Correct Tracking in FFAGs

J. Scott Berg

Brookhaven National Laboratory; Building 901A; P. O. Box 5000, Upton, NY 11973-5000

Abstract. Fixed field alternating gradient accelerators have many features which require careful modeling in simulation. They accept beams over an extremely large momentum range, generally at least a factor of 2. They often use magnets whose lengths are comparable to their apertures. The beam often makes large angles with respect to the magnet axis and pole face normal. In some applications (muons in particular), the beam occupies a substantial fraction of the magnet aperture. The longitudinal dynamics in these machines often differ significantly from what one finds in more conventional machines such as synchrotrons. These characteristics require that simulation codes be careful to avoid inappropriate approximations in describing particle motion in FFAGs. One must properly treat the coordinate system geometry independently from the magnetic fields. One cannot blindly assume that phase space variables are small. One must take magnet end fields properly into account. Finally, one must carefully consider what it means to have a "matched" distribution that is injected into these machines.

Keywords: Fixed Field Alternating Gradient Accelerator; Simulation; Tracking
PACS: 29.27.Bd

INTRODUCTION

Accelerator tracking codes do not generally lay out magnetic fields in a global coordinate system and integrate particles exactly in that coordinate system. They generally work locally about a reference curve, and assume that deviations from that reference curve will be relatively small. This helps make the layout of beamlines more straightforward, improves efficiency, and in some cases improves accuracy. Many of the assumptions that allow one to do this with a high degree of accuracy for more conventional machines are slightly or significantly less accurate for FFAGs.

This paper outlines some features that should be treated properly in tracking codes that will be used for FFAGs. The features that are emphasized are those which are often not included in conventional tracking codes. The purpose of discussing them here is to aid an FFAG designer in evaluating existing tracking codes for their application, and to point out features that they should consider including in tracking and analysis codes that they write. There are few, if any, new results here: this paper should be seen as a review of existing knowledge, some of which may not be generally well known.

TRUNCATED POWER SERIES

A truncated power series (TPS) of order n is a function of the d-dimensional vector $\boldsymbol{z}$ of the form

$$\sum_{\substack{j_i \geqslant 0 \\ j_1+\cdots+j_d \leqslant n}} a_{\boldsymbol{j}} z_1^{j_1} \cdots z_n^{j_n}. \tag{1}$$

Much analysis in accelerator physics is done using TPSs. For instance, a set of first order TPS is used to represent a linear map, which then gives the Courant-Snyder functions, the dispersion, and the momentum compaction. Chromaticities, tune shifts with amplitude, and resonance driving terms are calculated from higher-order TPSs.

The TPSs used in accelerator physics are generally a function of deviations of phase space variables from reference values. Rapid convergence of a TPS as its order increases relies on these deviations being small relative to some characteristic values. If one ignores coupling, these characteristic values are the magnet aperture (for transverse dimensions), the RF frequency (for time), and a "reference" momentum (for transverse momentum and energy divided by the speed of light c). The "reference" momentum for these purposes is some momentum within the designed operating range of the machine. In most synchrotrons, for example, it is an excellent approximation to assume that the deviations are small: typically the largest variables relative to their characteristic values are the transverse positions, generally being a couple percent of the pipe aperture.

