Fast symplectic map tracking for the CERN Large Hadron Collider

Dan T. Abell*
Brookhaven National Laboratory, Upton, New York 11973, USA

Eric McIntosh† and Frank Schmidt‡
CERN, CH-1211 Geneva 23, Switzerland

(Received 12 July 2002; published 23 June 2003)

Tracking simulations remain the essential tool for evaluating how multipolar imperfections in ring magnets restrict the domain of stable phase-space motion. In the Large Hadron Collider (LHC) at CERN, particles circulate at the injection energy, when multipole errors are most significant, for more than 10^7 turns, but systematic tracking studies are limited to a small fraction of this total time—even on modern computers. A considerable speedup is expected by replacing element-by-element tracking with the use of a symplectified one-turn map. We have applied this method to the realistic LHC lattice, version 6, and report here our results for various map orders, with special emphasis on precision and speed.

DOI: 10.1103/PhysRevSTAB.6.064001 PACS numbers: 41.85.–p

I. INTRODUCTION

Since 1982, when maps were first introduced to the accelerator community [1], the question has been, "Can they be used to replace element-by-element tracking?" It became clear that in the presence of strong multipolar fields, maps can be used only if one applies some kind of symplectification scheme [2]. One such scheme was proposed by Irwin [3] in 1989 and later tested for the CERN Large Hadron Collider (LHC) [4]. Unfortunately, it was found to be (i) only a factor of 2 faster and (ii) insufficiently precise for determining the dynamic aperture. The application of this technique was therefore discontinued, until recently, when a significant improvement was proposed [5,6]. This improved scheme produces a symplectic map, called a Cremona map, that is guaranteed to agree with element-by-element tracking through a given Taylor order *with minimal additional terms.*

Here we apply Cremona map tracking to a realistic model of the LHC to see if this new technique can yield a sufficiently precise determination of the dynamic aperture, but with a tenfold gain in speed compared to direct tracking. To this end we study 60 different configurations (called "seeds" in the following) of randomly distributed multipole errors.

A sketch of the underlying theory, with references to more detailed literature, appears in Sec. II. Then, in Secs. III and IV, we describe the performance and optimization of the code CTRACK and the production of the required Cremona maps. The results obtained for a model of the current LHC optics (now under construction at CERN) are reported in Sec. V. We conclude the paper with a brief summary, Sec. VI.

II. THEORETICAL BACKGROUND

A. The symplectic condition

The study of dynamical systems leads naturally to the concept of transfer maps $\mathcal{M}$, which describe a system's change in phase space from initial conditions z^i to final conditions z^f during a given fixed interval of time (or the timelike variable): $\mathcal{M}z^i = z^f$. For Hamiltonian systems, transfer maps obey the very special *symplectic condition.* In a $2n$-dimensional phase space, with points $z = (z_1 \cdots z_{2n}) = (q_1 \cdots q_n, p_1 \cdots p_n)$, where q_i and p_i are the particle's positions and canonical momenta, this condition reads [1,7,8]

$$\tilde{M}JM = J. \tag{1}$$

Here M denotes the *Jacobian matrix*, which has entries $M_{ab} = \partial z_a^f / \partial z_b^i$; $\tilde{M}$ denotes the transpose of M; and J denotes the fundamental $2n \times 2n$ symplectic matrix $\left(\begin{smallmatrix} 0 & I \\ -I & 0 \end{smallmatrix}\right)$, where I is the $n \times n$ identity matrix. If throughout phase space $\mathcal{M}$ obeys Eq. (1), then $\mathcal{M}$ is called a *symplectic map.*

For most systems of interest we cannot write the transfer map in closed form; hence we usually resort to the use of approximations—often in the form of a Taylor map: each phase-space variable in the final state is expressed as a (truncated) Taylor series expanded in terms of the initial conditions. When applied to the Taylor series representation of a map, the symplectic condition Eq. (1) comprises a set of nonlinear relations between Taylor series coefficients of different orders. Hence the unavoidable truncation of the Taylor series (almost always) violates the symplectic condition. For some purposes this (one hopes small) violation will have little effect; but for others, e.g.,

*Present address: Tech-X Corp., 5541 Central Avenue, Suite 135, Boulder, CO 80301.
Electronic address: dabell@txcorp.com
†Electronic address: Eric.McIntosh@cern.ch
‡Electronic address: Frank.Schmidt@cern.ch

long-term tracking using many iterations of the map, the consequences can be severe. Moreover, numerical studies have shown that these undesirable consequences often do not derive from the loss of high-order information in the truncated Taylor map but from the symplectic violation itself [9].

B. The factorization theorem

Dragt and Finn have shown that one can cast the Taylor map derived from a Hamiltonian system into a form that maintains its symplectic nature. To do this they use the Poisson bracket $[\,,\,]$ to define the *Lie operator* $:f:$ associated with any function f:

$$:f:g = [f, g] = \sum_{i=1}^{n}\left(\frac{\partial f}{\partial q_i}\frac{\partial g}{\partial p_i} - \frac{\partial f}{\partial p_i}\frac{\partial g}{\partial q_i}\right) \tag{2}$$

for any function g. And the corresponding *Lie transformation* is the operator

$$e^{:f:} = \sum_{k=0}^{\infty}\frac{1}{k!}:f:^k = 1 + :f: + \frac{1}{k!}:f:^2 + \cdots. \tag{3}$$

(Here $:f:^2 g = [f,[f, g]]$, and similarly for higher powers.) These operators have the very special property that for *any* function f, the Lie transformation $e^{:f:}$ denotes a symplectic map [1,10,11].

The Dragt-Finn *factorization theorem* shows that for any symplectic map $\mathcal{M}$ which has a convergent Taylor series representation, one can use an order-by-order procedure to convert that representation to the form

$$\mathcal{M} = e^{:f_1:}e^{:f_2:}e^{:f_3:}e^{:f_4:}\cdots, \tag{4}$$

where each f_k denotes a homogeneous polynomial of degree k in the initial phase-space variables z^i [7,10]. The f_k are called the *Lie generators* of the map $\mathcal{M}$, and the Taylor series terms through order $k-1$ determine uniquely the Lie generators $f_1, \ldots, f_k$. Among the virtues of this representation are that one can relate the coefficients of the f_k to the various optical properties of the dynamical system; each factor corresponds to successively higher-order information about the dynamical system; and truncating the product does not violate the symplectic condition Eq. (1). Hence the factorization theorem gives us a method for symplectifying Taylor maps.

The one drawback of the representation Eq. (4) is the time required to compute its action. The first two factors pose no difficulty: $e^{:f_1:}$ generates the constant terms in the Taylor map and $e^{:f_2:}$ generates the linear terms (the matrix part). It is the remaining, nonlinear factors

$$\mathcal{N} = e^{:f_3:}e^{:f_4:}\cdots \tag{5}$$

that present a problem: each $e^{:f_k:}$ leads to Taylor map terms of degrees $k-1$ and higher. A straightforward use of Eq. (3) can be time-consuming to compute to machine precision, and truncating that power series again (usually) violates the symplectic condition. See [6] for a more complete discussion. One approach to addressing the difficulty just described uses the concepts of Cremona maps, the jolt representation, and jolt decomposition.

