BNL-101499-2014-TECH
AD/AP 9;BNL-101499-2013-IR

Comparison of accelerator codes for a RHIC lattice

J. Milutinovic

May 1988

Collider Accelerator Department
Brookhaven National Laboratory

U.S. Department of Energy
USDOE Office of Science (SC)

DISCLAIMER

ACCELERATOR DEVELOPMENT DEPARTMENT

BROOKHAVEN NATIONAL LABORATORY
Associated Universities, Inc.
Upton, NY 11973

Accelerator Physics Technical Note No. 9

COMPARISON OF ACCELERATOR CODES FOR A RHIC LATTICE

J. Milutinovic and A.G. Ruggiero

May 1988

Abstract

We present the results of comparisons of performances of several tracking or/and analysis codes, available in the AP Division at BNL, on a RHIC lattice. The basic purpose of this program was to assess reliability and accuracy of these codes, i.e. to determine the so-called "error bars" for the predicted values of tunes and other lattice functions as a minimum and, if possible, to discover potential difficulties with underlying physical models in these codes, inadequate algorithms, residual bugs and the like. Not only we have been able to determine the error bars, which for instance for the tunes at dp/p = +1% are $\Delta\nu_x = 0.002735$, $\Delta\nu_y = 0.001010$, but also our program has brought about improvements of several codes. Things that we now understand better, as a result of this program, are equations of motion in the bend as well as its edge focusing effects. It is worthwhile to mention that all of the codes exposed one kind of weakness or another. In most cases, corrective actions have been taken by us or our collaborators. We feel that the benefits of this program justify further work, especially in view of the circumstances that there are still many aspects of these codes which are poorly understood and, furthermore, some of the deficiencies have not been eliminated yet.

Acknowledgement

The results reported in this note are more a consequence of collective efforts than just efforts of two individuals. Here are given the names of various collaborators, arranged in alphabetic order, who participated in this program. They are: G. F. Dell, E. Forest (LBL), S. Y. Lee, G. Parzen and S. Tepikian. These individuals ran the codes in which they are more proficient than the authors of this report and then patiently ran them again and again to accommodate changes that were requested for various reasons. In addition, most of them actively participated in various discussions relevant for the program or collaborated in other manners. Without their abundant help the completion of this program would not have been possible. The authors would like to express their sincere thanks to all of them. Finally, they would like to thank E. Courant, a non-participant in this program, who nevertheless provided valuable help in terms of various suggestions, discussions and advising.

Introduction

Computer programs for particle tracking and various lattice analyses have been around for quite some time. Their importance in accelerator physics, for accelerator design and improvements, cannot be overemphasized. Their predictions have proved to be an invaluable guide not only during the process of lattice designing, but also even later during the subsequent processes of better understanding, improving and upgrading of an accelerator. However, such computer programs are characterized by a variety of serious limitations.

First and foremost, these programs do not handle computation and analysis on an accelerator; rather they do all this on a model of an accelerator. Therefore, they are limited by the validity of models that contemporary accelerator theory can offer to account for what has been observed in practice. A good example of a model limitation is treating of fringe field effects for the bend, in the so-called hard-edge approximation.

Second and almost equally important, these programs are not capable of handling even the accepted models exactly, with the exception of TEAPOT which can handle exactly any lattice fully reduced to a series of drifts and thin lenses.[7] In particle tracking, for instance, generally the most accurate approach would be to take the exact equations of motion, dictated by the adopted physical model, and integrate them numerically through the lattice. Then the better numerical integration methods and the better computer used, the closer the results would be to the exact, i.e. to those precisely dictated by the underlying physical model in question. However, barring some very special cases, this is completely ruled out by computational demands imposed on a computer. Consequently, one has to resort to various approximations, even within approximate validity of underlying physical models, and as a result all these codes are replete with various approximations. An example of such kind of approximation is the kick approximation for the sextupole. We know the exact Hamiltonian for an ideal sextupole, we can derive the exact equations of motion, but we nevertheless have to resort to an approximation, in this case replacing the actual sextupole by a thin lens element of the same integrated strength.

Given the array of codes on the market, and given the fact that there are hardly such notions as the "best model" or the "best approximation," it is expected that by using different codes one gets different numerical predictions for the same physical quantity. So for each computed quantity, the results will vary over a certain range of values. Knowledge of these ranges, for a specific accelerator, is a valuable guide to the user in his judgement as to how much confidence to place in the results of a particular run with a particular code on that lattice. Such knowledge can be acquired only by running several codes with the same lattice input conditions and by comparing the results. With this in mind, we proceeded with the program we describe in this note.