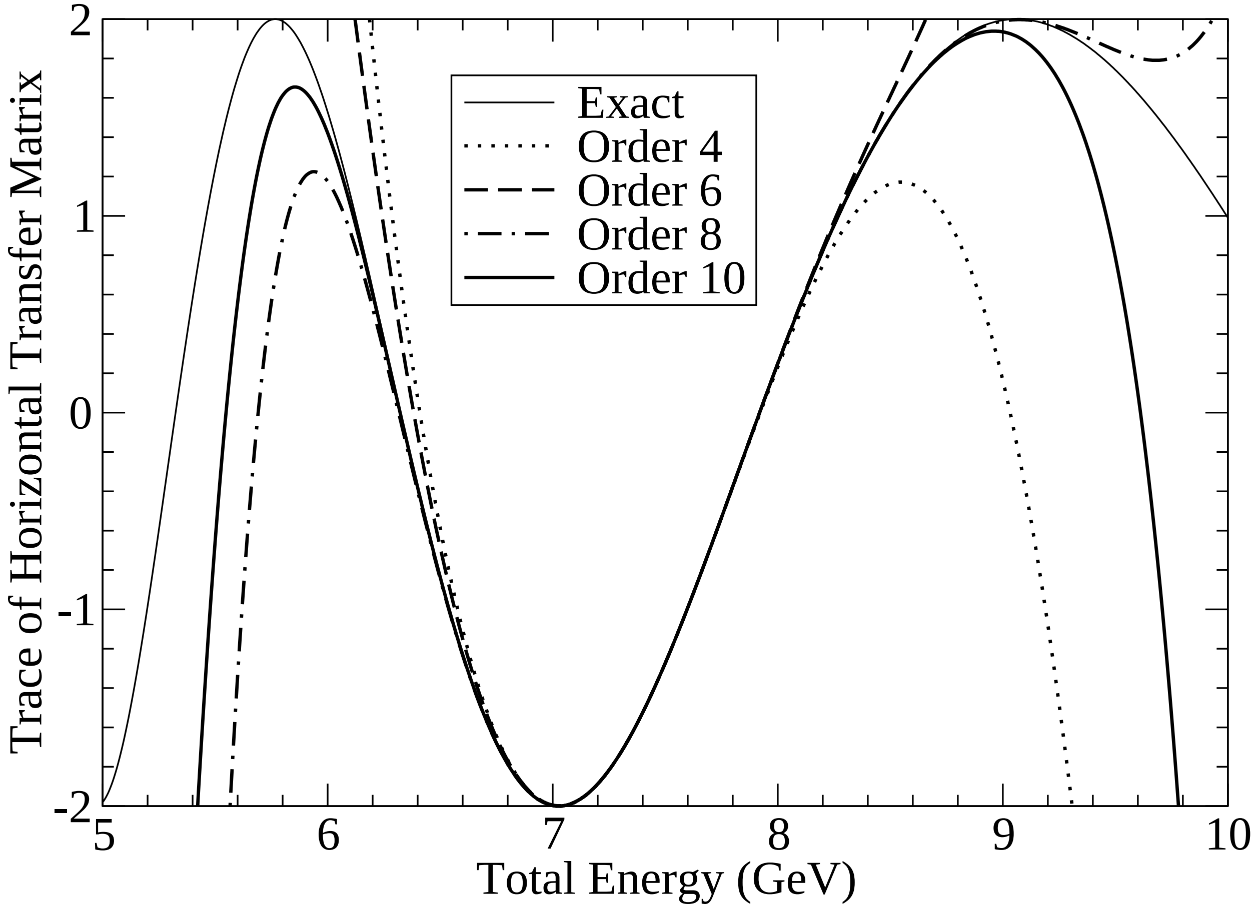

FIGURE 1. Trace of the horizontal transfer matrix for 10 cells as a function of energy for a non-scaling FFAG, computed using COSY INFINITY for several different map orders.

FFAGs clearly do not have small phase space variables in one particular area: energy deviation. Since FFAGs generally accelerate by a factor of 2 or more in momentum, the energy deviation relative to the central energy is usually at least 33%, if not more. Thus, one needs to be very careful about using a TPS with energy deviation as one of the variables. Furthermore, smaller FFAG rings can have large angles relative to magnet axes, both because of the small number of cells and the fact that FFAG cells generally bend both forward and backward over some part of their energy range. Also, since the closed orbit deviation in an FFAG generally moves over a relatively large range in the magnet, often the fraction of the magnet aperture (at least horizontally) occupied by the beam can be very large.

There are some codes, such as COSY INFINITY [1, 2, 3] and MARYLIE [4], which perform their analysis by constructing a single power series to represent the particle motion through a section of beam line. If used blindly, these types of codes will probably have difficulties representing the behavior in FFAGs, especially because of the large energy range. For example, Fig. 1 shows that even computing the tunes over the entire working range of an FFAG can be problematic for some lattices: even a 10th order computation fails to converge over the operating range of the FFAG. However, if the computation is done for a single cell instead of 10 cells, the computation converges to the correct answer. One can construct examples that fail to converge over the working range even for a single lattice cell. Off-energy tracking tends to be especially problematic when representing the transfer map using a single power series. Figure 2 shows strongly non-symplectic behavior in the tracking (note the particles converging toward the origin). Even if the tracking were “symplectified,” the results would still be highly inaccurate.

