DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. Reference herein to any social initiative (including but not limited to Diversity, Equity, and Inclusion (DEI); Community Benefits Plans (CBP); Justice 40; etc.) is made by the Author independent of any current requirement by the United States Government and does not constitute or imply endorsement, recommendation, or support by the United States Government or any agency thereof.

BNL-229007-2025-TECH
EIC-ADD-TN-142

The TRANFT User Manual version 2.0

M. Blaskiewicz

September 2025

Electron-Ion Collider

Brookhaven National Laboratory

U.S. Department of Energy

USDOE Office of Science (SC), Nuclear Physics (NP)

DISCLAIMER

The TRANFT User Manual version 2.0

M. Blaskiewicz*
BNL, Upton NY 11973, USA

The Fortran program TRANFT has undergone significant revision. While many of the original features remain others have been modified and new capabilities have been added. A second transverse direction is now tracked so that the tune and force distributions associated with the beam beam force and nonlinear space charge are correct. Also, general mutibunch instabilites are now treated. For symmetric coupled bunch modes one can track one bunch for purely transverse modes or a few contiguous bunches to include longitudinal modes as well. Another option which tracks all the bunchs in the ring with provision for a clearing or abort gap is also available. In addition to this a new algorithm to model the transverse voltage on crab cavities is introduced which includes the effect of various feedbacks.

I. INTRODUCTION AND THEORY

Coherent instabilities are of significant concern for a wide variety of planned and existing accelerators. The theory of these phenomena has been advancing steadily for decades [1–15]. A theoretical treatment involving all the relevant pieces appears very difficult whereas simulation using particle tracking is conceptually straightforward [16].

The algorithm involves single particle evolution and multi-particle kicks. First consider the single particle motion. The single particle longitudinal update for one turn is given by

$$\bar{\epsilon} = \epsilon + \frac{q}{mc^2}[V(\tau) - V_s] + \delta\epsilon - T_0\epsilon/T_r \tag{1}$$

$$\bar{\tau} = \tau + \frac{T_0\eta}{\beta^2\gamma_0}\bar{\epsilon} \tag{2}$$

where τ is the arrival time of the particle with respect to the synchronous phase, $\epsilon = \gamma - \gamma_0$ is proportional to the energy deviation, γ_0 is the reference Lorentz factor for a particle of mass m and charge q, $V(\tau)$ is the RF voltage, V_s is the synchronous voltage due to both acceleration and radiation, $\beta = v/c$, T_0 is the revolution period, η is the frequency slip factor, $\delta\epsilon$ is a quantum excitation random kick, T_r is the longitudinal radiation damping time, and the updated variables are $\bar{\tau}$ and $\bar{\epsilon}$.

Only one transverse variable is subjected to collective forces and it will be referred to as x. The single particle transverse update, without radiation, for one turn is

$$\bar{x} = x\cos\psi + p\sin\psi \tag{3}$$

$$\bar{p} = -x\sin\psi + p\cos\psi \tag{4}$$

$$\psi = \psi_0 + \frac{2\pi\xi}{\beta^2\gamma_0}\epsilon + o_x(x^2+p^2) + o_y(y^2+p_y^2) \tag{5}$$

where p is the transverse momentum variable, ψ_0 is the on-momentum phase advance, and ξ is the chromaticity. We refer to the other dimension with variables y and p_y. Octupolar detuning from both dimensions is included in x motion. Transverse radiation damping and quantum excitation are also included for x and y,

$$\bar{x} = \left(1 - \frac{T_0}{T_x}\right)x + \delta x \tag{6}$$

$$\bar{p} = \left(1 - \frac{T_0}{T_x}\right)p + \delta p, \tag{7}$$

where T_x is the transverse radiation damping time, and δx and δp are random variables. While equations (1) though (7) are written for one turn, TRANFT allows the user to choose the number of updates per turn.

*Electronic address: blaskiewicz@bnl.gov

The single bunch multiparticle forces are associated with three Green's functions that are referred to as wake potentials. The longitudinal voltage is

$$V_s(t) = -\int_{-\tau_b}^{\tau_b} W_s(\tau) I_b(t-\tau) d\tau, \tag{8}$$

where τ_b is the bunch length, $W_s(\tau)$ is the longitudinal wake potential, and $I_b(t)$ is the instantaneous beam current. Note that $I_b(t)$ is the linear superposition of the current impulses from each of the individual macro-particles. The short range transverse voltage is

$$V_x(x,t) = \int_{-\tau_b}^{\tau_b} \left[x W_d(\tau) I_b(t-\tau) + W_x(\tau) D_x(t-\tau) \right] d\tau, \tag{9}$$

where $W_d(\tau)$ will be called the detuning wake [17–19], $W_x(\tau)$ is the usual transverse wake potential, and $D_x(t)$ is the instantaneous dipole density. Note that $D_x(t)$ is the product of the instantaneous current and the instantaneous value of x.

II. ALGORITHMS

All calculations, but wakefields, are done to machine precision using straighforward implementations of the equations already introduced. As an example of the wakefield calculations consider the longitudinal voltage, equation (8). Taking the instantaneous current to be a series of delta functions one obtains the first order approximation

$$V_{s,1}(t) = -\sum_{k=1}^{N} \hat{q} W_s(t-\tau_k), \tag{10}$$

where there are N macroparticles of charge $\hat{q}$. There are two problems with using (10) as it stands [22]. Firstly, since N is small compared to the actual number of particles within the bunch, there can be large statistical fluctuations in the applied voltage. This is especially worrisome since short range wake potentials tend to be very large. The net effect is that one can have a significant, unphysical, blow-up in the longitudinal emittance. The second problem, not fully unrelated to the first, is caused by the discrete time steps between updates. A typical particle makes a step $2\pi Q_s \sigma_\tau$ each turn, where Q_s is the synchrotron tune and σ_τ the rms bunch length. When length scales less than $2\pi Q_s \sigma_\tau$ are important in the wake potential then it is possible for macro-particles to pass each other without interacting via the short range wake. Both of these problems can be alleviated by convolving (10) with a smoothing function of characteristic scale $\Delta\tau \gtrsim 2\pi Q_s \sigma_\tau$ and, since convolution is commutative and associative, we may consider a smoothed wake potential $\hat{W}_s(\tau)$. This leads to a second approximation for the voltage that is physically reasonable