C. Cremona maps

A *Cremona map* is a map that is both symplectic and polynomial. The set of all such maps obviously includes all linear symplectic maps, but it also includes many nonlinear maps: Consider a Lie generator g that depends only on the coordinates q. Then $:g(q):^2 z_a = 0$ for all a, and

$$e^{:g(q):}z_a = z_a + :g:z_a = z_a + [g, z_a] \tag{6}$$

is a polynomial. Hence the Lie transformation of any polynomial in the coordinates alone is a Cremona map. As these affect only the momenta p, they are known as *kick maps*. Now let $\mathcal{L}$ denote any linear symplectic map. One can show that $:\mathcal{L}g(q):^2 z_a$ also vanishes for all a, and hence any map of the form $\exp(:\mathcal{L}g(q):)$ is also a Cremona map. These are called *jolt maps* [5,6].

D. The jolt representation

The jolt representation of a nonlinear symplectic map seeks to approximate the truncated version of Eq. (5),

$$\mathcal{N}_P = e^{:f_3:}e^{:f_4:}\cdots e^{:f_{P+1}:}, \tag{7}$$

by a product of N Cremona maps (here jolts),

$$\mathcal{J}_P = \exp(:\mathcal{L}_1 g^1(q):)\cdots\exp(:\mathcal{L}_N g^N(q):), \tag{8}$$

in such a way that the Taylor expansions of $\mathcal{N}_P$ and $\mathcal{J}_P$ agree through terms of order P. (The number of jolts N will, of course, depend on P.) If one starts from a Taylor map of order P and extracts $\mathcal{N}_P$ using the factorization theorem, then $\mathcal{J}_P$ constitutes an easy-to-evaluate symplectification of the nonlinear part of the original Taylor map.

The particular form of the polynomial generators g^j is described in Sec. II F. Here we simply note that they are *not* homogeneous: each contains terms of degrees 3 through $P+1$, and each factor in Eq. (8) is therefore a nonlinear polynomial (symplectic) map of order P. The polynomial expansion of Eq. (8) will, of course, contain many higher-order terms, terms that serve to symplectify the original Taylor map. Assuming Taylor map terms of degrees $P+1$ and higher do not contribute to the dynamics, then those high-order terms in the polynomial expansion of Eq. (8) ought to be small. Since those high-order terms involve products of various coefficients in the generators g^j, we ask that the coefficients of the g^j's be as small as possible.

Irwin first proposed the representation Eq. (8) in 1989 [3], using $\mathcal{L}_j$'s that perform (uncoupled) rotations in the

transverse planes, but numerical tests yielded poor results [12]. Later research has shown the results were better when selecting the $\mathcal{L}_j$'s from a broader class of linear symplectic maps. Moreover, that research showed that the quality of the representation Eq. (8) depends very sensitively on the choice of $\mathcal{L}_j$'s, and showed how to recognize and construct optimal sets of $\mathcal{L}_j$'s [5,6]. The $\mathcal{L}_j$'s so constructed are the ones used in the work reported here.

E. Jolt decomposition

1. Notation

We use a special normalization for monomials of degree ℓ in z:

$$G_r^{(\ell)} = \frac{q_1^{r_1} \cdots q_n^{r_n} p_1^{r_{n+1}} \cdots p_n^{r_{2n}}}{\sqrt{r_1! \cdots r_{2n}!}}, \tag{9}$$

where $r_1 + \cdots + r_{2n} = \ell$. On the space spanned by these monomials, the $USp(2n)$ *inner product* $\langle\ ,\ \rangle$ is defined to act according to the rule [5,7]

$$\langle \alpha G_r^{(\ell)}, \beta G_{r'}^{(\ell')} \rangle = \alpha^* \beta \delta_{\ell,\ell'} \delta_{r,r'} \tag{10}$$

for any scalars α and β. In addition to the *general* monomials $G_r^{(\ell)}$, we define the q monomials

$$Q_k^{(\ell)} = \frac{q_1^{k_1} \cdots q_n^{k_n}}{\sqrt{k_1! \cdots k_n!}}, \tag{11}$$

where $k_1 + \cdots + k_n = \ell$. In terms of these monomials, we define two further quantities:

(1) The *sensitivity vectors* σ^r with elements

$$\sigma_{jk}^r = \langle G_r^{(\ell)}, \mathcal{L}_j Q_k^{(\ell)} \rangle. \tag{12}$$

Here the superscript r labels the different vectors, and the subscript jk is treated as a single index labeling the vector elements.

(2) The *Gram matrix* $\Gamma(\ell)$ with elements

$$\Gamma(\ell)_{rs} = \sum_{j=1}^{N} \sum_{k=1}^{M(\ell,n)} \frac{w_j}{M(\ell,n)} \sigma_{jk}^r \sigma_{jk}^s. \tag{13}$$

Here

$$M(\ell,n) = \binom{\ell + n - 1}{\ell} \tag{14}$$

denotes the number of monomials of degree ℓ in n variables, and the w_j denote a set of N *weights* associated with the N linear symplectic maps $\mathcal{L}_j$. (Equally weighted $\mathcal{L}_j$'s of course have $w_j = 1/N$.)

2. Decomposition

The conversion of a symplectic map from the factored product representation Eq. (7) to the jolt representation Eq. (8) relies on the solution to the following problem: Given a homogeneous polynomial of degree ℓ in z,

$$h_\ell = \sum_{r=1}^{M(\ell,2n)} c_r^{(\ell)} G_r^{(\ell)}, \tag{15}$$

and a set of N linear symplectic maps $\mathcal{L}_1, \ldots, \mathcal{L}_N$, with associated weights $w_1, \ldots, w_N$, find the smallest possible *jolt coefficients* $a_{jk}^{(\ell)}$ such that

$$h_\ell = \sum_{j=1}^{N} \frac{w_j}{M(\ell,n)} \sum_{k=1}^{M(\ell,n)} a_{jk}^{(\ell)} \mathcal{L}_j Q_k^{(\ell)}. \tag{16a}$$

The solution, derived elsewhere [5,6], is

$$a_{jk}^{(\ell)} = \sum_{r,s} c_r^{(\ell)} [\Gamma(\ell)^{-1}]_{rs} \sigma_{jk}^s. \tag{16b}$$

The above way of writing h_ℓ is called a *jolt decomposition* because each term in Eq. (16a) can act as the Lie generator of a jolt map.

3. Special case

When one variable is a constant of the motion, the jolt decomposition can be done in 1 less degree of freedom. Suppose, for example, that p_n is the given constant of the motion; then its conjugate variable q_n will not appear in the Lie generators, and hence the polynomials h_ℓ have the form

$$h_\ell = \sum_{\lambda=0}^{\ell} \sum_{r=1}^{M[\ell-\lambda,2(n-1)]} c_r^{(\ell,\lambda)} G_r^{(\ell-\lambda)} p_n^\lambda.$$

Note that the general monomials $G_r^{(\ell-\lambda)}$ in this expression have the form Eq. (9), but in $n-1$ degrees of freedom. Consider the coefficient of p_n^λ:

$$h_{\ell,\lambda} \equiv \sum_{r=1}^{M[\ell-\lambda,2(n-1)]} c_r^{(\ell,\lambda)} G_r^{(\ell-\lambda)}.$$

A jolt decomposition of this polynomial will be a sum of terms of the form $\mathcal{L}_j Q_k^{(\ell-\lambda)}$ in $n-1$ degrees of freedom. Now observe that $(\mathcal{L}_j Q_k^{(\ell-\lambda)}) p_n^\lambda$ can also act as the Lie generator of a jolt map. It follows that a jolt decomposition of $h_{\ell,\lambda}$ yields directly a jolt decomposition of h_ℓ. Since jolt decomposition is *much* easier in fewer degrees of freedom, we gain a considerable computational savings whenever we can use this technique.