Truncated power series should not be abandoned entirely. They should instead be used carefully and appropriately. On can, for instance, find a closed orbit at a given energy without using TPS methods (symplectic integration, for example), and then compute a linear map about that closed orbit. This will give one energy-dependent tunes and Courant-Snyder functions which will not suffer from errors related to the nonzero energy deviation.

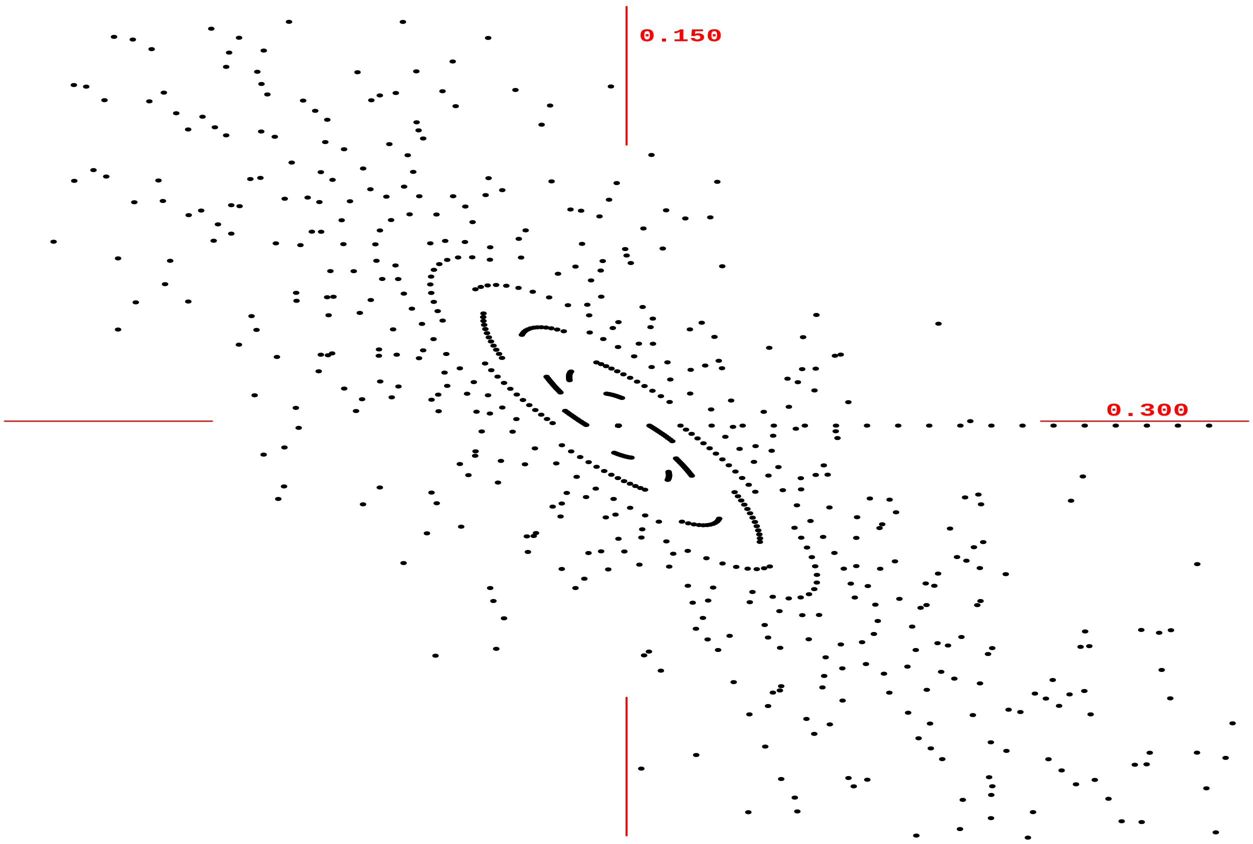

FIGURE 2. Tracking using COSY INFINITY for a single FFAG lattice cell at the minimum working energy for the lattice.

