Magnetic Field Maps for the RFOFO Muon Cooling Ring

Steve Bracker
s_bracker@hotmail.com

University of Mississippi

March 27, 2003

MuCool Note #271

Generating magnetic field maps for the RFOFO Muon cooling ring has proven challenging. The problem is divided into several parts: (1) The ***primary field generator*** which starts with the coil geometry and computes **B**(**X**) for an arbitrary point in the muon-accessible volume of the ring from basic physical principles, (2) A ***field map builder*** which uses the primary field generator to compute the field on a lattice of points, stored on disk for later use, and (3) a ***secondary field generator*** which loads the field-on-a-lattice into memory and interpolates in the field map to deliver **B**(**X**) to simulation programs as quickly as possible. Extensive tests must be performed to validate the results at every stage; these validations took far more time to program and run than did the field generators being checked. The current release of the field map and the software to access it is believed to be correct to about a part in ten thousand.

Contents:

General Statement of the Problem

The RFOFO Cooling ring [1,2,3] is a design (actually an ensemble of closely related designs) for an apparatus which receives intense pulses of muons from an upstream generator, reduces the emittance of the muon ensemble by absorption cooling and concurrent reacceleration, and emits the cooled muon ensemble into downstream consumers. While traveling around the cooling ring, the muon trajectories are steered by vertical dipole-like fields which provide the "bend" and solenoidal fields that provide focusing and confinement. In the present design, both fields are generated by the same coils -- short large-aperture solenoids which are slightly tilted about a horizontal axis to provide the bending field.

The RFOFO ring is divided into twelve identical 30-degree cells. Each cell has two solenoids symmetrically placed around the cell center. The two coils of each cell are tilted in opposite directions, and the current flows through them in opposite directions.

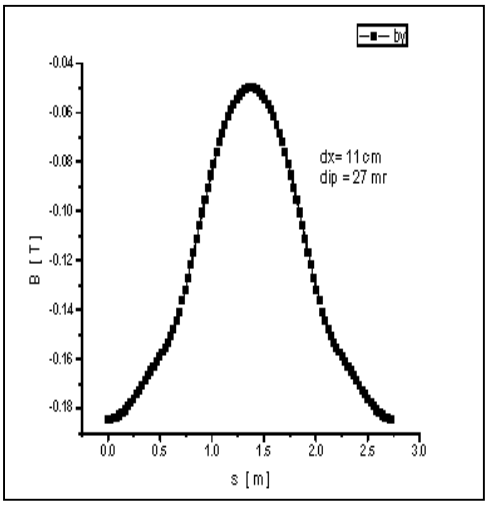
There are some complexities in generating the fields for small rings which are much less important for straight cooling channels. The coils being used in this design are large-diameter, short, and carry high-current; the maximum field is several tesla. The ring is small, a little over 5 meters in radius. From every point in the active (muon-accessible) region of the ring, a particle is significantly affected by the field generated by all the coils, even the ones on the other side of the ring. Hence one must develop a single global field map rather than assuming that a particle sees field components only from the closest coils.

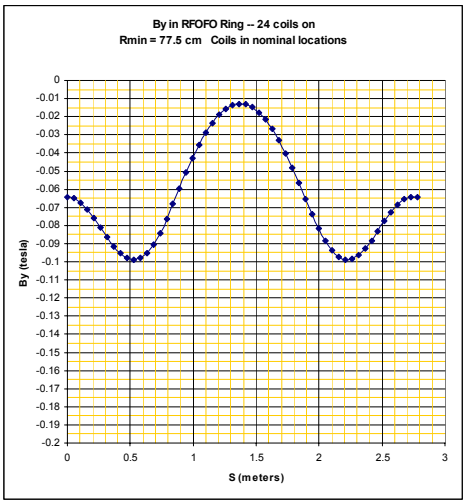
It has not proven easy to get a clear statement from ring designers regarding how accurately the field must be known. This is hardly surprising; the mechanisms of cooling are complex and still being intensively studied. For now, if cooling performance claims are to have credibility, it is important to know the field to high accuracy, so that even rough qualitative arguments will suffice to show that it is good enough. Our goal has been to know the components of $\mathbf{B}(\mathbf{X})$ to 1 gauss averaged over the active volume, and no worse than 3 gauss at any point in the active volume. Given the fact that the maximum field strength is several tesla and the field gradients are very steep, this is a challenging requirement.

Phase 1 -- The Original Field Discrepancy Problem and its Resolution

Upon arriving at the University of Mississippi in October 2002, I learned that there was a discrepancy between two magnetic field generation programs, both attempting to model the magnetic field in the active (muon-accessible) region of the RFOFO cooling ring. There were some odd aspects to the discrepancy. In transport coordinates, both the longitudinal component of the field Bs' and the horizontal transverse component Bx' agreed very closely, better than 1%. This offered some assurance that the simulations were not attacking completely different problems.

However, there was a very large discrepancy in the Y' (vertical transverse) component of the field. Plots of the two By' fields for points along the nominal beam trajectory are shown for one 30-degree cell of the ring; the other eleven cells have identical fields. The BNL field (left, Oct. 16, 2002) and the Cremaldi field (right, mid-October, 2002) are plotted at roughly the same scale.





Each cell contains two coils, equidistant from the cell midpoint, having opposite polarity and X' tilt. The coil centers are at about S' = 0.55 and 2.20 meters. It is the tilt about the X' axis that produces the bending field which orbits the muons around the ring. It is important that the around-the-ring average By' have the correct value (about 0.125 tesla) to produce stable orbits for 0.200 GeV/c muons. As plotted above, the BNL field, with coils tilted 27 milliradians, satisfies this requirement. The Cremaldi field By' does not reach 0.125 tesla anywhere, and the average By' is much less. If the Cremaldi field shape were correct, then the coils would need to be tilted far more than 27 milliradians to produce the necessary bending field. One could not rule out the possibility that a tilt this large would have serious repercussions on the cooling performance of the ring.

Both simulators agree that By is a minimum at the cell midpoint and increases symmetrically about the midpoint as one approaches the coils. However, the BNL field continued to increase as the observation point passes from the cell midpoint through the coils and approaches the cell boundaries, whereas the Cremaldi field reached a maximum near the coil centers and then began to fall again approaching the cell boundaries.

It is easy to make a first-order argument that the Cremaldi field is the more plausible of the two. Suppose that you "straightened the ring" by placing the coils along a single straight axis, and then shifted the coils along this axis so that they were spaced at equal intervals. In this case, symmetry arguments lead one to believe that By' should be rather like a sine wave, with nearly equal maxima at the coil centers and nearly equal minima at points midway between coils.

If every other coil is now shifted along the axis, so that inter-coil gaps alternate large, small, large, small . . ., then By' in the small gaps should become more like the field in the coil centers, while By' in the large gaps should approach zero more closely. This is exactly what the Cremaldi field does. In contrast, the BNL By' becomes very strong in the cell-boundary inter-coil gaps, even though the inter-coil spacing there is only a little smaller than the intercoil spacing between the two coils of a single cell. The shape of the field-strength plot should not change radically if the coil placement axis is bent into a circle, as long as the radius of the circle is large compared to the inter-coil separation.

Of course it is easy to mislead oneself with this sort of first-order argument. I was surely not confident enough in either field to go forward with simulations based upon it. Therefore, I decided to write a third field generator, independent of the other two, that would not only provide a "tie-breaker vote", but embody visualization tools and consistency checks that would hopefully settle the matter to everyone's satisfaction. So it came to pass that there were three more-or-less independent field generators -- BNL, Cremaldi and Bracker.

First Steps -- Biot-Savart Fields for Various Coil Configurations

Whenever there is uncertainty about the validity of a complex system, my approach is to begin work with a "strategic withdrawal" back to a simple system which can be well described analytically, often employing symmetry arguments. First the approximation methods needed for the complex system are made to work well on the simple symmetric systems; then the symmetries are broken and the complexities added back in slowly, checking along the way that to ensure that changes are thoroughly understood.

(1) The Biot-Savart field generator was first checked by computing the axial field of a single-turn circular coil and comparing the generated field to the well-known analytic solution. Even with this greatly simplified configuration, it is possible to start studying issues such as "Is 32-bit arithmetic likely to suffice for the complete problem?" and "How many segments must each turn be divided into in order to achieve acceptable accuracy?" In this case, I decided that single-precision arithmetic was probably not going to suffice, and that the 100 segments per turn previously used at Ole Miss would probably be too coarse-grained. It is always well to address these matters as early as possible.

(2) The slightly-off-axis behavior of the field generator was explored. There are analytic approximations and of course obvious symmetry arguments that can be employed. Once again, I tried to ensure that everything was working extremely well on this simple task before moving on.

(3) The "real coils" were built up by superimposing multiple radial layers and longitudinal layers of turns. Since these coils are simply superpositions of single-turn coils, they are analytically solvable to the same degree as is a single-turn coil. As a preliminary solution, each coil was composed of 50 longitudinal layers and 11 radial layers of turns -- a total of 550 turns, each 1 cm x 1 cm in cross-section. (All the current

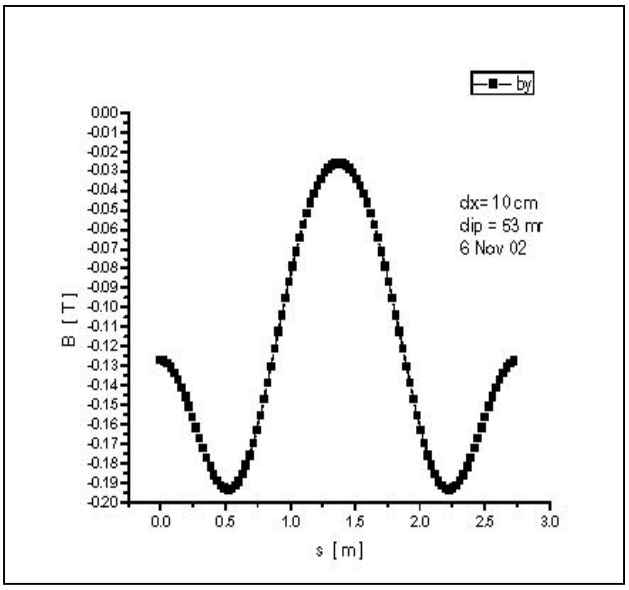
in the turn is assumed to flow at the center of the turn.) Each turn was divided into 360 (1-degree) straight segments. Initially the current segments had their end-points on the winding center, so a turn consisted of a 360-sided polygon inscribed within the circular winding. It proved to be important, even with 360 segments per winding, to use a polygon midway between the inscribed and circumscribed polygons.

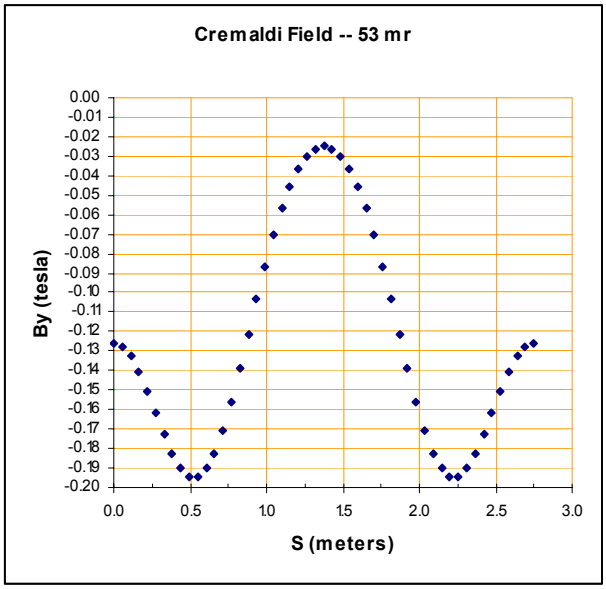
(4) I placed two of these 550-turn coils on a single axis, separated by a specified longitudinal distance, and computed the field along the axis. The coils had opposite polarity, so that the longitudinal component of the field was zero midway between the two coils. The axial field can still be computed analytically at this point, though this is the last configuration for which it can be done simply. From this point on, one must proceed not knowing the "right answer" exactly, though there are still continuity and symmetry arguments that can increase one's confidence that the results are correct.