F. Conversion to Cremona map

The process for converting a map from the factored product representation Eq. (7) to the jolt representation Eq. (8) has been derived elsewhere [3,6]. Here we simply indicate the algorithm:

(1) For the lowest Lie order, 3, set $h_3 = f_3$, and jolt decompose h_3 using Eqs. (16a) and (16b):

$$h_3 = \sum_{jk} \frac{w_j}{M(3,n)} a_{jk}^{(3)} \mathcal{L}_j Q_k^{(3)} = \sum_j \mathcal{L}_j g^j(3).$$

Then write

$$J_2 = \exp(:\mathcal{L}_1 g^1(3):) \cdots \exp(:\mathcal{L}_N g^N(3):).$$

(2) Now suppose one has computed the jolt representation through Lie order m to obtain

$$J_{m-1} = \exp(:\mathcal{L}_1 g^1:) \cdots \exp(:\mathcal{L}_N g^N:).$$

(3) To go to the next order, expand J_{m-1} as a Taylor series and then refactor it as

$$J_{m-1} = \exp(:f_3:) \exp(:f_4:) \cdots \exp(:f_m:) \exp(:\tilde{f}_{m+1}:) \cdots.$$

(4) Set $h_{m+1} = f_{m+1} - \tilde{f}_{m+1}$, and jolt decompose h_{m+1} using Eqs. (16a) and (16b):

$$h_{m+1} = \sum_{jk} \frac{w_j}{M(m+1,n)} a_{jk}^{(m+1)} \mathcal{L}_j Q_k^{(m+1)} = \sum_j \mathcal{L}_j g^j(m+1).$$

(5) Then set

$$g^j = g^j(3) + \cdots + g^j(m) + g^j(m+1) = \sum_{\ell=3}^{m+1} \frac{w_j}{M(\ell,n)} \sum_{k=1}^{M(\ell,n)} a_{jk}^{(\ell)} Q_k^{(\ell)},$$

and write

$$J_m = \exp(:\mathcal{L}_1 g^1:) \cdots \exp(:\mathcal{L}_N g^N:).$$

(6) Steps (3)–(5) can be repeated for successively higher orders until one obtains J_P.

G. Numerical evaluation

For rapid evaluation of the Cremona map J_P, we make use of a slight rewriting of Eq. (8): A very useful, indeed defining, fact about the Lie operators given in Eq. (2) is that for *any* symplectic map $\mathcal{A}$

$$:\mathcal{A} f: = \mathcal{A}:f:\mathcal{A}^{-1}.$$

We may therefore rewrite Eq. (8) in the form

$$J_P = \mathcal{L}_1 \exp(:g^1:) \mathcal{L}_1^{-1} \mathcal{L}_2 \exp(:g^2:) \mathcal{L}_2^{-1} \cdots \mathcal{L}_N \exp(:g^N:) \mathcal{L}_N^{-1} = \mathcal{A}_0 \exp(:g^1:) \mathcal{A}_1 \exp(:g^2:) \mathcal{A}_2 \cdots \exp(:g^N:) \mathcal{A}_N, \tag{17}$$

where the $\mathcal{A}_0 = \mathcal{L}_1$, $\mathcal{A}_1 = \mathcal{L}_1^{-1}\mathcal{L}_2$, $\mathcal{A}_2 = \mathcal{L}_2^{-1}\mathcal{L}_3, \ldots, \mathcal{A}_N = \mathcal{L}_N^{-1}$ all denote linear symplectic transformations. This rewriting of Eq. (8) says we may compute the action of J_P as a sequence of matrices and kicks. Moreover, evaluating a given kick, $e^{:g^j:}$, requires, according to Eqs. (2) and (6), simply calculating the associated gradient of g^j with respect to its variables, and this can be computed as the product of a matrix of coefficients (determined and stored in advance) with a corresponding vector of monomials. Fast numerical evaluation of the Cremona map J_P therefore requires efficient algorithms for just two essential routines: (1) multiplying a vector by a matrix, and (2) computing a vector of monomials (in two or three variables for our case and through the Taylor order of interest).

Before leaving this section, we note that the procedure described in Sec. II F is not actually applied to the map $\mathcal{N}_P$ of Eq. (7) but to a linearly normalized version of $\mathcal{N}_P$. In other words, the similarity transformation used to bring the linear part of $\mathcal{M}$ to normal form is applied also to $\mathcal{N}_P$, thus putting the coordinates and momenta on an equal footing. The linear maps thus "removed" from $\mathcal{N}_P$—along with the $\exp(:f_1:)\exp(:f_2:)$ from Eq. (4)—are then folded in with the maps $\mathcal{A}_0$ and $\mathcal{A}_N$ of Eq. (17) to build our Cremona approximation to the complete map $\mathcal{M}$.

III. CREMONA MAP PRODUCTION

The LHC optics database, version 6, comprises a design (ideal) lattice together with 60 different seeds representing the range of imperfections expected to exist in the real machine. To build Cremona maps for each of these rings, we first used the available SIXTRACK [13] tools to construct tenth-order Taylor maps representing one turn around each of these 60 rings. Then the code CREMONA was used to construct the corresponding Cremona maps of (Taylor) orders 2, 4, 6, 8, and 10. In the six-dimensional case (with the rf cavity on) the map constructed carried particles around the ring from just after the rf cavity to just before the cavity. The rf cavity was then treated as an exact kick. The virtue of this approach was that the "one-turn" map did not change the momentum deviation $\delta = (p - p_o)/p_o$, and we could therefore use the technique described in Sec. II E 3 to greatly simplify the Cremona symplectification.

IV. CTRACK PERFORMANCE AND OPTIMIZATION

The implementation of Cremona map tracking requires three principal tools: (1) a means for constructing the Taylor map for a given machine lattice; (2) a routine for

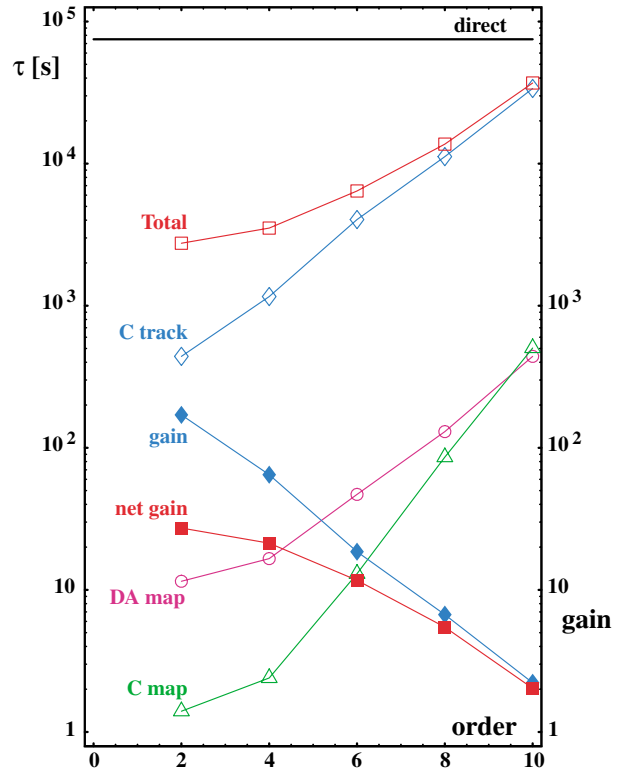

FIG. 1. (Color) This plot shows execution times (hollow symbols, left-hand ordinate) and speed gains (solid symbols, right-hand ordinate) versus map order for different aspects of Cremona map tracking as compared to direct tracking. Execution times are shown for creating a differential algebra map (DA map), converting it to a Cremona map (C map), doing the actual tracking (C track), and the sum of these times plus the CTRACK overhead (Total). The gains shown are just the ratios of the times required for direct tracking (shown by the horizontal line at top) and Cremona map tracking—either "C track" (gain), or "Total" (net gain).

