EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN-SL-Note-2002-036 AP

Practical User Guide for HEADTAIL

G. Rumolo and F. Zimmermann

20 November 2002

Abstract

This note describes the program HEADTAIL for the simulation of transverse and longitudinal single bunch phenomena, with special emphasis on the instability and emittance growth induced by an electron cloud.
All input parameters as well as the standard output files are described here. The note is intended to provide potential users with the guidelines for the installation and the use of the program, starting from a basic knowledge of its internal structure.

Geneva, Switzerland

1 General

Single bunch collective phenomena associated with regular impedances and/or induced by the interaction with an electron cloud [1] can be simulated with the HEADTAIL code. The simulation model and all the underlying physical mechanisms have been widely discussed in Ref. [2]. The simulation uses a Particle-In-Cell scheme to calculate the fields through which a positively charged particle beam and an electron cloud lumped at one or more locations along the beam orbit interact.
As input numbers, the code requires various beam and machine parameters, which will be commented in detail in the following section. The program computes the evolution of the bunch centroid and rms-size as functions of time over an adjustable number of turns.
To obtain the program from the web, one has to connect to the SL/AP electron cloud web page

'http://wwwslap.cern.ch/collective/electron-cloud/'

and follow the link HEADTAIL. From here it is possible to download a tarred file containing the source code, all the library files necessary to compile it and the input file. The file can be untarred then by simply typing

'tar -xvf headtail.tar' .

The files 'headtail.c', 'example.cfg', 'Makefile', 'fourtrans2.c' and the subdirectory 'fftw' will be thus created. Now the 'Makefile' must be edited in order to set the C compiler and compiler flags. Then the executable 'xheadtail' will be created by simply typing 'make'.

2 Input file

The goal of this section is to provide a thorough explanation of all the variables that must be set in the input file 'example.cfg'. Editing the file will display a sequence of lines containing short descriptions of the parameters followed by numbers, which need to be adjusted according to the case that is going to be simulated. The descriptions always show within brackets the physical dimension of the parameter they refer to (if no dimension is indicated, the parameter is dimensionless). The variable identifiers in the source code to which the input numbers correspond can be easily tracked by editing the source code and looking into the 'read_data ()' subroutine. For sake of clarity, these variable names are also highlighted in parentheses before each of the extended descriptions and comments given below.
To begin with, the following variables are not input numbers, but the user needs to be aware of their meanings to appropriately set them in each run (they are

included in the definition of constants in the source code): 'NPR' and 'NEL', number of bunch macroparticles and number of macroelectrons, respectively; 'NBIN', number of slices into which the bunch is subdivided (a criterion is to choose 'NPR' and 'NBIN' such that 'NPR'/'NBIN'$> 10^3$); 'NDC', number of cells used to build distribution functions from macroparticle data; 'NPR1', maximum number of macroparticles expected to fall into one single bunch slice during the simulation (if the number of macroparticles in one slice happens to become larger than 'NPR1', the program execution is immediately stopped and a warning is printed on screen).

- `Flag_for_bunch_particles_(1->protons_2->positrons_3->De+):`
  (ipr_pos)
  It sets the kind of particles the bunch is made of: protons (1), positrons (2) or single charged Deuterium ions (3).
- `Average_electron_cloud_density_along_the_ring_[1/m3]:`
  (dcloud)
  This is the density of an electron cloud supposed to be uniformly distributed along the ring. If the electron cloud is known to be local (like the electron cooler case), the number that must be input at this line is its density multiplied by the ratio between the length of the section filled with an electron and the total ring circumference (like for instance in the simulations of a bunch interacting with a strongly detuned electron cooler [3]). In practical application, this density is inferred from independent simulations of the electron cloud build up during the passage of a bunch train, e. g. using the code ECLOUD [4].
- `Number_of_particles_per_bunch:`
  (npr0)
  It sets the bunch population number N_b.
- `Average_horizontal_beta_function_of_the_ring_[m]:`
  `Average_vertical_beta_function_of_the_ring_[m]:`
  (betax0, betay0)
  The average beta functions $\beta_{x,y}$ have to be input at these lines if the electron cloud is uniformly distributed all over the ring. If the electron cloud is local, it is sufficient to use here the actual values of the beta functions at the location where the cloud is (or an averaged value along that region if they change significantly).
- `Bunch_length_(rms_value)_[m]:`
  (sz0)
  Note that the rms-bunch length σ_z is required in m.
- `Horizontal_beam_size_(rms_value)_[m]:`
  `Vertical_beam_size_(rms_value)_[m]:`
  (sx0, sy0)

The average transverse betatron beam sizes $\sigma_{x,y}$ are input at these lines. These are related to the emittances via the simple relations:

$$\sigma_{x,y} = \sqrt{\epsilon_{x,y}\bar{\beta}_{x,y}}$$

In addition, the ring might have a non-zero average horizontal dispersion, which is accounted for on an input line further down. If the electron cloud is localized, here the values of the bunch rms-sizes at that specific section have to be input.

- `Longitudinal_momentum_spread:`
  (dp0)
  Longitudinal momentum spread or energy spread $\delta p/p_0$ (in rms units). Its value must be matched to the bunch length, synchrotron tune and momentum compaction factor when simulating a stationary bunch with single particles performing synchrotron oscillations. The matching condition involves also the parameters which are input on the next three lines, and reads:

$$\frac{R_0|\eta|\delta p/p_0}{Q_s\sigma_z} = 1$$

- `Synchrotron_tune:`
  (nus)
  The synchrotron tune Q_s is the number of longitudinal oscillations per turn. It is typically much lower than the particle transverse (or betatron) tunes.
- `Momentum_compaction_factor:`
  (alpha)
  The momentum compaction factor α depends on the optics of the machine. It is used in this code to calculate the slippage factor

$$\eta = \frac{1}{\gamma^2} - \alpha \ ,$$

  which determines the particle longitudinal dynamics (below or above transition).
- `Ring_circumference_length_[m]:`
  (circ)
  Circumference length: $C = 2\pi R_0$.
- `Relativistic_gamma:`
  (gammar)
  The relativistic factor γ which must be input on this line can be derived from the beam energy.
- `Number_of_kick_sections:`
  (nkick)
  This parameter sets the number of kicks received by the bunch per turn to model its interaction with the electron cloud. A typical number is 1 to 4.

- `Number_of_turns:`
  (nturn)
  The number of simulated turns N_t sets the total simulation time $T_{sim} = N_t C/\mathrm{c}$.

- `Multiplication_factor_for_pipe_axes:`
  `Multiplication_factor_for_pipe_axes`
  (ielsizex, ielsizexy)
  This couple of input parameters is used to set the dimensions of the vacuum chamber or the simulation grid in units of rms beam sizes defined above. In simulations where electron cloud is switched off, through these lines one can set the size of the vacuum chamber. When the electron cloud is to be accounted for, then these lines set the size the simulation grid, which need not be equal to the pipe size. In this case we are just simulating a defined small region around the beam, and the loss conditions explained in the following do not really apply, but they are kept active because it anyway does not make sense to keep calculating fields when large numbers of particles exceed the grid boundaries. A warning is printed on screen if this occurs.
  When bunch particles exceed the boundaries specified in these lines, they are assumed to be lost. When electrons hit the boundary, they are reflected back but with a very low energy around 10 eV. The electron cloud is typically distributed over 90% of this space and its specific distribution function which can be selected from several options in one of the following input lines.

- `Horizontal_tune:`
  `Vertical_tune:`
  (htune, vtune)
  The transverse tune of the machine, $Q_{x,y}$, are set on these lines.

- `Horizontal_chromaticity:`
  `Vertical_chromaticity:`
  (csix, csiy)
  Horizontal and vertical chromaticities can be specified on these lines in $Q'_{x,y}$ units.

- `Flag_for_synchrotron_motion_(0->off_1->on_2->deb):`
  (isyn)
  This flag allows simulating a frozen bunch (0), a bunch subject to a linear rf-force (1), or a debunching bunch (2), that means a bunch in which particles run freely according to their initial momentum distribution. By setting this flag to (3) we can also simulate a bunch in a barrier bucket. In the latter case, the length of the bucket is $4\sigma_z$ and the bunch particles coast freely between the extremes of the bucket. When they hit the bucket wall they are elastically reflected.

- `Scale_factor_for_electrons_size:`
  (imesize)
  This parameter needs to be specified only if further down the soft-Gaussian approximation is chosen to simulate the interaction between electrons and bunch. Obsolete.