There is also an additional benefit arising from these comparison runs. An out-of-range result is a clear indication of a problem with the code that generated it. It could be an inadequate physical model, an inadequate algorithm or a plain simple bug. Therefore, should this happen, an opportunity to improve the code arises and this is precisely what was occurring during the implementation of this program.

Codes Subject to Comparisons

We have compared several codes which are available at BNL. Some of these are being used almost on daily basis, others have been used only occasionally. In an alphabetic order they are: FASTRAC,[1] MAD,[2] ORBIT,[3] PATRICIA,[4] PATRIS,[5] SYNCH[6] and TEAPOT.[7] We would like to comment briefly on these codes and on the specific versions we used in the testing.

FASTRAC[1] is a tracking/analysis code developed at LBL by E. Forest and B. T. Leemann. The starting code from which it has evolved was RACETRACK[8] of A. Wrulich, then at DESY. It is powerful but very difficult to use, especially in a black-box running manner. It handles linear elements in a formalism of two decoupled 3 x 3 matrices and nonlinear elements in the kick approximation. It can correct a distorted closed orbit by employing an algorithm based on the Fermilab correcting scheme,[9] provided by H. Nishimura who had installed the scheme in his version of RACETRACK. From the transfer map it finds, FASTRAC is able to generate its Lie algebraic representation and to supply the necessary Lie algebraic polynomials to be used by MARYLIE,[10] for many cases where MARYLIE is not able to generate the transfer map by itself (an example: a lattice with multipoles higher than duodecapole).

MAD[2] has been developed by C. F. Iselin from CERN and his collaborators. It is rather a conglomerate of codes, for various purposes, than a single code. We have tested both versions 4.03 and 6.01, with barely noticeable differences. It has been seldom used in the AP Division at BNL, but some other departments of the Laboratory have been using it more frequently.

ORBIT[3] is a tracking/analysis code written from scratch by G. Parzen from BNL. It has the structure and capabilities to some extent similar to those of PATRICIA[4] (see the next paragraph). This means that linear elements are described by similar matrices, without the two degrees of freedom being coupled, while nonlinear elements are simulated by kicks. Little is known about this code beyond scant information provided by its author in private discussions. The code has been used extensively by its author at BNL and it has supplied a vast amount of tracking data for the AGS Booster and RHIC.

PATRICIA[4] is a tracking/analysis code developed by H. Wiedemann. The version used in the AP Division at BNL was obtained from the version made in the fall of 1980 by S. Kheifets and H. Wiedemann. After that, the version has evolved independently under development done by its principal user G. F. Dell. The code has been notoriously known to have some difficulties with finding a closed orbit, but its various versions have been extensively used at several institutions. The version of the code that we tested has delivered a huge amount of tracking results for the SSC, the RHIC and the AGS Booster. It handles both linear and nonlinear elements in a manner conceptually similar to that of FASTRAC, i.e. a decoupled set of two 3 x 3 matrices is used for linear elements and the kick approximation for nonlinear ones.

PATRIS[5] is a tracking/analysis code developed by one of us (A. G. R.), from H. Wiedemann's PATRICIA.[4] The development started in 1975 and has resulted in a completely rewritten code that barely resembles its ancestor PATRICIA. The formalism of the two decoupled 3 x 3 matrices for linear elements has been superseded by a single 7 x 7 matrix formalism, which is capable of describing the coupling between all degrees of freedom. All difficulties with the closed orbit finder have been eliminated. Furthermore, the code includes many powerful features, such as for instance tracking with synchrotron oscillations, statistical treatment of closed orbit distortions in the kick approximation, and realistic closed orbit analysis where, which is almost unique in such type of code, it correctly handles errors associated with thick elements. It can also correct distorted closed orbit, by employing an algorithm based on the Fermilab correcting scheme.[9]

SYNCH[6] is primarily a lattice design code. It was originally conceived by A. Garren from LBL and subsequently developed and improved by Garren and several collaborators, including E. Courant who is in charge of SYNCH here at BNL. The version we tested was the currently defunkt CDC version. It did not incorporate some of the more recently made changes at Berkeley,[11] but we have been assured[12] that they would not affect our results. The code is huge and structurally very different from other codes we tested. Some of its predictions, as for instance edge focusing for an off-momentum trajectory, might be the most accurate among all of the codes [F1]. Here at BNL its role has been instrumental in designing the lattices for the AGS Booster and the RHIC.