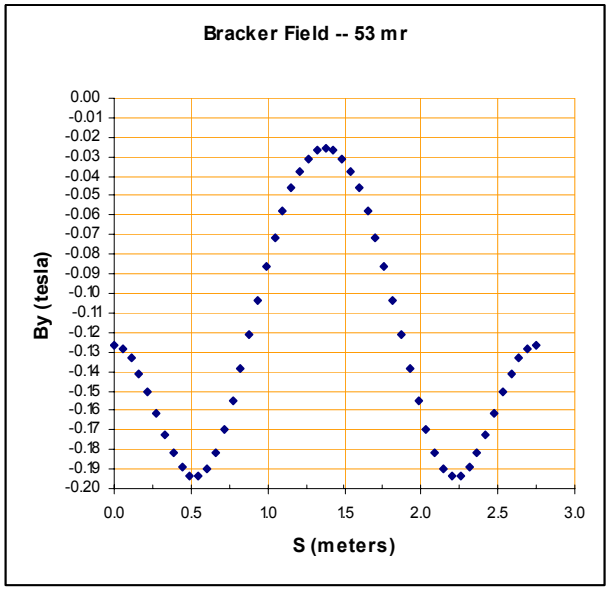
(5) I started tilting the coils about a horizontal axis, one clockwise and the other counter-clockwise, so as to simulate the tilting done to produce the ring's bending field. (Note that the rotations around a vertical axis, needed to position the coils around the ring, have not yet been made.) Special attention was devoted to the behavior of the Y (vertical) component of the field as the tilt angle, starting at 0, was increased. It was discrepancies in By, after all, that had prompted this whole effort.

(6) I put three magnets (alternating polarity and tilt) at equal intervals along a straight axis and studied the near-axis behavior to ensure that the expected symmetries were produced. Then I slowly moved the center magnet axially to the left, so that the nearest-neighbor magnet spacings were no longer the same. Next, I slowly shifted the magnets and tilted them about a vertical axis to place them on a coil-placement circle of the proper radius. At every step, coarse field maps were generated and plotted until I felt that I had a good semi-quantitative understanding of the field's behavior as the coil placements were slowly varied. Finally, all of the remaining 21 coils in the ring were placed and turned on to ensure that they did not produce first-order changes in the By components being examined.

Upon completion of this program, the field map I generated agreed very closely (better than 1%) with the Biot-Savart field produced by Lucien Cremaldi and disagreed very substantially with the field map generated at BNL. As a final check, I generated the field for a ring similar to the nominal ring, but with the coils equally spaced around the ring at 15 degree intervals. As expected, the axial By became very like a sine wave, distorted (but not severely) as the coils were pushed back to their proper positions around the ring. Hence we concluded that the Cremaldi field was likely to be nearly correct and the BNL field incorrect, and in the end this proved to be the case. Errors were found in the BNL field generator, and calculations done at BNL quickly concluded that the tilt angles must be increased to about 53 milliradians to produce the bending field required. This reconciliation of the three field generators (BNL, Cremaldi, Bracker), at least to the 1% level, marked the end of the first phase of work.







Upper left: New BNL Field (Y' component) with 53 milliradian tilts
Upper right: Cremaldi Biot-Savart field with new (53 milliradian) tilt angle
Lower: Bracker Biot-Savart field with new (53 milliradian) tilt angle

A Comparison of Field Map Generating Techniques

The BNL field was generated using BSHEET / ELLTHREE / RC -- a well-tested package deployed in ICOOL and elsewhere. In this approach, a coil is represented by a small ensemble of coaxial cylindrical current sheets. The field at an observation point is computed by summing field contributions from each current sheet. The field contribution from a current sheet is (insofar as I understand it) calculated by the iterative solution of a small system of coupled equations involving elliptic integrals.

The Cremaldi and Bracker fields were generated using independent implementations of a Biot-Savart summation technique. A coil is represented by a large number (typically 550) of circular turns; the turns are divided into a large number (typically 360) of straight segments. The field at an observation point is computed by summing field contributions from each segment. The field contribution from a segment is computed using the Biot-Savart law.

In both cases, computing the field for the entire ring requires that the coils be placed at prescribed positions around a coil placement circle, typically a circle concentric with the nominal beam trajectory and ~10 cm larger in radius. Then the coils must be rotated and tilted appropriately. The whole-ring field at some observation point within the active (muon-accessible) volume of the ring is computed by summing field contributions from each of the 24 tilted coils in the ring.

For all the field generators, one provides a method for computing the field from a single coil, and another method for combining the single-coil fields into the 24-coil field for the ring as a whole. In summary:

To find the total 24-coil field for any observation point in the ring:

```
Zero the total field accumulator
Position the coils around the ring
For all coils in the ring . . .
  Compute the field contribution from one coil
  Add the coil's contribution into the total field accumulator
```

To find the field due to a single coil (Biot-Savart)

```
Zero the coil's total field accumulator
For all turns in the coil . . .
  For all segments in the turn . . .
    Compute the field contribution from one segment
    Add the segment's contribution into the coil's total field accumulator
```

To find the field due to a single coil (BSHEET)

```
Zero the coil's total field accumulator
For all current sheets in the coil . . .
  Compute the field contribution from one sheet
  Add the sheet's contribution into the coil's total field accumulator
```

The position of turns, segments and sheets within each coil is specified in a coordinate system fixed in the coil. One axis of the coil coordinate system is the axis of the cylindrical coil. To get to a coordinate system that encompasses the entire ring -- the ring cartesian coordinate system for GEANT, the transport coordinate system for ICOOL -- you must position each coil around the coil placement circle and rotate it around two axes. The transforms involved are straightforward translations and rotation matrices, but like all such transforms, there are myriad opportunities for dropped terms, sign errors, etc.

For the Biot-Savart coordinate transforms, I made no effort to calculate an overall transform matrix combining all the simple rotations and translations. Instead, I performed the elementary transforms one at a time, and stored vector components for all the intermediate coordinate systems. This made it possible to use Excel dynamic scatterplots to exercise and graphically validate each simple rotation and translation to ensure that it was working as expected. No coordinate transform was installed in the field generation program until it was rigorously checked in this manner.

Biot-Savart Field Generation -- A Brief Summary

The RFOFO ***ring*** has 12 pie-slice ***cells***. Each ***cell*** has two ***coils***. Each ***coil*** consists of (typically) 550 ***turns*** (11 radial layers, 50 longitudinal layers). Each ***turn*** consists of (typically) 360 straight current ***segments***. Each ***segment*** vector has a ***head***, a ***tail***, and a ***midpoint***.

The segment's ***head***, ***tail*** and ***midpoint*** coordinates are transformed from the ***tilted-coil frame*** to the ***untilted-coil frame*** to the ***ring frame***. An ***observation point*** is specified in the ring frame. The magnetic field at the observation point due to each current segment is computed using the Biot Savart law. The field vector contributions from all current segments are summed to produce the ***magnetic field at the observation point*** in the ring frame. If it is necessary to express the field in ***transport coordinates***, then the field vector is rotated as needed.

Typically, we need to know the magnetic field at an ***ensemble of observation points***. The procedure outlined above is simply repeated for each observation point in turn.

Pros and Cons of the Two Primary Field Generation Methods

It might be well to digress briefly to spell out explicitly the advantages and disadvantages of these two primary field generation methods. The Biot-Savart field generator is much easier to understand; anyone with a passing acquaintance with Halliday & Resnick knows exactly how to employ the method to generate a field given a large ensemble of current segments whose currents, positions and directions are known. The BNL sum-of-sheets method is considerably more complex. As we shall see, there are some "gotchas" associated with the algorithm that one must be careful to take into account. Hence from the standpoint of simplicity -- very important in the early development of a simulation algorithm -- the Biot-Savart summation method is the clear winner.

However, Biot-Savart summation does have one major drawback; it is excruciatingly slow -- probably about 100 times slower than the sum-of-sheets method computed to equivalent accuracy. It quickly became clear that the Biot-Savart summation algorithm's role would become that of a "gold standard" against which to check other primary field generators over a limited ensemble of test points; it would not be used for production of a full-blown field grid, for example. Therefore, as soon as the Phase 1 difficulties were resolved, we kept a parallel software development track going, which began with sum-of-sheets software provided by Rick Fernow and involved various local tunings, tweakings and wrappings of that software.

Both field generation methods are approximations. A real coil is not a small set (traditionally three) of current sheets, nor is it a set of polygonal windings with current concentrated in the winding center. Neither Biot-Savart summation nor solving the sum-over-sheet equations is an exact algorithm. Whenever approximate methods must be used, one must always question whether the scale of the approximation is adequate to the task. Is the sum-of-sheets method using enough sheets to represent a coil? Is the algorithm's iteration procedure converging to sufficient accuracy? Is the Biot-Savart model using enough windings, and dividing the windings into enough segments? Are both algorithms working to sufficient numeric precision? I found it much easier to think about and test these questions using the Biot-Savart field generator, but opinions may vary.

Finally, a word about coordinate systems. The work being done at Ole Miss was aimed toward providing a field map for GEANT-based simulations of the ring. The GEANT simulation envelopes the entire detector in one large cartesian coordinate system with its origin at the center of the ring. Both positions $\mathbf{X}$ and the field components of $\mathbf{B}$ are represented in these coordinates. The sum-of-sheets work was being done mostly in support of ICOOL simulations, and ICOOL uses "transport coordinates", a coordinate system whose origin translates and axes rotate relative to the cartesian frame. Transport coordinates are not cartesian. It isn't hard to write the transforms between the two systems, but in order to compare results from the two field generators, one must do the transformations exactly correctly. Symmetries that are obvious and helpful in understanding results in one coordinate frame don't work at all in another, which can sometimes lead to confusion.

The Many Tasks that Phase 1 Left Undone

The completion of phase 1 left us feeling satisfied that we had a way (indeed two quite different ways) of calculating a field for the RFOFO ring, and that there were no major outstanding disagreements about the shape of the field. However, agreement at that level is no more than a ledge to rest on before resuming the climb, and much work remained to be done.

What didn't we have? First, we had compared the various fields for only a small ensemble of test points, and those test points were distributed at equal intervals along the nominal beam trajectory. Even with that limited check, the agreement between the various methods was only about the 0.1% to 1.0% level (depending on the field component). Clearly the field maps needed to be compared to a much higher level of accuracy, and over a large randomly-selected ensemble of test points within the ring's active volume.