- `Switch_for_wake_fields:`
  (iwake_flag)
  If this flag is non-zero, an additional 3D kick is given to the bunch particles turn by turn. The kick sums up the effect of all the trailing fields left behind by the preceding bunch slices that have crossed a resonant structure.

$$\Delta x' = \frac{e^2}{E_0} \sum_{j=1}^{k-1} N_j \bar{x}_j W_1 \left[-(k-j)\Delta l\right]$$

$$\Delta y' = \frac{e^2}{E_0} \sum_{j=1}^{k-1} N_j \bar{y}_j W_1 \left[-(k-j)\Delta l\right]$$

$$\Delta \left(\frac{\delta p}{p_0}\right) = \frac{e^2}{E_0} \sum_{j=1}^{k} N_j W_0' \left[-(k-j)\Delta l\right]$$

  The resonant structure is characterized by the typical resonator impedances:

$$Z_\perp^1(\omega) = \frac{\omega_{r\perp}}{\omega} \frac{Z_t}{1 + iQ_\perp \left(\frac{\omega_{r\perp}}{\omega} - \frac{\omega}{\omega_{r\perp}}\right)} \ ,$$

$$Z_\parallel^0(\omega) = \frac{R_{sh}}{1 + iQ_\parallel \left(\frac{\omega_{r\parallel}}{\omega} - \frac{\omega}{\omega_{r\parallel}}\right)} \ ,$$

  where the shunt impedances Z_t and R_{sh} (having dimensions of Ω/m and Ω, respectively), the resonance frequencies $\omega_{r\perp}$ and $\omega_{r\parallel}$, and the quality factors $Q_\perp$ and $Q_\parallel$ are set in the following lines.

- `Switch_for_pipe_geometry_(0->round_1->flat_2->h-off):`
  (i_pipe)
  This switch allows the choice between a round (0) or flat (1) chamber geometry. Option (2) switches artificially off the horizontal wake while keeping the vertical on. Following Yokoya's diagrams (that are rigorously true for reistive wall impedance but can be extended to broad band impedances induced by flat structures, too) [5], we set equal wakes in x and y for a round chamber, and we add an incoherent quadrupole field for a flat chamber ($(a-b)/(a+b) > 0.2$, $2a$ and $2b$ being the dimensions of the chamber, rectangular or elliptical:)

$$\Delta x' = \frac{e^2}{E_0} \sum_{j=1}^{k-1} N_j 0.33 (\bar{x}_j - x) W_1 \left[-(k-j)\Delta l\right]$$

$$\Delta y' = \frac{e^2}{E_0} \sum_{j=1}^{k-1} N_j 0.66 (\bar{y}_j + \frac{1}{2} y) W_1 \left[-(k-j)\Delta l\right]$$

Option (3) allows simulating the effect of a resistive wall wake field. The expression for the fields is taken from Ref. [6]:

$$W_m(z) = -\frac{1}{2\pi^2 \epsilon_0 r_p^{2m+1}(1+\delta_{m0})} \sqrt{\frac{c}{4\pi\epsilon_0\sigma}} \frac{1}{|z|^{0.5}} C \ ,$$

where σ is the conductivity of the chamber. The order 0 is used for the longitudinal field ($\propto dW_0(z)/dz$), and the order 1 for the transverse dipole field. Before using this option, the user should make sure that the bunch slicing is such that we fall into the range of applicability of the above formula (see Ref.[6]; the condition is $\Delta z_{sl} >> \chi^{1/3} r_p$ with $\chi = 1/(Z_0 \sigma r_p)$).
Option (4) corresponds to a pure energy loss for bunch particles turn by turn, e. g. to model the effect of synchrotron radiation. The value of the momentum loss per particle can be adjusted to the desired value by editing the source file and assigning a fixed value to the variable 'kick_z' in the part of the code which computes the interaction of the bunch with non-ecloud wake fields.

- `Res_frequency_of_broad_band_resonator_[GHz]:`
  (freqr)
  The resonant frequency $\omega_{r\perp}/(2\pi)$ of the transverse broad-band impedance is entered in GHz.
- `Transverse_quality_factor:`
  (merit)
  The quality factor of the transverse resonator $Q_\perp$ is typically in the order of unity.
- `Transverse_shunt_impedance_[MOhm/m]:`
  (zetat)
  The value of the transverse shunt impedance Z_t is input in MΩ/m on this line.
- `Res_frequency_of_longitudinal_resonator_[MHz]:`
  (freqrz)
  The resonant frequency $\omega_{r\parallel}/(2\pi)$ of the longitudinal impedance is assigned in MHz.
- `Longitudinal_quality_factor:`
  (meritz)
  The quality factor of the transverse resonator $Q_\parallel$ can be much higher than unity for cavity beam-loading. The simulated bunch must be assumed to be far enough from other bunches not to feel the effect of trailing fields induced by preceding bunches in the cavities, which are anyway not taken into account in this simulation.

- `Longitudinal_shunt_impedance_[MOhm]:`
  (rshz)
  The value of the longitudinal shunt impedance R_{sh} is input in MΩ on this line.

- `Flag_for_the_tune_spread_(0->no_1->space_charge_2->random):`
  (ishift)
  A tune spread can be introduced by setting the flag on this input line. Setting the flag to 1 causes an incoherent space charge tune spread which is recalculated for each macroparticle at each time step, according to its actual location along the bunch, by using the well-known Laslett formula:

$$\Delta Q_{u,sc,inc} = -\frac{r_c \lambda(z) R_0^2}{Q_{0u}\gamma^3 \sigma_u(z)(\sigma_x(z)+\sigma_y(z))} \ ,$$

  where u stands for x or y and $\lambda(z)$ is the line density along the bunch.
  This tune shift is incoherent because the rotation matrix is applied to the macroparticle's transverse coordinates relatively to the local bunch centroid (that is the centroid of the bunch slice where the macroparticle is located at the time when its tune is evaluated), and the centroid coordinates are then added back after the rotation. As it should be, this tune spread alone cannot cause decoherence of the centroid motion, but it affects the beam size evolution. If the ring is perfectly linear, it appears clearly in the spectra of $\sigma_x(t)$ and $\sigma_y(t)$, but it is not detectable in the centroid spectra of a transversely kicked bunch $< x > (t)$ and $< y > (t)$.
  Setting the flag for the tune spread to 2 generates a random tune spread independent of z as large as the maximum space charge tune spread at the start time of the simulation. This is a coherent spread and can be seen also in the spectrum of the transverse coherent centroid motion.

- `Flag_for_the_e-field_calc_method_(0->no_1->soft_Gauss_2->PIC):`
  (i_method)
  The interaction with the electron cloud can be carried out in different ways. First option is to switch it off by setting this flag to 0.
  The value 1 causes the interaction between bunch and electron cloud to be calculated considering all electrons as finite size macro-particles and their fields given by the soft-Gaussian approximation. The beam field is also calculated slice by slice with the soft-Gaussian approximation. As this computation is very time-consuming, it is recommended that if this option is chosen, the variables 'NEL' and 'NPR' in the source code be both set to 10^4 as maximum value.
  The value 2 uses a Particle In Cell calculation of beam and cloud fields. The PIC module was originally written by D. Schulte for his GUINEA-PIG code that simulates the beam-beam interaction in linear colliders [7], and then adapted to the electron cloud-beam interaction and incorporated in the

HEADTAIL code. At each time step, it smears all particles on a grid (which is chosen to be 10% larger than the cloud) and then evaluates the resulting fields at the grid points. After interpolating the fields back from the grid points to the actual particle positions, the particles are pushed on to the next time step. This method is much quicker than the soft-Gaussian, and therefore numbers of particles in the order of 10^5 can be used. Typically 'NPR' (number of macroparticles representing the bunch) is also determined by the required 'NBIN' (number of slices into which the bunch is subdivided). The criterion is to set these numbers such that at least 1000 bunch particles are contained in each bunch slice.

- `Magnetic_field_(0->no_1->dipole_2->solenoid_3->combined):`
  (i_dip)
  This number is used to choose the magnetic field configuration in which the electrons are made to evolve during the interaction with the passing bunch. Option 0 means simply field-free space.
  Option 1 freezes the electron motion along the vertical direction. This models a strong vertical dipole field. The cyclotron motion of the electrons around the field lines is neglected for sake of speed: to resolve the quick oscillations of the electrons due to strong magnetic fields we would in fact need time steps up to 100 times smaller than those normally used.
  Option 2 uses an embedded Runge-Kutta algorithm to solve the electron motion in a weak solenoid field. The value of the solenoid field B_z is to be specified on one of the following lines.
  Option 3 confines the electron motion on the field lines of a combined function magnet. This approximation can only work if the dipole component of the field is much larger than its quadrupolar component. Values for the field coefficients are those of the PS ring at CERN, that is $B_0 = 1.2$ T and $G = 5.2$ T/m. If needed, these values can be easily changed by editing the source file and modifying the lines where the two variables 'bdip' and 'bgrad' are assigned.

- `x-kick_amplitude_at_t=0_[m]:`
  `y-kick_amplitude_at_t=0_[m]:`
  (xkick, ykick)
  The initial distribution of bunch particles is transversely centered around the values input at this line. The default values are (0,0) (bunch not transversely kicked).

- `Flag_for_the_proton_space_charge:`
  (i_space)
  Setting this parameter to a non-zero value will cause an additional kick on the bunch particles due to the bunch transverse self-field. This models a beam-beam kick but may be rough for a space charge effect for high intensity bunches.

- `Flag_for_the_sc-rotation(0->local_centroid_1->bunch_centroid):`
  (i_rot)
  The space charge rotation can be executed around the centroid of the bunch slice where the particle we are transporting is sitting (which is correct for space charge) or around the bunch centroid (which is more correct for the beam-beam).