Magnetic fields are often well represented by TPSs. The fact that the magnetic fields can be well represented by TPSs while the resulting map cannot comes from the truncation of higher order terms in map composition: the composition of two nth order TPSs results in a n^2th order TPS, and the truncation to nth order results in significant information loss. This is also why a single cell will be more well represented than a group of cells or an entire ring: the single cell map has had less truncation occur on it than the group of cells. One could thus imagine several levels of TPS use

1. Doing direct integration where magnetic fields are represented using TPSs.
2. Constructing TPS maps for individual elements.
3. Constructing TPS maps for sections of beamline (cells, for example).

I have demonstrated that at least in some cases, the third approach is problematic for FFAGs (Fig. 2). The first method is almost certainly workable, and the second method may have some utility (it is used in some cases in MAD [5, 6].

GEOMETRY

Most standard accelerator analysis and tracking codes define their coordinate system by the fields in the magnets. Thus, the curvature h of the coordinate system is related to the midplane vertical magnetic field B_y by $h = qB_y/p_0$, where q is the "reference particle" charge and p_0 is a "reference momentum."

For FFAGs, however, it is important to separate the coordinate system from the magnetic field. The geometry should be that dictated by the geometry of the magnet [7, 8, 9]. Doing this is one of the express goals of the PTC routines [10]. For example, consider a scaling FFAG, where the field in the midplane is of the form $B_y(\theta)r^k$. Figure 3 shows the closed orbits, which are clearly not at a constant radius from the center of the machine. Using a coordinate system about one of these closed orbits would add unnecessary complexity to computing the magnetic fields. The