Second, we had not made even the most rudimentary tests to see whether the field obeyed the most basic physical requirement -- that it obeyed Maxwell's Equations. The current density is 0 throughout the active (muon-accessible) volume of the ring, so both the divergence and curl of the B field should be zero everywhere. One must of course recognize that there many Maxwellian fields, and that only a very small subset of these can be generated by the coils we are using in their assigned positions. Obviously one should have more confidence in a field which obeys Maxwell's Equations than in one that does not, but there are many other checks, many already made in Phase 1, to satisfy oneself that ours is the One True Maxwellian Field generated by our coil configuration.

Third, in the Biot Savart field generator, we did not have a method that would generate a fine-grained field map in finite time. Even during Phase 1, I started developing a BSHEET-based field generator, using slightly modified BNL-provided code and our own coordinate transform algorithms, to complement the Biot-Savart "gold standard". More work remained to be done, but already by the end of Phase 1 we had written a field generator in which either the Biot-Savart method or the BSHEET method could be switched on, while the remaining code (e.g. coordinate transforms) was largely common to both methods.

Fourth, we had no secondary field generator at all -- a method fast enough to be usable within a simulation environment without slowing the simulation down to a quite unacceptable degree. The Biot-Savart method was hopeless in this capacity, but even the BSHEET-based primary generator was many times too slow. I intended to precompute the field on a rectangular lattice of points, download this field map into the simulator, and supply software that would accept an observation point from the simulator and interpolate in the field map grid to find the field at that point. That this scheme would deliver the kind of accuracy we required with a lattice small enough to fit in computer memory was far from clear. However, there were other options -- e.g. expressing the field in

orthogonal functions of some sort, and then writing a routine to quickly calculate the field at any given point using stored coefficients.

Finally, with the completion of Phase 1 we bid farewell to the Cremaldi field. I therefore broke my longstanding ban on reading the code, and concluded that although the details of our implementation were different, and there were some very minor problems with the coil geometry, the Cremaldi field and the Bracker field should be extremely similar, as indeed they are.

Phase 2 -- Mostly Curls and Divergences

Rick Fernow emphasized that it would be wise to check all of our fields against Maxwell's Equations. To begin our effort, I modified our field generator to compute the field at a somewhat complicated set of observation points. There were ***primary observation points***, selected (at first) as before at points along the nominal beam trajectory. Around each primary point were six ***secondary observation points***, at a distance ±***dxyz*** from the primary point along the X, Y and Z axes. The secondary observation points made it possible to compute the partial derivatives needed to compute the curl and divergence of **B** at the primary observation point.

I started work on a curl-divergence checking program, went home for the evening, and read my email the next morning, only to find (in addition to the usual offers to refinance my mortgage, grant me a diploma, enlarge various body parts, and view pictures of others who had already done so) a note from BNL, reporting that the curls and divergences of the current BNL field -- a field which I believed to be nearly identical to our own -- were not even close to zero. Arghhhh!!!

I rushed my own curl/div checker to completion and ran it for a small ensemble of points along the nominal beam trajectory. The curls and divergences, while not precisely equal to zero, seemed very low -- a part in 10^5 below the derivatives comprising them. (To those of us not jaded by too much experience in such matters, the disappearance of the curl components and the divergence seems almost magical, especially given the fact that in many cases the near-zero value is produced from the difference between two very large numbers.)

The next couple of days were very confusing. Eventually it transpired that there was a problem with the BNL method for computing curls and divergences. (It should be noted that calculating the curl/div in transport coordinates is considerably harder than our calculation, which is done in cartesian ring coordinates.) Once the error was fixed, then all methods for doing curl/div checks, on all fields, in all states, yielded satisfactory results for small ensembles of points along the nominal beam trajectory.

At this point -- having computed and checked the curls and divs for a few dozen points along the nominal beam trajectory -- I decided to let the matter rest. This turned out to be a mistake. A few days later, BNL reported that although the curls and divs were indeed

zero along the nominal beam trajectory, they were not zero for points (for example) along the coil placement circle. Having been a bit lazy about making a thorough test myself, I was in no position to refute this claim immediately. Once again the curl/div checker was unlimbered. This time I did what I should have done right at the outset, and generated the field for a huge ensemble of primary points randomly distributed within the ring's active volume, and their associated secondary points. The results: both good and bad. Good because the curls and divergences were very close to zero almost everywhere. Bad, because they were not close to zero absolutely everywhere, and the points at which they were bad seemed not to be systematic in any recognizable way. Usually when problems of this kind arise, there is a region around the really troublesome points in which behavior gets worse and worse as you sneak up on the troublesome point. This seemed not to be the case, and that was extremely troubling.

At least we could report that the curls and divs were zero almost everywhere. It didn't take BNL long to find another problem with their curl/div calculator, once again due to the non-cartesian transport coordinate system, and soon everyone was satisfied that the curls and divergences were close enough to zero in all coordinate systems and on both sides of the Mason-Dixon line.

However, I decided not to ignore the few troubling points at which the curls and divergences were bad (different from zero by a part in a thousand relative to the local field gradient). Often enough I have found that pursuing this sort of anomaly, even if it has no significant effect on the results being sought (which is, after all, the propagation of particles, not the perfection of fields), leads to a better understanding of problems that are of significance. So it proved in this case.

By this time, the local field generator was readily switchable between sum-of-sheets (BSHEET) mode and the Biot-Savart mode. Because the sum-of-sheets field generator is so much faster, and because it had been compared reasonably carefully with the "gold standard" Biot-Savart field generator, I used it whenever possible. The fields for the large curl/div test were generated using the sum-of-sheets method. However, when I switched to the Biot-Savart field generator, the anomalous seriously-nonzero divs and curls went completely away. Hmmm....

A day or two of investigation followed. Finally, I found that the test points with anomalous divs and curls shared one common feature: they lie very close to a projection of a current-sheet cylinder into the active volume of the ring.

It is hardly surprising that a semi-analytical method for finding a magnetic field doesn't behave very well when working very close to a current-carrying sheet. Because the model current sheets are infinitely thin, the fields and field gradients get very extreme when you are close to the cylindrical sheet. However, it seems that the BSHEET routine is not only unhappy when generating the field near a sheet; it also behaves badly when the test point is anywhere on the longitudinal projection of the cylindrical current sheet, even meters away from the sheet itself. In a straight beamline, this would be of no concern. In a ring, and especially in a ring with tilted coils, the projections of the coils' current sheets do

enter the active volume. If a test point happens to be very close to a projected current sheet, the calculated field becomes very inaccurate.

Finally understanding what was special about the bad points, and knowing that the Biot-Savart method indicated that bad behavior was not inherent in the physics itself, I was able to explore the behavior of the sum-of-sheets algorithm in the immediate vicinity of the troublesome areas. I found that if you examine the field generator's performance approaching trouble points with a powerful enough microscope, you do see progressive deterioration of the field accuracy as measured by curls and divergences or by comparison with the Biot-Savart field.

Changing the sum-of-sheets routines to double precision reduced the volume of space seriously affected but made the problem even more striking within the reduced volume that remained. Changing the convergence criteria within the algorithms reduced but did not eliminate the problem. Finally, I decided that rather than monkey with an algorithm I didn't fully understand, I would simply sidestep the problem. I expanded the wrapper routine already surrounding BSHEET (mostly to center coordinates on the coil center) to recognize observation points too close to the current sheet projections, calculate the field at two nearby points on opposite sides of the sheet projection, and average them. I compared this solution to results from the Biot-Savart field generator, and examined the curls and divergences of formerly troublesome points. All seemed well, and this work-around is still in use today.

The milestones that marked the end of Phase 2 were (1) providing two field generators that agreed with each other (for a large ensemble of randomly-selected points within the ring's active volume) to about a part in 1000, (2) writing and employing curl/div checking software to ensure that the curls and divergences nearly vanish (again for a large ensemble of randomly-selected points) and (3) circumventing the bad behavior of the sum-of-sheets field generator at test points near the projection of a current sheet.

The Tasks that Phase 2 Left Undone

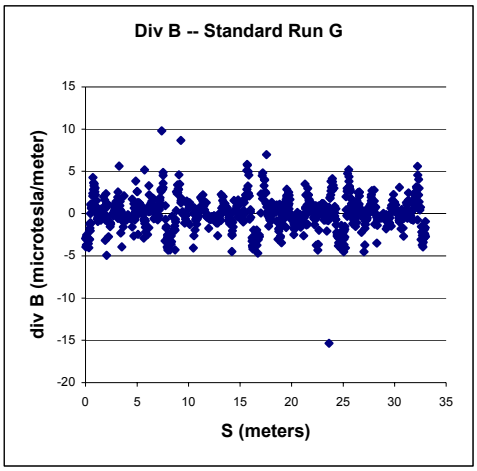
There was still work to be done:

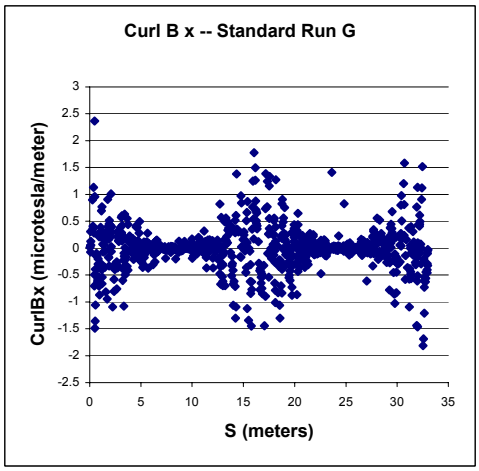
1. Agreement between the fields at a part per thousand was deemed insufficient. Errors of that magnitude might not impact cooling simulations in significant measure, but it was hard to be sure. The goal was to achieve fields whose RMS accuracy (over the entire active volume of the ring) was a part in ten thousand.

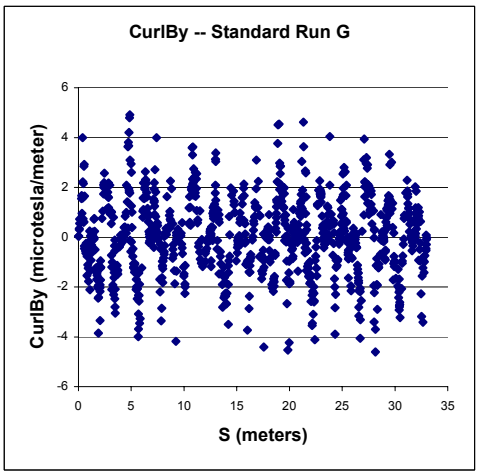
2. There was still no secondary field generator fast enough to be incorporated into a simulation program.

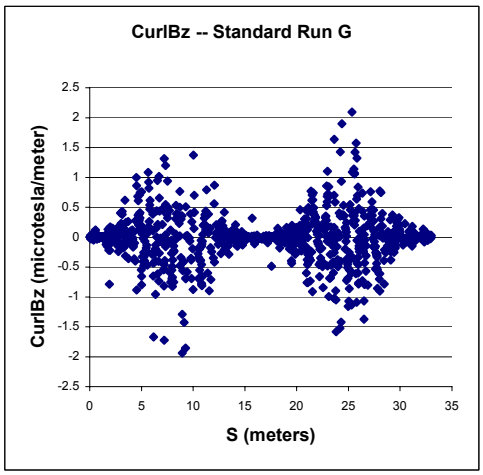
3. There was still no local study showing that this field would even orbit particles, let alone a demonstration that plausible uncertainties in the field would not compromise measures of a ring's cooling performance.

Example: Curl components and divergence for the G field (11-sheet double-precision BSHEET-derived) Note that the fields are a few tesla, whereas the vertical scale of the plots is microtesla/meter.