$$V_{s,2}(t) = -\sum_{k=1}^{N} \hat{q} \hat{W}_s(t-\tau_k). \tag{11}$$

To update the particles equation (11) needs to be evaluated for $t = \tau_1, \ldots \tau_N$ and a naive algorithm requires $O(N^2)$ operations. Instead of incurring this computational penalty, it was decided to use an approximate technique. First, a uniform grid of points spaced by $\delta t \lesssim \Delta\tau/5$ is generated. Next, the macroparticles are placed on the grid via linear interpolation. A fast Fourier transform (FFT) is applied, multiplied by the FFT of $\hat{W}_s$, and an inverse FFT completes the calculation of $V_{s,2}$. The total number of grid points is a power of 2 and the total grid length is at least twice the total bunch length to eliminate "phantom" of "ghost" forces [23]. There are two sources of error involved with this computation. The first is due to the application of linear interpolation in gridding the system and the second involves using numerical integration (via FFT) to evaluate the sums. The net effect is easily tested by cutting δt in half and rerunning the simulation until the answer converges. For many cases the smoothing length scale associated with a full turn update is to large. For these cases TRANFT allows the turn to be cut into pieces allowing for shorter range effects.

Now consider the coupled bunch algorithms. In the first it is assumed that there are M identical, equally spaced bunches interacting with coupled bunch mode number s. While the code allows for tracking a few continguous bunches to treat longitudinal CBMs this discussion is limited to a single tracked bunch. The tracked bunch is given

index 0. Consider a fixed location in the ring and let $x_k(n)$, $p_k(n)$, τ_k(n) denote the coordinates and arrival time of macroparticle k on turn n. For this turn define the two functions

$$D_x^0(t,n) = \sum_{k=1}^{N} x_k(n)\hat{\delta}(t-\tau_k(n))$$
$$D_p^0(t,n) = \sum_{k=1}^{N} p_k(n)\hat{\delta}(t-\tau_k(n)), \tag{12}$$

where $\hat{\delta}(t)$ is a Gaussian smoothing function. Note that these functions are nonzero only over the interval occupied by the first bunch. Define the coupled bunch mode number so that at a fixed point in Minkowski time there is a phase shift of $-2\pi s/M$ between one bunch and the next. The key assumption of this approximation is that equations (3) and (4) with $\psi = \psi_0$ are adequate to characterize the bunch motion over a single turn. Then one can define the dipole moment for turn n,

$$\begin{aligned} D_x(t,n) \quad &= \sum_{m=0}^{M-1} D_x^0(t-mT_0/M,n)\cos(m\psi_0/M - 2\pi ms/M) \\ &+D_p^0(t-mT_0/M,n)\sin(m\psi_0/M - 2\pi ms/M). \end{aligned} \tag{13}$$

Notice that the dipole moment for the turn are just shifted linear combinations of D_x^0 and D_p^0. For $D_x^0(t,n) = \cos(n\psi_0)\delta(t-nT_0)$, $D_p^0(t,n) = -\sin(n\psi_0)\delta(t-nT_0)$ one has

$$D_x(t) = \sum_n D_x(t,n) = \frac{M}{T_0}\sum_{p=-\infty}^{\infty} \cos[(pM+s-Q_x)\omega_0 t], \tag{14}$$

where Q_x is the betatron tune and $\omega_0 = 2\pi/T_0$. The frequency spectrum is typical for a coupled bunch mode and the equation above can be used to find the most unstable value of s for a given impedance.

We now limit the allowed transverse wakes to the form $W_x(\tau) = Re(\tilde{W}(\tau))$, with

$$\tilde{W}(\tau) = H(\tau)\sum_{\ell=1}^{L} W_\ell \exp(-\alpha_\ell \tau) \tag{15}$$

where L is the number of long range transverse wakes and H is the Heaviside function. For a given n the complex force is given by

$$F_\ell(t) = W_\ell \int_{-\infty}^{t} dt_1 D_x(t_1)\exp(-\alpha_\ell(t-t_1)), \tag{16}$$

which satisfies the differential equation

$$\frac{dF_\ell}{dt} = -\alpha_\ell F_\ell + W_\ell D_x(t) \tag{17}$$

Suppose F_n is known at the start of the turn. In the very beginning it is just 0. One accumulates arrays representing $D_x^0(t,n)$ and $D_p^0(t,n)$ as was used for the single bunch forces. The first bunch occupies the interval $[0, T_0/M]$. Equation (17) is used to update $F_n(t)$ over this interval supplying the force for bunch 0. Additionally one generates the numbers

$$\begin{aligned} U_\ell &= \int_0^{T_0/M} dt_1 D_x^0(t_1) W_\ell \exp(-\alpha_\ell(T_0/M - t_1)) \\ V_\ell &= \int_0^{T_0/M} dt_1 D_p^0(t_1) W_\ell \exp(-\alpha_\ell(T_0/M - t_1)) \end{aligned} \tag{18}$$

We have calculated $F_\ell(T_0/M)$ by integrating equation (17) over the first bunch. The most bullet proof algorithm to obtain the force at the end of the turn uses the iteration

$$F_\ell((k+1)T_0/M) = F_\ell(kT_0/M)\exp(-\alpha_\ell T_0/M) + U_\ell\cos(k\psi_0/M - 2\pi ks/M) + V_\ell\sin(k\psi_0/M - 2\pi ks/M). \tag{19}$$

While it is possible to use a geometric series to sum this there is always the risk of small denominators. Given there are usually less than a few thousand bunches this is a low cost safety measure.

The second transverse algorithm tracks a larger number of bunches simply using eq (17) throughout. It is worwhile to mention that the bunches can occupy regions far smaller than T_0/M and that between the bunch grids one simply propagates the decaying exponentials.

For longitudinal coupled bunch modes there is no simple formula for motion over a turn. In this case we track N_s contiguous bunches with M/N_s an integer. The wakes are limited to exponentials as in the transverse case. The bunches for the remainder of the turn are copies of the first N_s. With N_s 4 or 5 one can get coupled bunch modes if the resonant frequencies are chosen correctly. The transverse modes are updated using equation (19) with $M \to M/N_s$. As with the transverse modes it is possible to track using eq (17) alone.