- `Solenoid_field_[T]:`
  (bsol)
  As explained above, this line allows the assignment of the amplitude of a solenoid field present in the region where the electron cloud is. This only makes sense if previously a solenoid type of magnetic field configuration had been chosen. In other cases, the value input on this line does not change anything in the program execution.

- `Switch_for_amplitude_detuning:`
  (i_oct)
  Detuning with amplitude might also cause a spread in the particle tune distribution. By choosing 1 on this line, this spread is activated according to:

$$\Delta Q_x = Q_{pxx} I_x + Q_{pxy} I_y \ ,$$
$$\Delta Q_y = Q_{pyx} I_x + Q_{pyy} I_y \ ,$$

  where $I_x = 1/(2\beta_{x0})(x^2 + \beta_{x0}^2 x'^2)$ and $I_y = 1/(2\beta_{y0})(y^2 + \beta_{y0}^2 y'^2)$ are the single particle actions and the Q-coefficients are initialized in the source code as those from the SPS with standard octupole settings [8], $Q_{pxx} = -849.8\text{m}^{-1}$, $Q_{pxy} = Q_{pyx} = 1756\text{m}^{-1}$, $Q_{pyy} = -2310\text{m}^{-1}$. These values can be changed by simply editing the source code and modifying the assignments of the variables 'qpxx', 'qpxy' and 'qpyy' in the 'initialization' subroutine.

- `Coherent_centroid_motion_(0->off_1->on):`
  (i_incoh)
  In order to isolate pure emittance growth from coherent dipole instability, this line gives the option to artificially switch off the bunch centroid coherent motion. The flag input at this line must then be set to 0.

- `El_distrib_(1->Rect_2->Ellip_3->[1_strp]_4->[2_strp]):`
  (elflag)
  The bunch can interact with an electron cloud having an initial distribution which can be uniform inside the beam pipe, or have one or two denser stripes located at some horizontal positions.
  Option 1 gives a uniform electron cloud inside a rectangular boundary.
  Option 2 gives a uniform electron cloud inside an elliptical boundary.
  Option 3 gives a 1-stripe distribution of the electron cloud. The stripe stretches between -2σ_x and 2σ_x around the beam and occupies the whole area in the vertical direction.

Option 4 gives a 2-stripes distribution of the electron cloud. Horizontal location and thickness of these stripes are assigned down below. Vertically they stretch all over the beam pipe.

- `Linear_coupling_switch(1->on_0->off):`
  (i_coup)
  Linear coupling between the transverse planes can be introduced by setting the flag on this line to 1.
- `Linear_coupling_coefficient_[1/m]:`
  (key_xy)
  This number sets the strength of the linear coupling. A typical number is $0.015\mathrm{m}^{-1}$ for the CERN SPS when the coupling has not been properly corrected with the skew quadrupole setting [9].
- `Average_dispersion_function_in_the_ring_[m]:`
  (dispx)
  The effect of dispersion can also be studied by giving on this line a number for the average dispersion function along the ring. When this number is nonzero, the horizontal positions of the macroparticles are changed before interaction with the cloud and/or with the wake field and get correlated to their momenta at that time step. After the interaction, the dispersion contribution to the σ_x is subtracted out again.
- `Position_of_the_stripes_[units_of_sigmax]:`
  (strpos)
  The value on this line defines the horizontal position of the center of the two stripes in units of σ_x starting from the beam center. It is only used if a 2-stripes distribution for the electron cloud had been chosen beforehand.
- `Width_of_the_stripes_[units_of_sigmax]:`
  (strsig)
  The value on this line defines the horizontal width of the stripes in units of σ_x.
- `Kick_in_the_longitudinal_direction_[m]:`
  (z0_kick)
  Like the kick in either transverse direction, this line gives the opportunity to define a value for a longitudinal kick given in terms of bunch center displacement in the bucket area.
- `Number_of_turns_between_two_bunch_shape_acquisitions:`
  (n_diag)
  Bunch shapes are acquired every 'n_diag' turns.

3 Output Files

We list here all the output files, how they are named and the information that they contain.
We assume that the used input file was called 'example.cfg'. The output files are all in ascii format. In what follows we will often refer to *blocks* of data within the same file: by our definition, we assume *blocks* to be always sets of data separated by two blank lines. This feature can be easily removed or changed by editing out in the source file the lines where the blank lines are entered or just replacing them with other preferred separators.
The following files are generated:

- 'example_prt.dat'; This is by far the most important of the output files because it contains the most information on the bunch evolution over the simulation time. The file is made up of 16 columns which we comment below.

  (1) time
  (2) x averaged all over the bunch particles (horizontal centroid position)
  (3) x' averaged all over the bunch particles
  (4) y averaged all over the bunch particles (vertical centroid position)
  (5) z averaged all over the bunch particles (longitudinal centroid position)
  (6) $\delta p/p_0$ averaged all over the bunch particles
  (7) rms horizontal beam size σ_x
  (8) rms vertical beam size σ_y
  (9) rms bunch length σ_z
(10) or (12) Horizontal beam emittance (not normalized) ϵ_x
(11) or (13) Vertical beam emittance (not normalized) ϵ_y
  (15) yz correlation
  (16) fraction of the beam not yet lost at the pipe walls.
  (17) fraction of beam macroparticles included in the calculation after the slicing process. It is actually the fraction of bunch particles having longitudinal position between $z_0 - 2\sigma_z$ and $z_0 + 2\sigma_z$.

  The first column contains the time points at which the corresponding values on the other columns have been calculated. Therefore, plotting for instance 2 versus 1 will show the time evolution of the horizontal coordinate of the bunch centroid, and so on.

- 'example_hdtl.dat'; This file contains ('nkick'×'nturn')/'n_diag' blocks consisting of 5 columns and 'NBIN' rows.
  The first column provides the longitudinal position along the bunch.
  The second column contains the horizontal difference signal from a pick-up

monitor, that is the product $N(z)* < x > (z)$ along the bunch.
The third column contains the vertical difference signal from a pick-up monitor, that is the product $N(z)* < y > (z)$ along the bunch.
The fourth column contains the horizontal difference signal from a quadrupole pick-up monitor, that is the product $N(z) * \sigma_x(z)$ along the bunch.
The third column contains the verical difference signal from a quadrupole pick-up monitor, that is the product $N(z) * \sigma_y(z)$ along the bunch.
The signals provided in this file always stretch between $< z > -2\sigma_z$ and $< z > +2\sigma_z$ because the bunch slicing is performed anew at each turn within this interval. In our computation, the longitudinal dependence of centroids and rms-sizes derives in fact from the slice by slice local centroids and rms-sizes.

- 'example_bunchds.dat'; This file contains ('nkick'×'nturn')/'n_diag' blocks consisting of 2 columns and 'NDC' rows.
  The n-th block contains the longitudinal distribution function of the bunch at the time n×'n_diag'. This file can be used to construct a mountain range diagram of the bunch evolution, particularly useful when the longitudinal wake field is also taken into account in the calculation and therefore the longitudinal bunch shape can significantly change. The mountain range diagram can be easily plotted by displaying bunch profiles at subsequent times on top of each other.
- 'example_inph.dat;' This is a memo file that simply displays in few lines some basic simulation and beam parameters ('NPR', 'NEL' and 'NBIN'; beam and electron cloud sizes; initial space charge tune shifts; percent of beam loss at the end of the simulation).
- 'example_prb.dat'; This can be a huge file, which contains the bunch phase space sampled every 'n_diag' turns. The bunch phase space is obtained by writing to this file the 6-dimensional phase space coordinates of 2000 macro-protons. The phase space at each sampled turn is a block of the file 'example_prb.dat'. The phase space coordinates are recorded in the following order: x, x', y, y', z, $\delta p/p_0$.
- 'example_sample.dat'; This file is made of 7 columns with time plus the six phase space coordinates of a sample bunch particle. Data are recorded at each time step. First column contains time, the other six columns list the phase space coordinates of the chosen particle in the same order as specified in the description of the previous output file. By default, the particle whose motion is tracked in this file is the one having index 'NPR-1'. When a particle hits the beam pipe, it is considered lost and therefore swapped with the one having the lowest index available. Its motion is then frozen in the following turns. Thus, by choosing to track the particle 'NPR-1' we make sure that we are looking at an active particle's motion throughout the whole simulation time. However, this index can be easily changed in the source file

by editing the condition to write to this specific output file and recompiling the program.

4 Some examples of use

The program HEADTAIL has been used to simulate the electron cloud instability in the CERN Proton Synchrotron [10]. The parameters used for one of the runs in the simulation campaign can be read in Table I.

Table 1: PS parameters used in the simulations. As far as the beam emittance is concerned, the rms physical value is quoted here.