Phase 3 -- The Secondary Field Generator

The work of phases 3 and 4 partially overlapped each other in time, but it is convenient to divide them into distinct tasks here.

By now I had enough confidence in the field map to want to try it out in a real simulation. Romulus Godang had written a partial implementation of the ring in Geant 3, and had orbited particles in a small region of phase space near the nominal reference particle -- a 200 MeV/c muon with initial transverse position = 0 and transverse momentum = 0 in transport coordinates -- using a constant By field. I decided that it was time to incorporate our field map in this simulation and study its behavior.

As noted above, neither primary field generator (sum-of-sheets and Biot-Savart) was nearly fast enough to support a simulator. Therefore I decided to use a field-on-a-lattice approach. First, I would define a cubic lattice within the active volume of one cell of the ring. Second, I would use the sum-of-sheets field generator to generate a field map at each point on the lattice. Third, I would produce code, to be linked into the simulator, which would read the field map into memory at the start of the job and interpolate to find the field at an arbitrary point in the active volume of the cell whenever the simulator required it. Fourth, if needed, I would provide additional routines to propagate the field from one cell of the ring to the entire ring. (There was a belief abroad at the time that Geant would handle this for us; this turned out not to be right.)

I had serious misgivings as to whether this approach would work. The active volume of the ring is quite large (owing to the need to handle beam of high initial emittance), and one could easily imagine that a lattice fine-grained enough to provide a field of the required accuracy would be require more memory than could be provided, even in today's world of cheap memory. Nonetheless, I decided to begin with a 1 cm cubic lattice and study the matter.

In the ring coordinate frame used by Geant, even listing the lattice points in the active volume is not entirely trivial. (This is a much easier problem in transport coordinates, but if you use transport coordinates, then you are faced with interpolating in a non-cartesian grid; I decided to deny myself that pleasure.) In the end I used a brute-force approach. I enclosed the active volume of the cell in a large box, plenty large to enclose the active volume and some extra volume around the edges. For every lattice point in the box, I tested to see whether the point was close enough to the nominal beam trajectory to be included in the active volume. I simply wrote a list of the hundreds of thousands of points at which the field would be calculated. (Calculating the field at every point in the box would have taken several times longer.)

Excel plots were used to convince myself that the active volume of the cell (plus some extra volume around the edges) was adequately covered with lattice points. Then a job was set up to read the lattice file, compute the field at each lattice point, and store the position and field value in a huge text file. Using the sum-of-sheets field generator, this took an overnight run.

The monstrous formatted text file was converted to an unformatted (binary) file to save both space and field-loading time. Since unformatted files are not platform-independent, I wrote a platform-independent fortran routine to do this conversion. To produce the unformatted field file on a new platform, you transfer the formatted file and the conversion routine source code to the new platform, compile and link the conversion program, and run it to produce the binary file in a locally-acceptable format.

For incorporation into the simulation program, there is ***LoadField*** which reads the unformatted field map file into a large array, ***FindField*** which interpolates within the array to deliver the (approximate) field for any point in the active volume of the ***cell***, and ***FindFieldAnywhere*** to deliver the (approximate) field for any point in the active volume of the entire ***ring***. Normally, the simulator calls LoadField once near the beginning of the simulation (before any particle tracking is undertaken) and uses FindFieldAnywhere to deliver the field at any point required by the simulator. In Geant, routine GUFLD delivers field values to the particle propagation routines; GUFLD calls ***FindFieldAnywhere*** and does some unit conversions (centimeters to meters, tesla to kilogauss). Note carefully that ***FindFieldAnywhere*** expects test points to be specified in units of meters (Geant uses centimeters) and returns field components in tesla (Geant expects kilogauss).

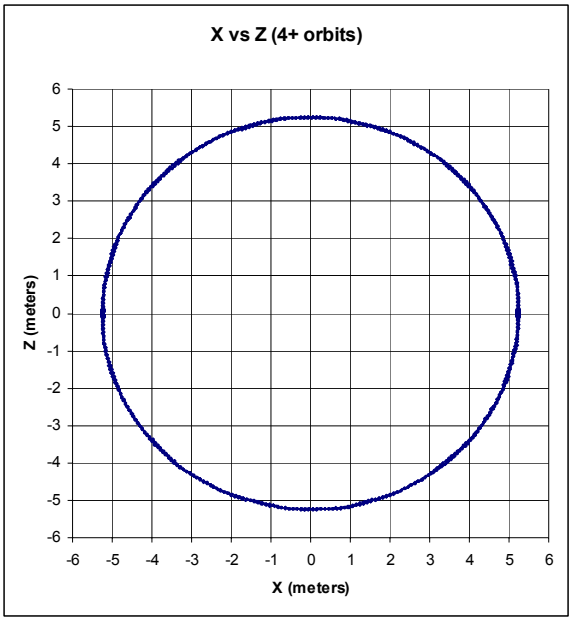
In due course, this software was incorporated (with much assistance from Romulus Godang) into the partially-completed Geant simulation, and particles were orbited successfully. Romulus then undertook some studies to determine how far (in position-momentum phase space) a particle could depart from the nominal reference particle and still remain confined in the field; he has reported those results elsewhere.

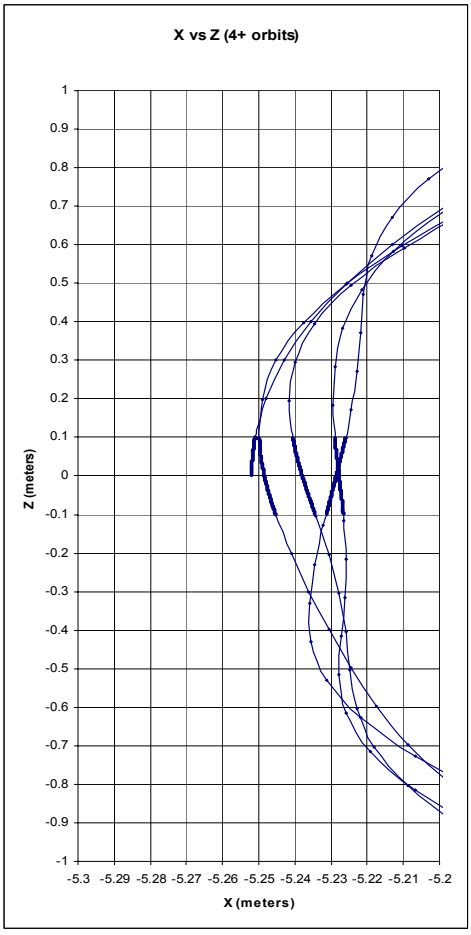
Interpolation in the field lattice is done using CERNLIB routine FINT, which is a multi-dimensional linear interpolator. I had grave misgivings as to whether a linear interpolator would be sufficient. Clearly there is no point in honing a primary field generator to very high precision, only to have the precision squandered by an inadequate secondary field generator. So once again, the large ensemble of randomly-selected points within the active volume of the ring was dusted off, and a large-scale comparison was made between the field as computed by the primary generator and the interpolated (secondary) field. The results were encouraging; the RMS discrepancies between the primary field and the interpolated field were very low, even with a 1 cm lattice and a linear interpolator. (See appendix for details.)

It should be noted that FINT does have one drawback. It expects that the input lattice be organized into a cubic array, which means that all of the points in the box circumscribing the cell's active volume, not only the points in the active volume, must be present. This makes for very large field arrays. It probably would not take too much effort to rid FINT of this requirement, but I haven't done so. The lattice points at which the field has not been calculated, all outside the proper range of muons, have field components of -9999. If muons accidentally transgress outside the proper active volume, they are flung out of the ring in a spectacular fashion.

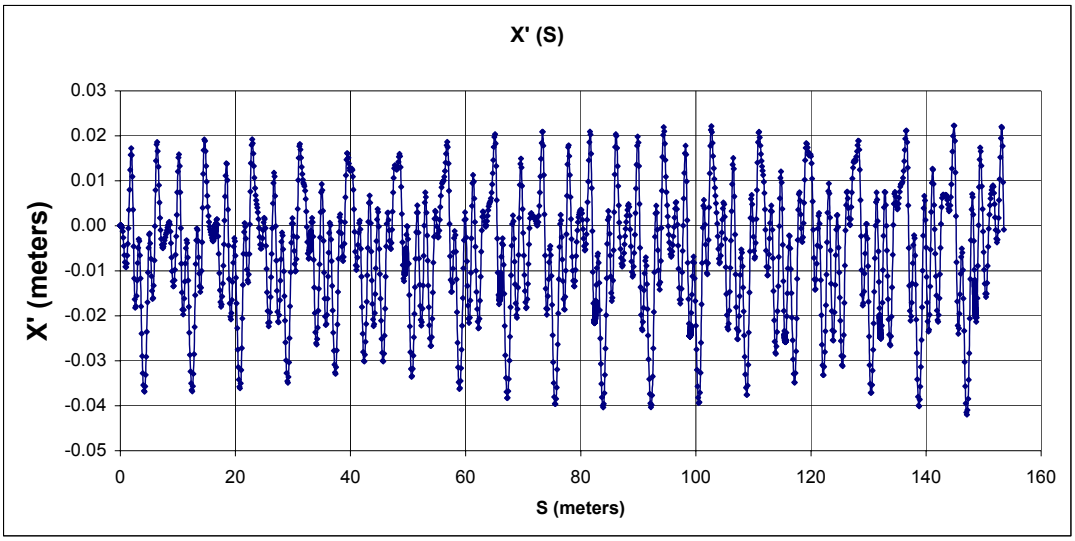
Before incorporating the field into the Geant package, I incorporated it (for testing purposes) into a crude little particle propagator that we jokingly called CremaldiCool, and used the package to orbit particles. This proved to be an instructive exercise, and I only abandoned CremaldiCool when I realized that in order to make more progress, I was going to have to rewrite large parts of Geant (or at least ICOOL), which seemed pointless. Suffice it to say that CremaldiCool provided additional confidence that if you deliberately distort the field map to a degree comparable to the uncertainty in the field map, particle trajectories are not greatly altered. Quite obviously this does not demonstrate that the cooling performance of a real simulation is not significantly altered by similar changes to the field map. CremaldiCool has now been abandoned.

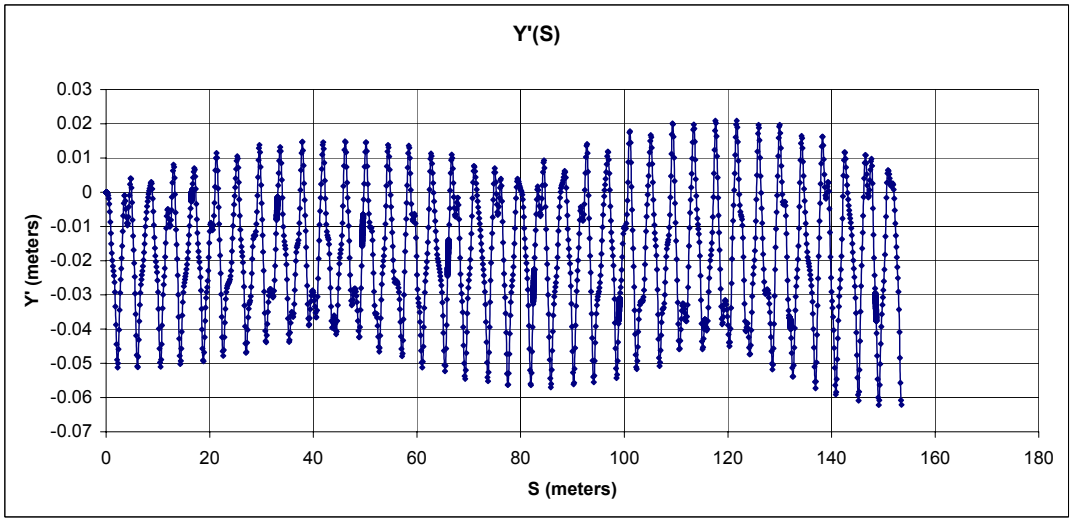
CremaldiCool Orbits within the 53 milliradian field:
Left: Full ring, about 4.3 orbits
Right: Left edge of ring magnified; changes in the magnetic field comparable to the uncertainty in the field (not shown here) seem not to change the trajectories significantly.