converting that Taylor map into a corresponding Cremona map; and (3) a routine for tracking with the resulting Cremona map. Tool 1 was already available in SIXTRACK's suite of tools; tool 2 was implemented in the program CREMONA, which makes extensive use of Forest's LIELIB package [14]; and tool 3 was implemented in the program CTRACK. Because tracking consumes so much more time than map production (see Fig. 1), only the second of the two new tools, CTRACK, was subjected to intensive optimization, as described in this section.

Code development and optimization were initially carried out on a Digital Equipment Corporation (DEC) computer with an Alpha EV5 processor. Performance and portability tests were then run on an IBM Power PC, and on a Linux Pentium III PC with both the GNU and Portland Group compilers. The intention was to find the best overall source code, hopefully the same for each platform and compiler, and not to compare systems with significantly different price and performance. Once the best compiler switches had been chosen, an execution-time analysis was performed. The results for a standard test case, ten particles for 50 000 turns in 6D mode, are shown in Table I. Based on our observations at the end of Sec. II G, it should come as no surprise that CTRACK spends almost all its time ($\sim 90\,\%$ or more) in the subroutines computing the dot product of a matrix with a vector and evaluating the required vector of monomials in the routines named `MdotV` and `evalMonoms`, respectively; see Table I.

In code where a loop has a fixed, and not too large, number of iterations, it sometimes pays to "unroll" the loop. This technique involves writing explicitly each iteration of the loop, thus eliminating the overhead. For example, the (trivial) code

```
do i=1,2
a(i)=b(i)+c(i)
enddo
```

would become simply

```
a(1)=b(1)+c(1)
a(2)=b(2)+c(2)
```

eliminating the time spent initializing, incrementing, and testing the loop index. Even on systems with vector or pipelined architecture, this can still be worthwhile, especially for small numbers of loop iterations.

Since the matrices and vectors have fixed dimensions, typically 6 or smaller, all the loops in `MdotV` and `evalMonoms` were unrolled in the new routines mydotu and `myMonoms`. In a further version of `MdotV` for longer vectors, called `mydotv`, the innermost loop was not unrolled in order to make use of pipelining.

Finally, the new, C-optimized, subroutines were translated to FORTRAN. (The bulk of the code remained in C.) The results in Table II show a further improvement in

TABLE I. Percentage of time spent in the two principal CTRACK subroutines.

System C compiler	DEC cc	IBM PPC xlc	Intel PIII pgcc	Intel PIII gcc
`MdotV` (%)	53.5	79.3	48.0	63.6
`evalMonoms` (%)	43.0	10.2	45.3	33.0
Total (%)	96.5	89.5	93.3	96.6

TABLE II. Comparison of timing results (in seconds), by version and platform.

System FORTRAN/C compiler	DEC f77/cc	IBM PPC xlf/xlc	Intel PIII pgf77/pgcc	Intel PIII g77/gcc
C	234	593	180	152
C, optimized	168	464	112	111
FORTRAN	130	422	121	137

TABLE III. Percentage of time spent in selected FORTRAN-optimized subroutines.

System FORTRAN/C compiler	DEC f77/cc	IBM PPC xlf/xlc	Intel PIII pgf77/pgcc	Intel PIII g77/gcc
`mydotv` (%)	67.0	74.7	62.7	74.0
`mydotu` (%)	7.6	5.8	8.2	11.0
`myMonoms` (%)	16.5	5.6	8.7	10.9

performance on the proprietary systems but not on the Intel PC.

Speedup factors of between 1.4 and 1.8 were obtained.

As confirmed by Table III the most time-consuming procedure is now the matrix by vector product, mydotv, with three rows and 165 columns in the standard test case (6D, Taylor order 8, 32 jolts/turn, for 5×10^5 turns). The procedure is called 16 000 000 times to perform 3×165 multiplies and adds, using about 67 s on the DEC system, giving a respectable 236 mega-floating point operations per second on a 350MHz processor.

Figure 1 shows our final measure of performance: a comparison of the total computer time used in tracking with Cremona maps (as optimized above) against the time required by direct tracking. To make realistic tracking speed comparisons between direct and map tracking, one must include the time required to produce the necessary maps; but this time becomes relatively less important as one tracks for more and more turns. Since our typical dynamic aperture study for one LHC lattice involved tracking particles at four different amplitudes and five initial angles, each 6×10^6 times around the ring—a total of 120×10^6 turns—we used this total amount of tracking as our basis for comparison.

The principal results of our timing study are given in Fig. 1 by the curves labeled *gain* and *net gain*. The curve *gain* compares pure tracking speed, showing gains of about 170 at order 2, 19 at order 6, and 2.2 at order 10. The curve *net gain*, which includes the cost of map creation and the overhead of invoking CTRACK, shows a gains of about 27 at order 2, 12 at order 6, and 2.0 at order 10.

The overhead cost of invoking CTRACK was not optimized—indeed CTRACK was invoked for each of the twenty particles tracked—and it constitutes the principal difference between *gain* and *net gain*. One could easily improve this behavior by tracking particles simultaneously.

TABLE IV. LHC parameters.

Length (m)	26 658.883
Injection energy (GeV)	450 (studied here)
Collision energy (GeV)	7000 (not studied)
Triplet errors	Not studied
Luminosity ($\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$)	1.0×10^{34}
Tune (Q_x/Q_y)	64.28/59.31
Low-order systematic resonances	Avoided by choice of tunes
Maximum β in the arc (m)	183
Natural chromaticity	-89
Chromaticity caused by b_3 errors	600
Dipole and quadrupole multipole components (systematic & random)	(b_3–b_{11}) and (a_3–a_{11})
Linear imperfections	Not considered
Correction system for dipole errors	b_3 and b_5
Nonlinear detuning with correction at $8\,\sigma$	5×10^{-3}