TEAPOT[7] is an exact thin-lens code developed at the SSC CDG by L. Schachinger and R. Talman. By "exact" we mean that once the actual lattice has been approximated by a completely thin-lens model of the lattice, no further approximations are being made. A serious conceptual deficiency is the code's inability to handle the vertical degree of freedom properly. To be precise, this means that the code can simulate magnet errors in the vertical plane by kicks, but cannot describe a vertical bend as a sandwich of thin lenses. In a planar ring, such as for instance the RHIC lattice used for the comparisons, it is immaterial. The version we used has been developed from its original by E. Forest. It includes many fancy analytic tools, but unfortunately the code's ability to supply the results of linear optics analysis has been eliminated by its principal user, interested in developing some other abilities of the code, such as for instance generating transfer maps for MARYLIE. For that reason, TEAPOT was subject only to limited comparisons with the other codes.

Three different computer systems were used to run these codes. FASTRAC, PATRICIA, PATRIS and TEAPOT were run on the CRAY X-MP, ORBIT and SYNCH on the CDC which is not in operation any more, while both versions of MAD were run on the VAX.

The codes we tested were run by several people. FASTRAC was run by J. Milutinovic. MAD was handled by S. Tepikian. ORBIT was run by G. Parzen. PATRICIA was run by G. F. Dell and later also by J. Milutinovic. PATRIS was run by A. G. Ruggiero and J. Milutinovic. SYNCH was run by S. Y. Lee. Finally, TEAPOT was run and analyzed by E. Forest.

Choice of Lattice for Comparison

Although one could have contemplated creating special kinds of inputs for the purpose of code testing, this would not have been a very productive approach. The primary purpose of the whole program was not code development and safeguarding them against any contingency one could envision; the purpose was to test how these codes work in the environment they have been, are being, and will be used, i.e. how well they fare on the lattices they are supposed to handle. Therefore the natural choice was to run these codes either on an AGS Booster lattice or on a RHIC lattice.

The Booster was mentioned in an independent and rather rough comparison attempt by Z. Parsa.[13] However, we did not contemplate resorting to the Booster for a simple reason. Its lattice is, in our opinion, simply too "smooth" to provide the framework for a realistic code comparison. This is not only obvious by inspection of the ring,[14] which contains only sector magnets that all bend inward and is probably too large to allow curvature-related nonlinearities to come into play, this has also been amply confirmed by comparing two closed orbit calculations which were done independently of, but about the same time as, this code comparison program. By contrast, a RHIC[15] lattice is sufficiently "coarse" to cause even slightly different algorithms and procedures employed by these codes to display noticeable differences.

The RHIC lattice (Fig. 1)[15] we used was the currently considered lattice at the outset of this program. It was later abandoned in favor of another lattice, with sector bends instead of rectangular ones, but we adhered to the original lattice which was better for the purpose of testing, even though it turned obsolete in the midst of our program implementation. It was the so-called ARH3NEW lattice, with all of its dipoles rectangulars, and with symmetric entrance and exit angles, except in the case of the two bends (BC1) on each side of each intersection (Fig. 2), which are unlike other bends shared by the two beams.

What the Codes Supply and What Was Compared

The codes we have tested are able to perform a wide variety of calculations. Using one or more of these codes, one can for instance track particles without or with synchrotron oscillations, perform a closed orbit analysis, obtain a Lie algebraic representation of the transfer map, perform linear optics calculation, (re)design a lattice and many other things. However, by and large these capabilities are not shared among all of these codes. To test them, it would mean comparing selected subgroups of these codes, with specific groups being formed according to specific shared capabilities. Time limitations have, of course, prevented us from testing everything. Therefore, we have selected a computational feature common to all of these codes (except to some extent to our version of TEAPOT). That was calculation of various lattice functions, such as for instance tunes, closed orbit, beta functions at various locations, and momentum dispersion and/or closed orbit function at the same locations as beta functions. The locations were the following three points of interest: the middle of an inner arc, the first following in-to-out crossing (Fig.2) point and the next crossing point which was out-to-in. In the subsequent text, these three points will be named SYM, CRIO and CROI, respectively.