X' and Y' as a function of S'; CremaldiCool orbits with 53 milliradian field.
The results from Geant simulations are very similar. One turn is 33 meters of S'





The benchmarks that concluded phase 3 were (1) delivery of a secondary field generator to the Geant simulation running under Linux, (2) orbit-stability tests of the secondary field generator using CremaldiCool, and (3) primary-to-secondary field comparisons that show that linear interpolation within an only moderately fine grained field lattice are almost certainly sufficiently accurate.

The Tasks that Phase 3 Left Undone

1. Agreement between the fields at 0.001 tesla was still deemed insufficient. Errors of that magnitude might not impact cooling simulations in significant measure, but it is hard to be sure. The goal was to achieve fields whose RMS accuracy (over the entire active volume of the ring) was about 1 gauss

2. There was still no convincing proof that the remaining uncertainties in the field were small enough to ensure that they would not corrupt cooling performance measures.

Phase 4 -- The Pursuit of the Last Factor of Ten in Accuracy

There were still discrepancies at the level of a few gauss between the two primary field generators. These discrepancies were not random noise; they were systematic, and plots of the field discrepancies often had about the same shape (only a thousand times smaller) as the fields themselves. This strongly suggested that the discrepancies related in some way to the adequacy of the coil models.

An enormous number of possible causes were explored. Very few made any significant difference. A couple that did make a noticeable difference were a bit embarrassing, for example a place in the code where the radius of the nominal beam trajectory was defined to inadequate precision.

I went through all the software and carefully parameterized all of the ring geometry, so that (for example) the ring radius and the ring circumference would be self-consistent to ten significant figures. No routine was allowed to have its own ring geometry or numeric constants.

The most important changes were simply increasing the number of current sheets, turns and segments in each coil. Three current sheets were not enough; we now use eleven. 360 segments per turn were not enough; we decided to use 1000. At each improvement, the curls and divergences got closer and closer to zero, and the discrepancies between the two primary field generators became smaller and smaller, leading to increased confidence that the changes being made were sensible.

In comparing two fields at the 1-gauss level, I found it helpful to permit field scaling -- multiplying all the components of one of the fields by a constant close to 1. This is legitimate because it the shape of the field that we care about at the 1 gauss level, not the overall magnitude. If we built a set of coils and found that the magnetic field had the correct shape but was 0.01% too high everywhere, in all components -- and this mattered to cooling performance -- we would simply turn the coil currents down by 0.01% and carry on. The scale factors were chosen, to yield the best-looking difference plot. The scale factors used were all very close to 1, typically 0.9999 or 1.0001 or closer.

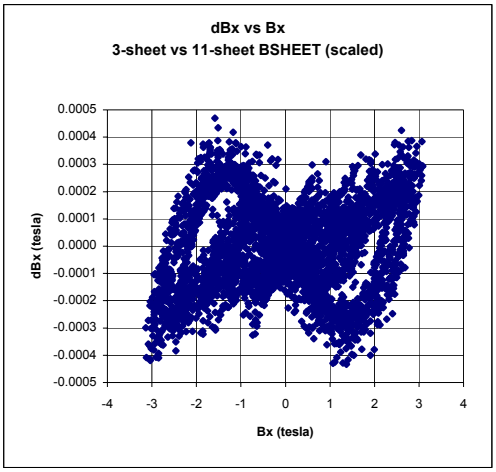
Finally, enough was enough. I convinced myself that RMS errors in the primary fields were probably below 1 gauss and that the primary-to-secondary discrepancies were of about equal size. Assuming they are uncorrelated, the two errors add in quadrature to produce a final field error (RMS over the whole active ring volume) of about 1.4 gauss, not quite the 1 gauss precision we had as a goal, but almost surely adequate to the task at hand. I ran a few tests with very fine-grained approximations of the coils (22 current sheets, 5000 segments per turn, etc.) to convince myself that there was no need to go further down this road.

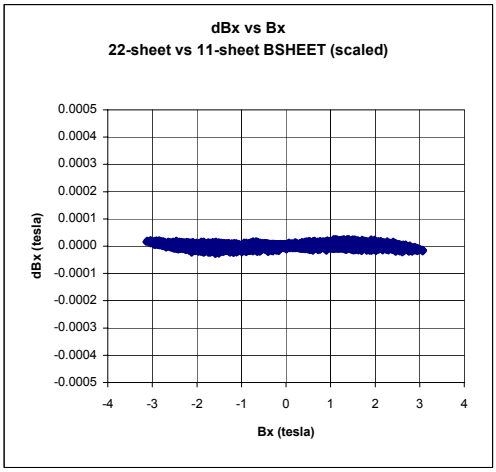
A final version of the field map (field3) was produced, and that is still the field map we use today.

Example -- For a large ensemble of random points in the active area, the Bx difference between two sum-of-sheet fields with different numbers of current sheets.

Left: Difference between a 3-sheet coil and an 11-sheet coil
Right: Difference between an 11-sheet coil and a 22-sheet coil

The results suggest that a 3-sheet approximation is not adequate, but an 11-sheet approximation almost certainly is. See appendix for details.





What Remains to be Done?

As I migrate north once again, there is still some field-related work to be done.

The Geant-based simulation of the RFOFO ring has never been completed, so the present field map is still untested in support of a complete simulation with RF and wedges. When and if the day comes that the test can be performed, the most important task will be to do

sensitivity studies. If one perturbs the magnetic field slightly, how significantly does cooling performance vary?

Presumably the trajectories of particles are not strongly affected by small-spatial-scale noise in the magnetic field. The extra bump that a muon receives in one step will be counteracted by another bump in the other direction in the next few steps. But field perturbations that are systematic over long spatial scales are another matter. Especially since the particles go through the field again and again (12 cells, ~20 turns), small anomalous kicks can build up until they really do have the potential to disrupt cooling behavior. It is very important to test the sensitivity of cooling performance measures to the worst plausible kinds of magnetic field errors, both because it provides a means to assess the ability of the field map to support reliable assessment of simulated cooling performance, and because it allows one to estimate how sensitive a real ring might be to manufacturing errors, survey errors, etc. Of course one must always avoid the temptation to use good cooling behavior to instill confidence in the adequacy of the field map.

The latest word I've heard (indirectly) from BNL is that their magnetic field yields good cooling when served to the simulator via a secondary field generator based on multi-pole expansion, but that cooling is much reduced or non-existent if the same primary field is served to the simulator through a secondary field generator based on interpolation within a grid. I had hoped to apply the same field-checking measures I have used here to the BNL field maps, but I've run out of time. I think it would be extremely useful to compare our primary field to the BNL primary field and the BNL primary field to both BNL secondary field generators (multipole expansion, grid interpolation) over a large ensemble of randomly chosen points in the active volume of the ring. It is not impossible that I could set up and run some field comparisons from British Columbia should it prove useful.

What Lessons Might be Drawn?

A few observations regarding software development efforts of this kind:

Do not begrudge the time to make careful, exhaustive checks at every step of development, starting from simple easily-verified configurations. It should not be surprising if writing and using verification software takes three or five or ten times longer than writing the software being checked. It's still cheap compared to building a cooling ring that doesn't work.

Whenever possible, use the simplest possible coordinate system -- e.g. a stationary cartesian reference frame -- for finding curls and divergences, interpolating, etc. If necessary, transform from your local coordinates into a cartesian frame, do your calculating, and then transform back to your local coordinates. It may take longer to run, but it's much easier to code.

Adopt one standard set of routines for moving between the cartesian frame and any other coordinate system you need, check them exhaustively, and always use them.

Coordinate transforms are "simple" but also easy to do wrong. Visualization tools that allow you to plausibility-check coordinate transforms after each simple translation or rotation are extremely valuable if systematically employed.

Parameterize the configuration geometry into the smallest possible set of constants, from which all others are derived. Exhaustively comment them. Include them in all the code.

If in doubt, go to double precision at the outset and build high precision into the code every step of the way.

Don't leave anything to chance in handling possible arithmetic faults, e.g. dividing by zero or taking square roots of negative numbers. Modern computers are so fast that any arithmetic problem that can arise will arise. Unless an operation is metaphysically certain to be safe, protect it.

Appendix: Verifying the RFOFO Ring Field At Randomly Distributed Points
Steve Bracker
December 12, 2002
RFOFO Field Map 27.doc

As part of our ongoing effort to assess the quality of our RFOFO field map, I have generated an ensemble of 10000 points pseudo-randomly distributed within the active (accessible to muons) volume of the RFOFO ring. For these purposes, the active volume consists of the Nominal Beam Trajectory (a circle 33 meters in circumference) and all points in space within 25 cm of the NBT.

The 10000 points are uniformly distributed in S (or equivalently, position angle b) and in X' and Y'; see the drawing below. Hence the points are almost (but not quite) uniformly distributed within the active volume. Plots showing the point distributions are below. I've made a file listing the test point positions, one line of text per point. The listed quantities are:

X, Y, Z -- position of the point in cartesian ring coordinates
S, b, X', Y' -- position of the point in accelerator coordinates (S and b redundant)

Our various field generating programs can read this file, generate the field at each test point, and output a line for each point specifying the position (from the input file) and the field components. Our field components are in cartesian ring coordinates, but these are easily converted back and forth to accelerator coordinates.

To date we have run three field generators against the first 5000 points of the 10000-point file of standard test points:

A -- The Biot-Savart field (24 coils, 50 x 11 turns per coil, 300 segments per turn) that remains our reference field. It exhibits very low divergences and curls, and the program that generates it is very simple and easily understood. It is very slow.

B -- Our standard BSHEET-based field (24 coils, 11 sheets per coil). This field is rapidly approaching the quality of Field A as we refine the algorithm and our use of it. It is very much faster than the Biot-Savart generator. For example, it can generate a full field map for 1 cell of the RFOFO ring on a 1 cm lattice in about ten hours of running on a single PC.