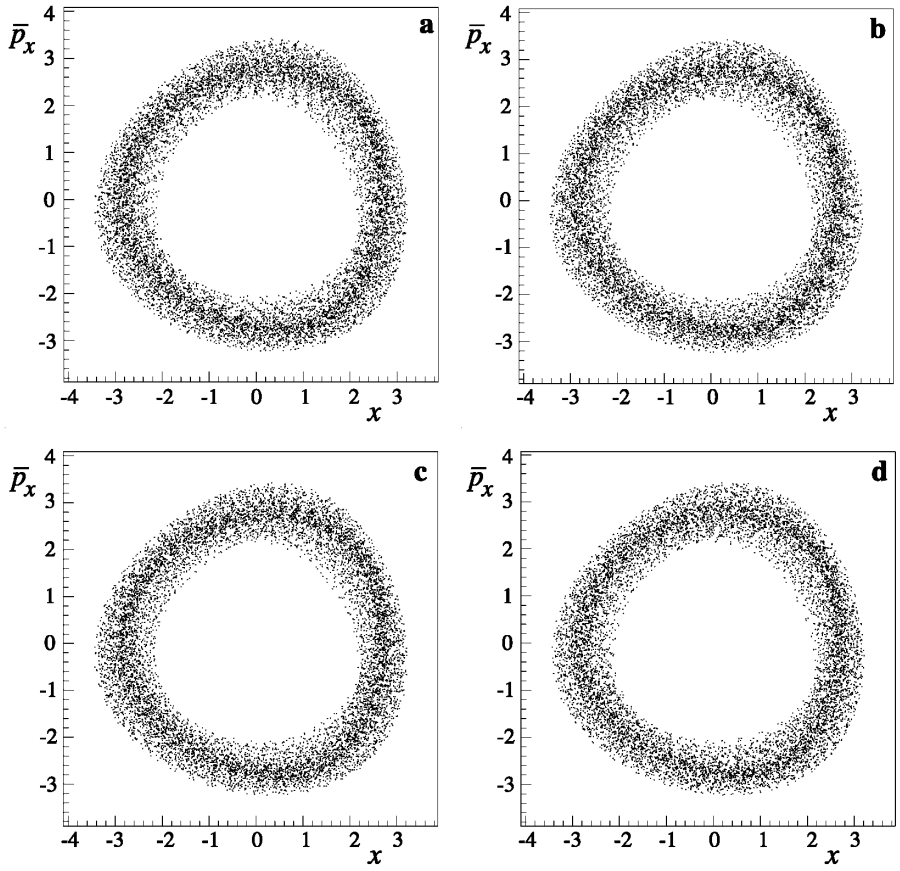

FIG. 2. Horizontal phase-space portraits generated at amplitude 10σ for the LHC optics, version 6, seed 39. The portraits shown derive from direct tracking (a), and Cremona map tracking at orders 10 (b), 8 (c), and 6 (d).

V. RESULTS AND ANALYSIS FOR THE LHC OPTICS, VERSION 6

The LHC is being constructed in the tunnel of the former Large Electron-Positron Collider and will accelerate protons (bidirectional) to 7 TeV using superconducting 2-in-1 dipole and quadrupole magnets for bending and focusing purposes, respectively. The LHC parameters relevant for these studies are shown in Table IV.

To assess the utility of Cremona maps for long-term tracking studies of the LHC, one might compare phase-space portraits generated by direct (element-by-element) and by Cremona map tracking. Figure 2 shows just such a comparison of horizontal phase-space portraits at a very large amplitude (10σ) for several different map orders. (In what follows the order of a map will always mean the corresponding *Taylor* order.) This figure shows only very subtle differences in the phase-space portraits and suggests that for our present purpose we must use more detailed and more quantitative measures to assess the utility of Cremona maps.

A. Quantitative measures

For more quantitative comparisons between Cremona map tracking and direct tracking, we looked at the following measures: (1) differences in the amplitude-dependent tune, or detuning error; (2) differences in the predicted dynamic aperture (DA); (3) one-turn tracking errors; and (4) multiturn phase errors. We describe each of these in turn.

From particle tracking data one can determine the tunes of a given particle. Figure 3 shows the results of just such an analysis on particles launched at five different angles (in x-y space) and at ten different amplitudes (from 1 to 10σ), and followed using direct tracking around one particular LHC lattice (seed 39). Using symbols, we say, for example, that each point of the left-hand plot in Fig. 3 represents a value for the horizontal tune $Q_h(A, \theta, r, \mathcal{X})$ of a particle launched with amplitude A and angle θ, and tracked around the LHC lattice, seed r, using direct tracking, indicated here by the "exact" map $\mathcal{X}$. Figure 4 shows how the results based on Cremona map tracking differ from those based on direct tracking. In, say, the upper plot of that figure, each point represents an average—over the 60 different seeds r and the five different angles θ—of the absolute error

$$|Q_h(A, \theta, r, C_n) - Q_h(A, \theta, r, \mathcal{X})|,$$

where C_n denotes the Cremona map of (Taylor) order n.

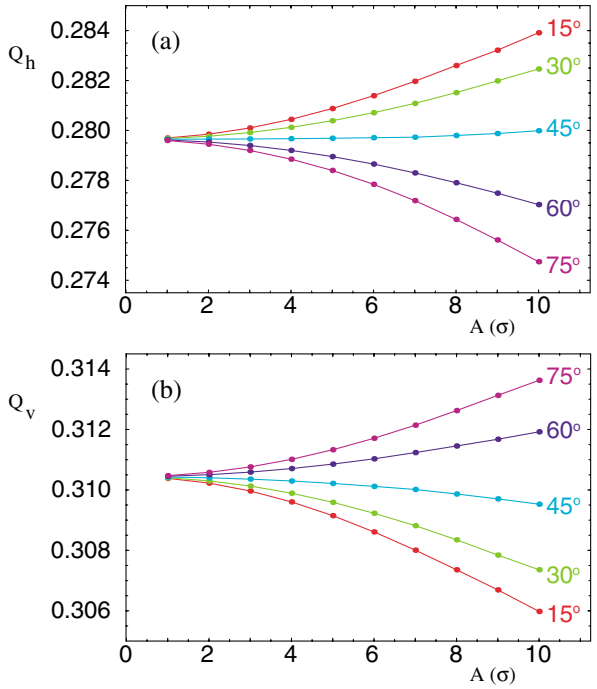

FIG. 3. (Color) Example (seed 39) of horizontal and vertical detuning versus amplitude for five initial angles. The upper and lower plots correspond to the horizontal and vertical planes, respectively.

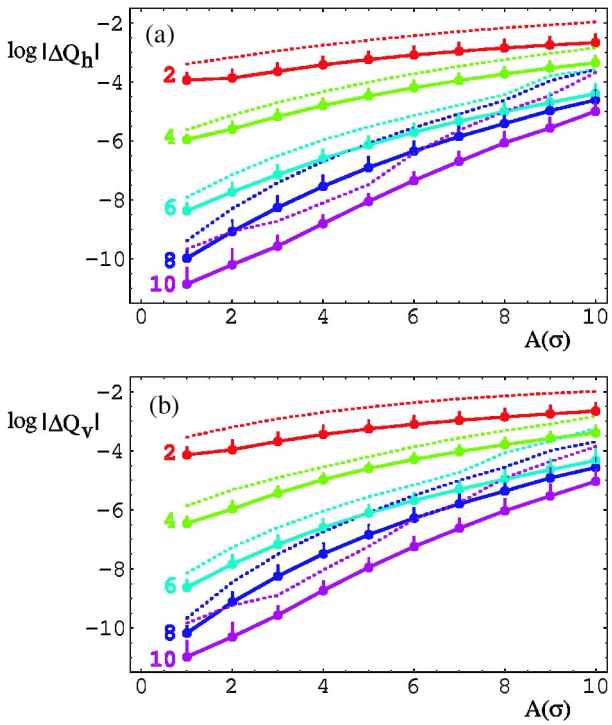

FIG. 4. (Color) Tune error ($\log_{10}$) versus amplitude, parametrized by Cremona map order. The upper and lower plots correspond to the horizontal and vertical planes. The solid and dashed lines show, respectively, the average and maximum (over 60 seeds and five angles) of the absolute difference in tunes measured from tracking results based on direct and Cremona map tracking.