A nonlinear space charge force has been implemented. It starts by calculating smoothed dipole moments and longitudinal densities

$$D_x(t) = \sum_{k=1}^{N} x_k \hat{\delta}(t-\tau_k), \qquad \lambda(t) = \sum_{k=1}^{N} \hat{\delta}(t-\tau_k). \tag{20}$$

With these one creates $\bar{x}(t) = D_x(t)/\lambda(t)$. The beam is assumed to be a round Gaussian with the same rms as the initial macroparticle distribution. The force in x is approximated as:

$$F_x(x,y,t) = K\lambda(t)[x-\bar{x}(t)]\frac{1-\exp\left[-\frac{(x-\bar{x}(t))^2}{2\sigma_b^2}-\frac{y^2}{2\sigma_b^2}\right]}{(x-\bar{x}(t))^2+y^2} \tag{21}$$

where K is a constant chosen to give the correct tune shift for small amplitude particles. A similar formula is used for the beam-beam kick, which is applied once a turn.

III. IMPLEMENTATION AND USE

The fortran source code resides in the file `tranft2_0.f` and is liberally commented. Common blocks and some parameters are in `tranft2_0_c.f`. The code employs some routines from numerical recipes[24]. Some small modifications were needed. The top of subroutine `SORT2.FOR` needs to be changed from

```
      SUBROUTINE SORT2(N,RA,RB)
      DIMENSION RA(N),RB(N)
```

to

```
      SUBROUTINE SORT2(N,RA,RB)
      DIMENSION RA(N)
C NEXT DECLARATIONS FOR INTEGER TAG-ALONG ARRAY
C BASED ON QUICKSORT
      INTEGER RB(N),RRB
```

Along with `sort2` the code uses `ran3` and `four1`. The code may be obtained by contacting the author `blaskiewicz@bnl.gov`. The code has been tested with the intel fortran compiler `ifort`. The compilation command is

```
ifort -132 -r8  -o tranft2_0 numrec.f tranft2_0.f
```

where the three numerical recipes routines are in `numrec.f` and the executable is `tranft2_0`.

The code uses four input files. The file `tranft2_0.in` contains data with regard to the beam and the lattice while `imped3.in` contains data with regard to collective forces. Numerically calculated wakefields can be incorporated via `impedfile.dat` and parameters defining the crab cavities are in the file `crab.in`. The files are not entirely independent and using the code will be outlined with an example. Consider `tranft2_0.in` given by

```
600  10000  10 171487 10  nturns,ndim,nwrite,iseed,nperturn  fiducial proton input
22.75 3833.94 25.37 -40.e3 315  -40.e3 2 gammat,circ,gamma0,vrf,nharm,vrf2,nharm2
29.23  30.25  1.368e-3  0.e-3 .0 0 0 tunex,tuney,ampx,xinject,chrom_init,chrom2d,dispavg
2.8e11 1. 1. 5.e-10 3.5 15.e-9 0. 100 5.4e9 0. pnumber,aatom,qatom,taupart,power,tauhat,phisynch,
                                                nturnon,harmbtf,phibtf
```

```
3000 21 19  0.e-8  83.e-9 20  nresamp,mbunch,nsim,tlob,thib,mode
-1 1 1 1    tradperp,sigperp,tradlong,siglong
0.00   -0.0 0.000  0.0  2.7e-4  0.0  dgain, alfaxx, alfaxy,bbtune,bbrad,dgainz
1.23778e9 0 0 315 15 19 omegadamp, idononlinsc, idosimfill, ncrabupdate, longrangemult, iavgsum
```

The code reads only the numbers on the lines (free format) and the character strings to the right are the corresponding variables in the source code. No text is required, but the author finds it very helpful. For the case above `nturns = 600` is the total number of turns simulated. The parameter `ndim=10000` controls the number of macro-particles per bunch. Writes to the screen and output files are done every `nwrite=10` turns. Setting `nwrite` to a negative number calculates more beam properties every `|nwrite|` turns and increases computional time. The random number generator is `ran3` from numerical recipes[24] and `iseed =171487` is the random seed. Radiation damping, quantum excitation, beam-beam, and all long range wakes are applied once per turn. All other algorithms are applied `nperturn=10` times per turn. For the `nperturn` updates of the transverse variables the angle in the rotation matrix corresponds to the tune `tune_dyn = tune - int(tune)+1`. This tune gives the correct betatron sidebands and a monotonic increase in the phase angle on the x, p plane throughout the turn. It can be removed upon request.

The transition Lorentz factor is `gammat`, the machine circumference is `circ` meters. The central Lorentz factor of the beam is `gamma0`. The primary RF voltage amplitude is `vrf` volts and `nharm` is the primary harmonic number. For stable motion, the product `vrf*qatom` is positive below transition and negative above. Since electron clouds are not included, switching the signs of `vrf` and `qatom` lead to identical dynamics. When `vrf2=vrf` the second harmonic voltage cancels the linear part of the first harmonic, `nharm2` is the ratio of the higher harmonic to the main harmonic.

The betatron tunes `tunex` and `tuney` and the average beta function is defined to be `circ/(2*pi*tunex)` with `pi = 3.141...`. The beam is round and the rms transverse amplitude at average beta is `ampx` meters and the particle is (eventually) given a kick of amplitude `xinject`. The unnormalized chromaticity is `chrom_init` and the second order chromaticity is `chrom2d`. The average dispersion referenced to the average beta is `dispavg`. It is best left 0.

The actual, physical bunch contains `pnumber` particles of atomic mass `aatom` and atomic number `qatom`. Set `aatom` $= m_e/m_p$ for electrons. The wake potential smoothing is controlled by the parameter `taupart`. The smoothing employs a gaussian function with equivalent length `taupart` $= \sqrt{2\pi}\sigma$ seconds. The initial bunch has half length `tauhat` seconds and its shape is controlled by `power`. The initial bunch is matched to the low current rf bucket with shape $|H_s(\text{tauhat}, 0) - H_s(\tau, \epsilon)|^{\texttt{power}}$ where $H_s(t, \epsilon)$ is the longitudinal Hamiltonian obtained from equations (1) and (2) without radiation damping. The synchronous phase is `phisynch` radians. (The next line should be appended to the one above.) Over the course of `nturnon` turns the longituinal wakes are linearly ramped up. Then, a transverse kick of amplitude `xinject` with angular frequency `harmbtf` and phase `phibtf` is given and the transverse wakes are turned on. By running with `phibtf=0` and `phibtf = 1.5707` ($\pi/2$) the transverse beam transfer function can be extracted [28].