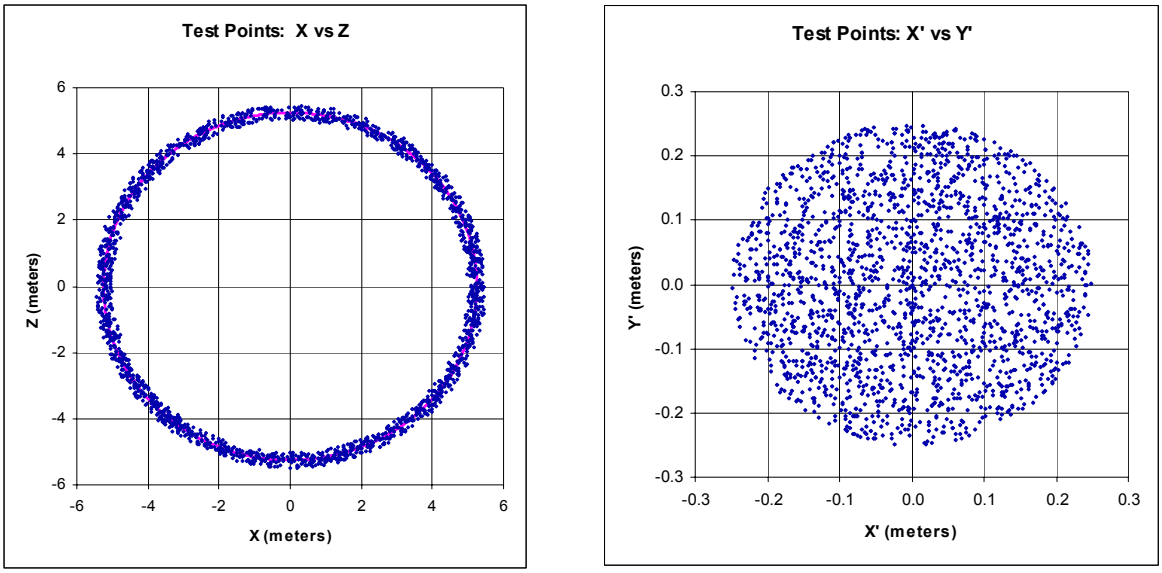
C -- A field generated by interpolating from a field map grid, which in turn was generated by a procedure that is a slight variant of the present Field B generator. As one would expect, this generator runs much faster than the Field B generator, but since the lattice spacing is fairly wide (1 cm in all three directions) there is naturally concern that it might be too coarse-grained to yield accurate field values.

When we compare field maps, it is consistent field shape that we are after. There are various reasons that a field generator may deliver a field that has the right shape but not quite the correct scale. This is of no consequence in an operating ring, since errors of

scale can be mitigated by adjusting the coil currents. It turns out that the scale factors that must be applied to maximize agreement between the various fields is extremely low; the largest scale factor being used is 1.00011. I am not quite ready to report final results yet, but the preliminary analysis is extremely encouraging. The field discrepancies seem to be at the part per ten-thousand level everywhere within the active volume of the ring.

I have a set of programs and Excel templates for making the field comparisons and presenting the results. They will be ready for routine use within 2-3 days. I am hoping that all fields being used for RFOFO simulations -- both the original field generators that go into making grids or computing coefficients for parameterized fields, and the field finders that couple directly to the simulation packages -- will be run on the file of standard test points. I am willing to do the comparisons that will assure us that the various simulation efforts are seeing the same field throughout the ring's active volume.

I am about to generate a second file of standard test points. This file presents 1000 randomly distributed primary test points and for each primary, 6 secondary test points located half a millimeter from the primary. The secondaries are used for computing curls and divergences at the primary points. It will be very reassuring to know that fields we use have nearly-zero curls and divergences for randomly chosen points throughout the entire active area of the ring. Again I have a set of programs that input the field files, calculate the curls and divergences, and present the results. Again these are almost but not quite ready for routine use.



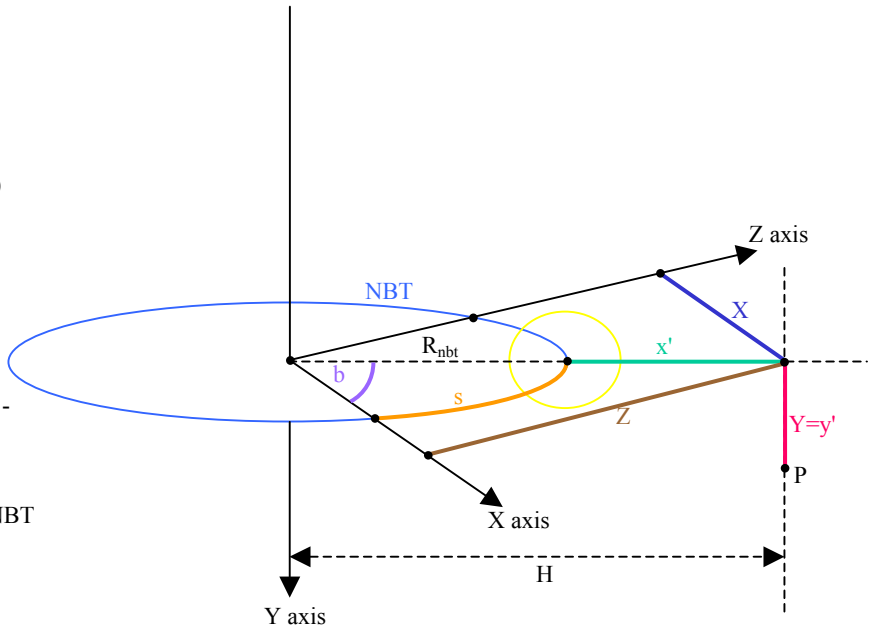
Distributions of test points around the RFOFO Ring; the first 1000 points are plotted. The (barely visible) magenta line in the left plot is the Nominal Beam Trajectory.

$s = R_{nbt}\ b$
$H = R_{nbt} + x' = \text{sqrt}(X^2 + Z^2)$

$X = (R_{nbt} + x')\ \cos b$
$Y = y'$
$Z = (R_{nbt} + x')\ \sin b$

$b = \text{atan2}\ (Z, X)$
$y' = Y$
$x' = H - R_{nbt} = \text{sqrt}(X^2 + Z^2) - R_{nbt}$

Unit Vectors:
e_b -- parallel to tangent to NBT
$e_{y'}$ -- vertically downward
$e_{x'}$ -- radially outward



Appendix: Three-Way RFOFO Ring Field Comparison
Steve Bracker
December 14, 2002
RFOFO Field Map 28.doc

In this writeup I compare six field maps for the RFOFO Cooling Ring. For all six field maps, the following ring configuration parameters remain the same:
24 solenoids centered on the Coil Placement Circle, +-6 degrees from cell boundaries
Solenoid length = 0.5 meters
Solenoid inner radius = 0.77 meters
Solenoid outer radius = 0.88 meters
Solenoid current-turns = 5239850 amp-turns
Solenoid tilt angle = 53 milliradians
Coil Placement Circle 0.10 meters larger in radius than Nominal Beam Trajectory
Circumference of the Nominal Beam Trajectory = 33.0 meters

All six fields are generated for the same pseudo-random distribution of 5000 test points within the active area of the RFOFO ring, which for these purposes is defined to be all points within 0.25 meters of the Nominal Beam Trajectory. The location of the test points are recorded in a disk file; see RFOFO Field Map 27.doc for details.

The differences:

Field Map A: Generated with the Ole Miss Biot-Savart field generator. Each coil has 50 x 11 = 550 1 cm x 1 cm turns. Radial layers of turns are centered at R = 0.775, 0.785 ... 0.875 meters from the coil axis. 300 current segments per turn. The Biot-Savart field is relatively straightforward to code; only location points must be transformed from tilted coil coordinates to ring coordinates. The fields appear directly in ring coordinates (which is the form our GEANT simulation wants them), and inasmuch as the ring coordinate system is cartesian, it is straightforward to compute curls and divergences for ensuring that the field conforms to Maxwell's Equations. The primary disadvantage of the Biot-Savart field generator is that is slow. It would takes weeks of computing time to generate an adequately fine-grained field map grid, for example. However, it is perfectly practical to apply it to small ensembles (thousands) of points, and to check the performance of other field map generators against it.

Field Map B: Generated with the Ole Miss coordinate rotations, but finding the field for each coil using 11 current sheets (radii as above) and a local version of BSHEET / ELLTHREE / RC modified from one provided to us by Rick Fernow. BSHEET is a much more difficult routine to understand; there are behaviors around certain "magic regions" in a coil that have to be handled with some care. (Magic regions include the sheet center and any point on the axial extension of the sheet cylinder, even well beyond the sheet itself.) There are fairly complex issues of numeric precision and convergence criteria that must be looked after. Assessing the accuracy of the field calculated is not entirely straightforward. However, BSHEET is very much faster than the Biot-Savart field

generator; it can compute an eleven-sheets-per-coil field grid for a cell (1 cm lattice) in just a few hours. That fact has made the 11-sheet BSHEET field our standard "workhorse" field generator. A field map grid based on it is used in our GEANT simulations, and it is the generator of choice whenever more than a few thousand field points must be calculated. By comparing Field Map B with Field Map A, we can determine whether the "tuning" of BSHEET to the task at hand has been successful.

Field Map C: Generated by interpolation within a 1 cm x 1 cm x 1 cm grid of points, where the field at each grid point was computed a version of BSHEET very similar (but not quite identical) to that used to generate Field Map B. Now mostly of historical interest, this field map is included here only because it remains the field map used by GEANT, but will be replaced within a few days by . . .

Field Map D: Generated by interpolation within a 1 cm x 1 cm x 1 cm grid of points, as in Field Map C. However in this case the version of BSHEET used to generate the fields at the grid points is exactly the same as that used in Field Map B. The field at an arbitrary point is interpolated using multi-dimensional linear interpolator FINT. In GEANT, the field grid for one cell (1/12 of the ring) is loaded into memory from a binary disk file by routine ***LoadField*** (in a few seconds). As particles are stepped around the ring, the needed field values are computed by routine ***FindFieldAnywhere***, which rotates the particle location into an equivalent position in Cell 1, calls ***FindField*** to compute the field there, and then rotates the field components back to the cell where the particle actually is. ***FindFieldAnywhere*** is very fast, about 30 times faster than calling BSHEET directly, but since the grid is fairly coarse and the interpolator is only linear, the question naturally arises as to whether Field Map D is sufficiently accurate. By comparing Field Map B with Field Map D at a suitable sample of test points, we can answer that question.

Field Map E: Identical to Field Map B except that each coil is represented by only three current sheets instead of the eleven used in Field Map B. By comparing Field Maps B and E and their respective discrepancies from Field Map A (Biot-Savart) we can assess whether it is appropriate to save some execution time, when generating large field grids, by reducing the number of sheets per coil.

Field Map F: Identical to Field Map B (and E) except that each coil is represented by 22 current sheets instead of the eleven used in Field Map B. It is appropriate to wonder whether the field map is actually converging as the sheet count goes up. Comparing Field Maps B, E and F helps answer that question.

Scaling the Fields

Each field may be viewed as a ***field shape*** times a ***field scale***. In a working ring, if the field ***scale*** were found to be slightly different than expectations, one would fix this by increasing or decreasing the coil currents. (Remember that there is no iron in this design.) Finding that the fields had a ***shape*** consequentially different from expectations might be

much harder to fix. It might require anything from realigning the coils to introducing correction elements to facing catastrophe.

All field generating procedures discussed here are approximations. In the Biot-Savart generator, the simulated turns are larger than real turns, and the circular arcs of the coils are approximated by many short straight segments. In the BSHEET generator, there are fewer sheets per coil than there would be layers of winding, and the calculation of the elliptic integrals is done iteratively. Hence it should not be surprising that there are small differences between the fields generated by the two methods and their variants. To the degree that these differences are only small differences of field ***scale***, they may be ignored. To the degree that they are differences of field ***shape***, they are potentially more serious.