Bunch population (N_b)	8×10^{10} protons
Transverse rms emittances ($\epsilon_{x,y}$)	0.09/0.09 μm
Chamber half-aperture (h_x)	70 mm
Chamber half-aperture (h_y)	35 mm
Tunes ($Q_{x,y,s}$)	6.25/6.25/0.0015
Bunch rms length (σ_z/c)	2.5 ns
Rms momentum spread	4.4×10^{-4}
Longitudinal 2σ emittance (ϵ_l)	0.35 eVs
Circumference	628 m
Dipole field (B_{y0})	1.256 T
Field gradient (G)	5.2 T/m
Relativistic γ	27.7
T_{rev}	2.2 μs
Number of turns	2000
Average beta-functions ($\beta_{x,y}$)	16/16 m
Average Dispersion (D_x)	2.56 m
Mom. compaction (α)	0.027
Chromaticities ($\xi_{x,y}$)	corrected in both planes
Electron cloud density	$3 \times 10^{12}\mathrm{m}^{-3}$

With this set of parameters, the beam was found to be strongly unstable. As the name of the input file is 'curr4.cfg', the bunch evolution over 2000 turns can be monitored by plotting columns 2 through 16 versus column 1 from the output file 'curr4_prt.dat'. Figs. 1 to 9 show respectively:

- Figure 1: horizontal (column 2) and vertical (column 4) centroids versus time (column 1). There is evidence for a dipole mode instability, which is stronger in the vertical plane, causing an exponential growth of the centroid amplitude oscillation.

- Figure 2: longitudinal mean position (column 5) and momentum spread (column 6) versus time (column 1). They are both basically zero until beam loss occurs. Then the bunch gets longitudinally displaced because of the localized beam loss, and a synchrotron oscillation is excited.
- Figure 3: σ_x (column 7), σ_y (column 8) and σ_z (column 9) versus time (column 1). The transverse beam rms size also increases exponentially, and there is a large oscillation due to mismatch. The longitudinal rms-size exhibits at first a very small amplitude $2Q_s$ oscillation, then it drastically changes because of beam loss.
- Figure 4: ϵ_x (column 10 and 12) and ϵ_y (column 11 and 13) versus time (column 1). The emittance grows exponentially in both planes.
- Figure 5: yz correlation (column 14) versus time (column 1). As expected, the correlation between the vertical and the longitudinal plane (which is driven by the electron cloud) increases exponentially. Its values are normalized to the instantanous bunch rms-size.
- Figure 6: fraction of bunch particles which have not yet been lost (column 15) and fraction of bunch particles that are longitudinally still within $(-2\sigma_z, 2\sigma_z)$ (column 16) versus time (column 1). We start losing particles at 2.5 ms and at the end of the simulation (2000 turns, which correspond to about 4.2 ms) we are left with half of the initial number of bunch particles. Even after the beam loss has occurred and the bunch is deformed, more than 92% of the total amount of macroparticles left is still used in the slicing process (with respect to the initial almost 96% coming from the Gaussian distribution truncated at 2σ).

Using the output file 'curr4_bunchds.dat', we can reconstruct the bunch shape evolution throughout our 4.2 ms simulation. As in the input file 'n_diag' had been set to 80, the bunch profile has been acquired every 80 turns. This makes a total of 25 bunch shapes recorded all along the 2000 turns simulation and plotted in the mountain range diagram of Fig. 7. In this simulation the bunch was perfectly matched to the bucket and no longitudinal wake field was activated. Therefore we did not expect to see drastical changes in the bunch shape during the simulation. Only when beam loss occurs, we can see a small bump in the right tail of the Gaussian distribution. Also, as the beam is being lost, the distribution relaxes because its integral changes (the distribution is not normalized but proportional to the number of particles). To show these two effects even more clearly, we plot initial and final bunch distribution in Fig. 8.

Finally, from the output file 'curr4_hdtl.dat' the dipolar and quadrupolar difference signals can be viewed at different times. The data are acquired at the same times at which the bunch shapes are written to a file, that means every 'n_diag' turns. The 25 traces coming from our PS run are plotted in Fig. 9 for the vertical dipole and quadrupole Δ signals (blocks 0 through 24 of columns 3

and 5 versus column 1). These diagrams are more interesting than the horizontal ones, because, unlike the latter, they clearly show a coherent behaviour along the full bunch length, which develops in time. When the coherent motion has reached large amplitudes, the correlation between subsequent traces becomes more difficult to detect because of the sparse sampling applied in this example. Note also that traces from the last part of the bunch evolution (those in the top) stretch over a shorter length due to beam loss.

As second example of use of the code, we show the evolution of a 26 GeV SPS bunch under the effect of a transverse broad band impedance as inferred from our measurements [11]. The broad band impedance parameters have been estimated from theory and further adjusted to better match the observed evolutions ($Q = 1$, $Z_t \approx 10\text{M}\Omega/\text{m}$, $\omega_r = 2\pi \times 1.3\text{GHz}$). The vertical motion of the centroid can be damped or unstable according to the sign of chromaticity (see Fig. 10). From the Fourier analysis of these time signals, growth and decay rates as function of chromaticity have been computed and compared to those measured. Figure 11 shows the excellent agreement between simulation and experiment. The deviation of the straight line from the origin, which we observe both in measurements and in simulations, is likely to be explained as a space charge induced effect which can stabilize the bunch even at low negative chromaticities (incoherent tune spread).

Simulation of debunching an intense bunch in the SPS can also reveal very interesting features. Figure 12 displays mountain range diagrams of the debunching beam for the two cases where the longitudinal wake field from the 200 MHz cavities ($Q = 140$ and $R_s = 5\text{M}\Omega$) is either taken into account or not. The longitudinal impedance causes a microwave instability, which can also affect the transverse motion.

Finally, we show the results of the simulation of a 26 GeV bunch in the SPS in a regime of strong space charge (10^{11}, $\sigma_{x,y} = 2$ mm, $\sigma_z = 0.14$ m) and with amplitude dependent detuning. Apparently, the space charge can significantly affect the centroid motion, if the single particle tunes vary with the amplitude oscillation. Decoherence is slowed down or even inhibited by the space charge forces (Figs. 13). The sign of the detuning with amplitude, which does not affect the centroid evolution without space charge, appears to have a significant influence if space charge is taken into account. In Fig. 14 we show the proton vertical phase space at the end of the simulation for all three cases plotted in Fig. 13. When space charge is not included in the simulation, the bunch simply decoheres and occupies all the phase space available around the kick amplitude (picture on top). In the cases with space charge, amplitude detuning the beam no longer filaments quickly despite of the detuning with amplitude, but there is halo formation evidenced by a fraction of particles reaching large amplitudes. The phase space plots are obtained from the output file 'namefile_prb.dat'.

5 Acknowledgements

The authors would like to thank D. Schulte, F. Ruggiero, H. Burkhardt, E. Frulloni, M. Hayes, S. Redaelli, R. Tomàs, E. Métral, M. Giovannozzi and E. Benedetto for fruitful discussion, contributions and encouragement to develop the program in all its features and to write up this guide.

References

[1] G. Rumolo, F. Ruggiero, F. Zimmermann, "Simulation of the Electron-Cloud Build Up and its Consequences on Heat Load, Beam Stability, and Diagnostics", PRST-AB 012801 (2001).

[2] G. Rumolo and F. Zimmermann, "Electron Cloud Simulations: Beam Instabilities and Wake Fields", in Proceedings of the Mini-Workshop on Electron-Cloud Simulations for Proton and Positron Beams, CERN Geneva, 2002, edited by G. Rumolo, F. Zimmermann (Yellow Report CERN-2002-01).

[3] O. Boine-Frankenheim, I. Hofmann, G. Rumolo and F. Zimmermann, "Driving the Electron Cloud Instability by an Electron Cooler", in Proceedings of the Mini-Workshop on Electron-Cloud Simulations for Proton and Positron Beams, CERN Geneva, 2002, edited by G. Rumolo, F. Zimmermann (Yellow Report CERN-2002-01). CERN-SL-2002-035

[4] G. Rumolo and F. Zimmermann, "Practical User Guide for ECLOUD" CERN-SL-Note-2002-016

[5] K. Yokoya, Part. Acc., **41**, 221, (1993).

[6] A. W. Chao, "Physics of Collective Beam Instabilities in High Energy Accelerators" J. Wiley & Sons, (1993).

[7] D. Schulte, "Beam-Beam Simulations with GUINEA-PIG", in Proceedings ICAP98, Monterey, CA (1998).

[8] M. Hayes, F. Schmidt and R. Tomàs, "Resonance Driving Terms experiment in the SPS at 26 GeV in 2001", SL-Note-2001-037 MD

[9] R. Tomàs, private communication.

[10] M. Giovannozzi *et al.*, "Electron Cloud Build up and Instability: Numerical Simulations for the CERN Proton Synchrotron" submitted to Phis. Rev. ST-AB and in Proceedings of the Mini-Workshop on Electron-Cloud Simulations for Proton and Positron Beams, CERN Geneva, 2002, edited by G. Rumolo, F. Zimmermann (Yellow Report CERN-2002-01).

[11] H. Burkhardt, G. Rumolo and F. Zimmermann, "Measurements of SPS Single Bunch Coherent Tune Shifts and Head-Tail Growth Rates in the Year 2001" CERN, SL-Note-2001-043 (MD), 2001 (unpublished).

