

# Numerical simulations of a proposed hollow electron beam collimator for the LHC upgrade at CERN

V. Previtali, G. Stancari, A. Valishev  
 FERMILAB, 60510 Batavia, IL (US)  
 S. Redaelli  
 CERN, Geneva, Switzerland

July 12, 2013

## Abstract

In the last years the LHC collimation system has been performing over the expectations, providing the machine with a nearly perfect efficient cleaning system[1]. Nonetheless, when trying to push the existing accelerators to - and over - their design limits, all the accelerator components are required to boost their performances. In particular, in view of the high luminosity frontier for the LHC, the increased intensity would ask for a more efficient cleaning system. In this framework innovative collimation solutions are under evaluation[2]: one option is the usage of an hollow electron lens for beam halo cleaning. This work intends to study the applicability of an the hollow electron lens for the LHC collimation, by evaluating the case of the existing Tevatron e-lens applied to the nominal LHC 7 TeV beam. New e-lens operation modes are here proposed to standard enhance the electron lens halo removal effect.

## 1 Introduction

The electron lens [3] is a device able generating an electron beam which travels in the same direction of the proton beam. The electron beam shape can be controlled by the shape of the emitting cathode, in order to serve different purposes, i.e. abort gap cleaning (flat-shaped), beam-beam compensation (gaussian-shaped) and collimation (hollow-shaped). Past experience at Tevatron has proven that the usage of an hollow beam can efficiently drive the beam halo onto the standard collimation system, by leaving the beam core unaffected [4, 5]. The applicability to the LHC is, anyway, not completely straightforward: not only for the different beam energy (7 TeV w.r.t. 980 GeV) and different particles (proton instead of antiproton), but also for the different lattice structure. The Tevatron was, in fact, a strongly

coupled machine with strong non-linearities: the non linearities generated by the e-lens were adding up to the machine non linearities, thus generating an unstable region in correspondence with the electron beam position [6]. The optics lattice of the LHC, on the other hand, is almost completely uncoupled and fairly linear, so the effectiveness of the e-lens still had to be proven. For these reason an e-lens element was included in the standard code for collimation (Sixtrack) and a full simulation campaign has been performed. Here we present the results of these simulations.

## 2 The electron lens: the device

The electron lens (e-lens) is a device which comprises three main sections: the gun (section 1), the main solenoid (section 2) and the collector (section 3). In the first section, the hollow beam is generated by thermionic emission of an hollow cathode (inner radius  $R_{1c}$  and outer radius  $R_{2c}$ ) and accelerated with an accelerating voltage of typically few KV. In section 2, the beam overlaps with the circulating beam (active region). In the third section, the beam is extracted and collected.

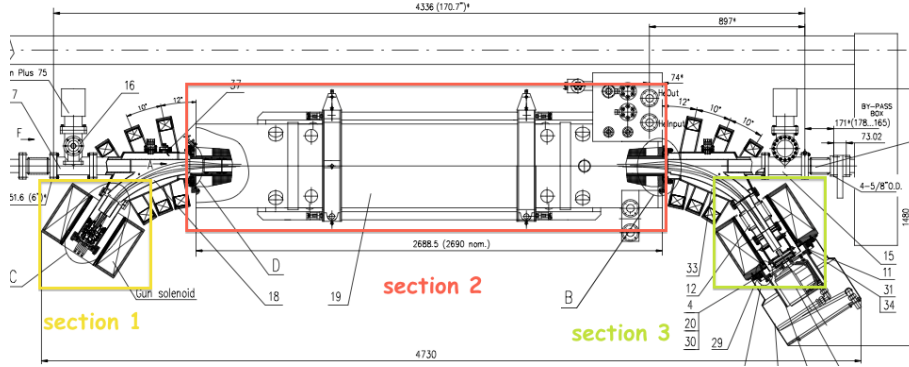


Figure 1: Sketch of the e-lens device.

In each section an independent magnetic field is provided by solenoidal magnets, generating the solenoidal fields which confine and transport the electron beam through the device ( $B_1, B_2, B_3$ ). When generated, the electron beam has the same geometrical dimensions of the emitting cathode. Since the electrons spiral around the magnetic lines, the dimension of the electron beam can be changed by acting on the magnetic solenoid field ratio, i.e. the ratio between the solenoidal field at the gun ( $B_1$ ) and the solenoidal field that confines the electron beam ( $B_2$ ). Imposing the magnetic flux conservation within the external radius of the electron beam, we have:

$$B_1 \cdot R_{2c}^2 = B_2 \cdot R_2^2 \quad (1)$$

which is obviously also valid for the inner radius

$$B_1 \cdot R_{1c}^2 = B_2 \cdot R_1^2 \quad (2)$$

It must be noticed that, even if absolute value of  $R_1$  can be modified to fit the desired beam size, the ratio  $g = R_{1c}/R_{2c}$  is fixed by the geometry of the cathode. On the collector side, on the other hand, the magnetic field  $B_3$  is imposed equal to the cathode field  $B_1$  so to restore the initial dimensions and easily collect back the electron beam. For further details on the hollow beam generation and technical specifications please refer to [3].

### 3 Kick computation and simulation modelling

Despite the effort of producing a perfectly symmetric electron beam, the real beam always present some asymmetry due to the interplay of geometrical imperfections and collective plasma effects. This can generate multipole fields within the inner radius of the electron beam which could be affect the core emittance. From beam tests conducted at Tevatron, it has been measured a negligible effect on the beam core. However, dedicated simulation with Particle In Cell plasma codes are being conducted to quantify a possible impact on the LHC beam core[7]. The simulation of the e-lens effect on the beam core are complementary to the the work presented here, but are not treated in this report.

The purpose of this work is to quantify through simulation the removal rate of halo particles. For this reason, we consider an ideal electron distribution, which consists of a azimuthal-symmetrical distribution of electrons contained between internal radius  $R_1$  and external radius  $R_2$ . Only the active part of the device is considered (section 2).

For the calculation of the transverse electric and magnetic fields we will approximate the finite-lenght cylinder with an infinite one, which is legitimate since the transversal space of interest is of the order of few millimeters, while the total device lenght is typically  $L = 2$  m. The fringe fields are neglected.

The electron beam radial profile can be either considered a uniform distribution between  $R_1$  and  $R_2$  (simplified electron lens model), or a more realistic beam profile can be implemented (realistic e-lens model). The realistic profile is based on a laboratory measurement of the electron beam generated by a cathode mounted in the e-lens test stand in Tevatron ( Figure 2).

In both the simplified and the realistic model, the normalized function  $f(r)$  is considered independent on the angular coordinate and it is defined as:

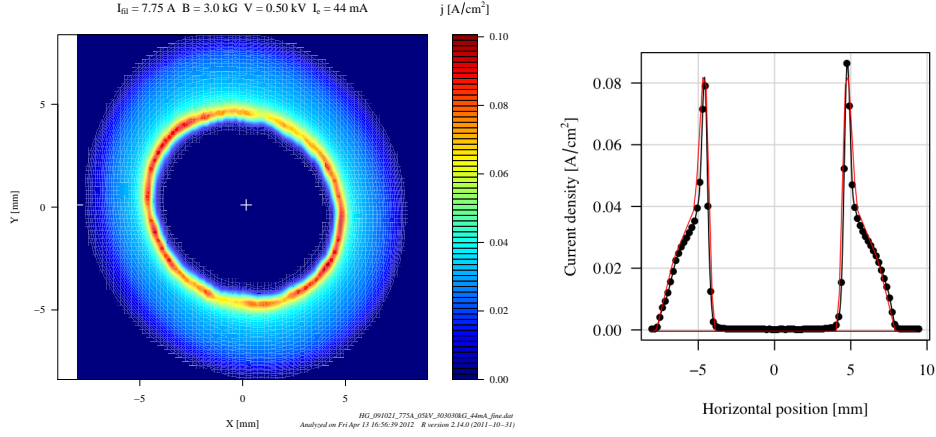


Figure 2: On the left hand side, 2D hollow beam measured profiles for a total current of 44 mA,  $V=0.5$  KV. On the right hand side the profile in the  $y=0$  plane is shown, both experimental data (dotted line) and fit (red curve).

$$f(r) = I(r)/I_T = \frac{2\pi}{I_T} \int_{R_1}^r r \rho(r) dr \quad (3)$$

where  $I(r)$  is the current enclosed in a radius  $r$ ,  $I_T$  is the total electron beam current and  $\rho(r)$  is the electron beam density distribution; in the following the function  $f(r)$  will be referred as shape function. In case of perfect electron lens the shape is simply:

$$f(r) = \begin{cases} 0 & r < R_1 \\ \frac{r^2 - R_1^2}{R_2^2 - R_1^2} & R_1 < r < R_2 \\ 1 & r > R_2 \end{cases} \quad (4)$$

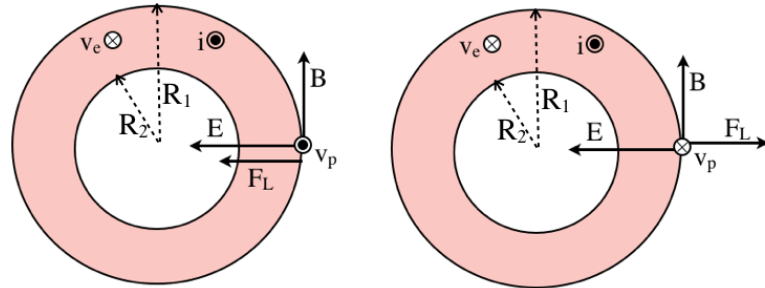


Figure 3: Electrostatic and magnetic force for different proton directions.

Thanks to the cylindrical symmetry, the EM field in the space enclosed by the ideal electron beam is perfectly zero, thus not affecting the beam

core. On the contrary, protons with transverse radius  $r = \sqrt{x^2 + y^2} > R1$  will feel both the electrostatic force and the magnetic force for the whole interaction length  $L$ . As shown in figure 3, the two forces will sum up when the versus of the proton velocity is opposite to the electron one; otherwise the two forces will have opposite versus.

