implementation of tracking procedures, the PliB code is able to calculate finite lifetimes caused by parasitic beam-beam crossings based on the tracking simulation of $\approx 10^{10}$ particles-turns, which, as it produces measurable quantities, can be viewed as a distinct advantage over dynamical aperture results produced by other tracking codes. While we are able to reproduce the signatures of anti-proton loss in the injection stage of the Tevatron, the model is not complete.

The possibly most important physics we are missing is magnet nonlinearities. The natural chromaticity of the Tevatron due to sextupole elements is extremely high (it is compensated down to 8 units, however). While we have implemented the effective global lattice chromaticity, the high local sextupole content may lead to strong non-linear effects. To handle this effects, we have implemented a fast truncated power series evaluator and a chain of tools, involving MAD and COSY infinity, which allows us to extract high-order maps between interaction points and compile them into PliB.

REFERENCES

- [1] G. A. Erskine M. Bassetti. *CERN-ISR-TH/80-06*, 1980.
- [2] J. A. C. Weideman. *SIAM J. Numer. Anal.*, 31:1497, 1994.
- [3] F. Matta and A. Reichel. *Math. Comp.*, 25:339, 1971.
- [4] W. Gautschi. *SIAM J. Numer. Anal.*, 7:187, 1970.

¹T. Sen, personal communication