The number of grid points used for the FFT is the smallest power of 2 that is at least as large as `nresamp`. The number of bunches for a symmetric fill is `mbunch`, the number of contiguous bunches to track is `nsim`. If nsim¡ mbunch there is a gap at the end of the bunch train. When symmetry is employed `mode` is the parameter is s in equation (13). To minimize problems with truncation the stable fixed point is at half an rf period $T_{rf}/2$. The interval used for the entire FFT domain is `[tlob,thib]`. One gains no increase in accuracy by having `thib` $-$ `tlob` greater than twice the bunch length, but a factor of 3 leaves a good safety margin without introducing a large computational overhead.

The transverse radiation damping time is `tradperp`, called T_x in equation (6). The rms, equilibrium beam size at average beta is `sigperp`. The longitudinal radiation damping time is `tradlong`, called T_r in equation (1) and `siglong` is the rms energy spread in units of γ. Setting `tradperp < 0` turns off radiation damping and quantum excitation.

The transverse damper kicks with angular frequency `omegadamp` with imaginary tune `dgain`. The octupole damping is controlled with `alfaxx` which is the average tune shift from the octupoles due to the x action and `alfaxy` which is the average tune shift due to y action. The central beam-beam tune shift is `bbtune`. This is implemented as a round beam with rms radius bbrad. There is a rigid longitudinal damper with imaginary tune `dgainz`. When `idononlinsc = 1` the nonlinear space charge algorithm is used. When `idosimfill=1` it is assumed that mbunch/nsim is an integer and equation (12) through (19) are used. The crab cavities are calculated using base band forumlas [29]. The number of baseband updates per turn is `ncrabupdate` which must be an integer multiple of mbunch. Another useful trick is to divide the true number of bunches by an integer. For the case under consideration the number of bunches for a symmetric fill is 315. The number `mbunch`=21 is a factor of 15 less than this. The parameter `longrangemult` multiplies all long range wakefields, by 15 in this case. In this way one can track fewer bunches while keeping most dynamics and not need to modify the impedance input file. The final parameter `iavgsum` is the number of bunches one chooses to use when writing `res.den` and creating averages for `csmon.out`, to be described later.

A sample impedance input file `imped3.in` is

```
3   3833 70.e-8 .035 0 0.  slenx,slens,rhoe,bpipe,wstep,detunefrac
-5.10e-6 2.e-10 47.8e6 0.078 0.e6 0  wallinduct,twall,scimped,scabrat, broad,curve_csr
```

```
2 0 0 5  npolex,npoles,npolexlong,npoleslong
4.5e15 0 2.5e9 0  broad band transverse
1.e15  0 5.0e8  0
  1.245950E+10 -2.317615E+05  1.207316E+04  6.490531E+08  5.160000E+05  2.688000E+04
  1.113201E+10 -1.664973E+05  1.807144E+04  1.208257E+09  3.080000E+05  3.343000E+04
  4.191200E+10 -7.263778E+05  3.010920E+04  1.737301E+09  6.960000E+05  2.885000E+04
  1.090688E+11 -2.344556E+06  4.440910E+04  2.065911E+09  1.228000E+06  2.326000E+04
  3.422371E+10 -3.789162E+05  2.742285E+04  2.476832E+09  6.240000E+05  4.516000E+04
```

The first line defines all the resistive wall quantities. The effective length for the transverse impedance and transverse detuning wake is `slenx` meters. Remember that all quantities are referenced to the average beta function `circ`$/(2*$`pi`$*$`tunex`) and that the relevant quantity for stability calculations is the beta weighted transverse impedance. The effective length for the longitudinal resistive wall impedance is `slens` meters. The electrical resistivity of of the wall material is `rhoe` ohm-meter. The transverse wake potential is calculated assuming a round pipe of radius `bpipe` meters. The transverse detuning wake is a fixed fraction `detunefrac` of the longitudinal wake. For vertical instabilities with a small vertical aperture and large horizontal aperture `detunefrac = 0.5`[18]. The parameter `wstep` is the value for the transverse step function wake (as in a stripline pickup) in volts per coulomb per meter.

The broad band wall inductance is `wallinduct` in Ohm-seconds. The very short range nature of this force warrants additional smoothing[25] and the rms length of the additional smoothing is `twall` seconds. The net voltage kick per turn is given by

$$V = -\texttt{wallinduct}\left.\frac{dI_{beam}}{dt}\right|_{\texttt{twall}},$$

where the subscript on the time derivative of the beam current denotes smoothing with a gaussian pulse of rms length `twall` seconds. For longitudinal space charge[26]

$$\texttt{wallinduct} = -\frac{T_0}{2\pi}\left|\frac{Z}{n}\right|_{sc} = -\frac{T_0}{2\pi}\frac{Z_0}{\beta\gamma^2}\ln\left(\frac{b}{1.5\sigma_b}\right), \tag{22}$$

where the beam has relativistic parameters β, γ, an rms (round) transverse size σ_b in a round pipe of radius b and $Z_0 = 377\Omega$. The magnitude of the direct, transverse space charge impedance is `scimped` in ohms per meter [26, 27],

$$\texttt{scimped} = \frac{Z_0\texttt{circ}}{2\pi\beta^2\gamma^2 a^2} \tag{23}$$

where $a = 2\sigma_b$ is the radius of a uniform equivalent beam[27]. The parameter `scabrat` $= a/b$ is the ratio of the equivalent beam radius to the beam pipe radius and is used to calculate the image space charge force. The broad band transverse impedance in Ohm/meter is `broad`, and `curv_csr` is the average curvature for the free space coherent synchrotron radiation wake. The last components defined in `imped3.in` are the resonators. There are `npolex` short range transverse resonators followed by `npoles` short range longitudinal resonators. Each resonator is defined by 4 real (2 complex) numbers. The input parameters on each line are w_r, w_i in Ohms per second; and α and ω_r in inverse seconds. The wake potential vanishes for $t < 0$ and is $Re(w_r + iw_i)\exp(-\alpha t - i\omega_r t)$ for $t > 0$. One may set either or both of `npolex`,`npoles` to zero and neglect resonator contributions to that plane. There are `npolexlong` long range transverse resonators and `npoleslong` longitinal resonators. For resonantors only the first four columns are read by the code. Here there are additional columns on some wakes denoting shunt impedance and quality factor which help the user keep track of things.