In the comparisons that follow, I have not hesitated to apply a scaling correction whenever it is helpful to minimizing the discrepancy between two field maps. What remains is the ***field shape discrepancy***, which is what we care about. However, I emphasize that the scaling corrections applied are very small; the largest one used below is 1.00015. It is a measure of the substantial consilience of the various field generating methods that adjustments this small remove discrepancies that otherwise would completely dominate those relating to field shape.

If a field is scaled, then all three components must be scaled by the same amount.

How Good is Good Enough?

To what accuracy must the fields be known? At this point, the quick answer is "well enough so that remaining errors in the field map do not materially affect the performance measures of competing ring designs". Yes, but what does that mean in terms of actual errors in the field? Is "no more than a 1% error at 95% of the points and no more than 3% error at any point" good enough? Should we strive for errors ten times smaller? One hundred times smaller?

At some point in time, if a cooling ring is ever built, there will have to be field sensitivity studies done. Magnet builders cannot wind coils or position coils to perfect accuracy, and if the cooling ability of the ring is jeopardized by minute variations in the coils, comparable to those expected from imperfect winding technology, surveying errors or temperature fluctuations, then the ring is too fragile to be built, regardless of its performance in an ideal world. The same kind of sensitivity studies will shed light on the accuracy required of the simulation. But at the moment, they do not exist, as far as I know. Hence for now it is reasonable to hold the simulated fields to a high standard, something like "accurate enough so that, lacking sensitivity studies, there seems to be a vanishingly small likelihood that field map errors will be a major contributor to uncertainties in the assessment of the ring's performance." I have somewhat arbitrarily taken this point to be per-field-component standard deviation of 0.0001 tesla (1 gauss) for a sample of points randomly distributed over the active (particle-accessible) volume of the ring.

I am willing to be argued with about this; I imagine (perhaps wrongly) that most people will consider this too stringent a requirement, and its pursuit a waste of time. However, as we shall see below, we are probably already at or near that level, without having done anything terribly heroic. In any case, if the project (as we hope) moves from conceptual design toward engineering, we shall have to place this matter on firmer footing than considered opinion, even the opinions of those who have considered such matters for a lot longer than I have.

Inputs and Outputs of the Comparison Process

All code is written in Fortran and is located in C:\Magnet and its subdirectories. All data are written in C:\Magnets and its subdirectories. Most data are in the form of "vanilla text files" but field grids are converted to binary files.

Project ***DefineTestPoints*** generates a data file of standard test points. The standard test point file is ***C:/Magnets/StandardTestPoints/StdTestPoints1.txt*** containing locations of 10,000 test points. The first 5000 test points are used in the comparisons that follow. For each point, the location of the point is given in Ring (cartesian) coordinates X, Y, Z, and in Accelerator coordinates S, b, X' and Y'. (S and b are redundant but it's convenient to have both.) There is one line of text per test point.

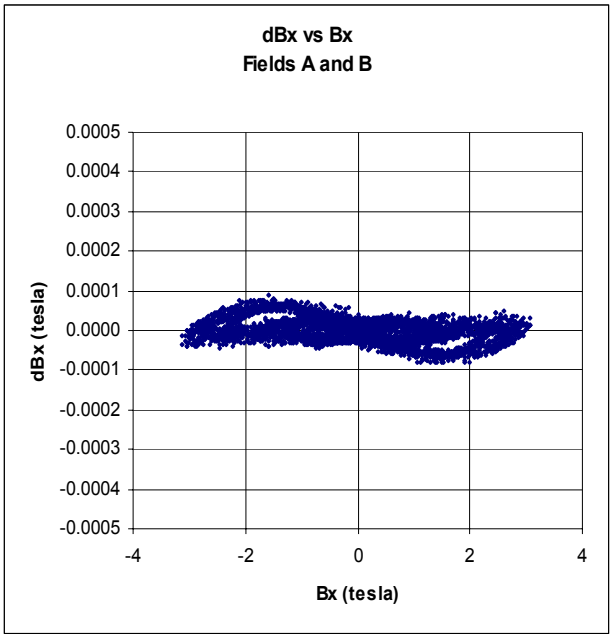
All field-generating program read the standard test point file, compute the field at each point, append the field components to the input line, and write it to an output file. At Ole Miss the fields are in Ring Coordinates and units of tesla, but other coordinates and scales may also be used; the field components are converted to Ring Coordinates before being submitted to the comparison program.

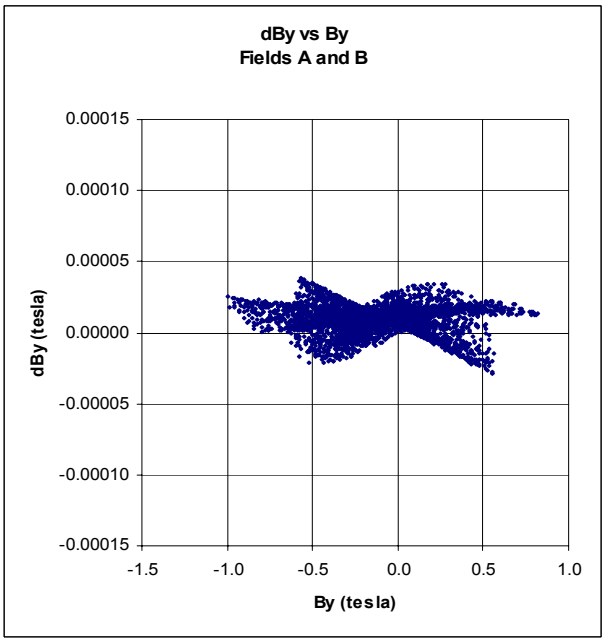
The results of test runs of lasting interest are given a letter (as in A-F above) and stored in ***C:/Magnets/StandardTestPoints/TestFieldReportA.txt*** etc. In some cases there is an accompanying ***TestFieldDiagA.txt*** with certain diagnostic information pertinent to the reported results.

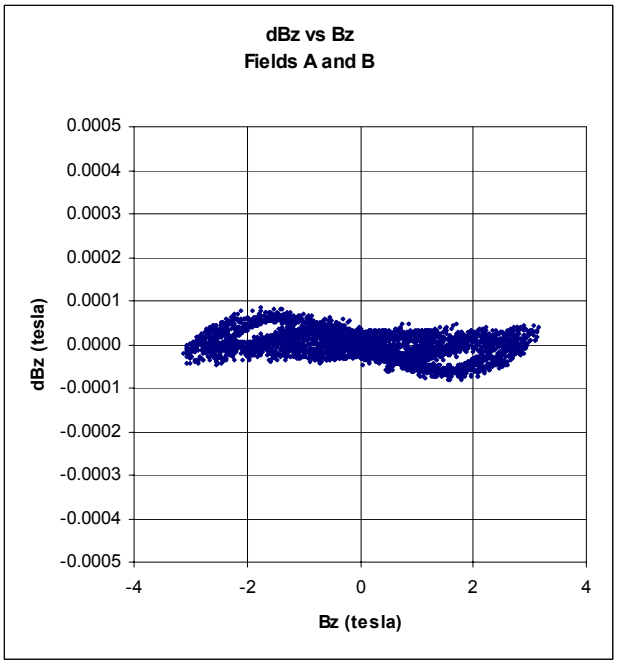
The comparison program is run on pairs of results files. The output of the comparison file is a ***report*** which contains histograms of unscaled field component discrepancies and a ***diagnostic*** file containing point-by-point location / field value / field differences, one line for each point. The diagnostic file (which has proven to be the more useful of the two) is imported into a standard Excel worksheet, where interactive scaling is performed if needed, discrepancy scatterplots are produced, and a few basic statistics are computed.

I have found it most useful to characterize the discrepancies component-by-component in the form of scatterplots, such as BX difference vs BX. This makes it easy to distinguish scale-dependent discrepancies (BXdif correlated with BX) from others, and to ferret out any "magic regions" in which the discrepancies are unusually high. As the overall measure of discrepancy I use the standard deviation of BXdif for the whole ensemble of test points.

The Biot-Savart Field A vs the Eleven-Sheet BSHEET Field B



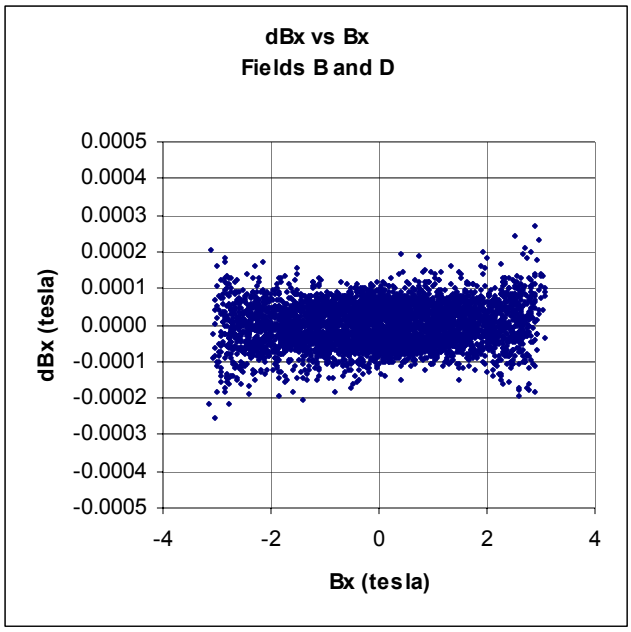


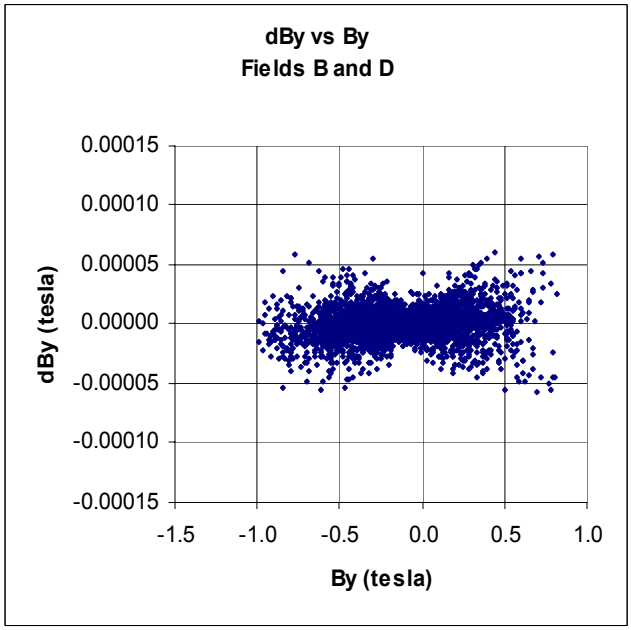


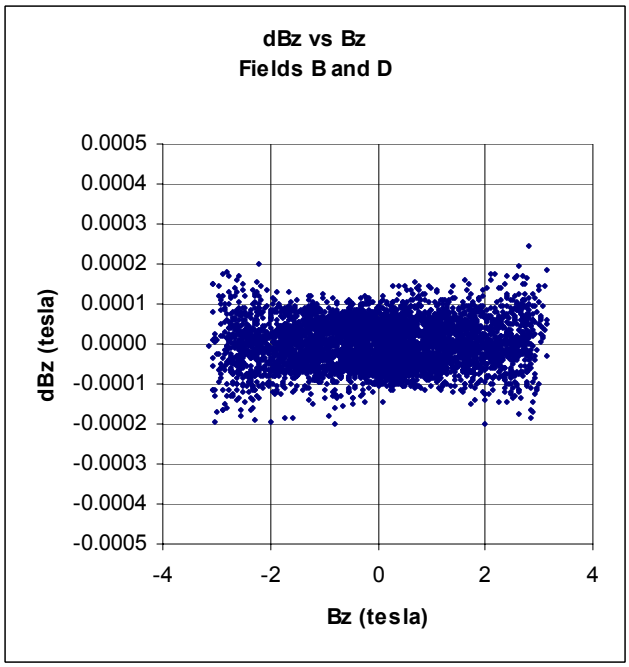
0.0000285 StDev dBx
0.0000092 StDev dBy
0.0000287 StDev dBz

The scaling factor is 1.00011. There is some remaining structure to the field differences, but the discrepancies are very low -- these two very different methods of generating the field for points around the entire ring, neither exact, nonetheless give highly consistent results.