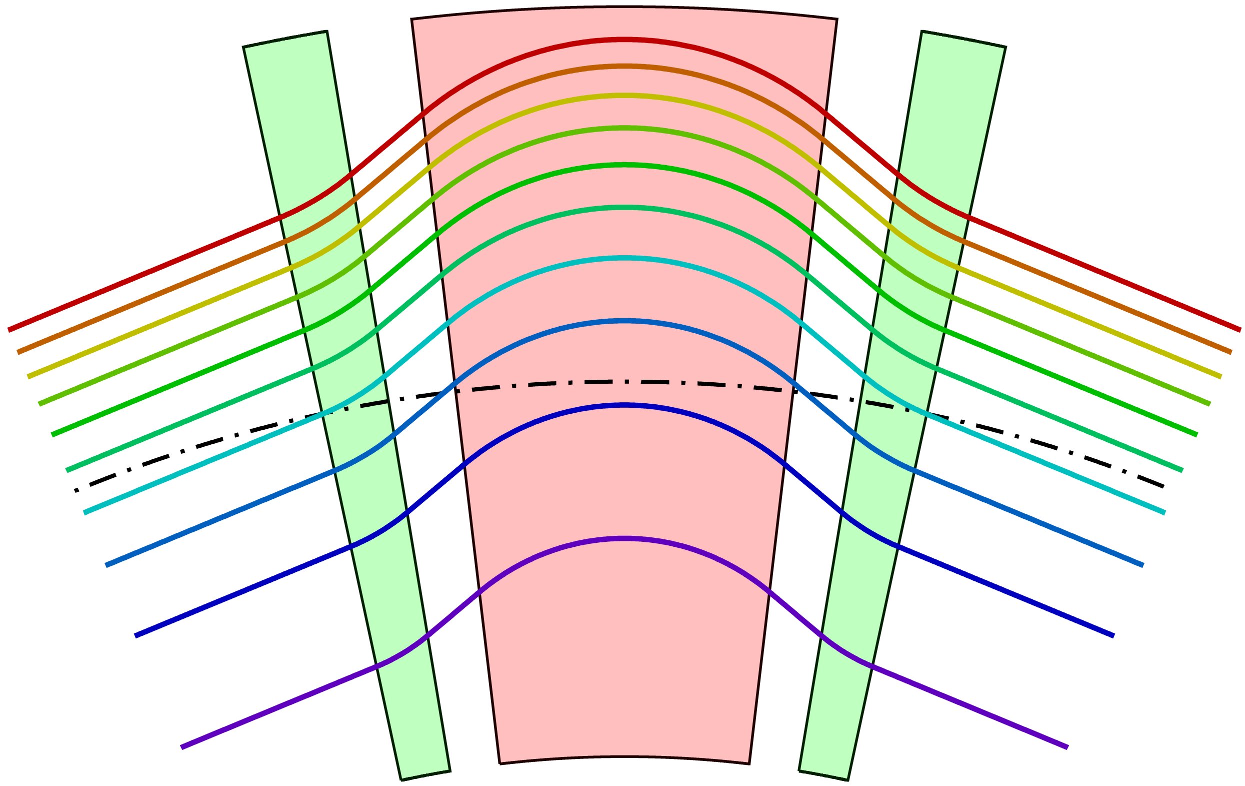

FIGURE 3. Closed orbits at different energies in a scaling FFAG, drawn over the magnets, with a dot-dash line showing a circular arc with a constant distance to the machine center.

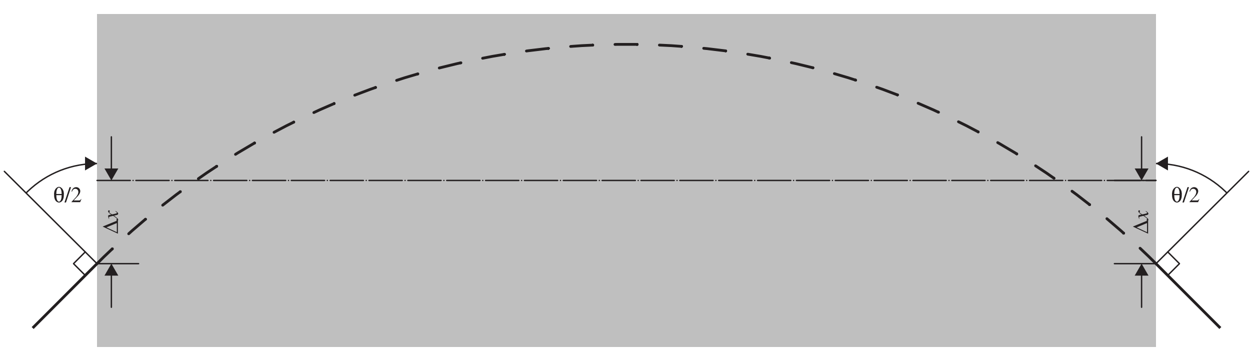

FIGURE 4. Orbit offset in a gradient rectangular bend.

closed orbits don't even make arcs of circles, making a description of the coordinate system difficult. Instead, using a cylindrical coordinate system is obviously straightforward and the optimal approach.

Another example is the gradient rectangular bend, a gradient magnet where the lines of constant field are straight lines. These are commonly used in non-scaling FFAGs. MAD [5, 6] and other codes often represent these as a "sector bend," a magnet where the lines of constant field are arcs of circles, with the end pole faces rotated to be parallel to each other. Due to the short length and large aperture of the magnets, one would generally not use a true sector bend in a non-scaling FFAG; a rectangular bend is more appropriate. The sector bend with rotated pole faces allows one the convenience of defining a coordinate system within the magnet based on the arc of a circle. However, in the true rectangular bend, there is no orbit which is the arc of a circle. To approximate the relationship between bend field, bend angle, and length that one finds in most tracking codes, one can set up the magnet so that if the orbit did bend in an arc of a circle with a radius of curvature defined by the bend field, then the integral of the gradient part of the field would be zero. The offset Δx (see Fig. 4) to accomplish this is

$$L\left(\frac{\theta}{8}\csc^2\frac{\theta}{2}-\frac{1}{4}\cot\frac{\theta}{2}\right), \tag{2}$$

where L is the length of the magnet along the axis and θ is the total bend angle of the arc[1]. Notice that this is independent of the gradient in the magnet, and allows one to construct an approximate relationship between the bend angle, the magnet length, and the dipole field in a gradient rectangular bend.

Implicit in the above is that there is no orbit whose transverse phase space coordinates are zero everywhere. Even if one had only rectangular gradient bends with the offset (2), there would still in general be no energy with a closed orbit whose coordinates, even in drifts, were zero (although it would be close if one used (2)). This is a necessary consequence of separating the geometry from the magnetic field.