The third input file is `crab.in`. Before describing the input file we overview the algorithm. The crab cavity transverse voltage is related to the dipole moment by

$$\ddot{V}_x + \omega_r^2 V_x + 2\alpha\dot{V}_x = \frac{R_x}{Q}\omega_r^2 D_T(t), \tag{24}$$

where V_x is the transverse voltage on the cavity, ω_r is the angular resonant frequency of the cavity, R_x is the transverse shunt impedance, Q is the loaded quality factor of the cavity, $\alpha = \omega_r/2Q$, and D_T is the total dipole moment driving the cavity. The dipole moment is given by

$$D_T(t) = D_B(t) + x_0 I_{LL}(t) - \frac{1}{\omega_r R_{eff}}\frac{d}{dt}\left(V_x(t - T_d) + V_{notch}(t)\right). \tag{25}$$

In equation (25) D_B is the beam dipole moment at the cavity as a function of time, $x_0 I_{LL}$ is the drive derived from the setpoints of the RF system, R_{eff} is the effective transverse shunt impedance for the feedback gain, T_d is the time delay for wide-band feedback which is equal to an integer number of RF periods. The time derivative operating on the voltages is automatic if one couples to the electric field of the cavity. The voltage V_{notch} is output from narrow band notch filters centered at the betatron sidebands.

TRANFT assumes that a typical voltage varies like $V(t) = Re[\hat{V}\exp(j\hat{\omega}t)]$. The code requires that $\hat{\omega} = k\omega_b$ with ω_b the bunching frequency and integer k. The CC voltage update on a given turn begins with creating the drive signal from the beam,

$$\hat{V}_B(n) = -j\frac{R_x}{Q}\frac{\omega_r^2}{\hat{\omega}}\int_{(n-1)\tau}^{n\tau} D_B(t)\exp(j\hat{\omega}t)dt \tag{26}$$

where $\tau = T_{rev}/M$ and the dipole moment for the turn resides in the interval $[0, T_{rev}]$. When the bunch spacing is large it can be advantageous to update the slow voltages more than once per bunch. This is done by padding zeroes after each nonzero value of $\hat{V}_B(n)$. Including this effect in the derivation introduces unnecessary complications and will be neglected. The wideband feed back drive is

$$\hat{V}_{eff}(n) = \frac{\omega_r\tau}{2Q_{feed}}\left[f_-\hat{V}_{cav}(n-n_-) + f_+\hat{V}_{cav}(n-n_+)\right]\exp(-j\hat{\omega}T_d), \tag{27}$$

where $n_- \le T_d/\tau \le n_+$ bound the true delay, Q_{eff} is the quality factor associated with feedback and $f_- = n_+ - T_d/\tau = 1 - f_+$. For the notch filters there are two stages of processing. First, the notches are formed

$$\hat{V}_\pm(n) = (1-\epsilon)\hat{V}_\pm(n-M)\exp(\mp j2\pi Q_x) + \epsilon\hat{V}_{cav}(m-K)\exp(\mp j2\pi Q_x), \tag{28}$$

where $K < M$ is an input delay, ϵ controls the width of the notch, and Q_x is the betatron tune. Next a low pass filter is applied.

$$\hat{V}_{notch}(n) = \frac{1}{1+\alpha_{notch}}\left\{\frac{\omega_r\tau}{2Q_{feed}}G_{notch}(\hat{V}_+(n) + \hat{V}_-(n)) + \alpha_{notch}\hat{V}_{notch}(n-1)\right\}, \tag{29}$$

where G_{notch} is the notch filter gain and α_{notch} controls the bandwidth of the notch filter. With all the various inputs defined the cavity voltage is updated via

$$\hat{V}_{cav}(n) = \exp[j(\omega_r - \hat{\omega})\tau - \alpha\tau]\hat{V}_{cav}(n-1) - \hat{V}_{eff}(n) - \hat{V}_{notch}(n) + \hat{V}_B(n). \tag{30}$$

The voltage is applied using linear interpolation. For particles in interval $[(n-1)\tau, n\tau]$ this is $V_x(t) = Re[(f_-V_{cav}(n-1) + f_+V_{cav}(n))\exp(j\hat{\omega}t)]$ where t is the particle arrival time and $f_- = n - t/\tau$. After the forces are updated one shifts $\hat{V}(n) \to \hat{V}(n-M)$ and the process is repeated.

We now discuss the input file `crab.in`.

```
13.8e11 3.e6 197077393. 3.196714e-7 500 10 .0625 .9460317 r197,q197,f197,tw197,gw197,gn197,
                                                          eps197,frac197
0       3.e6 394.e6       3.196714e-7 1000 10  .1 .9 eventual parameters of second crab cavity
0 0 500. lagrange idowake qnotch
```

The transverse shunt impedance in Ohm/m is $R_x =$ `r197`. This value is referenced to the average beta function and crab cavities are skipped when `r197` < 0. The loaded quality factor of the crab cavities is $Q =$ `q197` and the resonant frequency is `f197` $= \omega_r/2\pi$. The total delay for fast feedback is `tw197` $= T_d$ and the gain is chosen so that `gw197` $= QR_{eff}/R_x$. The notch filters have gain `gn197` $= G_{notch}$. (The next line should be appended to the line above.) The notch width in revolution lines is `eps197` $= \epsilon$ and the notch filter input is delayed by `frac197` turns. The second line is a place holder for the second crab cavity. On the third line the parameter `lagrange` may be implemented to control the order of Lagrange interpolation. It is not used currently and should be left 0. When `idowake` $= 1$ the crab cavity routine gives a single kick starting at the first turn. The output files can be used to get the effective impedance. Finally, the parameter `qnotch` is the quality factor of the low pass filter applied to the notch output. The interested reader can consult [29] for details.