The lower plot shows corresponding results for the vertical plane. The error bars on the points indicate 1 standard deviation above the mean values, and the dashed lines indicate the maximum values.

We define the DA for a given lattice as the largest amplitude below which all simulated particles remain in the accelerator for the duration of the simulation [15]. Figure 5 compares DAs computed using direct tracking with those computed using Cremona map tracking at orders 4, 6, 8, and 10. Each plot shows, for a given order and a given angle (in the x-y plane), the DAs computed by direct tracking for each of the 60 realistic LHC lattices (red curves), and the corresponding DAs computed by Cremona map tracking (blue curves). The green hatching serves simply to highlight the differences between the two curves.

The two measures described so far—detuning errors and differences in predicted DA—might be called "high-level" measures: they report how well Cremona map tracking performs the tasks we want it to do. By contrast, one might describe the next two measures as "low level": they report how well Cremona map tracking performs on a turn-by-turn basis.

Recall, for a moment, the origin of our Cremona maps: from a tracking code, corresponding to some exact map $\mathcal{X}$, we extract a truncated Taylor series map $\mathcal{T}_m$ containing terms through order m, so that $\mathcal{T}_m = \mathcal{X} + O(z^{m+1})$. This Taylor map is then symplectified by converting it to a Cremona map $\mathcal{C}_m$ that agrees with the Taylor map through the same order m; hence $\mathcal{C}_m = \mathcal{T}_m + O(z^{m+1})$. It follows that the three maps $\mathcal{X}$, $\mathcal{T}_m$, and $\mathcal{C}_m$ all differ from one another by terms that scale as z^{m+1}; but we would like to know just how big those differences are. To that end we define the *one-turn errors*

$$\varepsilon_m^{TX}(z) = \|\mathcal{T}_m z - \mathcal{X} z\|, \tag{18a}$$

$$\varepsilon_m^{CX}(z) = \|\mathcal{C}_m z - \mathcal{X} z\|, \tag{18b}$$

$$\varepsilon_m^{CT}(z) = \|\mathcal{C}_m z - \mathcal{T}_m z\|, \tag{18c}$$

where $\|\cdot\|$ denotes an appropriate vector norm on phase space. To determine in some meaningful way how the average errors vary with amplitude, we require at each amplitude a set of points over which to average. To generate a given set of points, we launch a single particle at the given amplitude (between 3 and 12 σ, where σ is defined in terms of the local lattice function values); and we use direct tracking to follow it for 1000 turns, thus generating the phase-space points

$$\{z^0, z^1, z^2, \ldots, z^{1000}\} = \{z^0, \mathcal{X} z^0, \mathcal{X} z^1, \ldots, \mathcal{X} z^{999}\}.$$

The first 1000 of these, 0–999, constitute the desired set of points. Applying also the maps $\mathcal{T}_m$ and $\mathcal{C}_m$ to these points, we can compute the one-turn errors defined in

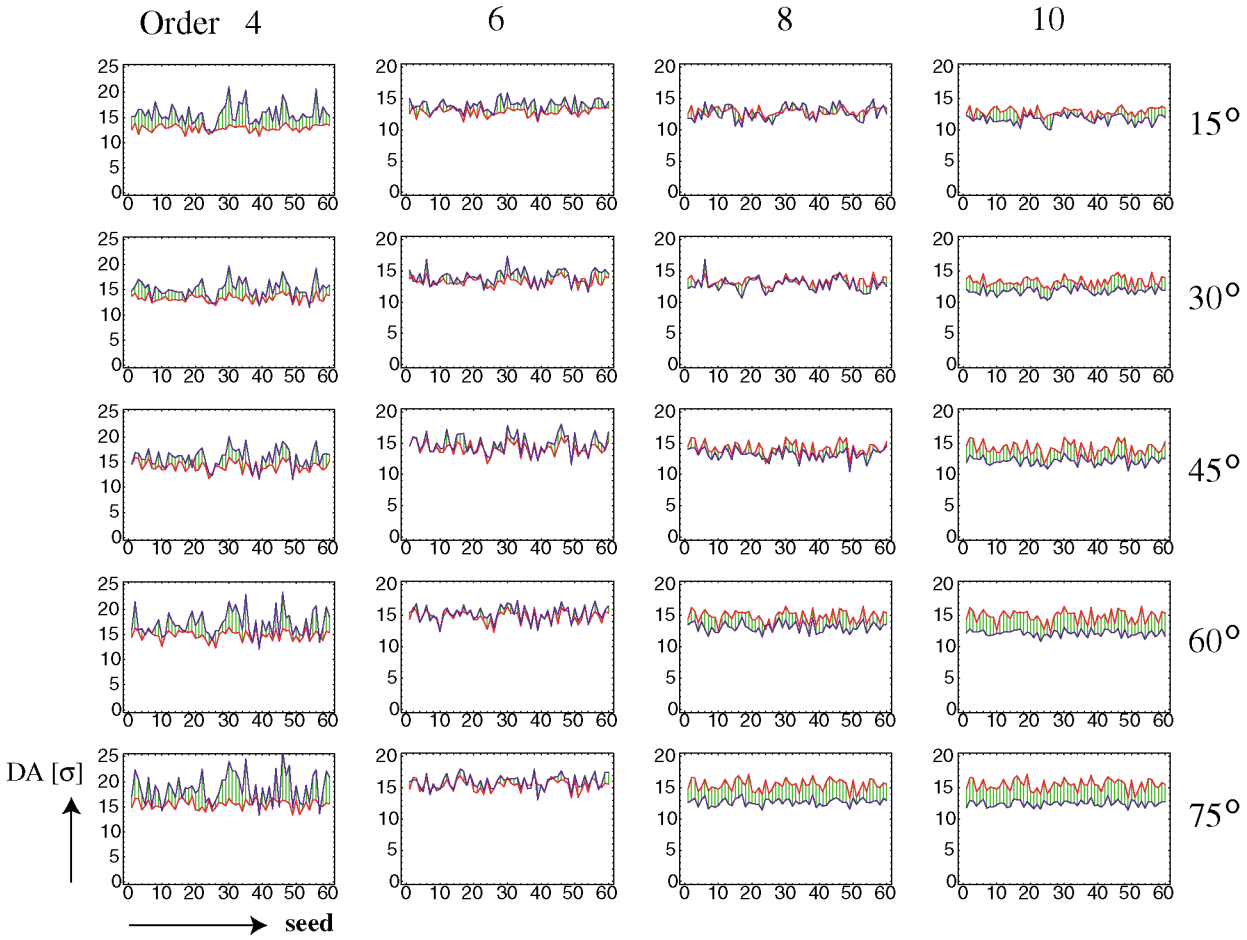

FIG. 5. (Color) Dynamic apertures (DA) computed from the results of direct and Cremona map tracking (orders 4, 6, 8, and 10) for the realistic LHC lattice, version 6. In each plot we show the DA for each of 60 different seeds. The red curves show the results obtained from direct tracking, and the blue curves show those obtained from Cremona map tracking. The comparisons are shown for five different angles. (Note that the vertical scales for order 4 differ slightly from those of the other orders.)