In our approximation the resulting force can be easily calculated by using the Gauss and the Biot-Savart laws:

$$\overline{F}(r) = \frac{I_r q_p (1 \pm \beta_p \beta_e)}{2\pi\epsilon_0 r v_e} \bar{j}_r \quad (5)$$

where  $I_r$  is the current encompassed by the radius  $r$ ,  $q_p$  is the proton charge,  $\beta_p$  and  $\beta_e$  are the relativistic  $\beta$  for the proton and the electron beam, and  $\bar{j}_r = (\bar{x} + \bar{y})/r$  is the radial inward direction. Using the definition of shape function  $f(r)$  and keeping in mind the definition of angular velocity  $\omega = \theta/t = (v_p/r)$ , the crossing time  $t = L/v_p$ , and the centrifugal force  $(1/r) = F/(mv_p^2)$ , it is possible to calculate the integrated kick for a particle which crosses the electron lens at transverse amplitude  $r$ :

$$\theta(r) = \frac{2L f(r) I_T (1 \pm \beta_e \beta_p)}{4\pi\epsilon_0 r (B\rho)_p \beta_e \beta_p c^2} \quad (6)$$

where  $(B\rho)_p = \gamma m_0 v_p / q = 3.3356(mv)_p [GeV/c]$  is the magnetic rigidity of the proton beam.

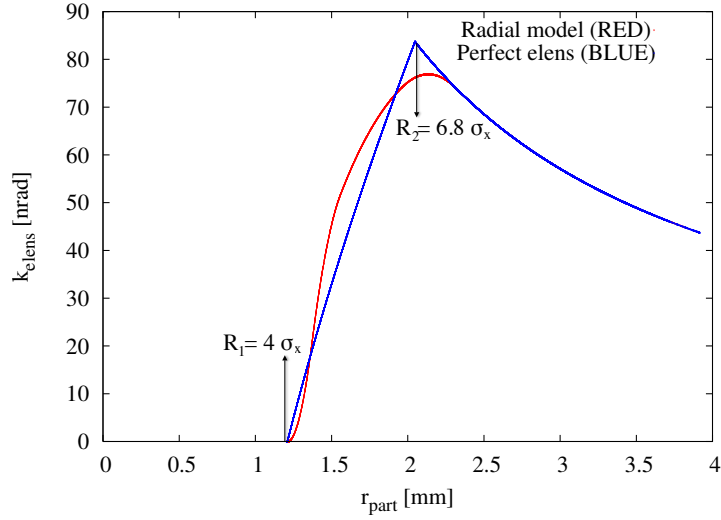


Figure 4: Radial kick given by the electron lens, for both simplified electron lens model and realistic profile model.

The typical kick generated by a 1.2 A electron lens on the 7 TeV LHC beam is presented in Figure 4; the details about simulation parameters are

described in in Section 4.1). It is interesting to notice how the realistic profile provides larger kick in the region between  $4.5\sigma_x$  and  $6\sigma_x$ , but it is less effective for low-amplitude particles. Both profiles have been implemented in the simulation and tested. Even though preliminary results have been calculated using the perfect electron lens model, the full statistic has been performed only for the realistic profile mode. Only results from simulations including the realistic profile are presented in this report.

Since the electrostatic force is always directed inward, and the electron beam versus is usually chosen such as the magnetic force adds up to the electric one, the kick given by the electron lens is always directed inwards (focusing), both for the vertical and for the horizontal plane. It follows that, if the kick given by the electron lens is acting constantly on the particle motion (continuous mode), the electron lens field becomes a part of the periodic lattice and the single particle invariant for stable particles must be re-defined. The electron lens is therefore expected to introduce a deformation of the phase space together with a positive tune shift of the particle, which would be larger for large particle amplitudes. On the other hand, the continuous mode is not the only possible mode of operation of the electron lens: given the extremely fast rising time of the cathode modulator (250 ns) it is possible to modulate the intensity of the electron beam and even to switch it ON-OFF on a turn-by-turn basis. This allows us to use the electron lens both continuously or with a specific modulation. In particular, three different operational modes have been identified:

- “continuous” mode: the electron lens is used in continuous.
- “resonant” mode: the electron lens current is modulated in order to achieve resonance condition with the particle betatron oscillation, including both a ”sinusoidal excitation” and ”turn skip” (which was used for abort-gap cleaning in Tevatron);
- “diffusive” mode: the electron lens is randomly switched on or off in order to steadily heat the beam.

With any of these modes a current jitter of  $\pm 2\%$  can be activated so to better reproduce experimental conditions. In the next chapters the details of the different operation modes and the effect on the LHC beam as predicted by simulations for the LHC case will be presented.

## 4 LHC simulations

### 4.1 Simulation parameters

For a future installation of the electron lens the LHC two possible locations, immediately downstream and upstream of IP4 (RB44-RB46). The location

has been selected because of the increased distance between the two beam pipes (420 mm), which would ease the installation of the e-lens. See Figure 5

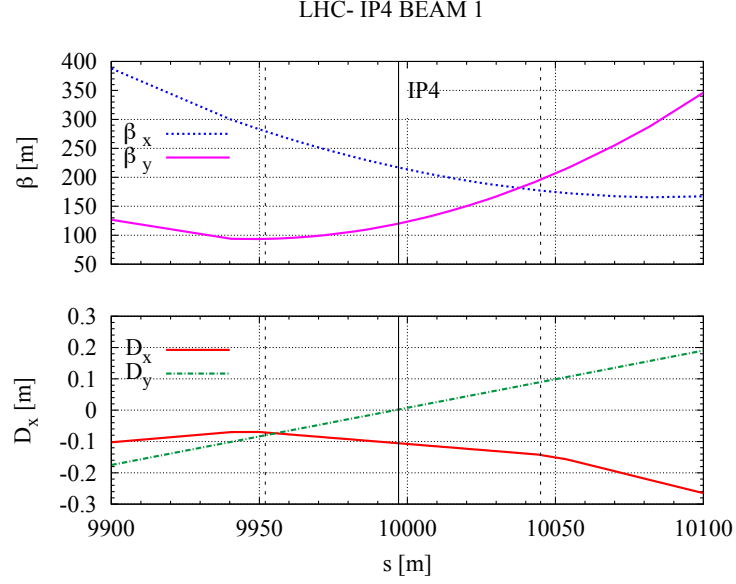


Figure 5: Beta and dispersion functions for the horizontal and vertical orientation at the LSS4, LHC, collision optics for Beam 1.

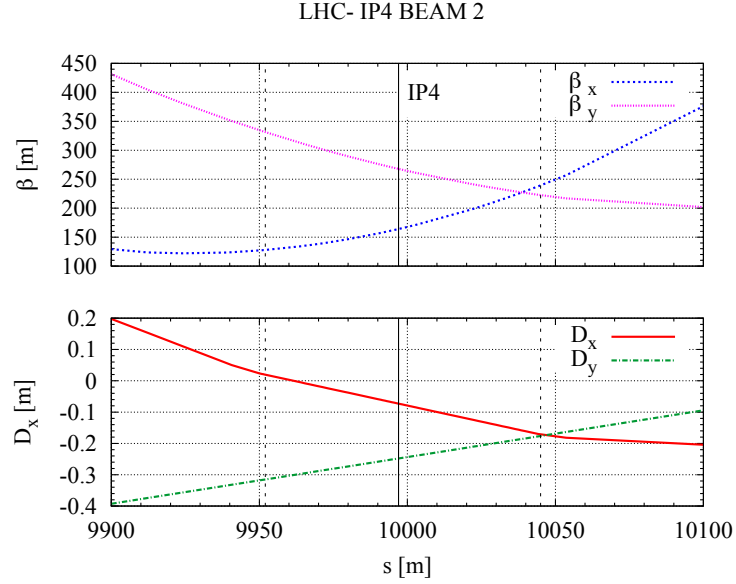


Figure 6: Beta and dispersion functions for the horizontal and vertical orientation at the LSS4, LHC, collision optics for Beam 2.

and Figure 6 for computed optical function plots (by **MADX**), both for Beam 1 and Beam 2. Dispersion is almost totally suppressed in the whole region: it is of the order of 10 cm both in horizontal and in vertical plane. Since in RB46 the  $\beta$ -functions are close to round beam both for Beam 1 and for Beam 2, we selected this location for our simulation campaign.

All the simulations assume typical parameters for the electron lens (current 1.2 A, extraction voltage 5 KeV). The nominal 7 TeV LHC squeezed optics has been simulated, sextupoles included, octupoles switched off. The case of purely horizontal (or purely vertical) halo has been studied, generating an on momentum halo with nominal normalized beam emittance of  $3.75 \cdot 10^{-6}$  m rad. Since the results do not substantially differ for the vertical and horizontal case, only the horizontal case is presented in this report. The machine is in storage mode, and no diffusive effects or beam beam interactions have been included. The simulation have been performed with **SixTrack** (collimation version)[8], where a model describing the ideal electron lens has been implemented as a new collimator-type element. For the technical details about the implementation of the new e-lens element in the code, please see Appendix A.

The inner radius of the electron lens has been fixed to  $4\sigma_{x|y}$  in case, respectively, of horizontal or vertical halo simulations. The absolute kick value given by the electron lens in function of the distance from the center is shown in Figure 4, both the simulation results and the analytical expectation.