The final input file is `impedfile.dat` which allows the user to input data from simulation codes such as Gdfidl, CST and ABCI. The top few lines of such a file are

```
  3.137e-12 -242 114   dtfile,nptfile,idex0
 -3.545098E-10  -8.046247E+12  0.000000E+00  4.023123E+12  tk filex(k),files(k),filed(k)
 -3.513726E-10  -4.601705E+12  0.000000E+00  2.300852E+12
```

The file defines three arrays of `nptfile` elements. The wakepotentials defined in these arrays are added to those calculated above. Setting `nptfile < 0` causes the subroutine to exit and no lines after the first are read. The array elements are spaced in time by `dtfile` seconds. The index corresponding to time equals zero is `index0`. It is strongly suggested that at least one negative time be included, even if all wake potentials vanish at that time. The second line on contain the array elements. The first column is the time lag in seconds. While this column is ignored by the code the author finds it useful for comparison purposes. columns 2,3 and 4 are the additional values of W_x, $-W_s$ and W_d. Note that the longitudinal wake is negative at the particle which corresponds to the sign convention used in most codes.

To run the code it is suggested that all the input files be put in a single use directory. When the code runs there are several lines describing relevant quantities. Some notable ones are the actual voltage used for the second harmonic cavity and the average beta function assumed for applying transverse wakes. Another important parameter is the ratio of bin length to the rms of the gaussian smoothing length, it should be $\lesssim$0.2. Also there is the Oide-Yokoya intensity factor I[13] in MKS units.

On the last turn of the simulation the individual particle tunes are calculated via least squares. Let a single particle's coordinates on this turn be $x_1, p_1, \ldots x_N, p_N$ with $N =$ `nperturn`. Consider the sum

$$\Gamma(a,b,c,d) = \sum_{k=1}^{N-1} \left(x_{k+1} - ax_k - bp_k\right)^2 + \left(p_{k+1} - cx_k - dp_k\right)^2 . \tag{31}$$

Minimizing Γ simultaneously with respect to a, b, c and d gives a least squares fit to the transfer matrix best describing the particle. Given the transfer matrix the single particle tune q satisfies $2\cos(2\pi q/N) = (a+d)/(ad-bc)$. The sign of q is the same as the sign of b. Fitting $q = \xi\epsilon/\gamma + q_0$ yields yields the chromaticity ξ. The effect of long range wakes are not included in the tune calculation.

As the code updates the turn number (`kturns`) and number of macroparticles per bunch (`np`) are displayed. There are also two quantities which allow the user to monitor the progress of a transverse instability. The parameter `csfull` is the average over particles of $x^2 + p^2$. This parameter includes information about the emittance as well as any coherent motion. The parameter `coherecs` is tailored to be a sensitive indicator of instability. Let $\bar{x}(t)$, and $\bar{p}(t)$ be smoothed average values of x and p as the bunch passes and let $I(t)$ be the smooth current pulse, then

$$\texttt{coherecs} = \frac{\int\limits_{bunch} dtI(t)\left[\bar{x}^2(t) + \bar{p}^2(t)\right]}{\int\limits_{bunch} dtI(t)}. \tag{32}$$

Along with writes to the screen there are several output files. The wake potentials in various forms are in `w.out`, `z.out`, and `ws.out`. The raw wake potentials before smoothing are in `w.out`. There are 4 columns: the time, $W_x(\tau)$, $W_s(\tau)$, and $W_d(\tau)$. All are in MKS units. The file `z.out` contains the Fourier transforms of the wake potentials and the frequency window used for smoothing. Defining Fourier transforms as

$$\tilde{F}(f) = \int\limits_{-\infty}^{\infty} F(t)\exp(2\pi i f t)dt,$$

the columns are f in Hz, $Re(\tilde{W}_x)$, $Im(\tilde{W}_x)$, $\exp(-\pi[f\texttt{taupart}]^2)$, $Re(\tilde{W}_s)$, $Im(\tilde{W}_s)$, $Re(\tilde{W}_d)$, and $Im(\tilde{W}_d)$ all in MKS. The final wake field file is `ws.out`. This file contains the smoothed wake potentials in the same format as `w.out`.

A note concerning the resistive wall quantities is in order. Only the simplest low frequency approximations for resistive wall quantities have been used, resulting in continuum forms that have singularities of various types. The transverse resistive wall wake is taken to be

$$W_x(\tau) = H(\tau)\frac{cL_x}{\pi b^3}\sqrt{\frac{Z_0\rho}{\pi c\tau}}, \tag{33}$$

where $Z_0 = \mu_0 c = 376.74\Omega$ is the impedance of free space, $L_x =$ `slenx`, $b =$ `rpipe`, $\rho =$ `rhoe`, and $H(\tau)$ is the Heaviside function. The value of W_x assigned to time $n\delta t$ is

$$W_{x,n} = \frac{1}{\delta t} \int_{(n-1/2)\delta t}^{(n+1/2)\delta t} W_x(t)dt. \tag{34}$$

This guarantees that the low frequency impedance behaves correctly and that the smoothed quantities converge rapidly to their continuum values. When the pipe radius or beta function vary with machine azimuth the appropriate value for $L_x =$ `slenx` satisfies

$$\bar{\beta}\frac{L_x\sqrt{\rho}}{b^3} = \oint \beta(s)\frac{\sqrt{\rho(s)}}{b(s)^3}ds, \tag{35}$$

where $\bar{\beta} =$ `circ`$/(2*$`pi`$*$`tunex`$)$ is the average beta function. The longitudinal resisitve wall wake potential it taken to be

$$W_s(\tau) = \frac{d}{d\tau}\left\{H(\tau)\frac{L_s}{2\pi b}\sqrt{\frac{\rho Z_0}{\pi c\tau}}\right\}, \tag{36}$$

where $L_s =$ `slens`. $W_s(\tau)$ has a functional near $\tau = 0$, but it is trivial to integrate as long as $\tau = 0$ is not one of the end points. The code does this in exact analogy to equation (34). The analog of equation (35) is

$$\frac{L_s\sqrt{\rho}}{b} = \oint \frac{\sqrt{\rho(s)}}{b(s)}ds. \tag{37}$$

Two output files describe the instantaneous dynamics of the beam. The file `res.den` is over written every `|nwrite|` turns and contains gridded data. The file `tran.full` is over written every `|nwrite|` turns if `nwrite < 0`. Both are always written when the simulation ends and the code is much faster with `nwrite > 0`. There are 8 columns of data in `res.den`. The first column is arrival time in seconds, with $T_{rf}/2$ corresponding to the stable synchronous phase. The second column is the instantaneous line density normalized to be equal to the number of macroparticles in $[t -$ `taupart`$/2, t +$ `taupart`$/2]$, which is useful for judging the statistical accuracy of the simulation. Columns 3 and 4 are $\bar{x}(t)$ and $\bar{p}(t)$ in meters, as defined in equation (32). Column 5 is the instantaneous current in amperes. Column 6 is the value of the kick to p due to W_x, column 7 is the longitudinal kick due to W_s in units of γ, and column 8 is the detuning kick. Multiplying column 8 by $-$`nperturn`$/4\pi$ yields the tune shift due to W_d as a function of longitudinal position within the bunch.The next two columns are the longitudinal and transverse kicks due to long range forces with the same units as columns 6,7. Columns 11 and 12 describe the nonlinear space charge force. Column 11 is $\bar{x}(t)$ and columns 12 is $K\lambda(t)$.