When one goes from a model with no end fields to one with true end fields, the coordinate system and layout of the magnets should not be changing by that modification alone. Similarly, displacing or mispowering magnets for the study of errors should not change the underlying coordinate system. Yet if one does not maintain an independence between the coordinate system and the fields in the magnets, these will end up being tied together, and this will complicate the analysis of the effects in question.

It is important to separate time of flight from geometry as well for FFAGs. Often codes will define all times relative to the aforementioned reference particle, and RF cavities will be synchronized to that reference particle. In the design of an FFAG for muon machines, one may design a machine based on the behavior of a particle at the central energy, but may want the RF cavities to be synchronized to the closed orbit of a particle with a different energy [11, 12]. One may in fact want to vary the energy at which the particle is synchronized to without changing the lattice design (this would involve slight changes in the RF frequency). The most straightforward way to handle this is to use the total time rather than the time relative to a reference particle (ZGOUBI [13, 14] and ICOOL [15] do this). If one chooses to use time relative to some fixed-velocity particle attached to the coordinate system (not necessarily an actual particle trajectory), one must either be able to set the velocity of that particle independently of a "reference momentum" that may have been used to define a coordinate system and/or fields (and to set that velocity greater than the speed of light if needed), or to specify a phase that advances by a fixed amount on each turn.

MAGNET END FIELDS

In accelerator analysis, we often imagine magnets as having a field that doesn't change longitudinally within the magnet, and abruptly goes to zero outside the magnet. In a real magnet, the field changes gradually from its intended value within the magnet to zero outside the magnet. Maxwell's equations will cause higher-order fields to appear as a result of this field variation. The combination of the non-square field profile and the "Maxwellian" fields it generates are known as "end fields" or "edge fields."

Since FFAGs tend to have magnets whose lengths are short compared to their aperture, the relative importance of these end fields tends to be larger in FFAGs than in conventional accelerators. Thus, tracking codes which model FFAGs must have some method for handling the end fields. There are three possible ways of doing this:

1. Field maps
2. Generating fields using an end profile and Maxwell's equations
3. Hard-edge approximations

Field maps are the best way to get a realistic field from a known magnet. However, they suffer from some practical difficulties:

- Obtaining a field map requires having a magnet design, which one is unlikely to have at the machine design stage.
- A field map may require keeping a large amount of data
- The interpolation method may introduce noise, particularly for linear interpolation, which is the most straightforward way of handling field map data.
- The interpolation method may not satisfy Maxwell's equations.

---

[1] This is probably the only new result in this paper, and is arguably trivial.

- To integrate symplecticity, one needs vector potentials, whereas most field maps will instead contain field data.
- One cannot easily manipulate the field data to do a parametric study of the magnet parameters.

One may instead specify the longitudinal profile of the desired component of the field (e.g., the dipole and quadrupole) and use Maxwell's equations to compute the higher order components (this is done in COSY INFINITY [1, 2, 3], ZGOUBI [13, 14], and ICOOL [15]). This gives a smoothness related to the smoothness of the representation of the longitudinal profile of the desired field component, which in principle can avoid the noise introduced in linear interpolation of field maps. One can straightforwardly vary the end field profile and see its effects. The vector potentials can be used directly to get symplectic tracking. Computing the higher order components is an order-by-order iterative process, however, so the computation must be stopped at some order, which will lead to fields that don't satisfy Maxwell's equations perfectly. Furthermore, this iterative computation can be slow.

Computing the fields from Maxwell's equations also requires one to specify a magnet symmetry; this is equivalent to choosing constants of integration in solving Maxwell's equations. Take a quadrupole as an example. One could say that $B_y(x,0,z) = B_2(z)x$ in the midplane for a $B_2(z)$ that one specifies. Or instead, one could specify that to lowest order in x and y, $B_y(x,y,z) = B_2(z)x$, $B_x(x,y,z) = B_2(z)y$, and A_z has only terms with $\cos 2\theta$ symmetry. These two are not equivalent; which one is correct depends on the design of the magnet. For example, to maintain the scaling condition in a scaling FFAG, one will try to design the magnets to maintain the r^k field profile in the midplane; thus, choosing the constants of integration to specify the field in the midplane is probably the correct representation. Many methods of designing superconducting magnets, on the other hand, try to create fields with given multipole components, and thus one should probably specify those rather than the midplane field profile for such magnets.