Table 1: Main optics parameters for the collimators simulated in Sixtrack.

name	s [m]	$\alpha_x$ [ - ]	$\beta_x$ [m]	$\alpha_y$ [ - ]	$\beta_y$ [m]	$\mu_x$ [ - ]	$\mu_y$ [ - ]
ELENSE	10037	0.318	181.8	-0.962	179.9	24.36	22.24
TCP.D6L7.B1	19789.2	2.120	158.9	-1.097	78.26	47.34	43.42
TCP.C6L7.B1	19791.2	2.051	150.5	-1.153	82.76	47.34	43.42
	$\sigma_x$ [ $\mu$ m]	$\sigma'_x$ [ $\mu$ rad]	$\sigma_y$ [ $\mu$ m]	$\sigma'_y$ [ $\mu$ rad]			
ELENSE	302.3	1.74	300.7	2.32			
TCP.D6L7.B1	282.54	4.17	198.3	3.76			
TCP.C6L7.B1	275.02	4.17	203.9	3.76			

## 4.2 Continuous mode

As discussed in Section 2, we expect the continuous mode of the electron lens to introduce a deformation of the particle orbit in the phase space and



a tune shift. This has been verified by simulations.

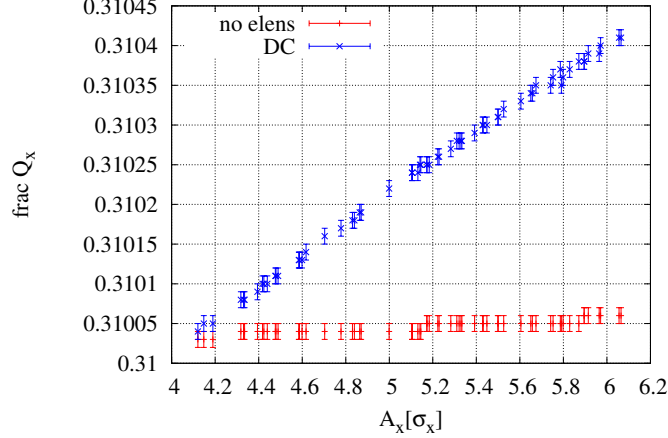


Figure 7: The fractional part of the horizontal tune in dependence on the particle average amplitude for a quasi-linear machine. In case of no electron lens, the slope is caused by the other non linear elements (sextupole) in the optics.

In case of quasi-linear machine (linear machine and sextupoles), the tune shift of the particle is of the order of few  $10^{-4}$  and depends on particles initial conditions, as shown in Figure 7: as expected the tune shift is always positive (because the electron lens is always focusing) and larger for particles with a larger betatronic amplitude. Since the electron lens kick amplitude depends linearly on the particle position, it follows that the dependency on the tune shift by the initial amplitude is also linear with the particle initial amplitude. The tune shift reaches a maximum value of about  $5 \cdot 10^{-4}$ . Given the LHC extremely stable working point, such a tune shift is not sufficient for driving the particle into a high order machine resonance. The jitter in time of the tune is also negligible, at least up to a precision of about  $1 \cdot 10^{-5}$ .

When including non linear elements such as the octupoles in the simulation, the particle tune is heavily affected: the whole tune range is shifted and the spread is about a factor 100 larger. By dedicate studies it was observed that the regular tune increase with respect to the particle amplitude is interrupted only at about  $5.7 \sigma_x$ . After further investigation, studying the phase space portraits of particles with about  $5.7 \sigma_x$ , it was found out that a high-order resonance sits there when the electron lens is present.

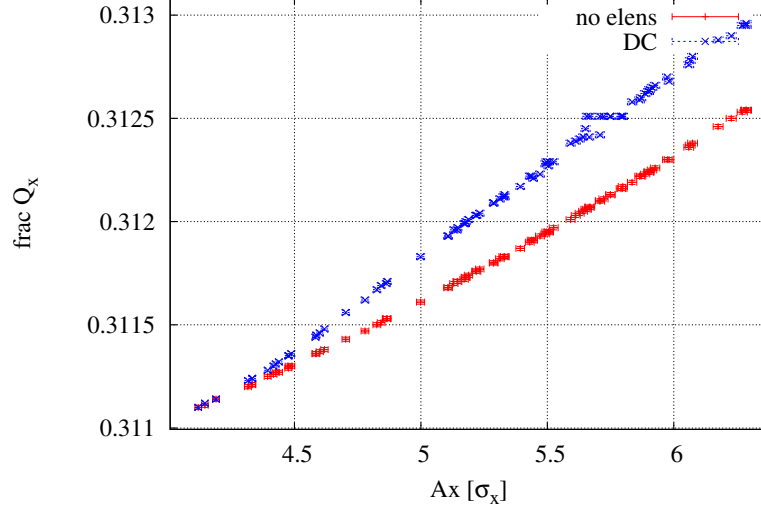


Figure 8: The fractional part of the horizontal tune in dependence on the particle average amplitude, when octupoles are included in the simulation.

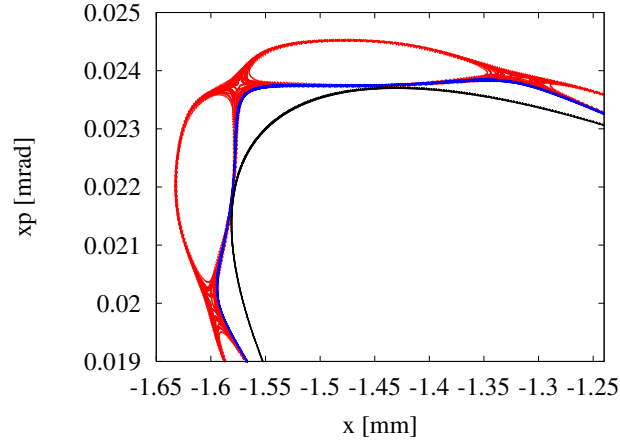


Figure 9: Close-in of the horizontal phase space for the same particle in case with (BLUE) and without electron lens (BLACK), for a particle which has a normalized amplitude of about  $5.7 \sigma_x$ . The RED line correspond to the case of electron lens with current jitter.

In Figure 9, the deformation of the phase space induced by the non linearity of the electron lens is presented for the electron lens in the non linear machine. A maximum amplitude oscillation of the order of few hundreds of sigma is introduced by the electron lens. When adding a jitter of  $\pm 2\%$  in the electron beam current, the electron lens begins to show a diffusive effect, as shown in Figure 9: in this case the amplitude change is enough to drive the particle in the high resonance sitting at about 5.7 sigma.

### 4.3 Resonant mode

In order to enhance the efficiency of the electron lens as a scraper, it was proposed to put the electron lens in resonance with the betatronic oscillation of the particles. The basic idea is to use the electron lens for giving a periodic kick which follows the periodicity of the betatronic tune. In order to find the right resonance frequency, it is useful to consider the equation of a purely horizontal (or vertical) motion of a particle at the electron lens location. Neglecting the non linear elements in the machine it is clear that, when the electron lens is not active, the equation of motion in the normalized coordinate space is the equation of an harmonic oscillator:

$$m\ddot{x} + kx = 0 \quad (7)$$

where the natural oscillation frequency  $\omega_0 = \sqrt{k/m}$  is the particle betatron tune in the considered plane. When the particle is subject to the electron lens force, the equation becomes:

$$m\ddot{x} + kx = -f(x) \quad (8)$$

where in general  $-f(x)$  is an highly non linear force. It is possible however to consider the simple case of a non-hollow flat lens and to study the motion for a limited transverse range, which is a good approximation because the available aperture is limited by the collimator aperture, which is usually lower than the electron lens outer radius. In this case the electron lens the force is nearly linear with the transverse position, and the equation of motion is:

$$\begin{aligned} m\ddot{x} + kx &= -k_C x \\ m\ddot{x} + (k + k_C)x &= 0 \end{aligned} \quad (9)$$

and the electron lens effect is simply to increase the particle oscillation frequency from  $\omega_0$  to  $\omega_C = \sqrt{(k + k_C)/m}$ . For focusing forces, i.e.  $k_C > 0$ , the frequency of the oscillation increases (i.e. increases the tune).

We shall now consider the case where the electron lens beam current is variable in time, so that  $f = f(x, t)$ . Being the e-lens force directly

proportional to the total e-beam current, we can decouple the spatial and time-dependent part. A positive modulation function  $g(t)$ , varying between zero and one, represents the e-lens pulsing waveform; it is natural to start with the simplest possible shape for such function, i.e. a pure harmonic with frequency  $\omega_r$ :

$$g(t) = (1 + \sin \omega_e t)/2 \quad (10)$$

When the electron lens is modulated, the particle motion equation is therefore:

$$m\ddot{x} + [k + k_C(1 + \sin \omega_e t)] x = 0$$

$$\ddot{x} + \left[ \omega_C^2 \left( 1 + \frac{k_C}{k + k_C} \sin \omega_e t \right) \right] x = 0 \quad (11)$$

Equation 11 is known, in physics mathematics, as Mathieus equation [9]. In case  $\frac{k_C}{k+k_C} \ll 1$ , it describes a parametric resonance which is peaked in  $\omega_e = 2\omega_C + \epsilon$ , with  $\epsilon$ :

$$-\frac{1}{2} \frac{k_C}{k + k_C} \omega < \epsilon < \frac{1}{2} \frac{k_C}{k + k_C} \omega \quad (12)$$

Considering the relation between the relation between  $k$  and  $w$ :

$$k + k_C = w_C^2 m \quad (13)$$

$$k = w_0^2 m \quad (14)$$

and therefore:

$$\frac{k_C}{k + k_C} = \frac{\omega_C^2 - \omega_0^2}{\omega_C^2} \quad (15)$$

Considering the fact that tune and frequency are directly proportional and using the tune values presented in Figure 8, we have that  $\omega_C = \omega_0 + \delta\omega$  with  $\delta\omega \ll \omega$ . The previous equation then is:

$$\frac{k_C}{k + k_C} = \frac{2\omega_0\delta\omega + \delta\omega^2}{\omega_0^2 + 2\omega_0\delta\omega + \delta\omega^2} \approx \frac{2\delta\omega}{\omega_0} \ll 1 \quad (16)$$

which demonstrate that the electron lens case can actually be considered a case of parametric resonance. The parametric resonance width  $\epsilon$  therefore results:

$$-\delta\omega < \epsilon < \delta\omega \quad (17)$$

meaning that the resonance is excited only if the difference between the applied frequency  $\omega_e$  and the exact resonance frequency  $2\omega_r$  is less than the induced tune spread.

