

Large Hadron Collider Project

LHC Project Report 134

Numerical Optimization of Collimator Jaw Orientations and Locations in the LHC

D.I.Kaltchev*, M.K.Craddock*[‡], R.V.Servranckx*, J.B.Jeanneret^{†,§}

Abstract

The collimation system of LHC will consist of flat collimator jaws distributed along the IR7 lattice with the aim of limiting the maximum combined amplitudes of secondary halo particles (born along the edges of the primary collimators). The code DJ (Distribution of Jaws) computes this amplitude using a quasi-analytic algorithm (no tracking), by which the maximum initial angles are found, corresponding to trajectories escaping all secondary jaws. We report the latest version of DJ, which contains the following enhancements: (1) the orientation of each pair of jaws is a free variable (instead of using only vertical, horizontal, or 45 degrees skew jaws); (2) the minimizing method used is "simulated annealing", which, for our case of a discontinuous function of up to 32 variables, always finds a global minimum. Different initial jaw distributions lead to different final ones, but they all give essentially the same maximum halo amplitude; this seems to depend only on the number of jaws and the lattice parameters, particularly the tune-split. We discuss lattice characteristics found favorable for collimation.

Paper presented at the Particle Accelerator Conference, PAC97, Vancouver, Canada, 12 May 1997.

*TRIUMF, 4004 Wesbrook Mall, Vancouver, B.C., Canada V6T2A3

†CERN-SL Division

‡Also at Dept. of Physics & Astronomy, UBC, Vancouver, Canada.

§bjb@mail.cern.ch

Administrative Secretariat
LHC Division
CERN
CH-1211 Geneva 23
Switzerland

Geneva, September 1997

1 INTRODUCTION

The betatron beam collimation system for the LHC will be installed in the IR7 insertion. It will consist of a set of primary collimators, followed by a number of secondary collimators arranged to limit the so-called secondary beam halo produced at the edges of the primaries, thereby protecting the LHC vacuum chamber from scattered particles. Each collimator will be composed of a pair of opposing flat jaws.

In [1] we presented the computer code DJ, which distributes secondary jaws along the IR7 lattice with the aim of minimizing the largest surviving combined amplitude of the halo. We now report some enhancements to DJ and the improved results which they have made possible.

2 DEVELOPMENT OF THE MODEL

2.1 Varying the jaw angles

In the code DJ, the production of halo particles is modeled by a set of point-like sources distributed along the borders of the primary jaws (straight lines in the transverse plane). The secondary jaws, assumed to act as black absorbers, are defined by their horizontal tune advance μ_x (within the collimation section IR7) and their rotation angle α around the longitudinal axis.

For a given jaw distribution, a mapping technique is used to isolate the fraction of trajectories escaping all secondary jaws, i.e. those passing between the two opposing jaws of all pairs. For these uncaptured halo particles, the code finds the maximum combined x - y betatron amplitude A_{max} , where $A = (A_x^2 + A_y^2)^{1/2}$ and A_x, A_y are the single-plane transverse invariants. This computation is fast (~ 1 s) as no tracking is needed. DJ further minimizes A_{max} as a function of the jaw distribution vector $(\mu_{x1}, \mu_{x2}, \dots, \mu_{xN}, \alpha_1, \alpha_2, \dots, \alpha_N)$.

In [1] four types of jaws are used – vertical, horizontal and two skew – with rotation angles $\alpha = 0^\circ, 90^\circ, 45^\circ, 135^\circ$ respectively. In the new version of the code DJ, the angle α is an independent variable, along with the jaw position, and may range over $0^\circ \leq \alpha \leq 180^\circ$; during minimization the jaw positions and angles are varied together. The improvement in A_{max} (expressed in terms of the r.m.s. amplitude σ) is shown in Table 1, for primary and secondary collimators set at 6σ and 7σ respectively. For 12 secondary collimators, allowing α to vary has the same effect as adding 4 more secondaries; for 16 secondaries the improvement is less dramatic.

Table 1:

Number of secondary pairs of jaws	A_{max} discrete $\alpha = 0^\circ, 90^\circ, 45^\circ, 135^\circ$	A_{max} $0^\circ \leq \alpha \leq 180^\circ$
12	9.4σ	8.7σ
16	8.6σ	8.4σ

With variable angles, the program module calculating A_{max} for a fixed jaw distribution remains unchanged. However, the minimization of A_{max} with twice as many variables called for some new numerical tools and solutions.