The quantities were compared as printed, i.e. the comparison was in fact limited by the code that provided the smallest number of significant digits for the compared quantity. Except in a single instance of PATRIS's closed orbit printout which, because of too few displayed digits, deceptively indicated a nonexistent problem, we did not try to lengthen the printouts for the sake of comparison. Not all the codes supplied everything we wanted to compare. If a particular code did not furnish a particular quantity of interest, it will be indicated at the appropriate place in the text.

Finally, we should mention that most codes evaluate transfer matrices with respect to the commonly used noncanonical coordinates (X, X', Y, Y'). FASTRAC and (presumably) MAD, on the other hand, evaluate the transfer matrix with respect to the canonical coordinates (X, P_x, Y, P_y). Therefore, their β-functions have to be divided by $(1 + \delta)$ to conform to the accepted standards. This important fact, however, is not mentioned in the code manuals.

Some Specific Problems Revealed by Comparisons

As mentioned, the primary goal of this program was not to hunt specifically for problems and bugs, but to determine the ranges of predictions for each computed quantity instead. However, a possibility of discovering some problems was realistically admitted, even though we had had some advanced knowledge of only one problem, that of PATRICIA's deficient closed orbit finder, to be specific. In the course of testing, all of the codes we compared exposed one weakness or another. We will explain the known causes of these weaknesses in the next section. Here we mention them in an alphabetic order.

FASTRAC revealed a bug in its edge focusing routine. Our attention to this bug was accidentally drawn by the observed (unrelated) discrepancy between the total length of the machine computed by the code and the actual length of the machine. Had the code supplied a correct length of the machine, the bug would not have shown up at all on a machine as big as the RHIC. However, on a small machine, such as for instance some light sources, this bug might have had even the potential of preventing the code from finding a periodic solution. This problem was easily fixed, but even then FASTRAC (and all other codes based on quadratically approximated Hamiltonians) should not be trusted to handle a very small ring.

MAD revealed a discrepancy in predicting tune dependence with momentum deviation δ, when compared with the rest of the codes we tested (Fig. U1 and U2.). The difference between MAD's prediction and that of any other code is manifestly quadratic in δ, in both planes. On the other hand, with the exception of PATRIS, the much smaller differences among other codes are all linear in δ. Both versions of MAD, i.e. 4.03 and 6.01, displayed the same incorrect tune versus momentum deviation behavior.

ORBIT displayed a discrepancy in predicting tune values even for on-momentum, i.e. $\delta = 0$ values, in both planes (Fig. U1 and U2). However, for off-momentum values the residual discrepancy is linear in δ, once the on-momentum discrepancy is subtracted out.

PATRICIA did not reveal any further serious difficulties, aside from its notorious problem with the closed orbit finder (Fig. U5). A minor nuisance was its inability to suppress the chromaticity sextupole fitting, hence the code entered the comparison with its own set of slightly different values of the sextupole strengths.

PATRIS initially displayed a big discrepancy in predicting the tune dependence on momentum deviation (Fig. U1 and U2). Like MAD's discrepancy, this one was also quadratic in δ, in both planes. But unlike the MAD case this one was quickly understood and corrected. Other minor problems with interpolation formulas for off-momentum values of closed orbit function and beta functions at arbitrary points in the lattice did not call for urgent corrective measures, since the problem can be simply dealt with by starting at another point in the lattice. The sources of troubles and the corrective measures will be discussed in the next chapter.

SYNCH strictly speaking did not display problems during this set of comparisons, with the exception of small discrepancies in the momentum dispersion when compared to the other codes. However, partly as a result of participating in these comparisons and related discussions, E. Forest[11] got interested in some aspects of the code, found some bugs and a conceptual error in the quadrupole algorithm, and later made some improvements together with A. Garren. These bugs, however, did not affect the validity of our comparison program. Later it was also found that SYNCH fails in handling very thin and very strong quads, but this is well understood[17] and did not affect our program.

TEAPOT did not display much trouble during these tests, except that we were quickly halted in our attempt to split the bend into more that 6 thin lenses. Beyond 6 the code crashed due to underflow/overflow condition. Also, to be meticulous, it was noticed that its closed orbit finder did not predict the closed orbit to be zero for the on-momentum case of a lattice without magnet errors. Despite the stringent exiting criterion for the iterating loop of its closed orbit finder, based on the Newton method, and our attempt with sufficiently many iterations, the residual closed orbit displacement was still $X \sim 10^{-7}$, which does not present any problem in practice, but should not have been so high on a machine such as CRAY X-MP.