At the earliest design stages, one does not even necessarily know what the magnet apertures will be, making the choice of an end field profile difficult. Thus, in cases where one does not know much about the magnet design, or where one wants to do rapid computations that take into account the magnet end fields, one can use so called hard-edge approximations [16, 17]. In these approximations, one integrates in the body of the magnet as if the field did not vary longitudinally, and applies a single transformations at the entrance and exit of the magnet. These transformations can be made correct to first order in the body field in the limit as the length of the varying magnet end field goes to zero [17]. One has to apply the same considerations of symmetry to these hard-edge end fields as for the end fields arising from a specified longitudinal field profile.

Most accelerator physicists are familiar with one kind of hard-edge approximation: the linear transformation associated with dipole pole faces that are not perpendicular to the reference orbit. Some codes treat this as the following transformation:

$$\Delta p_y = \mp q B_0 y \tan\phi, \tag{3}$$

where B_0 is the dipole field, q is the particle charge, and ϕ is the angle that the perpendicular to the pole face makes with respect to the reference orbit. If one assumes that the midplane field profile in the body of the magnet is $B_{y0}(x)$ and it remains proportional to that as the field goes to zero, one finds that the transformation at the edge (to lowest order) arises from the Lie generator [17, 18]

$$\frac{q y^2 p_x}{2\sqrt{p^2 - p_x^2 - p_y^2}} \Delta B_{y0}(x), \tag{4}$$

where the coordinate x is with respect to the magnet edge, not the reference orbit. Not only does this generator induce the transformation (3), but it also induces a (nonlinear) transformation in all the phase space variables except energy. Furthermore, for the transformation in p_y, it modifies the sense of (3): one should use B_y at the point where the particle exits the magnet, and one should use the angle that the actual particle (not the reference particle) makes with respect to the magnet pole face.

DISTRIBUTION MATCHING

When one is simulating a machine, one should in general start with a distribution that is matched (or nearly matched) to the Courant-Snyder functions of the lattice at the injection point, since that is what should occur in the real machine. Similarly, one should match the dispersion at the injection point as well, since FFAG lattices generally don't have dispersion-free sections. None of this is very surprising. However, in some FFAGs, one must also be aware of longitudinal matching. This may be straightforward in a low frequency FFAG where the frequency is matched to the revolution frequency of the beam's current energy. However, for high-frequency systems, such as some muon FFAGs, one must be aware of two aspects of longitudinal matching.

First of all, since there is dispersion in most of the machine, there will be a corresponding transformation in the time of flight: if the horizontal position and momentum (x,p) are related to coordinates about the closed orbit $(\bar{x},\bar{p})$ by

$$x=\bar{x}+x_0(E) \qquad\qquad p=\bar{p}+p_0(E) \tag{5}$$

then the time of flight is related to an uncoupled time of flight $\bar{\tau}$ by

$$\tau=\bar{\tau}+\frac{dp_0}{dE}\bar{x}-\frac{dx_0}{dE}\bar{p}. \tag{6}$$

If one has a large transverse emittance and high frequency RF, ignoring this correlation could lead to undesirable emittance growth due to mismatch.

Secondly, in fixed, high-frequency machines, finding the optimal longitudinal ellipse shape is actually a nonlinear problem [11, 12]. One should take this into account when choosing the distribution to track through the machine. One must of course also consider how one would create that distribution in the real machine. Ignoring the optimal distribution, or making an educated "guess" at the distribution, can give performance much worse than the capability of the machine.

CONCLUSION

This paper has reviewed some features that should be considered when writing or reviewing tracking and analysis codes for FFAGs. It has described why those features are of importance for correct tracking and analysis in FFAGs. These features should at least be given consideration when writing or evaluating code for use in FFAGs.

ACKNOWLEDGMENTS

This work has been supported by the United States Department of Energy, Contract No. DE-AC02-98CH10886.