The exact transposition to the case of the non-linear force generated by an hollow electron lens (Figure 4) is not straightforward. Instead of the well-know equation 9, the equation of motion would be:

$$m\ddot{x} + kx = -f(x) \quad (18)$$

where  $f(x)$  could be approximated with a high order polynomial. The solution of such a differential equation is not straightforward. In order to identify the optimal excitation frequency, we simulated the scraping effect of an el-lens driven by different multiples of the natural frequency  $n \cdot \omega_0$ , with the multiplying factor  $n$  in the range  $\{1, 2 \dots 10\}$ . However, when octupoles are present, it is difficult to define the natural frequency  $\omega_0$ : as shown in Figure 8  $\omega_0$  spans in a wide range. In order to overcome this complication, the e-lens excitation frequency is varied between  $n \cdot 0.3104$  and  $n \cdot 0.3136$ , in frequency steps of  $2 \cdot 10^{-5}$ , one step every  $10^3$  turns. The total number of turns simulated is  $2 \cdot 10^5$ .

Since the purpose of this simulation is to evaluate the efficiency of the electron lens and not to study the losses in the machine, the simulated collimation system has been reduced to only two elements: an e-lens in resonant mode (inner radius  $4 \sigma_x$ ) and a standard LHC horizontal primary collimator in IP7 (TCP.C6L7.B1, aperture  $6.2 \sigma_x$ ). The realistic electron lens profile has been used, and there is no jitter of the electron beam intensity. The primary collimator has been treated as a black absorber. The initial distribution is a uniform flat distribution between 4 and  $6 \sigma_x$ . The LHC optics is the standard LHC case at 7 TeV with octupoles at maximum strenght.

In order to qualify the efficacy of the electron lens as a scraper, we use the global scraping efficiency, defined as:

$$\eta_g(t) = N(t)/N_0 \quad (19)$$

where  $N(t)$  is the number of particles which have not been removed after  $t$  turns over the initial number of particles  $N_0$ . This quantity is obviously dependent on the initial distribution: for all the simulation presented in this report the initial halo distribution is a uniform flat distribution in the amplitude space between  $4\sigma$  and  $6\sigma$ . Only the simulations about horizontal plane are here presented, but the vertical plane presents no significant differences.

In Figure 10 the global scraping efficiency  $\rho_s(2 \cdot 10^5)$  is presented for different multiplication factors. It can be noticed that there is a different response to odd and even multiplication factors. The most effective resonancy frequency is  $2\omega_0$ , as expected in the case of simple parametric hoscillations.

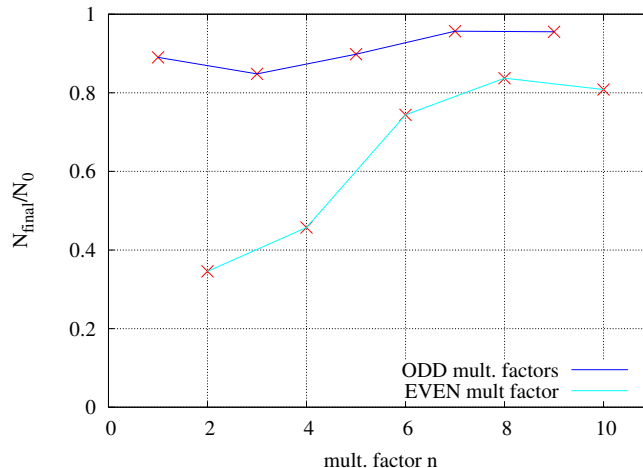


Figure 10: Relative number of survival particles after  $2 \cdot 10^5$  turns for different excitation frequency  $n \cdot \omega_0$  of the resonant electron lens.

#### 4.3.1 Optimization of frequency sweep

The electron lens simulated configurations can be divided in two groups: the "low step" series, corresponding to tune step of  $2 \cdot 10^{-5}$  and the "high step" series, with tune step of  $5 \cdot 10^{-5}$ . The total covered range varies from  $8 \cdot 10^{-4}$  to  $32 \cdot 10^{-4}$  in tune units. Considering the revolution frequency of the LHC  $f = 11.245$  KHz, the tune range and the e-lens frequency multiplication factor of two, this is equivalent to sweeping through the frequency range  $[6.983 : 7.051]$  KHz with steps of 0.45 Hz (low step case) or 1.12 Hz (high step case) every 89 ms (1000 turns).

A summary table of the tested range-step combinations is presented in Table 2. The corresponding global scraping inefficiencies are presented in Figure 11. Most cases do not significantly differ in global scraping efficiency, as far as the sweeping range is not too different from the tune range we want to cover. In fact we noticed that an optimal sweeping range is between one third and one half of the total tune range.

To study the amplitude distribution of the survival particles, a new qualifying parameter has been introduced, i.e. the local scraping inefficiency  $\eta(a, t)$ :

$$\eta(a, t) = dN(a, t)/dN_0(a) \quad (20)$$

Table 2: sweeping parameters

Label	$f_{min}$	$f_{max}$	$F_{avg}$	$\Delta f$	Tune Step	Turns per step
L8	.3116	.3124	.3120	.0008	$2 \cdot 10^{-5}$	$10^3$
L10	.3115	.3125	.3120	.0010	$2 \cdot 10^{-5}$	$10^3$
L16	.3112	.3128	.3120	.0016	$2 \cdot 10^{-5}$	$10^3$
L20	.3110	.3130	.3120	.0020	$2 \cdot 10^{-5}$	$10^3$
L32	.3104	.3136	.3120	.0032	$2 \cdot 10^{-5}$	$10^3$
H5	.31175	.31225	.3120	.005	$5 \cdot 10^{-5}$	$10^3$
H10	.3115	.3125	.3120	.0010	$5 \cdot 10^{-5}$	$10^3$
H15	.31125	.31275	.3120	.0015	$5 \cdot 10^{-5}$	$10^3$
H20	.3110	.3130	.3120	.0020	$5 \cdot 10^{-5}$	$10^3$
H30	.3105	.3135	.3120	.0030	$5 \cdot 10^{-5}$	$10^3$

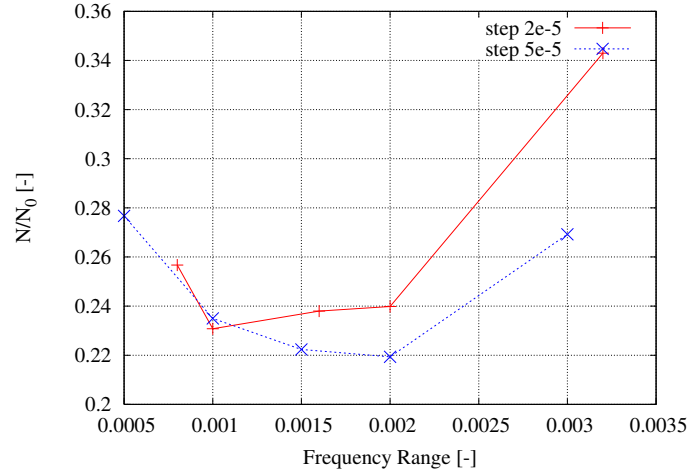


Figure 11: Global inefficiency versus total range scanned, for different steps in frequency.

where  $dN(a, t)$  is the number of particles with amplitude in the range  $[a, a + da]$  at the turn  $t$  and  $dN_0(a)$  is the correspondent number of particle at the initial turn.

The local scraping efficiency for the most performing case (i.e. H20) is shown in Figure 12 . It can be noticed that the e-lens is much more effective for high amplitude particles (after about  $4.5 \sigma_x$ ), where the scraping inefficiency is lower than  $10^{-2}$ - $10^{-3}$ . The low amplitude particles, on the other

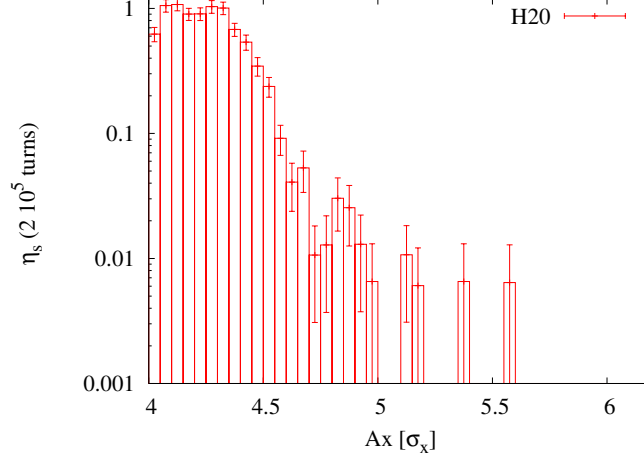


Figure 12: Local scraping inefficiency for the H20 case.

hand, are hardly affected.

#### 4.4 Diffusive mode

It is worth noting that the resonant mode requires accurate knowledge of the machine tune and a separate procedure for the horizontal and for the vertical case. An alternative use for the electron lens could be to use the device as a random diffuser acting only on the beam halo. It was already observed that, by including a beam current jitter in the continuous e-lens, a mild diffusive effect started to appear. To exploit and maximize this process, the kicks given by the electron lens should include a random component. This can easily be achieved by using a white noise generator to drive the electron lens current. Given the nature of the electron lens, the rms kicks will obviously be larger for high amplitude particles (see the full power kick in Figure 4).

Two different approaches were tested:

- Random mode: the electron beam current was modulated on turn by turn basis by a random multiplier in the range  $[0,1]$ ;
- ON-OFF mode: then electron beam was either reduced to zero (OFF) or at its full power (ON) randomly on a turn-by-turn basis.