2.2 Minimization

During minimization, the locations of the four primary jaws are fixed at maxima of the corresponding beta functions, their angles being: $0^\circ, 90^\circ, 45^\circ$ and 135° .

The optimization process has been developed in two stages: 1) conventional methods, which allow better insight, but show an unwelcome dependence on the initial conditions, with some runs ending up in local minima, and 2) simulated annealing.

A_{max} is not a smooth function of the jaw-distribution vector, because of screening effects by some secondary jaws on others. A step of *finite* length made in any direction of the $2N$ -coordinate space may result in unpredictable changes in the indices m, n of the two *maximum amplitude jaws* [1] and correspondingly in A_{max} . However, for small enough increments, the coordinates of the two maximum-amplitude jaws are the only ones whose variables affect A_{max} , so there are only four active variables $\mu_m, \alpha_m, \mu_n, \alpha_n$ (local smoothness).

Downhill-overstep methods [2] are based on the local smoothness of A_{max} . At each iteration, the LMDIF package routine is used with four variables up to the limits of the smoothness interval. After that, a step is made outside this interval, to pick a new pair of active jaws. The step is halved after each unsuccessful iteration (no downhill direction found).

Simulated annealing (SA) – a probabilistic optimization method [3], is a recent technique devised for solving difficult problems involving discontinuous multi-variable functions, but requiring large computing time. The Appendix offers a quick overview of SA in one dimension.

At early stages of minimization, if the percentage of accepted cases rises, then the range over which the code searches for an optimum increases, i.e. the SA algorithm keeps more than one local minimum in sight. As the “temperature” parameter is reduced, downhill moves are less likely to be accepted, more cases are rejected and SA focuses on the global extremum.

In several initial runs, appropriate values were chosen for the most important SA parameters – the initial temperature ($T_0 = 5$) and the temperature reduction factor (0.6). A typical SA run assumed fixed IR7 lattice functions, a sufficiently large number of source points along the edges of the four primary collimators, and 12 to 16 secondary collimators.

With these parameters fixed, SA runs made for random initial jaw distributions always resulted in essentially the same minimum value for A_{max} , as desired. The final jaw distributions, however, were by no means identical, although many were very similar (see §3.2 below). The secondary-halo cross-sections differed correspondingly, having different maximum single-plane invariants A_{xmax} and A_{ymax} (but the same amplitude near the diagonal in (A_x, A_y) space).

If DJ is modified to search only for jaw locations compatible with the rest of the hardware, then the computing time increases unacceptably. The alternative approach was taken of shifting the quadrupoles slightly to free locations at which jaws were needed.

3 RESULTS

Different IR7 tunes were explored for several recent versions of the LHC lattice, and for 16 secondary collimators A_{max} was found to be between 8.4σ and 9.1σ , depending on the lattice setting.

3.1 Optimum lattice setting

Optics criteria can be formulated in terms of $\mu_x(s)$ and $\mu_y(s)$ – the horizontal and vertical tune advances along the straight section, with $\mu_x(s_0) = \mu_y(s_0) = 0$ at the first primary collimator ($s_0 = 290$ m), or equally, in terms of the functions $\mu^\pm = (\mu_y \pm \mu_x)/2$. The average tune advance μ^+ is roughly proportional to the distance from the