[12] H. Burkhardt, G. Rumolo and F. Zimmermann, "Coherent Beam Oscillations and Transverse Impedance in the SPS" in Proceedings of 8th European Particle Accelerator Conference, Paris, 3-7 June 2002, p.1449

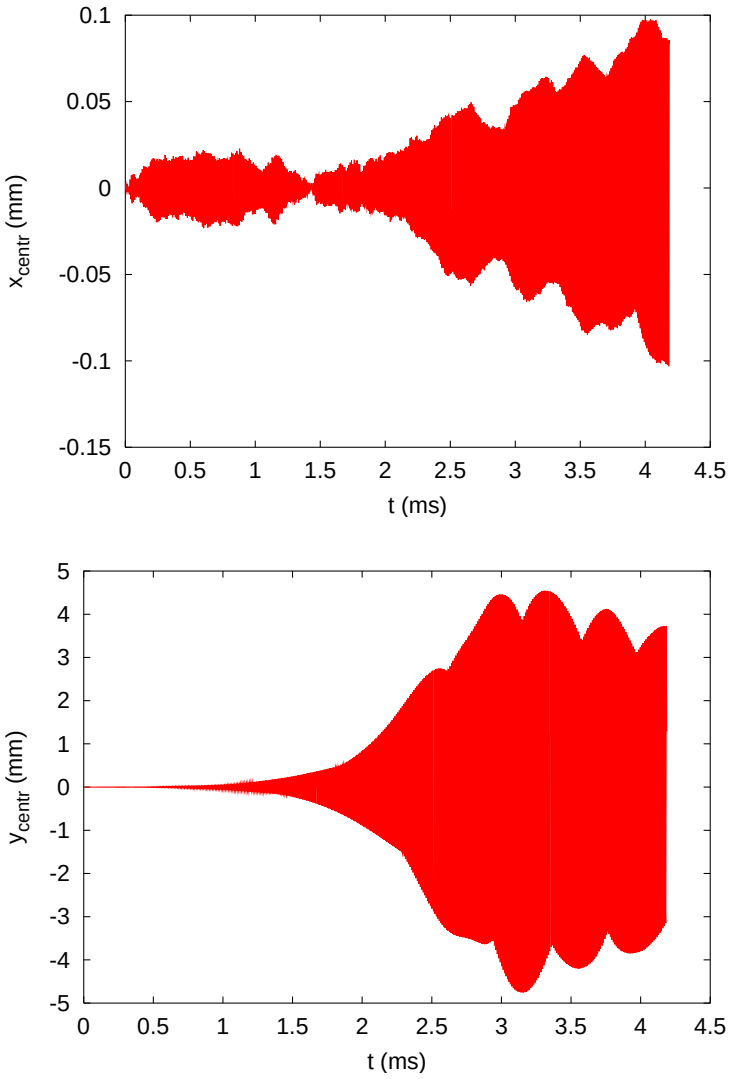

Figure 1: Time evolution of horizontal and vertical centroids.

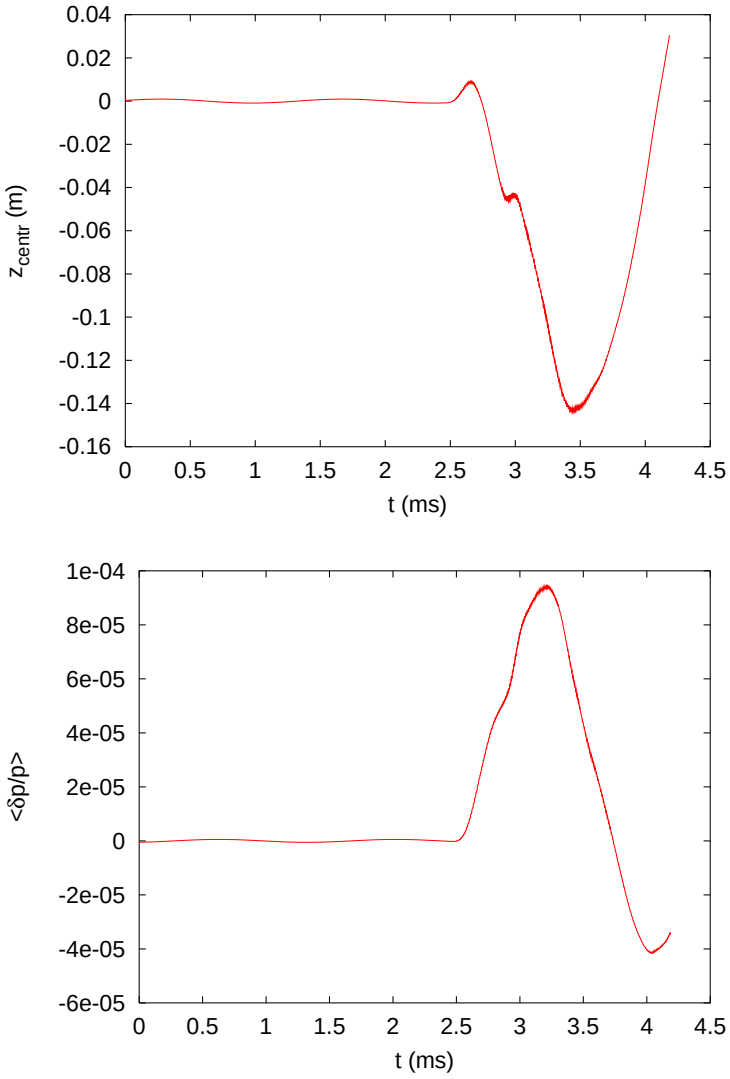

Figure 2: Time evolution of longitudinal average position and average momentum spread.

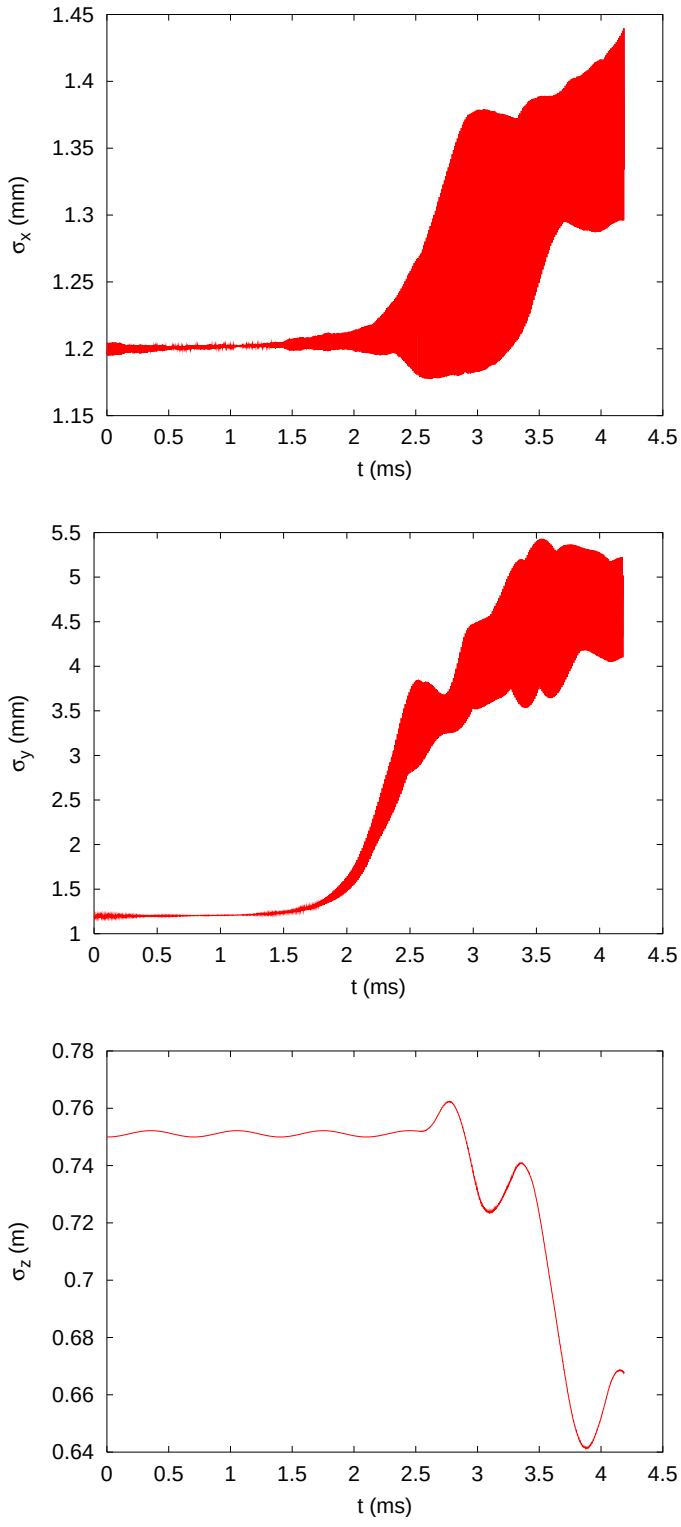

Figure 3: Time evolution of σ_x, σ_y and σ_z.