REFERENCES

1. K. Makino, and M. Berz, *Nucl. Instrum. Methods A* **427**, 338–343 (1999).
2. M. Berz, and K. Makino, COSY INFINITY version 8.1 user's guide and reference manual, Tech. Rep. MSUCL-1195, National Superconducting Cyclotron Laboratory, East Lansing, MI (2002), URL `http://cosy.pa.msu.edu/cosymanu/index.html`.
3. M. Berz, , J. Hoefkens, and K. Makino, COSY INFINITY version 8.1 programming manual, Tech. Rep. MSUCL-1196, National Superconducting Cyclotron Laboratory, East Lansing, MI (2002), URL `http://cosy.pa.msu.edu/cosymanu/index.html`.
4. A. J. Dragt, R. D. Ryne, D. R. Douglas, F. Neri, C. T. Mottershead, E. Forest, L. M. Healy, P. Schütt, and J. van Zeijts, MARYLIE 3.0 users' manual, Tech. rep., Department of Physics, University of Maryland, College Park, MD (2003).
5. H. Grote, and F. C. Iselin, The mad program (methodical accelerator design) version 8.19, Tech. Rep. CERN/SL/90-13 (AP) (Rev. 5), European Organization for Nuclear Research (CERN), Geneva, Switzerland (1996).
6. H. Grote, and F. Schmidt, "MAD-X – an Upgrade from MAD8," in *Proceedings of the 2003 Particle Accelerator Conference*, edited by J. Chew, P. Lucas, and S. Webber, IEEE, Piscataway, NJ, 2003, pp. 3497–3499, URL `http://mad.home.cern.ch/mad/`.
7. É. Forest, and K. Hirata, A contemporary guide to beam dynamics, Tech. Rep. KEK Report 92-12, National Laboratory for High Energy Physics (KEK), Tsukuba, Japan (1992).
8. É. Forest, "The Absolute Bare Minimum for Tracking in Small Ring (Minus Radiation)," in *Advanced Beam Dynamics Workshop on Effects of Errors in Accelerators, Their Diagnosis and Corrections*, edited by A. Chao, AIP Conference Proceedings 255, American Institute of Physics, New York NY, 1992, pp. 263–275.
9. É. Forest, M. F. Feusch, D. L. Bruhwiler, and A. Amiry, *Part. Accel.* **45**, 65–94 (1994).
10. É. Forest, F. Schmidt, and E. McIntosh, Introduction to the polymorphic tracking code, Tech. Rep. CERN-SL-2002-044 (AP) and KEK-Report 2002-3, European Organization for Nuclear Research (CERN) and High Energy Accelerator Research Organization (KEK), Geneva, Switzerland and Tsukuba, Japan (2002).
11. J. S. Berg, "Longitudinal Acceptance in Linear Non-Scaling FFAGs," in [19], pp. 1532–1534.

12. J. S. Berg, Minimizing longitudinal distortion in a nearly isochronous linear non-scaling ffag (2005), URL `http://www-mucool.fnal.gov/notes/notes.html`, submitted to Phys. Rev. ST Accel. Beams., `BNL-74966-2005-JA`, `MUC-PUB-ACCELERATION-326`.
13. F. Méot, and S. Valero, ZGOUBI users' guide, version 4.3, Tech. Rep. CEA DSM DAPNIA-02-395 (2002), CEA Saclay, Gif sur Yvette, France (2003).
14. F. Lemuet, and F. Méot, *Nucl. Instrum. Methods* **A547**, 638–651 (2005).
15. R. C. Fernow, "Recent Developments on the Muon-Facility Design-Code ICOOL," in [19], pp. 2651–2653, URL `http://pubweb.bnl.gov/people/fernow/icool/readme.html`.
16. É. Forest, and J. Milutinović, *Nucl. Instrum. Methods* **A269**, 474–482 (1988).
17. J. S. Berg, "Higher Order Hard Edge End Field Effects," in *Proceedings of EPAC 2004, Lucerne, Switzerland*, EPAC, European Physical Society Accelerator Group, 2004, pp. 2236–2238.
18. A. J. Dragt, and J. M. Finn, *J. Math. Phys.* **17**, 2215–2227 (1976).
19. C. Horak, editor, *Proceedings of 2005 Particle Accelerator Conference, Knoxville, Tennessee*, IEEE, Piscataway, NJ, 2005.