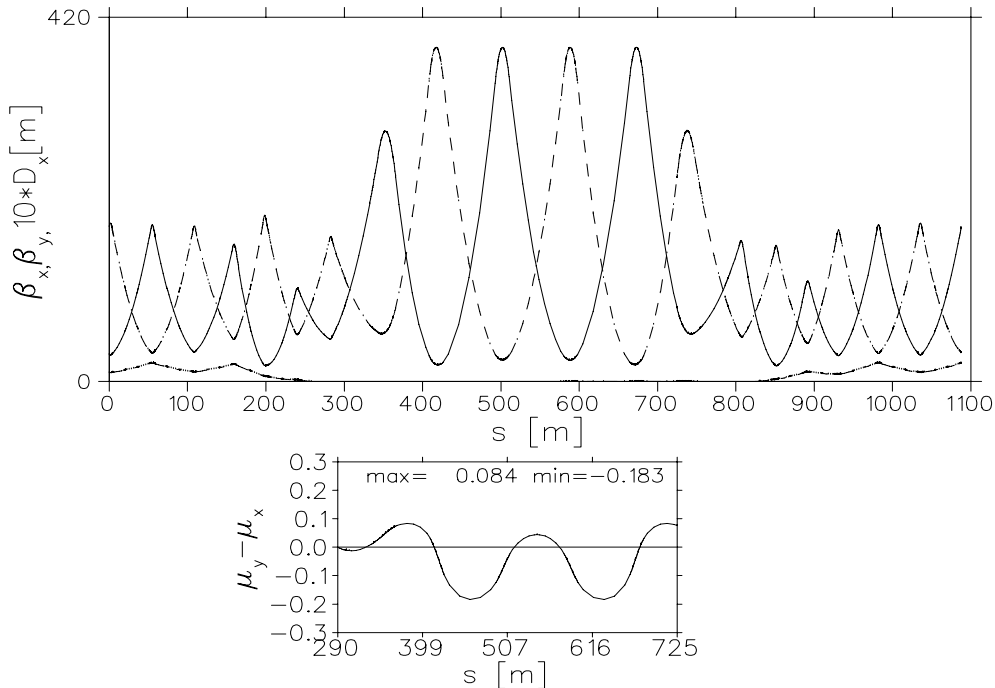


Figure 1: IR7 lattice and tune-split functions for LHC version 4.2, with IR7 quadrupoles tuned for high negative tune split, giving $A_{max} = 9.1\sigma$.

first primary: $\mu^+ \propto s - s_0$. Therefore, for a fixed length of the collimation section, the collimation quality can essentially be expressed in terms of the total μ^+ and the tune-split function $\mu^-(s)$.

The advantage of having the tune split vary along the beamline was first suggested, and confirmed by tracking, by Risselada [4]. For the case of circular collimators, initial studies have been carried out [5], aiming to explain the relation between the shape of $\mu^-(s)$ and the collimation quality, and a search for a rigorous theory is under way.

As reported in [1], larger oscillations in $\mu^-(s)$ give lower A_{max} , but we have also found dependence on the sign of μ^- . The figures below show the lattice and tune-split functions of IR7 for two cases recently studied: a tune giving large negative μ^- and $A_{max} = 9.1\sigma$ (Fig. 1), and a tune giving large positive μ^- and $A_{max} = 8.45\sigma$ (Fig. 2).

The following tune-split variation gave $A_{max} < 8.5\sigma$:

- almost everywhere positive and close to periodic, with three nearly equal maxima ~ 0.2 each (Fig. 2, bottom);
- one high peak in the middle ~ 0.25 (an abandoned lattice version, not shown).

On the other hand, a tune giving two large negative peaks in μ^- (Figure 1, bottom), gives a somewhat higher $A_{max} = 9.1\sigma$.

3.2 Optimum collimator phases

For a lattice optimized purely for collimation, some relation is to be expected between the horizontal and vertical betatron phases of perfectly located collimators. Even in realistic lattices, constrained by additional factors, the SA runs showed that this remains true, favouring certain jaw locations. We also found that μ^+ is a more relevant independent variable than s .