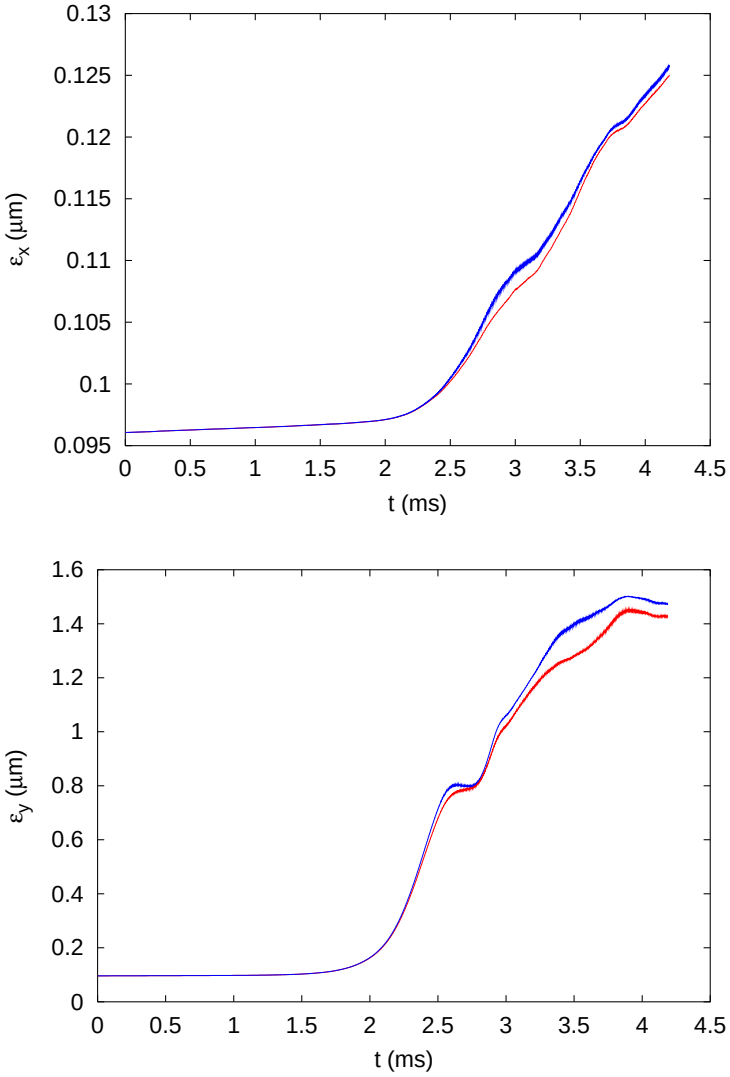

Figure 4: Time evolution of ϵ_x and ϵ_y

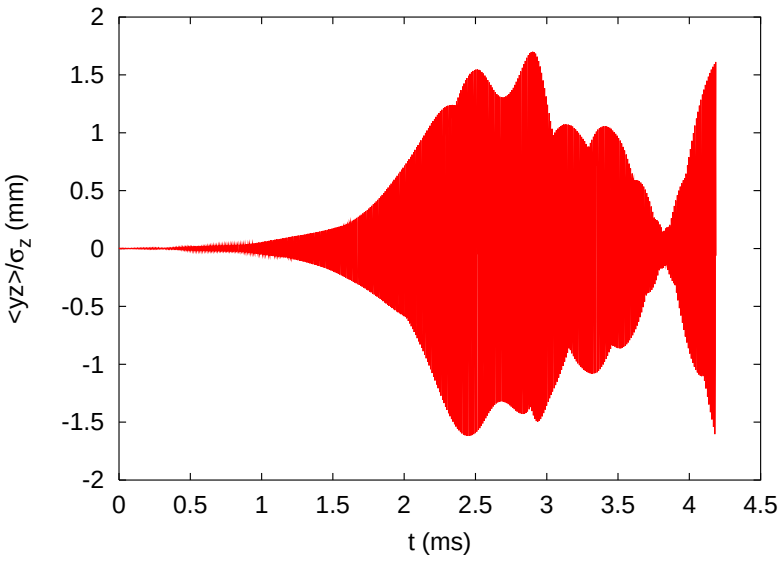

Figure 5: Time evolution of the yz correlation among bunch particles.

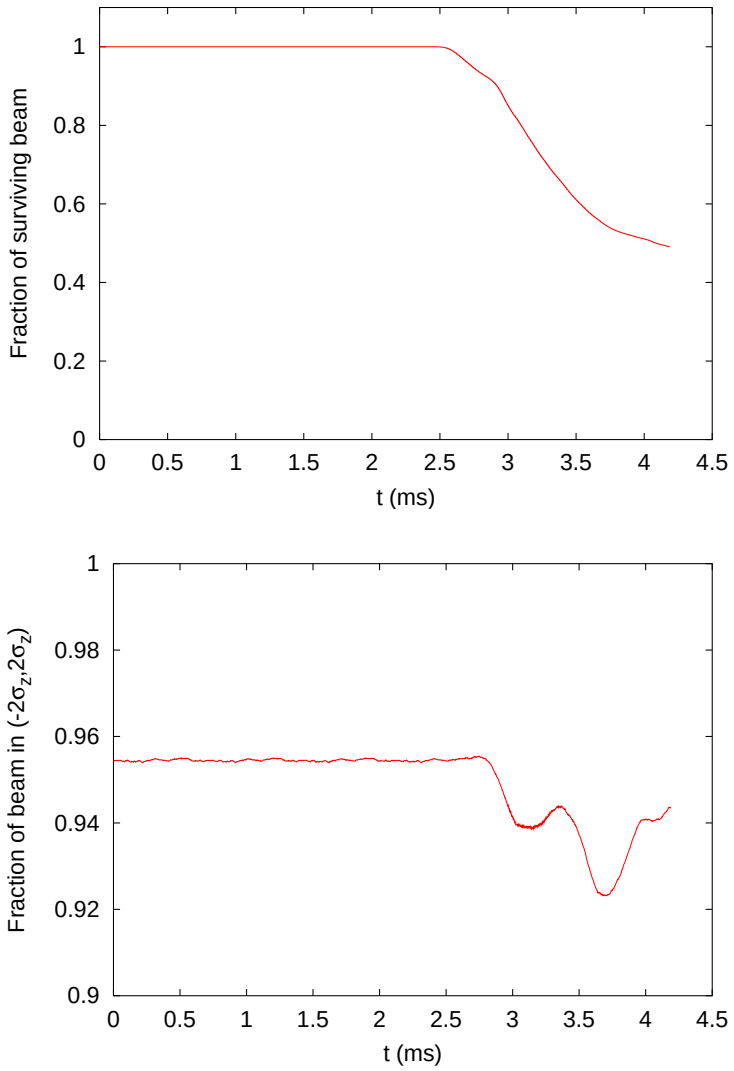

Figure 6: Time evolution of the fraction of bunch particles which have not yet been lost and the fraction of bunch particles that are longitudinally still within $(-2\sigma_z, 2\sigma_z)$

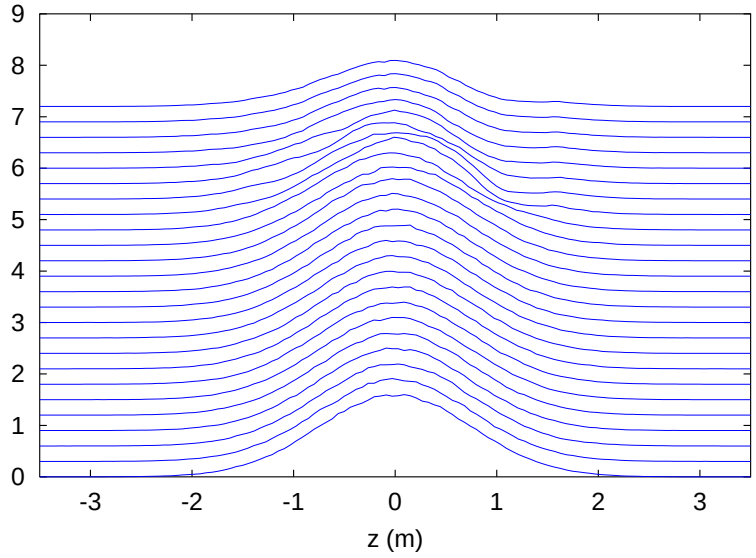

Figure 7: Mountain range plot of the bunch shapes all through the 2000 turn simulation.