The comparison between the two different methods can be appreciated



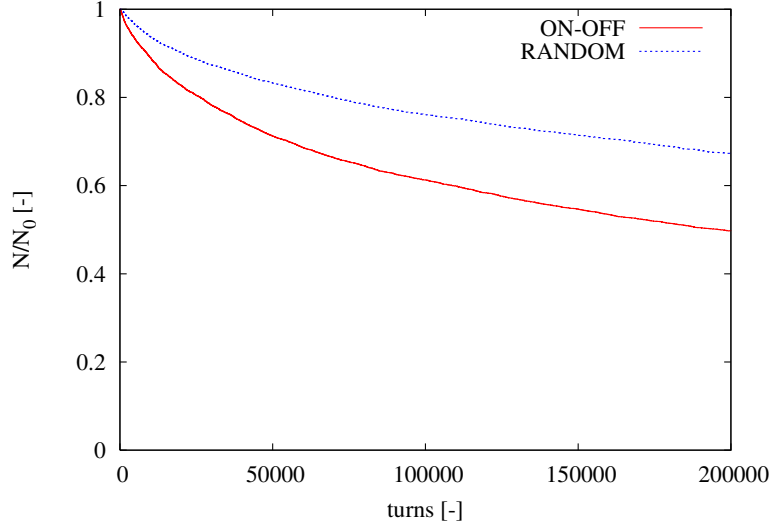


Figure 13: Normalized number of survival particles vs number of turns for the diffusive noise mode. Both ON-OFF and Random method results are presented.

in Figure 13: the global scraping efficiency after  $2 \cdot 10^5$  turns for the ON-OFF method is about  $0.50 \pm 0.01$ , while for the random method it is only  $0.67 \pm 0.01$ .

The huge advantage of this method is that it would not require perfect knowledge of the tune or complicated modulation, and would work both for the horizontal and for the vertical case at the same time.

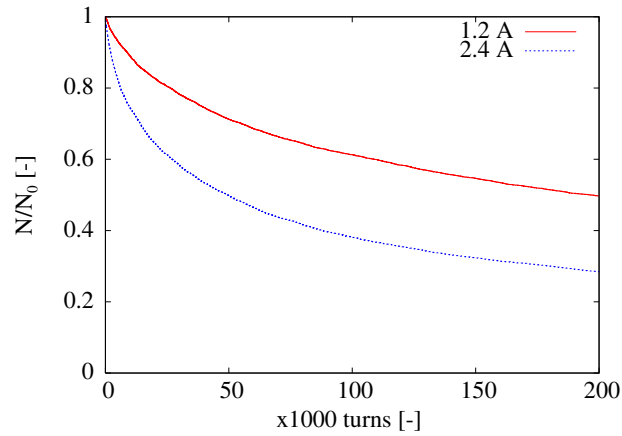


Figure 14: Random effect with double current.

The price to pay for an easier and more robust operation mode is, however, the scraping time. The random ON-OFF about a factor 5 slower than the optimal resonant mode: after  $2 \cdot 10^5$  turns, only 50% of the particles have been scraped. However, being the electron lens kick proportional to the total electron beam current, the scraping efficiency can be easily increased by increasing the beam current. In Figure 14, an electron lens with doubled beam current (2.4 A) is compared with the standard electron lens (1.2 A): in case of enhanced current the global scraping inefficiency is almost halved, reaching a cleaning of more than 70% of the halo particles in 20s.

## 5 Alternative halo scraping schemes

Even if the e-lens seems to be an effective solution for the LHC halo beam scraping, it is not the only candidate which have been proposed. If oscillating in resonance with the particle tune, also a quadrupole or a dipole could effectively drive the beam particles onto the collimator system. If the particle betatron frequency are sorted accordingly to their amplitude (as it is the case when octupoles are present, see Figure 8) it should be possible to resonate only a part of the beam, by selecting the appropriate resonance frequency.

In order to study the applicability of this method, a new element has been inserted in sixtrack. The tune modulation element can be either a dipole or a quadrupole, and oscillate in resonance with a specified tune range. In principle, both elements can oscillate with a oscillation frequency which is any integer multiple of the original betatron oscillation. For generating a resonance, however, the dipole and the quadrupole case have been chose to fit the resonance condition:

- In case of a dipole, the oscillation frequency is exactly the betatron oscillation frequency of the halo particles.
- In case of a quadrupole, the element oscillates with a frequency which is doubled with respect to the particle oscillation frequency (see Section 4.3).

### 5.1 Tune modulation with a dipole

The first test with a tune modulation dipole has been conducted using the beam feedback system (ADT)[11] with the nominal 7 TeV LHC beam, with a kick of  $0.2 \mu\text{rad}$  per turn. With this condition, the whole beam (both the halo and the core) were lost within about 200 turns. It was therefore tried to decrease the provided kick of a factor ten, but the full beam was still

lost in about  $10^5$  turns. From this preliminary results it looks like the tune modulation of a dipole is not a viable solution for halo beam scraping.

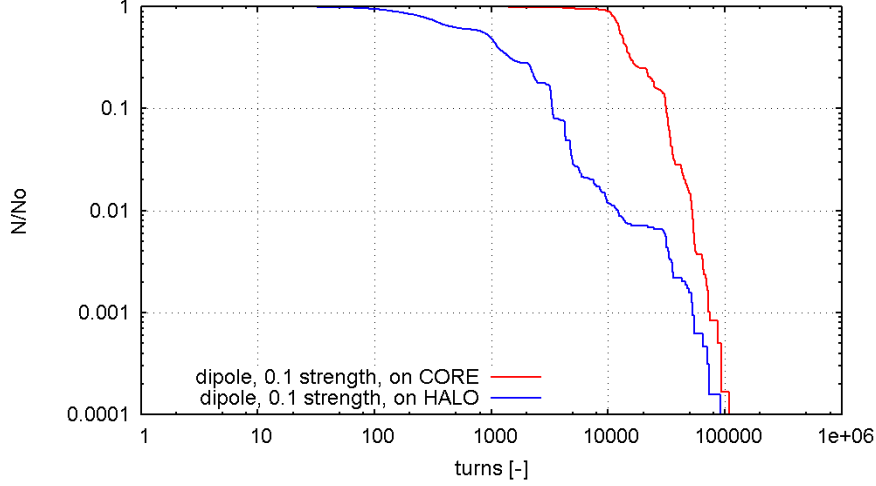


Figure 15: Survival particles in the halo and in the beam core for tune modulation dipole versus turn number.

## 5.2 Tune modulation with a quadrupole

Another considered option was to modulate a quadrupole elements to depopulate the beam halo. For preliminary studies, a thin quadrupole element was placed at the same location as the ADT system, with a gradient kick for length unit of  $2\mu\text{rad}/m^2$  and a length of 0.5m. This kick was chosen in order to obtain kicks (and therefore tune shifts) comparable with the e-lens. Two cases were studied: a quadrupole with gradient oscillating in the range  $[-2\mu\text{rad}; +2\mu\text{rad}]$  (denominated "inverse polarity") and the other one oscillating in the range  $[0; +2\mu\text{rad}]$  (denominated "one polarity"). The same tune range and tune sweeping pattern as the e-lens was applied.

In Figure 16 the number of halo particles not cleaned by the tune modulation element is shown. The simulation settings are identical to the e-lens settings described in Section 4.1. It can be noticed that, for the first case (inv. polarity) the halo removal efficiency is comparable with the e-lens one. When comparing the local scraping inefficiency for the two cases (Figure 17), it can be seen that the cleaning pattern is very different. While the e-lens is very effective for amplitudes larger than  $5\sigma_x$  and hardly affect lower amplitude halo particles, the tune modulation quadrupole has a non negligible effect also on  $4\sigma_x$  particles. The effect on the beam core still has to be checked.

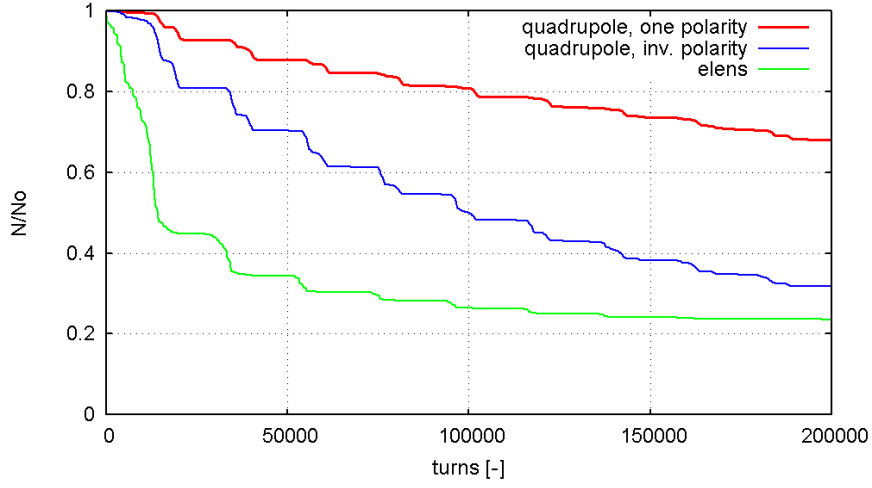


Figure 16: Survival particles in the halo for tune modulation quadrupoles compared with the e-lens case.

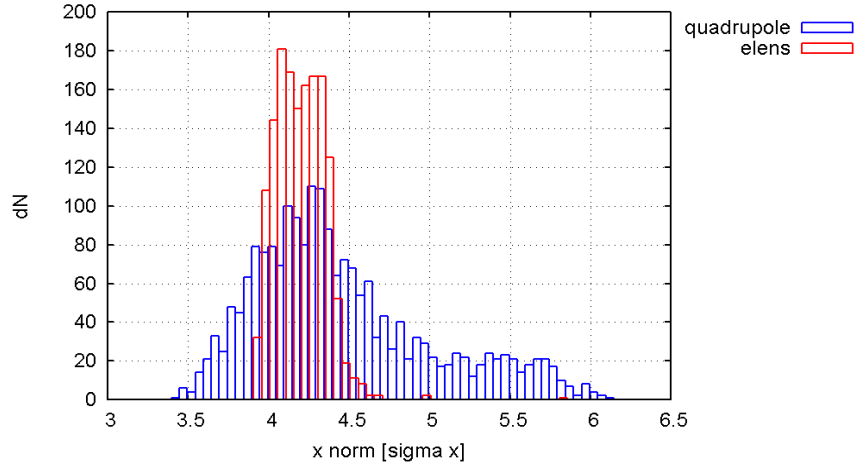


Figure 17: Local scraping inefficiency after  $2 \cdot 10^5$  turns for tune modulation quadrupole with inverted polarity compared to the e-lens one.