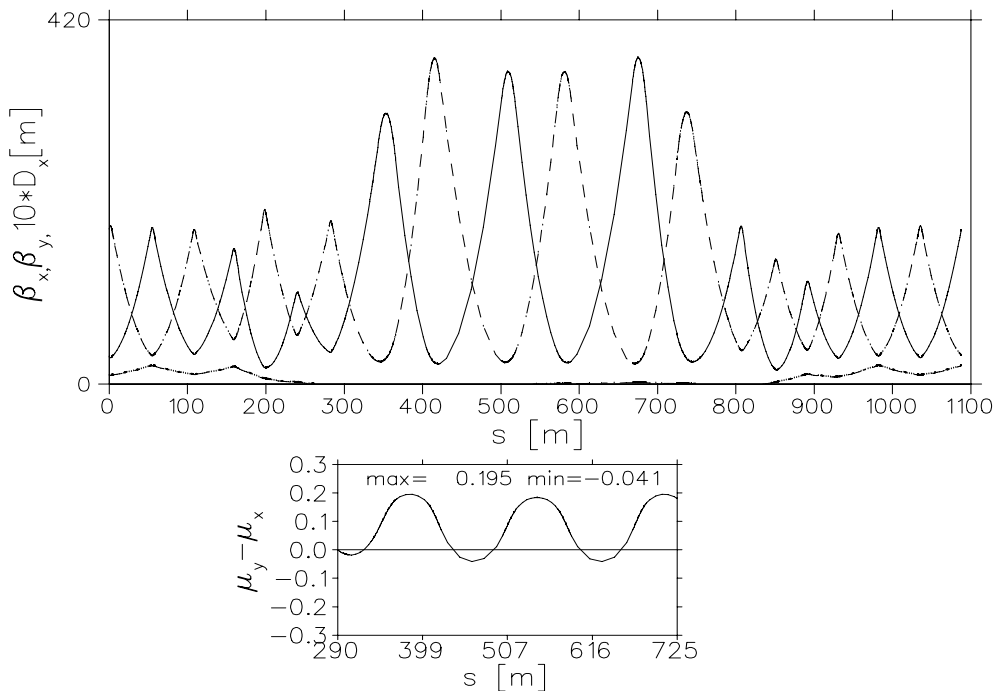


Figure 2: IR7 lattice and tune-split functions for LHC version 4.2, with the IR7 quadrupoles tuned for high positive tune split, giving $A_{max} = 8.45\sigma$.

Fifty SA runs were performed for nearly optimum conditions ($A_{max} = 8.45\sigma$) using the lattice shown in Fig. 2. Each run used a randomly generated initial jaw distribution, i.e. random angles and phases for 16 collimators. The values of α were then plotted against μ^+ (Fig. 3) for the resultant 50 jaw distributions, which all give nearly the same value of A_{max} : $8.4\sigma < A_{max} < 8.5\sigma$. The jaw locations tend to cluster near the extreme values of the function $\mu^-(\mu^+)$. On the other hand, for the lattice with $A_{max} = 9.1\sigma$ the pattern is more chaotic (Fig. 4).

References

- [1] D. Kaltchev, M.K. Craddock, R.V. Servranckx, J.B. Jeanneret, Optimization of Collimator Jaw Locations for the LHC, Proc. EPAC96 (Barcelona, June 1996), ed. S. Myers et al., (IOP, Bristol, 1996), 1432-4.
- [2] D. Kaltchev, M.K. Craddock, R.V. Servranckx, Progress Report on Optimizing Collimator Jaw Angles and Positions with the Code DJ (Distribution of Jaws), TRIUMF design note (in preparation).
- [3] A. Corana, M. Marchesi, C. Martini, S. Ridella, Minimizing multi-modal functions of continuous variables with the "simulated annealing" algorithm, ACM Transactions on Mathematical Software Vol. 13, No. 3 (Sept. 1987), 262-280.
- [4] T. Risselada, Optical Requirements for an LHC Cleaning insertion with Elliptical Collimators, SL Note 95-67, 1995.
- [5] T. Risselada, private communication.