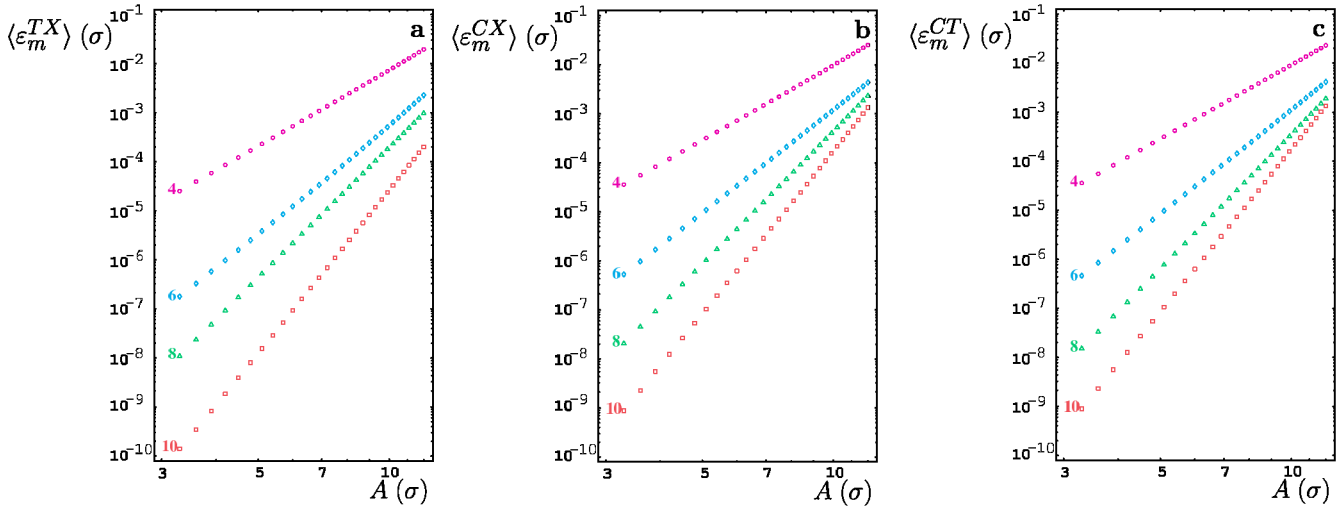

FIG. 6. (Color) Average one-turn error $\langle \varepsilon_m^{\alpha} \rangle$ versus amplitude, parametrized by map order m (indicated by 10, 8, 6, and 4 in the figure), for a relative momentum deviation $\delta = 0$. The different plots show results averaged over an orbit of 1000 turns for the one-turn error between (a) direct and Taylor map tracking, TX, (b) direct and Cremona map tracking, CX, and (c) Taylor map and Cremona map tracking, CT.

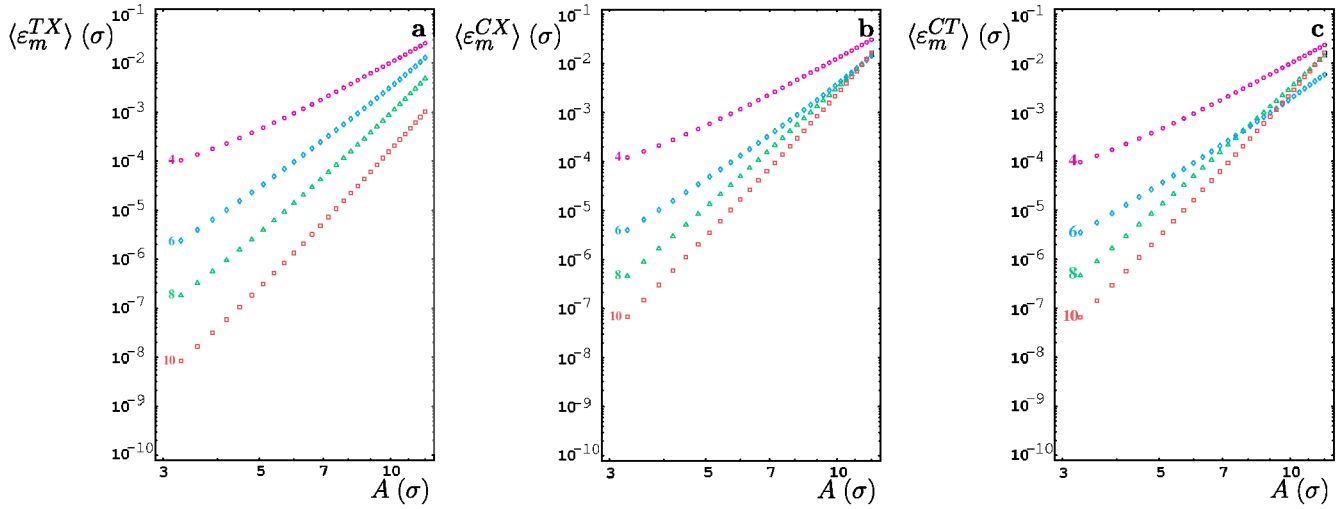

FIG. 7. (Color) Average one-turn error $\langle \varepsilon_m^{\alpha} \rangle$ versus amplitude, parametrized by map order m (indicated by 10, 8, 6, and 4 in the figure), for a comparatively large relative momentum deviation of $\delta = 7.5 \times 10^{-4}$. The different plots show results averaged over an orbit of 1000 turns for the one-turn error between (a) direct and Taylor map tracking, TX, (b) direct and Cremona map tracking, CX, and (c) Taylor map and Cremona map tracking, CT.

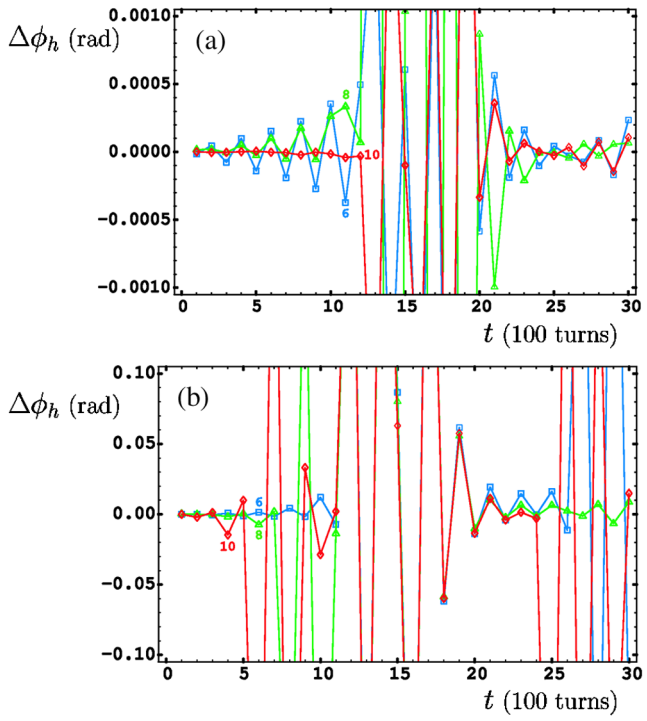

FIG. 8. (Color) Horizontal phase error $\Delta\phi_h$ versus turn number, in steps of 100 turns, parametrized by Cremona map order (6, 8, and 10), for a comparatively large relative momentum deviation of $\delta = 7.5 \times 10^{-4}$. The upper and lower plots show results obtained at amplitudes of 8σ and 12σ, respectively. That the phase error is largest for the *low*-order maps at 8σ but for the *high*-order maps at 12σ is consistent with the one-turn error results. The large intermediate excursions are due to the nonlinear phase-space topology.