## 6 Conclusions

A new routine describing the effect of the e-lens in the LHC collimation system has been implemented in Sixtrack. Simulations assuming to use a hardware similar to the one presently installed in the Tevatron have been performed. The first simulations, performed for the nominal 7 TeV LHC optics, include sextupoles and octupoles but do not include Beam Beam, diffusion or multiple errors. Since one possible option is to use the electron lens immediately before the squeeze, the beam beam has not been inserted. However other non linear effect could play a major role in increasing the electron lens removal rate, and should be evaluated in future. Three different operation modes, compatible with the hardware requirements, have been tested: while the DC mode does not provide noticeable scraping efficiency, both the AC and the diffusive modes appears to be a viable solution for the LHC scraping needs. While the AC modes is at present the fastest option (removing 75% of halo particles in about 20s), the diffusive mode has the undeniable advantage of not depending on particle tune, and therefore being more robust, of much easier implementation and usage. Moreover, by increasing or decreasing the e-beam current, the scraping efficiency can be adapted to the machine needs. Scraping performances comparable to the AC mode seems achievable by doubling the e-beam current.

Preliminary investigation on tune modulation scraping techniques have been conducted. While the dipole modulation seems to be a non viable option, the quadrupole modulation gives results which are similar to the electron lens. The effect of the core, anyway, still has to be verified.

# Appendices

## A E-lens in Sixtrack

The purpose of this appendix is to describe the electron lens model which has been implemented in the tracking software SixTrack.

### A.1 General code description

The electron lens is implemented in the code in the `thin6d` routine, which can be found in the fortran source file **track.f**. While cycling along the optics sequence, the code checks the element name as provided in the sequence input file **fort.2**. If the name is compatible with an electron lens name (meaning it begins with the letters **ELENS**) then the collimation routine is called. Following the treatment which has been implemented for all the collimator-type elements in Sixtrack, the element is always recognized by its denomination. Within the collimator routine, a collimation routine is called for electron-lens elements: the name of the subroutine is *collimate\_elens*. In the *collimate\_elens* a cycle over the treated number of particle is opened. Within this cycle, the radial position of the particle is checked. If the Radial position is larger than the electron lens inner radius (also denominated **aperture** for consistency with other collimators), then the subroutine describing the electron lens kick is called *elens\_kick*.

### A.2 E-lens modelling

The device is treaded as a thin lens giving a single kick at the middle of its length. In Sixtrack an ideal electron lens is considered, i.e. a cylindrical distribution of electrons with internal radius  $R_1$  and external radius  $R_2$ . For the kick computation and the different radial distributions available (simplified or realistic profile), see Section 3.

The routine can provide four different operational modes, which are the same mode already discussed in this report:

- continuous mode: variable *op\_mode=0* in the code. Also called 'DC' mode in the code.
- diffusive mode: *op\_mode=1*. The electron lens is randomly switched on or off on turn by turn basis. Also called 'random' mode in the code.
- resonant mode with sinusoidal excitation: *op\_mode=2*. Also called 'AC' mode in the code. When this mode is activated, the electron

beam total current is pulsed with a frequency which is twice the natural betatron frequency of the system. Since more than one tune should be checked, a range of tune can be scanned. The parameters for this scan are user defined (see resonant mode in the collimator database description)

- resonant mode with "turn skip": *op\_mode=3*. Also callend 'resonant' mode in the code. The electron lens is switched on only every  $n$  turn, which are user-defined.

### A.3 Electron lens database

The electron lens parameters are passed to sixtrack through the input files which defines the collimator features, i.e. the collimator database. The code has been adapted to read a specific format for the electron lens entry. In this section the new entry defined for an electron lens type collimator is described (see Table 3) and the physical meaning of each one.

The electron lens element is represented in the optics input file as a thin element, essentially a marker which is recognized to be an electron lens only by the element name.

The description of the different entries is here detailed:

- *name1, name2*: name of the element in the optics sequence. Name1 must be in uppercase and name2 must be written in lowercase;
- *aperture*: electron lens half inner radius, expressed in normalized units [sigma] in the plane specified as "collimation plane" (see below);
- *length*: electron lens active length [m];
- *collimation plane*: designed collimation active plane, expressed as the angle between the horizontal plane and the collimation plane itself [rad];
- *x center, y center*: horizontal and vertical coordinates of the electron lens center [m];
- *current*: electron beam total current [A];
- *voltage*: accelerating voltage [eV];
- *R2 to R1 ratio*: ratio between the inner and the outer electron beam radius. The electron beam shape can in fact be rescaled by acting on the solenoidal magnetic field that confines the electron beam, but the ratio  $g = R_1/R_2$  is fixed by the geometry of the cathode;

Table 3: New format for electron lens entry in the collimation database input file.

line	TYPE <i>description</i>	TYPE <i>description</i>	...	...	...	...
1	STRING <i>name1</i>					
2	STRING <i>name2</i>					
3	DOUBLE <i>aperture</i>					
4	DOUBLE <i>length</i>					
5	DOUBLE <i>collimation plane</i>					
6	DOUBLE <i>Xcenter</i>	DOUBLE <i>Ycenter</i>				
7	DOUBLE <i>current</i>	DOUBLE <i>voltage</i>				
8	DOUBLE <i>R2/R1</i>					
9	INT <i>operation mode</i>	DOUBLE <i>operation tune</i>	DOUBLE <i>Delta tune</i>	DOUBLE <i>tune step</i>	INT <i>turns per step</i>	INT <i>resonant turns</i>
10	LOGIC <i>jitter</i>	LOGIC <i>radial pro- file</i>				
11	DOUBLE <i>betaX</i>					
12	DOUBLE <i>betaY</i>					

- *operation mode*: variable defining the electron lens operation mode: 0=continuous, 1=diffusive, 2=sinusoidal, 3=turn skip;
- *operation tune  $\nu_0$* : central value of the scanned tune range (only for sinusoidal resonant mode);
- *Delta tune  $\Delta\nu$* : The half width of the tune range which must be scanned in sinusoidal resonant mode. The tune range is defined as:



$[\nu_0 - \Delta\nu : \nu_0 + \Delta\nu]$   
 (only for sinusoidal resonant mode);

- *tune step* : the steps in tune which are used to scan the tune range in sinusoidal resonant mode. (only for sinusoidal resonant mode);
- *turns per step*: number of turns which are between two tune steps (only for sinusoidal resonant mode);
- *resonant turns*: the electron lens is switched on only if the turns is a multiple of the variable *resonant turns* (only for turn skip mode);
- *jitter*: activate or de-activate a 2% jitter in the electron lens total current. It is compatible with all operation modes. [TRUE-FALSE];
- *radial profile*: activate or de-activate the measured current radial profile. If set to false, a uniform distribution of electrons in the circular region defined by the inner and outer radius is defined. If set to true, the shape of a measured distribution has been taken as reference. [TRUE-FALSE];
- *beta x - y*: horizontal and vertical beta functions at the electron lens locations. These values are considered only if the appropriate variable (do\_nominal) is selected in the fort.3 input file. [m]

#### A.4 Additional output

Two additional output has been introduced mainly for debugging purposes. The output names are called **elens.bin** and **elens.norm.bin** and provide the full 6D description of all the particles passing through the electron lens with a radial position larger than the inner electron lens radius, at each turn.

Being the outputs typically of the order of hundreds of Mb, it was chosen to write them unformatted (in binary) so to save some space. Appropriate fortran programs are accessible for converting them in text files, if desired, during the post processing. A dedicated flag (logical variable) in the main Sixtrack input file **fort.3** has been inserted in order to activate/deactivate such inputs (line 10 of the collimation block, column 10).

The file headers read as:

##### File **elens.bin**

---

1=sample 2=npart 3=nturn 4=x0[m] 5=xp0[rad] 6=y0[m] 7=yp0[rad]  
 8=kx[rad] 9=ky[rad] 10=kr[rad]

---

where the first two columns permits to identify the particle in Sixtrack, the third column is the turn number and the other columns contain the 4D coordinates in the physical space and the received corresponding kicks in horizontal, vertical and radial direction.

## File **elens.bin**

---

1=sample 2=npart 3=nturn 4=x0[ $\sigma_x$ ] 5=xp0[ $\sigma'_x$ ] 6=y0[ $\sigma_y$ ] 7=yp0[ $\sigma'_y$ ]  
8=kx[ $\sigma'_x$ ] 9=ky[ $\sigma'_y$ ] 10=kr[ $\sigma'_r$ ]

---

where the first two columns permits to identify the particle in Sixtrack, the third column is the turn number and the other columns contain the 4D coordinates in the nomalized space and the received corresponding kicks in horizontal, vertical and radial direction.