The second file describing instantaneous dynamics is tran.full. The first 4 columns are x, p, t, ϵ for each of the macroparticles, They are ordered in t. Column 5 is the betatron tune of the particles in the first bunch. The next column is the kick in γ due to the RF voltage. The next column is a running sum of ϵ within the bunch. The last value in each subarray is the average energy offset before the RF kick. With strong synchrotron radiation loss the last value in each subarray is close to the energy loss per turn. The next two columns are the offsets and momenta of the other transverse coordinate. The only collective force for these coordinates is beam-beam.

The summary output file is `csmon.out` which has 8 columns. Column 1 is the turn number. Column 2 is $< x^2+p^2 >$ averaged over the beam. Column 3 is $< y^2+py^2 >$ averaged over the beam. Column 4 is `coherecs` defined in equation (32). Column 5 is the averge rms bunch length in seconds. Column 6 is the average rms energy spread in units of γ. Column 7 is the effective synchrotron tune defined using the ratio of the rms energy spread to the rms bunch length. It is useful for long bunches or multiple harmonics. Column 8 is $< t >$ averaged over the bunches. Column 9 is the averge value of ϵ averaged over bunches. Column 10 is the rms value of x and 11 is the average value of x.

There are 6 other output files, `fort.21` contains output associated with the crab cavities. Each line is

$$t, \hat{V}_{cav}(n), \hat{V}_B(n), \hat{V}_+(n), \hat{V}_-(n), \hat{V}_{notch}(n-1)$$

Each line of `fort.22` are the real and imaginary values of the long range transverse wakes at the end of each turn. The file `fort.23` is the same for the longitudinal.

The file `fort.24` has 3 columns. The first is the turn number. Columns 2 and 3 are the cosine and sine Fourier harmonics of D_x evaluated at frequency `harmbtf`. When kicks are used these columns can be used to calculate beam transfer functions [28].

The initial particle coordinates are in `fort.66`.

$$n, t(n), \epsilon(n), x(n), p(n), y(n), py(n)$$

The initial RF parameters are in `fort.67`. Column 1 is the time in seconds, column 2 is the voltage in volts. Column 3 is $U(t) = C \int^t V_{rf}(t)dt$. The constant is chosen so that $\sqrt{U(t) - U_{usp}}$, with U_{usp} the value of U at the unstable fixed point, is the RF bucket in units of ϵ.

Consider the output using the input files described above, only the first line of `impedfield.dat` is needed. Figures 1 through 5 highlight key points. Please note this example is only for illustration.

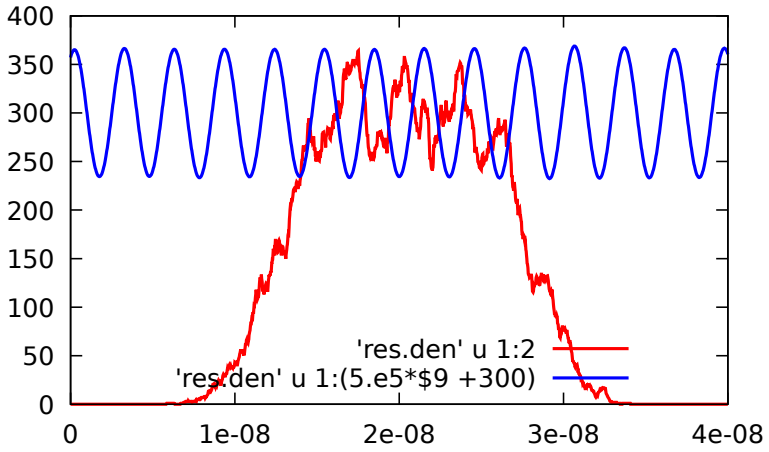

FIG. 1: Longitudinal profile of bunch 1 (red) and scaled and offset coherent longitudinal long range force (blue). Notice that the peaks in the line density correspond to losing energy to the wake.

Acknowledgements

Thanks to Michael Borland, Yong-Chul Chae, Sam Krinsky, Ryutaro Nagaoka, Thomas Roser, Todd Satogata and Gang Wang for useful discussions and encouragement.

---

[1] F.J. Sacherer, CERN/SI-BR/72-5, (1972).
[2] F.J. Sacherer, IEEE-TNS **NS-24**, # 3, p1393, (1977).
[3] A.W. Chao & C.Y. Yao, PEP-Note 321, (1979).
[4] J.L. Laclare, CERN 87-03, p 264 (1987).
[5] J.M. Wang, BNL 39057, (1986).
[6] A.W. Chao *Physics of Collective Beam Instabilities in High Energy Accelerators* Wiley, (1993).
[7] Y. Chin, K. Satoh & K. Yokoya, *Particle Accelerators* **13** , p45 (1983).
[8] Y. Chin, NIMA , **215** p501, (1983)
[9] K. Satoh & Y. Chin, *Nuclear Instruments and Methods* **207**, p309 (1983).
[10] G. Besnier, D. Brandt & B. Zotter, *Particle Accelerators* **17**, p51 (1985).
[11] S. Krinsky & J.M. Wang *Particle Accelerators* **17** , p109 (1985).
[12] Y. Chin, CERN SPS/85-2 (1985).
[13] K. Oide & K. Yokoya, KEK 90-10 (1990).