Eq. (18), and average over the 1000 points. For the vector norm used in Eq. (18) we take the Pythagorean length in the transverse linearly normalized phase space, and we show the results for relative momentum deviations $\delta = 0$ in Fig. 6 and $\delta = 7.5 \times 10^{-4}$ in Fig. 7.

Our last quantitative measure for comparing direct and Cremona map tracking is the horizontal *phase error* $\Delta\phi_h$. It measures the phase difference in (normalized) horizontal phase space between a particle tracked element by element and the same particle tracked using a Cremona map. Figure 8 shows the evolution of $\Delta\phi_h$ for map orders 6, 8, and 10 at two different amplitudes, 8σ and 12σ.

B. Analysis

The detuning error, shown in Fig. 4, behaves more or less as expected: the error grows with amplitude, and higher-order maps have significantly smaller detuning errors. Note, however, that the improvement seen in the detuning error as one goes to higher order becomes much smaller at the larger amplitudes. In particular, going from order 6 to order 8 shrinks the average error by more than a decade at amplitude 1σ but by less than a factor of 2 at amplitude 10σ.

The predicted DA, however, does not show the same consistent improvement as one goes to higher order: As seen in Fig. 5, Cremona map tracking at order 6 agrees very satisfactorily with direct tracking. But at higher orders Cremona map tracking tends to underestimate the DA—and by as much as several sigma for the larger angles. This fact, puzzling at first sight, becomes more understandable in light of the one-turn errors shown in Figs. 6 and 7.

Figure 6 shows that the average one-turn error has the correct scaling with amplitude: Consider, for example, ε_m^{TX} of Eq. (18a). The mth order Taylor map $\mathcal{T}_m$ makes errors of order z^{m+1}, hence $\varepsilon_m^{TX}(z) \sim O(z^{m+1})$, and similarly for the other one-turn errors, ε_m^{CX} and ε_m^{CT}. And in the log-log plots of Fig. 6 the errors labeled order m do indeed lie on lines of slope roughly $m + 1$.

The three plots in Fig. 6 show that the differences between the three maps—Cremona, Taylor, and exact—are all comparable, but closer inspection reveals more. A comparison of plots 6(a) and 6(b) shows that the error made by the Cremona map, ε_m^{CX}, is roughly 2 to 3 times that made by the Taylor map, ε_m^{TX}. And a comparison of plots 6(b) and 6(c) shows that ε_m^{CX} roughly equals the difference between the Cremona and the Taylor maps, ε_m^{CT}. In other words, the difference between the Cremona and the exact maps is due mostly to the process of symplectification. Observe also that the power-law scaling of the various orders implies that at some amplitude higher-order map tracking must lead to larger errors than lower-order map tracking.

The plots in Fig. 7 correspond to those in Fig. 6, except that here all the particles have a relatively large momentum deviation of $\delta = 7.5 \times 10^{-4}$, which masks the $O(z^{m+1})$ scaling at amplitudes below about 8 or 9σ. One may here make comparisons similar to those made above for Fig. 6. But it suffices to note [see plot 6(b)] that at this nonzero value of δ, relevant for the tracking studies, the amplitude at which higher-order tracking begins to give poorer results is less than 12σ.

The results shown in Fig. 8 reinforce the observations made above for Fig. 7: At some amplitude between 8σ and 12σ tracking at orders higher than 6 no longer produces smaller errors.

We conclude that the difficulty higher-order Cremona map tracking has (see Fig. 5) in predicting the dynamic aperture results from the small-amplitude loss of precision introduced by the symplectification process coupled with the steep power-law scaling of the one-turn errors. Ironically, one can expect better results when applying Cremona map tracking to a machine with larger errors and, hence, a smaller dynamic aperture. Indeed, in preliminary studies for an earlier version of the LHC lattice, we successfully used Cremona map orders higher than 6.

VI. SUMMARY

By replacing direct tracking of the LHC with Cremona map tracking at order 6 (which includes magnetic multipole components up through order 7), one can obtain a net gain in speed greater than ten; and one can predict DAs that agree very well with those of direct tracking. Moreover, for the amplitudes of interest, Cremona map tracking produces errors that are smaller at order 6 than at higher orders. We conclude that for the current version of the LHC, where multipole components beyond order 7 are less relevant, Cremona map tracking at order 6 is an attractive alternative for doing rapid systematic investigations.

To the potential client at other future accelerators, we recommend examining the one-turn errors ε_m^{α}, which can be done rapidly. From plots such as those in Fig. 7 one can determine whether or not Cremona map tracking will be useful.

ACKNOWLEDGMENTS

One of us (D. T. A.) would like to thank the AP group of the CERN SL division for their generous hospitality and support.

[1] A. J. Dragt, in *Physics of High-Energy Particle Accelerators*, edited by M. M. R. A. Carrigan and F. R. Huson, AIP Conf. Proc. No. 87 (AIP, New York, 1982), pp. 147–313.
[2] R. Kleiss, F. Schmidt, Y. T. Yan, and F. Zimmerman, CERN Technical Report No. SL/92-02(AP), DESY HERA 92-01, SSCL Report No. SSCL-564, 1992.
[3] J. Irwin, in *Accelerator Physics at the Superconducting Super Collider*, edited by Y. T. Yan, J. P. Naples, and M. J. Syphers, AIP Conf. Proc. No. 326 (AIP, New York, 1995), pp. 662–669, originally written in 1989 as Report No. SSC-228.
[4] R. Kleiss, F. Schmidt, and F. Zimmerman, Part. Accel. **41**, 117 (1993).
[5] D. T. Abell, Ph.D. thesis, University of Maryland, 1995.
[6] A. J. Dragt and D. T. Abell, in *Integration Algorithms and Classical Mechanics*, edited by J. E. Marsden, G. W. Patrick, and W. F. Shadwick, Fields Institute Communications Vol. 10 (American Mathematical Society, Providence, RI, 1996), pp. 59–85.
[7] A. J. Dragt, University of Maryland Physics Department Report, 1995.
[8] H. Goldstein, *Classical Mechanics* (Addison-Wesley, Reading, MA, 1980), 2nd ed.
[9] A. J. Dragt, I. M. Gjaja, and G. Rangarajan, in *Proceedings of the 1991 IEEE Particle Accelerator Conference* (IEEE, Piscataway, NJ, 1991), Vol. 3, pp. 1621–1623.
[10] A. J. Dragt and J. M. Finn, J. Math. Phys. (N.Y.) **17**, 2215 (1976).
[11] E. Forest, *Beam Dynamics: A New Attitude and Framework*, The Physics and Technology of Particle and Photon Beams Vol. 8 (Harwood Academic, Amsterdam, 1998).
[12] E. Forest (private communication).
[13] F. Schmidt, CERN Technical Report No. SL/94-56(AP), 1994.
[14] E. Forest, M. Berz, and J. Irwin, Part. Accel. **24**, 91 (1989).
[15] M. Böge and F. Schmidt, in *Beam Stability and Nonlinear Dynamics*, edited by Z. Parsa, AIP Conf. Proc. No. 405 (AIP, New York, 1996), pp. 201–210.