The Eleven-Sheet BSHEET Field B vs the Interpolated Field D







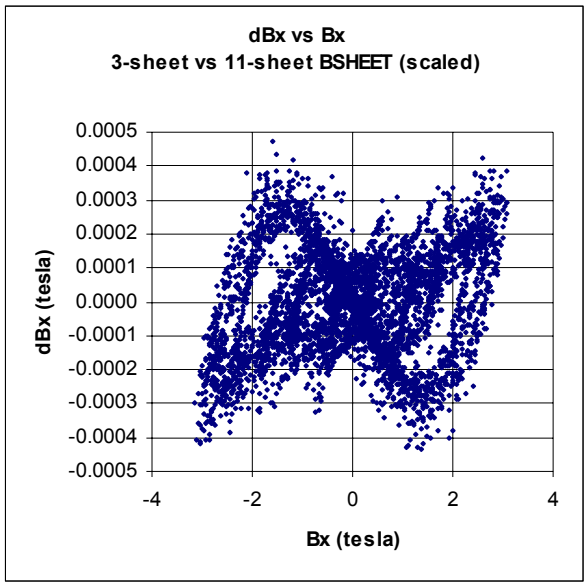
Bx1-SBx2
-0.00000091 Average
0.00005770 Std Dev
0.00013577 FWHM

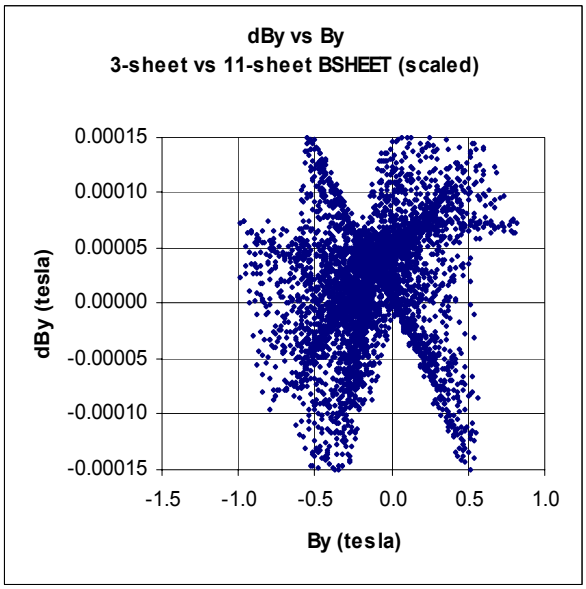
By1-SBy2
-0.00000026 Average
0.00001161 Std Dev
0.00002733 FWHM

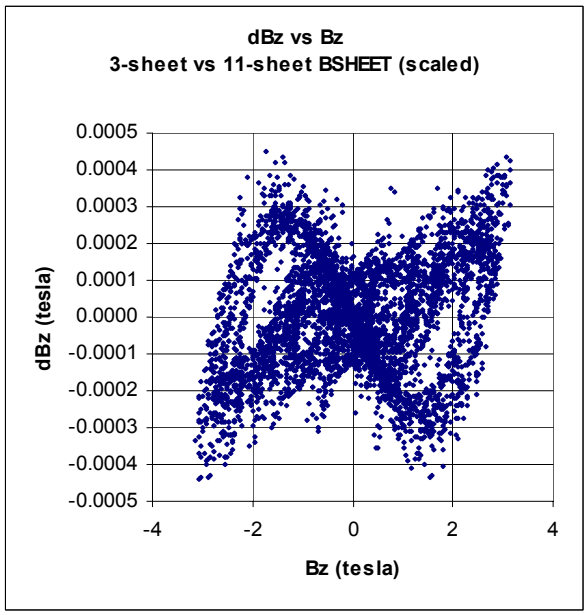
Bz1-SBz2
0.00000037 Average
0.00005757 Std Dev
0.00013547 FWHM

How does the field interpolated from the BSHEET-derived grid compare to the BSHEET field itself? Again, the results are very encouraging; very little accuracy is lost by resorting to the (much faster) interpolation-in-grid. The memory required to hold the grid is substantial, however.

The Three-sheet BSHEET Field vs the Eleven-sheet BSHEET Field







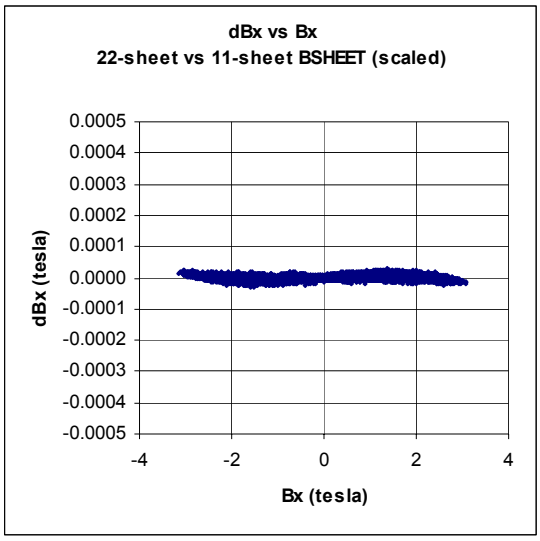
Bx1-SBx2
-0.00000536 Average
0.00015235 Std Dev
0.00035847 FWHM

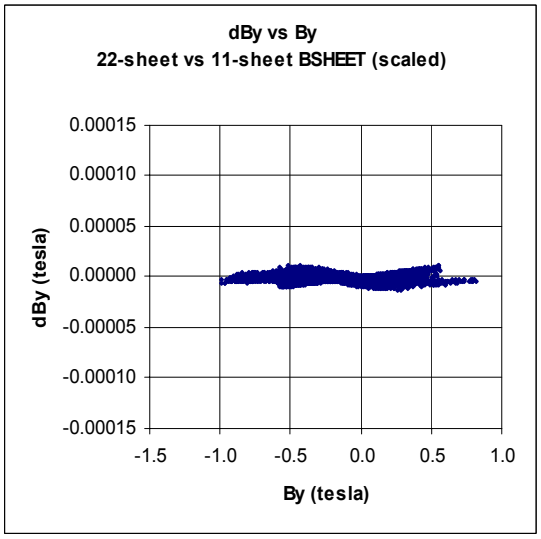
By1-SBy2
0.00001749 Average
0.00005715 Std Dev
0.00013447 FWHM

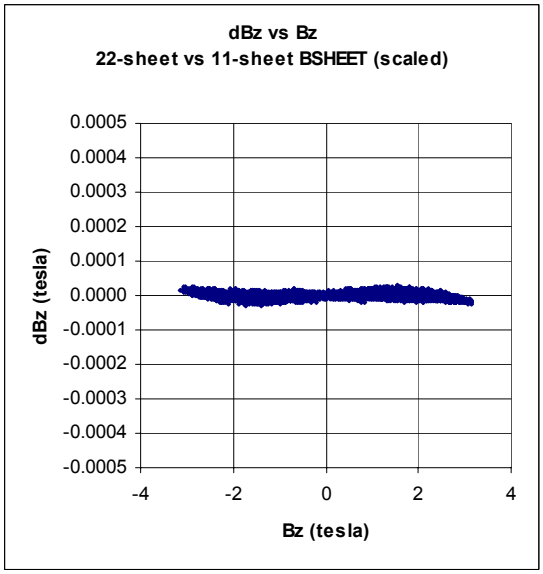
Bz1-SBz2
0.00000097 Average
0.00015506 Std Dev
0.00036487 FWHM

If we assume that the 11-sheet field is a lot closer to the real field than is the 3-sheet field, then the 3-sheet field is not really an acceptable approximation (given the criteria for acceptability stated above). In our case, generating a grid with an 11-sheet approximation takes only about 2 hours longer than generating with a 3-sheet approximation. For this comparison the scaling factor is 1.00015.

The Eleven-sheet BSHEET Field vs a Twenty-two Sheet Field







Bx1-SBx2
0.00000032 Average
0.00000905 Std Dev
0.00002128 FWHM

By1-SBy2
-0.00000118 Average
0.00000339 Std Dev
0.00000798 FWHM

Bz1-SBz2
-0.00000004 Average
0.00000917 Std Dev
0.00002158 FWHM

As expected, the 11-sheet field and the 22-sheet field are very close to each other; maybe we are even converging to the Truth. The scale factor here is 0.999990.

Next Steps

A second dataset is being prepared which has 1000 primary test points scattered around the ring, and for each primary, six nearby secondaries which enable the taking of derivatives that go into calculating the curls and divergence of the field. A very similar test has already been done to our fields A and B, but the new scheme will formalize the process in a way that makes it easy to check other fields as well.

Perhaps the most important thing to do is to perform the same tests, using the same ensemble of test points, on fields generated elsewhere, e.g. BNL. This is a very powerful consistency check, especially because BNL and Ole Miss have developed the transforms used to place and rotate coils almost completely independently That is one area where mistakes have been found in the past and problems could still be lurking. Such consistency checks as we have performed have been done mostly for small ensembles of points taken along the Nominal Beam Trajectory or the Coil Placement Circle. The tests being reported here are far more rigorous.

I understand from Rick that he is reworking a parameterization of his field map, an alternative method to interpolating within a field map grid. When that task has been completed, I would welcome the opportunity to plug that field into the comparison process as well. We are of course happy to participate in comparison processes that other devise, or to code and carry out comparison protocols suggested by others.

How sensitive is a ring's cooling performance to details of the magnetic field map? It is important to answer that question eventually, and perhaps not premature to start doing some serious work on it now. As soon as we have identified the parameters of a satisfactory (not necessarily optimal) cooling ring, it is fairly straightforward to alter the coil configurations and corresponding field maps and test the impact on the cooling effectiveness. What kind of alterations to the field are important to make? Presumably not random fluctuations on a point-by-point basis; these will be mostly averaged out as the particle passes thousands of field points. It is larger-scale alterations -- changes analogous to moving a coil or changing a current -- that are of the most interest.

References

[1] J. S. Berg, R.C. Fernow, and R. B. Palmer,
"An alternating solenoid focused ionization cooling ring,"
http://www-mucool.fnal.gov/mcnotes/public/pdf/muc0239/muc0239.pdf,
March 2002.

[2] M. M. Alsharo'a et al., "Status of Neutrino Factory and Muon Collider Research and Development and Future Plans," hep-ex/0207031.

[3] R. C. Fernow and J. C. Gallardo,
"Calculation of RFOFO fields using the off-axis expansion in ICOOL,"
http://www-mucool.fnal.gov/mcnotes/public/pdf/muc0268/muc0268.pdf,
January 2003.