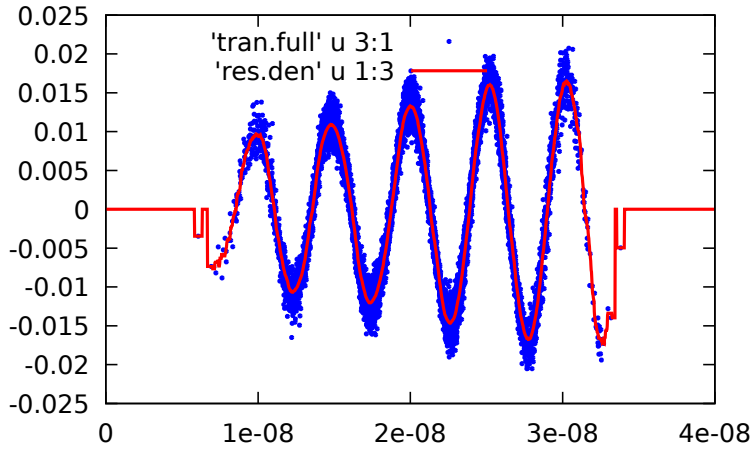

FIG. 2: x versus t of particles in bunch 1 (blue) and average offset along the bunch (red).

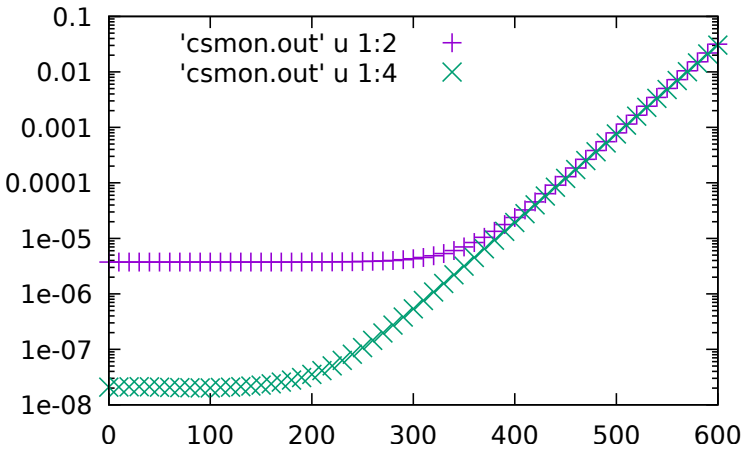

FIG. 3: Average of $< x^2 + p^2 >$ (purple) and coherent value from eq (32) (green).

[14] M. Blaskiewicz, W.T. Weng PRE **50** p4030, (1994)
[15] P. Kernel, R. Nagaoka, J.-L. Revol, & G. Besnier p1133, EPAC00.
[16] R.H. Siemann, AIP Conf 127, p368, (1983).
[17] R.L. Gluckstern, J. van Zeijts, B. Zotter, PRE **47** # 1, p656 (1993).
[18] K. Yokoya *Particle Accelerators* **41** , p221 (1993).
[19] A. Burov & V. Danilov, PRL **82** , #11 p2286 (1999).
[20] R.D. Kohaupt DESY-85-139 (1985).

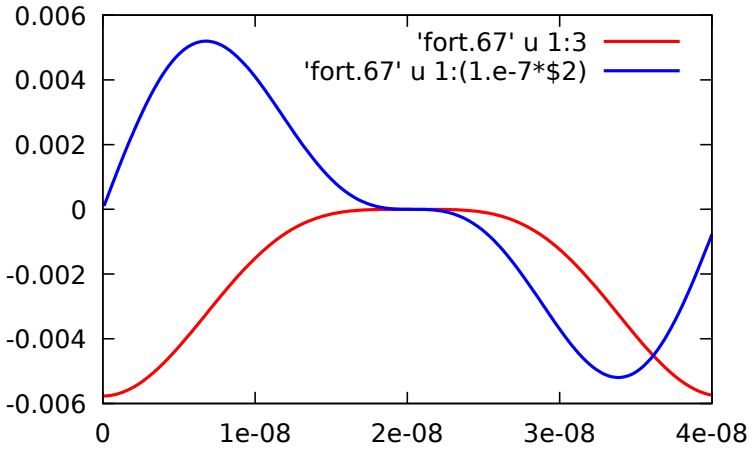

FIG. 4: RF potential (red) and scaled RF voltage (blue). The potential is -0.00577 at the unstable fixed point.

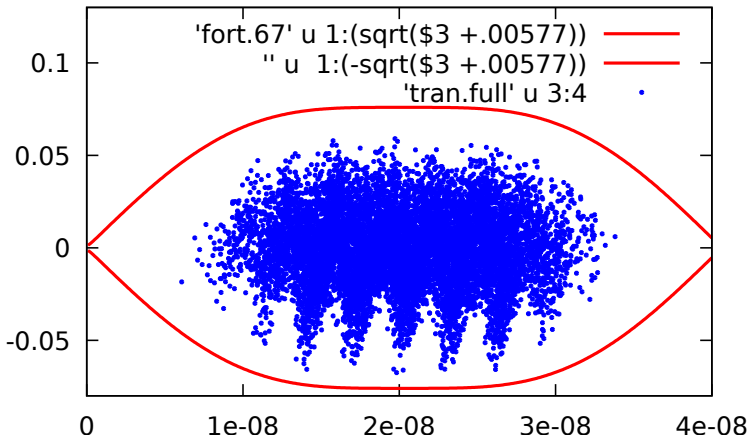

FIG. 5: RF bucket (red) and particles (blue).

[21] J.S. Berg, CERN SL 97-72 (AP) (1997).
[22] Shane Rupert Koscielniak, PhD Thesis, University of Oxford, (1987).
[23] R.W. Hockney, J.W. Eastwood *Computer Simulation using Particles* Hilger, Bristol, (1988).
[24] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery *Numerical Recipes in Fortran 77* Cambridge University Press (1992).
[25] J. Wei *PhD thesis* State University of New York at Stony Brook (1990).
[26] A.W. Chao, M. Tigner *eds Handbook of Accelerator Physics and Engineering* World Scientific (1999).

[27] M. Blaskiewicz PRSTAB **4**, 044202 (2001).
[28] M. Blaskiewicz, V. Ranjbar, PAC2013, froaa2, (2013).
[29] T. Masoridis *et. al.* PRAB **28**, 011001 (2025). see also EIC preliminary design manual, collective effects section (2025).