## A.5 Fortran code

### A.5.1 collimate\_elense subroutine

```
!*****
!  

! VALENTINA PREVITALI - 2012
! JCSMITH
!  

!  

! subroutine collimate_elense (elense_r_min, elense_curr,  

!      length, voltage, r2_ov_r1, center_x, center_y,  

!      op_mode, tune, Delta_tune, step_tune, step_turns,  

!      resonant_turns, elens_jitter, elens_radial,  

!      x_in, xp_in, y_in, yp_in,  

!      p_in, s_in, np, enom, lhit, part_abs, impact, indiv, lint, name)  

! Calculates radial kick of beam passing through electron lense. Assumes a  

! radially symmetric lense and no slicing. Does not address fringe field. Applies  

! kick at center of element, it's a thin lense, no actual tracking is performed.  

!  

! from: V. Shiltsev, et al. "LHC Particle Colimation by Hollow Electron Beams"  

! EPAC08 MOPC098  

!  

! Parameters:  

! npart      -- total number of particles := 64  

! maxn      -- 1000000  

!  

! Input:  

! elense_radius  -- double precision: electron lens inner radius (m)  

! elense_curr    -- double precision: electron lens total current (A)  

! length         -- double precision: electron lens lenght (m)  

! voltage        -- double precision: electron lens Voltage (kV)  

! r2_ov_r1       -- double precision: ratio between external and  

!                  internal cathode radius  

! center_x       -- double precision: center x coordinate  

! center_y       -- double precision: center y coordinate  

! op_mode        -- integer: electron lens operation mode  

!                  1- random mode  

!                  2- AC mode
```

```

!                                     3- resonant mode
!                                     otherwise the electron lens is operated in DC mode
! tune -- double precision: (for AC mode only) elens operational tune
! Delta_tune -- double precision: (for AC mode only) maximum variation in tune
! step_tune -- double precision: (for AC mode only) tune variation per step
! step_turns -- integer: (for AC mode only) n. of turns per step
! resonant_turns -- integer: (for resonant mode only) turn period
! elens_jitter -- logical: switch on/off a 2% jitter in electron lens current
! elens_radial -- logical: switch on/off a realistic radial profile i(r)
! x_in(npart) -- double precision: incoming particle horizontal position [m]
! xp_in(npart) -- double precision: incoming particle horizontal angle [rad]
! y_in(npart) -- double precision: incoming particle vertical position [m]
! yp_in(npart) -- double precision: incoming particle vertical angle [rad]
! p_in(npart) -- double precision: incoming particle momentum (GeV)
! s_in(npart) -- double precision: incoming particle longitudinal position [m]
! NOTE: all particles start at same longitudinal position!
! np -- integer: total number of tracked particles (.le. npart)
! enom -- double precision: reference energy (MeV)
! Output:
! x_in(npart) -- double precision: outgoing particle horizontal position [m]
! xp_in(npart) -- double precision: outgoing particle horizontal angle [rad]
! y_in(npart) -- double precision: outgoing particle vertical position (meters?)
! yp_in(npart) -- double precision: outgoing particle vertical angle [rad]
! p_in(npart) -- double precision: outgoing particle momentum (Gev)
! s_in(npart) -- double precision: outgoing particle longitudinal position [m]
! /info/ ! do not modify:
! ie: element number
! iturn: turn number
! nabs_total: not used
!
!*****

subroutine collimate_elense (elense_r_min, elense_curr,
& length, voltage, r2_ov_r1, center_x, center_y,
& op_mode, tune, mult_tune, Delta_tune, step_tune, step_turns,
& resonant_turns, elens_jitter, elens_radial,
& x_in, xp_in, y_in, yp_in,
& p_in, s_in, np, enom, lhit, part_abs, impact, indiv, lint, name)

implicit none

! Parameters defined yet again...
integer, parameter :: npart = 64

double precision elense_r_min, elense_curr, curr,
& length, voltage, r2_ov_r1, center_x,
& center_y, tune, mult_tune, Delta_tune, step_tune
logical elens_jitter, elens_radial
integer op_mode, step_turns, resonant_turns

double precision x_in(npart), xp_in(npart),
& y_in(npart), yp_in(npart), p_in(npart), &
& s_in(npart), enom, impact(npart), indiv(npart), &
& lint(npart), sigma

logical switch
real rndm4

c
double precision p_radius, p_angle
double precision, target :: coord(npart,6)
double precision, pointer :: icoord(:)

```

```

integer np, name(npart), lhit(npart), this_elense
integer part_abs(npart)
integer i_part

! p mass in GeV, defined yet again...

!Common blocks

integer ie, iturn, nabs_total, icoll

common /info/ ie, iturn, nabs_total

common /icoll/ icoll

!Big ugly GRD common block! WHY WHY WHY?
logical do_coll, do_select, do_nominal, dowrite_dist, do_oneside, &
&dowrite_impact, dowrite_secondary, dowrite_amplitude, radial, &
&systilt_antisymm, dowritetracks, cern, do_nsig, do_mingap
&, physical, diffusive
integer nloop, rnd_seed, c_offsettilt_seed, ibeam, jobnumber, &
&do_thisdis, n_slices, pencil_distr
double precision myenom, mynex, mdex, myney, mdey, &
&nsig_tcp3, nsig_tcs3, nsig_tcs3, nsig_tcla3, &
&nsig_tcp7, nsig_tcs7, nsig_tcs7, nsig_tcla7, nsig_tclp, nsig_tcli, &
&nsig_tcth1, nsig_tcth2, nsig_tcth5, nsig_tcth8, &
&nsig_tctv1, nsig_tctv2, nsig_tctv5, nsig_tctv8, &
&nsig_tcdq, nsig_tcstcdq, nsig_tdi, nsig_tcxrp, nsig_tcryo, &
&smin_slices, smax_slices, recenter1, recenter2, &
&fit1_1, fit1_2, fit1_3, fit1_4, fit1_5, fit1_6, ssf1, &
&fit2_1, fit2_2, fit2_3, fit2_4, fit2_5, fit2_6, ssf2, &
&emitx0, emity0, xbeat, xbeatphase, ybeat, ybeatphase, &
&c_rmstilt_prim, c_rmstilt_sec, c_systilt_prim, c_systilt_sec, &
&c_rmsoffset_prim, c_rmsoffset_sec, c_sysoffset_prim, &
&c_sysoffset_sec, c_rmerror_gap, nr, ndr, &
&driftsx, driftsy, pencil_offset, pencil_rmsx, pencil_rmsy, &
&sigsecut3, sigsecut2, enerror, bunchlength
character*80 filename_dis
character*24 name_sel
character*80 coll_db
character*16 castordir
double precision, PARAMETER :: Pi = 3.14159265
common /grd/ myenom, mynex, mdex, myney, mdey, &
&nsig_tcp3, nsig_tcs3, nsig_tcs3, nsig_tcla3, &
&nsig_tcp7, nsig_tcs7, nsig_tcs7, nsig_tcla7, nsig_tclp, nsig_tcli, &
!
&nsig_tcth1, nsig_tcth2, nsig_tcth5, nsig_tcth8, &
&nsig_tctv1, nsig_tctv2, nsig_tctv5, nsig_tctv8, &
!
&nsig_tcdq, nsig_tcstcdq, nsig_tdi, nsig_tcxrp, nsig_tcryo, &
!
&smin_slices, smax_slices, recenter1, recenter2, &
&fit1_1, fit1_2, fit1_3, fit1_4, fit1_5, fit1_6, ssf1, &
&fit2_1, fit2_2, fit2_3, fit2_4, fit2_5, fit2_6, ssf2, &
!
&emitx0, emity0, xbeat, xbeatphase, ybeat, ybeatphase, &
&c_rmstilt_prim, c_rmstilt_sec, c_systilt_prim, c_systilt_sec, &
&c_rmsoffset_prim, c_rmsoffset_sec, c_sysoffset_prim, &
&c_sysoffset_sec, c_rmerror_gap, nr, &
!
&ndr, driftsx, driftsy, pencil_offset, pencil_rmsx, pencil_rmsy, &

```

```

&physical,diffusive,
&sigsecut3,sigsecut2,enerror,
&bunchlength,coll_db,name_sel,
&castordir,filename_dis,nloop,rnd_seed,c_offsettilt_seed,
&ibeam,jobnumber,do_thisdis,n_slices,pencil_distr,
&do_coll,
!
&do_select,do_nominal,dowrite_dist,do_oneside,dowrite_impact,
&dowrite_secondary,dowrite_amplitude,radial,systilt_antisymm,
&dowritetracks,cern,do_nsig,do_mingap
! Use c_aperture as inner diameter

switch=.false.
curr=elense_curr
if (elens_jitter) then
c      write(*,*)"elens jitter ON"
      curr=elense_curr+elense_curr*(rndm4()/100*2)
endif

if (rndm4() .le. 0.5) switch=.true.

! Switch to elense frame
coord(:np,1) = x_in(:np)-center_x
coord(:np,2) = xp_in(:np)
coord(:np,3) = y_in(:np)-center_y
coord(:np,4) = yp_in(:np)
coord(:np,5) = s_in(:np) ! absolute s poosition!
coord(:np,6) = p_in(:np) ! absolute energy in Gev!
c      call coord_transformation (.true., coord)

! main loop over all particles
do i_part = 1, np
! don't loop over absorbed particles
if (part_abs(i_part) .ne. 0) cycle
! initialize parameters
impact(i_part) = -1d0
lint(i_part) = -1d0
indiv(i_part) = -1d0

icoord => coord(i_part,:)

!consider the particle coordinates at the middle of the elens
icoord(1) = icoord(1) + 0.5*length*icoord(2)
icoord(3) = icoord(3) + 0.5*length*icoord(4)

! find if particle hits e lense field
p_radius = sqrt(icoord(1)**2 + icoord(3)**2)
p_angle = atan(icoord(3)/icoord(1))
if (icoord(1).lt.0 ) then
p_angle=p_angle+4*atan(1.)
endif

if (p_radius .ge. elense_r_min ) then
call elense_kick(icoord,elense_r_min,p_radius,p_angle,
&      length,voltage,
&      r2_ov_r1, curr, op_mode, tune,mult_tune,
&      Delta_tune, step_tune, step_turns,
&      resonant_turns,elens_radial, switch
&      )

```

```

        lhit(i_part) = 100000000*ie + iturn
        indiv(i_part) = sqrt(icoord(2)**2 + icoord(4)**2)
        lint(i_part) = length
        part_abs(i_part) = 0 ! particles never absorbed
        impact(i_part) = p_radius - elense_r_min
    endif
    ! In order to work with track6d routine have to first track to end of
    ! collimator.
    icoord(1) = icoord(1) + 0.5*length*icoord(2)
    icoord(3) = icoord(3) + 0.5*length*icoord(4)

enddo

! Switch back to accelerator frame
c    call coord_transformation (.false., coord)

!transfer working coords to final coords
x_in(np) = coord(np,1) + center_x
xp_in(np) = coord(np,2)
y_in(np) = coord(np,3) + center_y
yp_in(np) = coord(np,4)
s_in(np) = coord(np,5)
p_in(np) = coord(np,6)

end subroutine collimate_elense