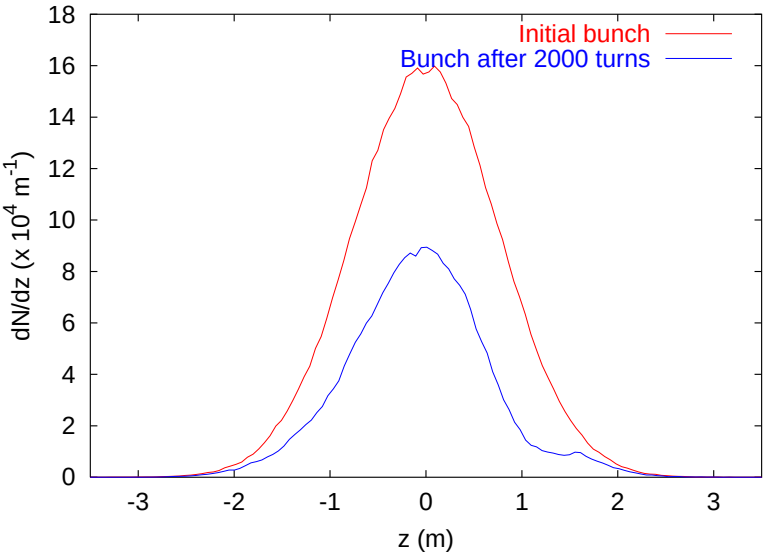

Figure 8: Two snap-shots of the bunch profile at the beginning and at the end of the simulation.

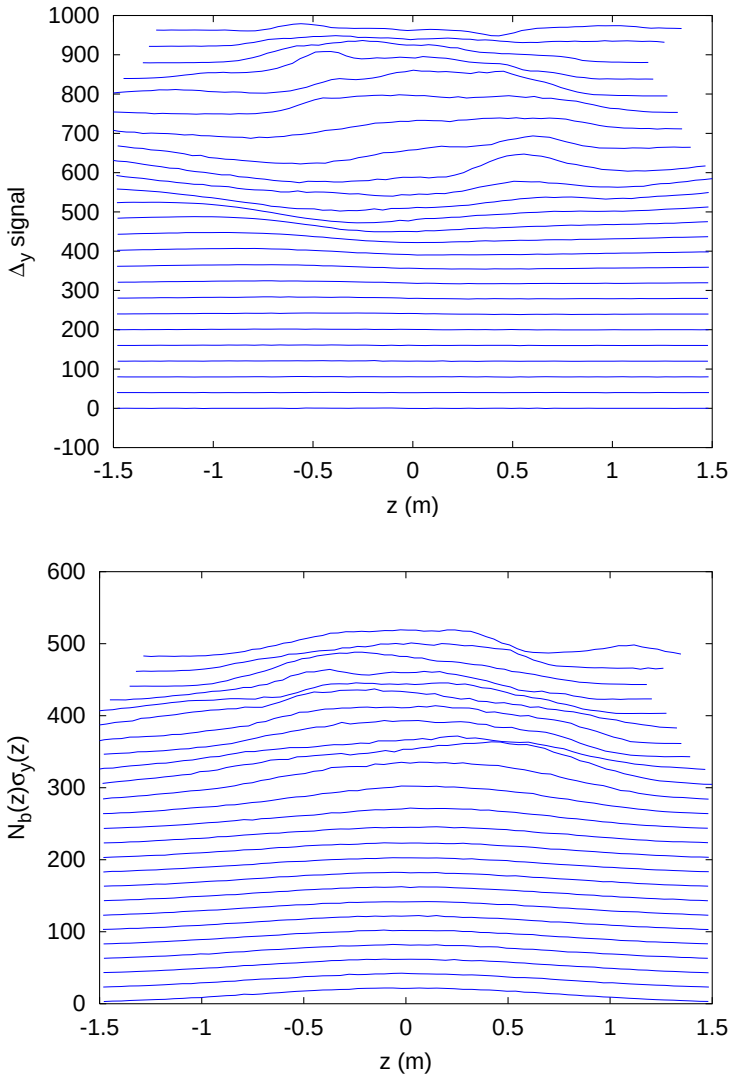

Figure 9: Mountain range diagrams of the vertical Δ signal along the bunch (above) and a quadrupolar pick-up signal (below), which shows the product $N_b(z)\sigma_z(z)$ at different locations along the bunch

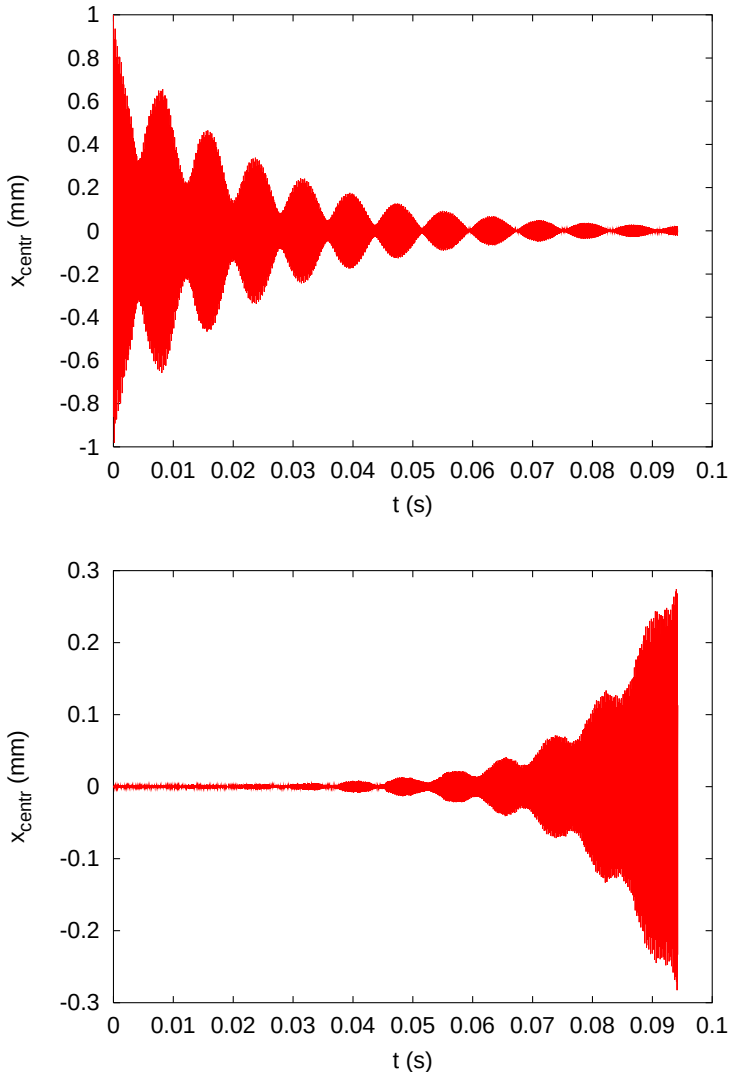

Figure 10: Vertical centroid evolution of an SPS bunch interacting with a broad band impedance. The motion is stable and the initial amplitude imparted through a kick is damped when the chromaticity is positive ($\xi_y = 0.1$, upper picture). The motion grows quickly unstable from noise when the chromaticity is negative ($\xi_y = -0.3$, lower picture).

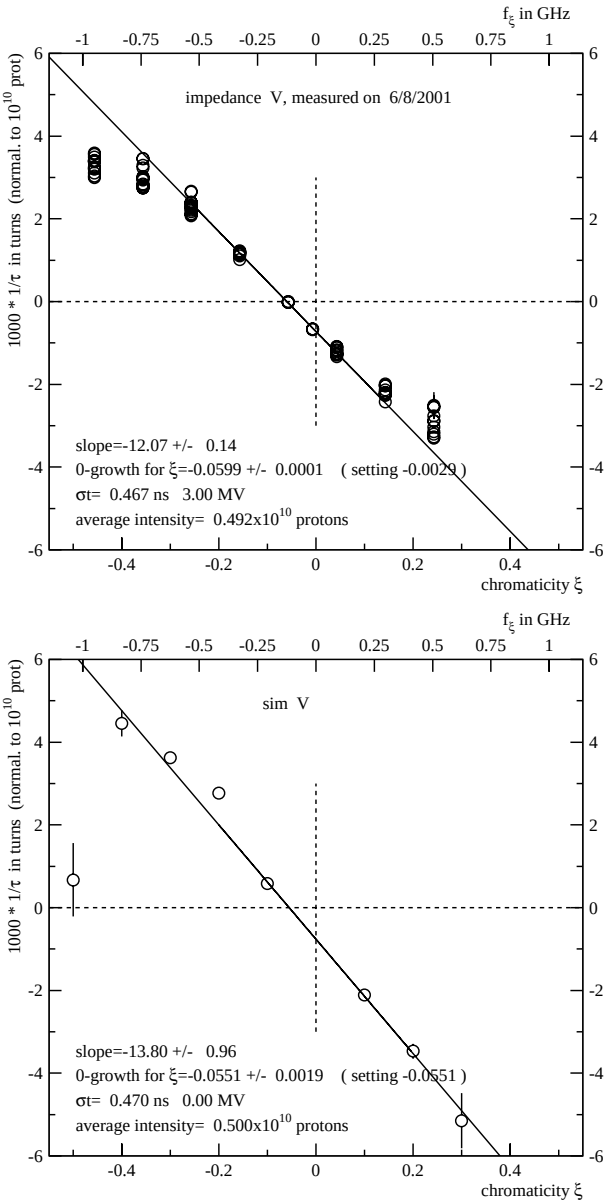

Figure 11: Growth and decay rates as a function of chromaticity in the experiment (above) and in the simulation (below). All data have been analyzed with the same sliding wondow technique over 4000 turns [12].