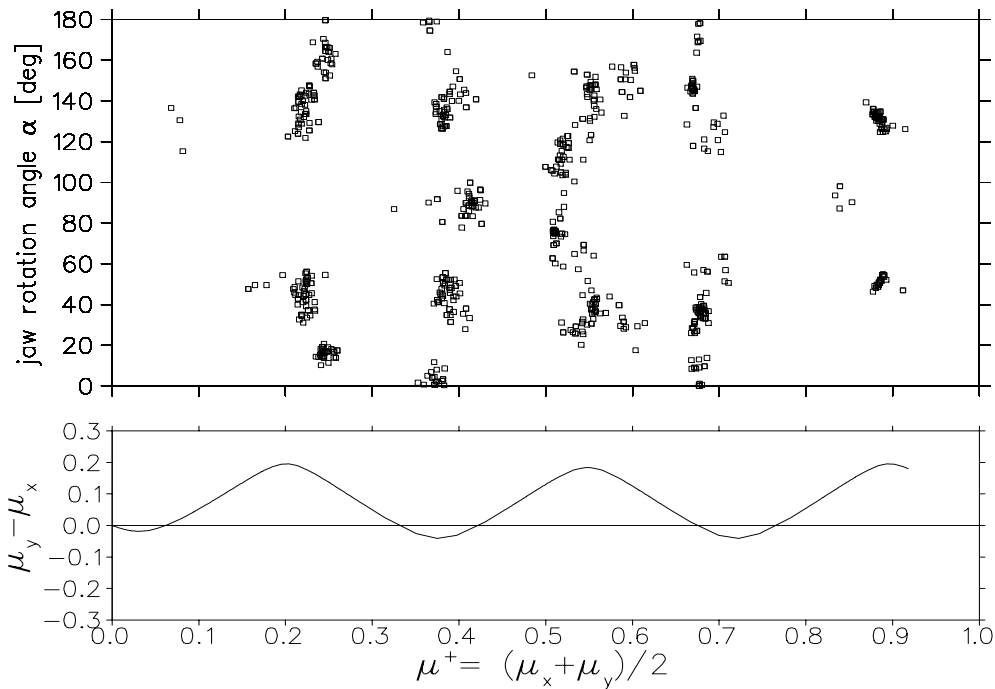


Figure 3: Collimator distributions from 50 SA runs for the lattice shown in Fig. 2 ($A_{max} = 8.45\sigma$) using 16 pairs of secondary jaws with random initial settings. Each point represents the rotation angle and the average betatronic phase μ^+ of one pair.

4 APPENDIX: EXAMPLE OF SIMULATED ANNEALING ALGORITHM IN ONE DIMENSION

In this example, the global minimum of $F(x)$ is searched for, within some interval (x_1, x_2) . The user supplies initial values for x ($x_1 < x < x_2$), the step dx and the temperature parameter T ($T > |x_1 - x_2|$).

At each iteration, 20 trial values x_{tri} are generated randomly in the interval $(x - dx, x + dx)$. If a trial value x_{tri} is downhill, i.e. $F(x_{tri}) < F(x)$, then it is accepted and, if $F(x_{tri})$ is lower than its previously lowest value, x_{tri} is recorded as a new optimum. An uphill x_{tri} can also be accepted with probability $P = e^{(F(x) - F(x_{tri}))/T}$ (Metropolis criterion). If the trial x_{tri} is out of bounds, then it is rejected and simply a new x_{tri} is generated in bounds. Each time the trial is accepted, x_{tri} replaces x (the centre x moves, but dx stays). At the end of the iteration, dx is scaled to some new length, which would have produced a roughly equal number of rejected and accepted trials; for instance, dx is increased if too many cases have been accepted.

The temperature is reduced by a factor 0.6 after each 5 iterations. The process is halted if the optimum found remains unchanged during several subsequent temperature cycles.

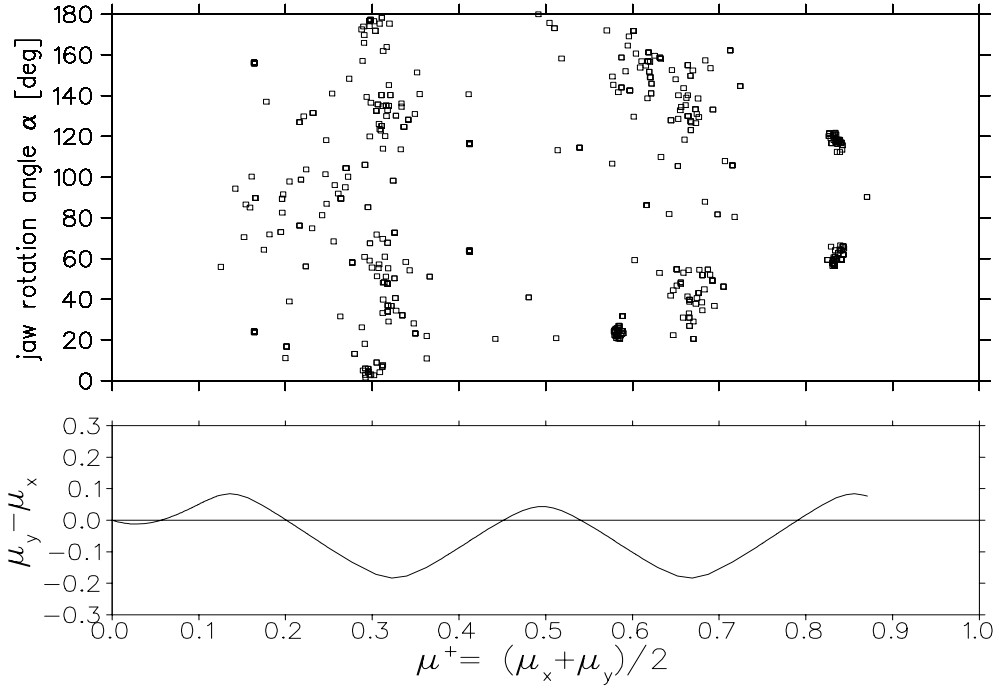


Figure 4: Collimator distributions for the lattice shown in Fig. 1 ($A_{max} = 9.1\sigma$).