```

### A.5.2 elense\_kick subroutine

```

!    ***
    subroutine elense_kick(coord, elense_r_min,p_radius, p_angle,
&        hit_length,
&        V, r2_ov_r1, curr,
&        op_mode, m_tune, mult_tune,
&        Delta_tune, step, period, repetition,
&        elens_radial,switch)

    implicit none

    double precision coord(6), r_kick, max_kick, Brho, hit_length
    double precision, intent(in) :: p_radius, p_angle,elense_r_min
    double precision gamma_pr, beta_pr, beta_el
    double precision coord2_temp, coord4_temp
    double precision phi
    double precision V,p_el, r2_ov_r1, curr !parameters for
        !electron lens simplified. cathode voltage [KV],
        ! electron beam momentum [KeV],
        !ratio R1/R2, current [A]
    double precision f !shape function for elense kick

c    RADIAL PROFILE
    logical :: elens_radial
    double precision r0,x0,y0
    double precision r1,x1,y1
    double precision r2,x2,y2
    double precision r3,x3,y3
    double precision r4,x4,y4
    double precision n0,n1,n2,n3,ntot

```

```

c      OPERATION MODE
!      LHC tune with octupoles H= .311->.312 ;
!      LHC tune no octupoles H= .31->.3105 ; V=.32 ->.3205
!      SPS tune no octupoles H= .13 ->      ; V=.18 ->
c
      integer op_mode

      logical :: AC
      double precision m_tune,mult_tune,Delta_tune,nstep, step
      double precision min_tune,op_tune
      integer period

c
      logical :: random
      logical :: switch

c
      logical :: resonant
      integer repetition

c
      logical write_c_out, write_SPS_out, write_elens_out
&      , write_TM_QUAD_out
      integer ie,iturn,nabs_total, icoll

      double precision, PARAMETER :: Pi = 3.14159265
      double precision, PARAMETER :: e0=.00000000000885418781762
      double precision, PARAMETER :: c=299792458.
      double precision, parameter :: pmap_gev = 0.938271998d0

      common /info/ ie,iturn,nabs_total
      common /outputs/ write_c_out, write_SPS_out, write_elens_out
&      , write_TM_QUAD_out

      p_el=V
      Brho = 3.33564095 * sqrt(coord(6)**2 - pmap_gev**2)
      gamma_pr = (coord(6)) / pmap_gev !convert momentum to GeV
      beta_pr = sqrt(1 - (1/gamma_pr**2))
      beta_el = sqrt(1 - 1/((511.+p_el)/511.)**2)
      max_kick = (2*hit_length*curr*(1+beta_el*beta_pr))
&      /(4*(4*atan(1.))*e0*p_radius*Brho*beta_el*beta_pr*c**2)

      AC=.false.
      random=.false.
      resonant=.false.
      if (op_mode .eq. 1) then
c          write(*,*) "random"
          random=.true.
      elseif (op_mode .eq. 2) then
c          write(*,*) "AC"
          AC=.true.
      elseif (op_mode .eq.3) then
          write(*,*) "resonant"
          resonant=.true.
c          write(*,*) "DC"

      endif

      ! three regions
      if (p_radius .lt. elense_r_min) then ! hits elense
          f=0
          write(*,*) "elense warning! this should not happen..."
      elseif (p_radius .le. (r2_ov_r1*elense_r_min)) then
          f=(p_radius**2-elense_r_min**2)/

```

```

&          ( (r2_ov_r1*elense_r_min)**2-elense_r_min**2)
else
    f=1
endif

if (elens_radial) then
c      write(*,*)"elens radial!!"
    r0=445./2.
    x0=1*elense_r_min
    y0=0.
    r1=505./2
    x1=r1/r0*elense_r_min
    y1=917.
    r2=574./2
    x2=r2/r0*elense_r_min
    y2=397.
    r3=729./2
    x3=r3/r0*elense_r_min
    y3=228.
    r4=853./2
    x4=r4/r0*elense_r_min
    y4=0.
    n0=((y1-y0)/(x1-x0)/3)*x1**3+(y0 -x0*(y1-y0)/(x1-x0))*x1**2/2
&    -(((y1-y0)/(x1-x0)/3)*x0**3+(y0 - x0*(y1-y0)/(x1-x0))*x0**2/2)
    n1=(y2-y1)/(x2-x1)*x2**3/3+(y1 - x1 *(y2-y1)/(x2-x1))*x2**2/2
&    -((y2-y1)/(x2-x1)*x1**3/3+(y1 - x1 *(y2-y1)/(x2-x1))*x1**2/2)
    n2=(y3-y2)/(x3-x2)*x3**3/3+(y2 - x2 *(y3-y2)/(x3-x2))*x3**2/2
&    -((y3-y2)/(x3-x2)*x2**3/3+(y2 - x2 *(y3-y2)/(x3-x2))*x2**2/2)
    n3=(y4-y3)/(x4-x3)*x4**3/3+(y3 - x3 *(y4-y3)/(x4-x3))*x4**2/2
&    -((y4-y3)/(x4-x3)*x3**3/3+(y3 - x3 *(y4-y3)/(x4-x3))*x3**2/2)
    ntot=n0+n1+n2+n3
    if (p_radius.lt.x0) then
        f=0
        write(*,*) "elense warning! this should not happen..."
    elseif (p_radius.lt.x1) then
        f= (((y1-y0)/(x1-x0)/3)*p_radius**3+(y0 - x0 *(y1-y0)/
&          (x1-x0))*p_radius**2/2 -(((y1-y0)/(x1-x0)/3)*x0**3+(y0
&          - x0 *(y1-y0)/(x1-x0))*x0**2/2))/ntot

    elseif (p_radius.lt.x2) then
        f= (n0 + (y2-y1)/(x2-x1)*p_radius**3/3+(y1 - x1
&          *(y2-y1)/(x2-x1))*p_radius**2/2 - ((y2-y1)/(x2-x1)*
&          x1**3/3+(y1 - x1*(y2-y1)/(x2-x1))*x1**2/2))/ntot

    elseif (p_radius.lt.x3) then
        f=(n0+n1+(y3-y2)/(x3-x2)*p_radius**3/3+(y2 - x2
&          *(y3-y2)/(x3-x2))*p_radius**2/2 - ((y3-y2)/(x3-x2)*
&          x2**3/3+(y2 - x2*(y3-y2)/(x3-x2))*x2**2/2))/ntot
    elseif (p_radius.lt.x4) then
        f=(n0+n1+n2+(y4-y3)/(x4-x3)*p_radius**3/3+(y3 - x3
&          *(y4-y3)/(x4-x3))*p_radius**2/2 - ((y4-y3)/(x4-x3)*
&          x3**3/3+(y3 - x3*(y4-y3)/(x4-x3))*x3**2/2))/ntot
    else
        f=1
    endif
endif

r_kick = - f*max_kick !minus is because the kick in inward

if (AC) then
    if (step.ne.0 .and. Delta_tune .ne. 0) then

```



```

        min_tune=m_tune-Delta_tune
        nstep=(2*Delta_tune/step)+1
        op_tune=
&         (min_tune+mod(floor(DBLE(iturn)/
&         DBLE(period)),(nstep))*step)
        else
            op_tune=m_tune
        endif
        phi=mult_tune*(iturn*op_tune*2*(4*atan(1.)))
        r_kick=r_kick*0.5*(1+cos(phi))

elseif (switch .and. random) then
    r_kick = 0
elseif (resonant .and. mod(iturn,repetition).gt.0) then
    r_kick = 0
endif

coord2_temp = coord(2)
coord4_temp = coord(4)
! Apply the kick in the appropriate ratios
coord(2) = coord(2) + r_kick*cos(p_angle)
coord(4) = coord(4) + r_kick*sin(p_angle)

end subroutine elense_kick_vp

```

## References

- [1] B. Salvachua *et al.*, “Cleaning Performance of the LHC Collimation System up to 4 TeV”, IPAC13 Proc., China, 2013
- [2] “Internal review of Tevatron hollow e-lens usage at CERN”, Nov. 21012 CERN, Geneva, Switzerland, <http://indico.cern.ch/conferenceDisplay.py?confId=213752>
- [3] V. Shiltsev *et al.*, “Tevatron Electron Lenses: Design and Operation,” Phys. Rev. ST Accel. Beams **11**, 103501 (2008).
- [4] G. Stancari *et al.*, “Experimental study of magnetically confined hollow electron beams in the Tevatron as collimators for intense high-energy hadron beams,” Conf. Proc. C **110328**, 370 (2011).
- [5] G. Stancari *et al.*, “Collimation with Hollow Electron Beams”, Phys. Rev. Lett. 107, 084802 (2011)
- [6] V. Previtali *et al.*, “Numerical Simulations of Transverse Beam Diffusion Enhancement by the Use of Electron Lens in the Tevatron Collider,” Conf. Proc. C **1205201**, 1113 (2012).
- [7] V. Moens *et al.*, “Electron Hollow Gun Collimation, Performance and Simulations”, APC seminar, April 18, 2013, FNAL (Batavia, IL).
- [8] G. Demolaize *et al.*, “A New Version of SixTrack with Collimation and Aperture interfaces”, PAC05 Proc., USA, 2005
- [9] L.D. Landau and E.M. Lifschitz, Mechanics, Nauka Publishers, Moscow, 1988.
- [10] G. Valentino *et al.*, “Beam Diffusion Measurements Using Collimator Scans in the LHC,” Phys. Rev. ST Accel. Beams 16-021003 (2013)
- [11] W. Hoeffle *et al.*, “The LHC Transverse Damper (ADT) Performance Specification”, SL-Note-99-055 HRF, CERN (1999)