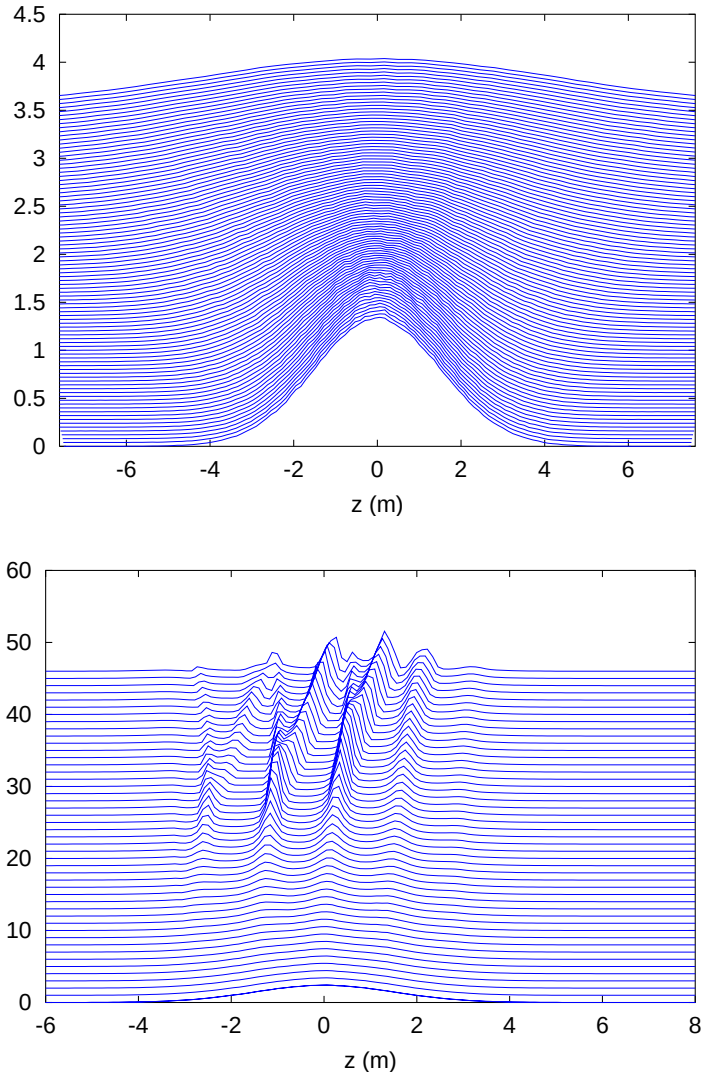

Figure 12: Simulated ountain range diagrams of bunch shapes for a debunching beam in the SPS. In the above picture no longitudinal wake is taken into account in the computation; the picture below shows a microwave instability arising from the longitudinal wake in the 200 MHz cavities.

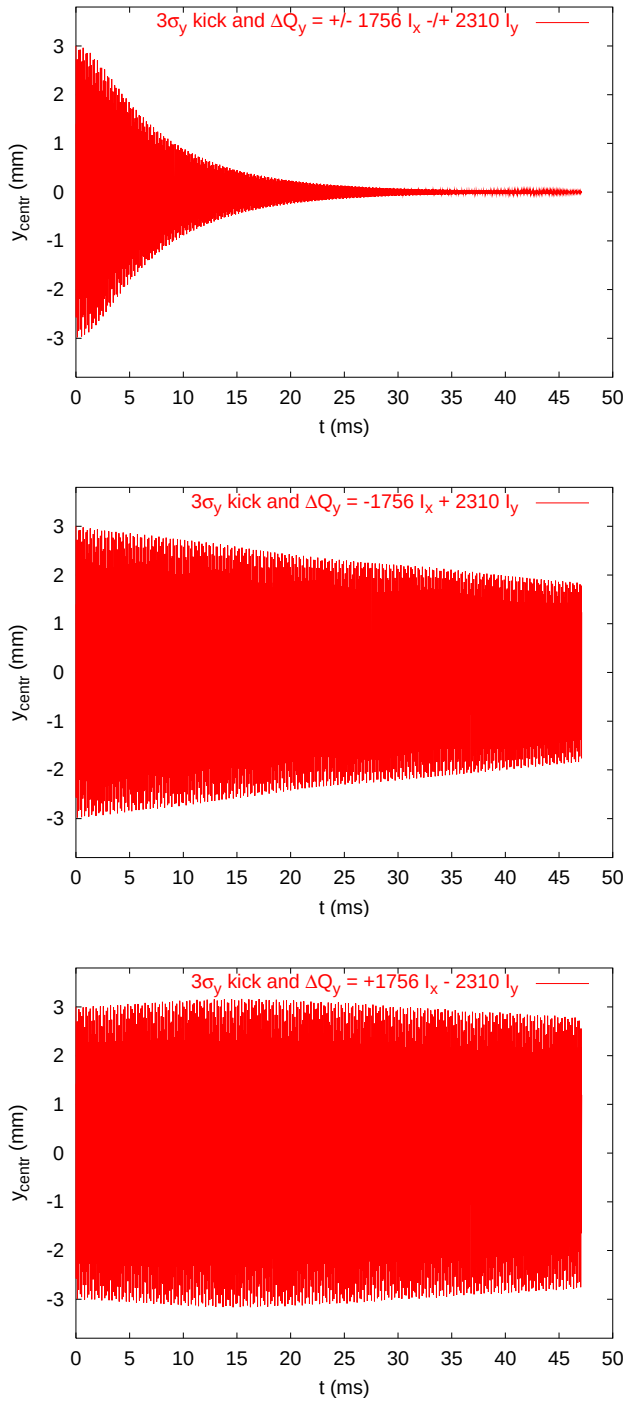

Figure 13: Vertical centroid evolution of an SPS bunch kicked to an amplitude of $3\sigma_y$. Detuning with amplitude is included in all the simulations; space charge only in the two lower pictures.

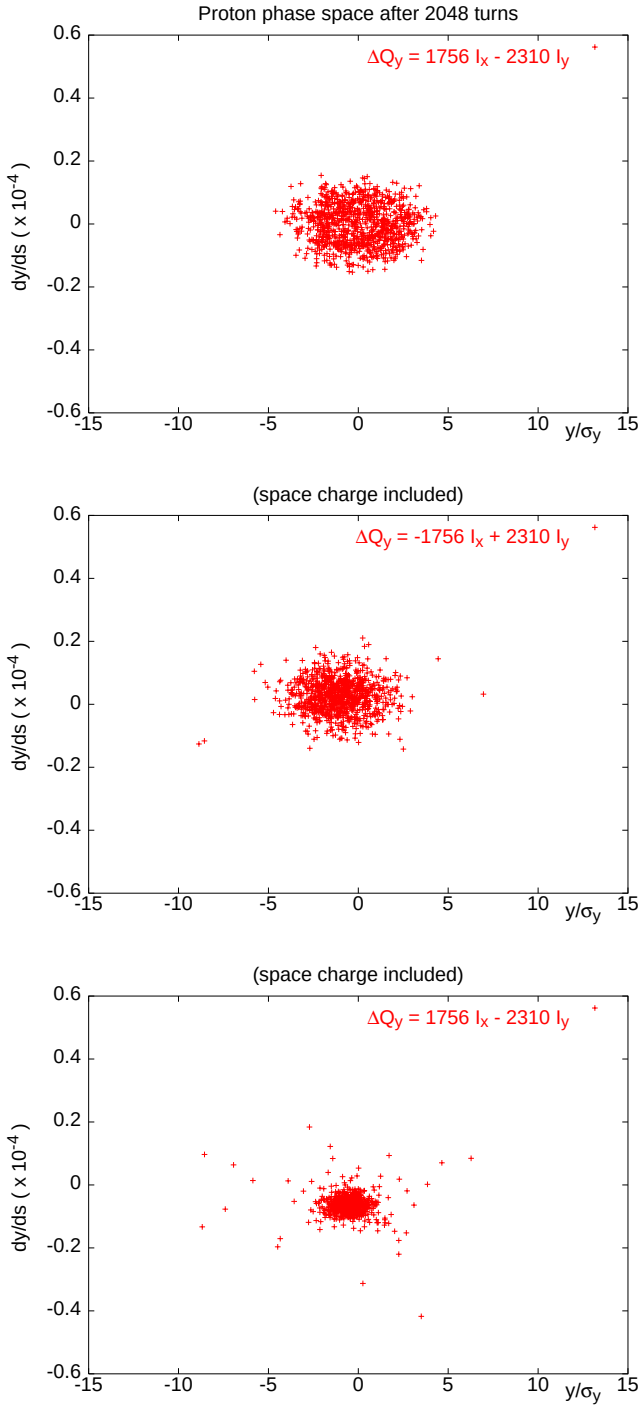

Figure 14: Vertical phase space of an SPS bunch initially kicked to an amplitude of $3\sigma_y$ after 2048 turns. Only amplitude dependent detuning (upper figure); with space charge and amplitude detuning of opposite signs (lower figures).