Explanation of Some Discrepancies. Corrective Measures

Here we will try to account at least for some discrepancies we have found in the course of our code testing venture. We have not yet examined the problem with MAD's prediction of tune versus momentum dependence. Nor have we attempted to understand the momentum dispersion discrepancy (linear in δ) between SYNCH and the rest of the codes we have tested. However, we can talk about FASTRAC, PATRICIA and PATRIS as a result of our own understanding of how these codes work. We can also say a few words about ORBIT and TEAPOT, primarily on the basis of information we have received from the two collaborators who have actively helped us in carrying out this program (E. Forest and G. Parzen). Some residual subtleties will be dealt with in Appendix.

In FASTRAC we found a small bug in the edge focusing routine. In the current version of the code, this routine was never activated for a rectangular bend, which was handled separately, as a block consisting of a combined edge-body-edge routine, with no bugs introduced. Therefore, the only relevance was for magnets with nonsymmetric entry and exit angles, such as for instance BC1 in the RHIC. This error had apparently survived from the old times of A. Wrulich's RACETRACK. The routine incorrectly used $\tan^{-1}(\epsilon)$ instead of $\tan\epsilon$ and it also added an extra length ℓ to the total length of the machine. This problem was fixed [F2]. We would like to reiterate that even though the problem was quite benignant on a big lattice such as the RHIC, this would not be true on a small machine where the bug might even be able to prevent the code from finding a periodic solution. One should also remark that the edge focusing routine does not get the angle ϵ on the input, it is being evaluated by the code as $\ell/2\rho$, which would match symmetric entry/exit configurations on a rectangular bend! Therefore, the user has to enter this expression with a fake length (or bending radius ρ) of the element the edge focusing is attached to, in order to adjust the angle correctly, or has to redesign the code as an alternative. For BC1, for instance, we had to replace ℓ by 2ℓ to get the correct angle. Also, ρ is here the on-momentum bending radius. We fixed the small bug, but did not consider worthwhile to spend time and effort to reprogram the code.

For MAD, as already mentioned, we have not yet attempted to find the source of manifestly quadratic deviation from the other codes' predictions of the tune dependence on momentum deviation. We nevertheless hope to be able to understand this at some future time. A small consolation to us is the fact that MAD has not been used to any appreciable degree in our Accelerator Physics Division.

ORBIT displayed on-momentum tunes different from the rest of codes (Fig. U1 and U2). We were privatly informed by G. Parzen that the reason for this discrepancy is the fact that the code handles all bends as rectangular magnets, with equal entrance and exit angles. Therefore, it erred on the BC1 magnets, which are rectangular in shape but nevertheless must not be handled in that manner since their entrance and exit angles are not equal. To determine to which extent Parzen's explanation can account for the observed discrepancies, we ran the corrected version of PATRIS (explanations of PATRIS's problems are given in the second paragraph after this one) on a fake lattice, in which we introduced false equal entrance and exit angles on the BC1 magnets. As a result of this trick, the on-momentum tune difference between ORBIT and PATRIS vanished (Fig. U3, U3a, U4, U4a) and the remaining off-momentum difference is small and linear in δ. This is because of slightly different transfer matrices for the bend in PATRIS and ORBIT, and also because of slight differences in the manner the two codes treat the edge focusing for $\delta \neq 0$. These differences are discussed in Appendix. This problem was beyond our control and as far as we know it has not been fixed yet, but one can always "renormalize" the results for the tunes by subtracting the known (another code has to be run in parallel with ORBIT) on-momentum differences.

PATRICIA had been well known for its problems with the closed orbit finder. We attempted to understand why. We have been able to find a bug in the sextupole routine. There, it is necessary to know the displacement of

the particle in order to evaluate the sextupole kick strength. At the sextupole location, it is given by

$$X = (A_{11}\eta + A_{12}\eta' + A_{13})\delta$$

where η is the so-called closed orbit function (denoted in this manner in PATRICIA and PATRIS, but generally denoted as X_{co} in the rest of this note), at the beginning of the lattice, such that at the starting point the closed orbit displacement is given by $X = \eta\delta$, with $\delta = dp/p$. In this expression, we found that the $A_{12}\eta'$ term was missing. This fact had been already known to G. F. Dell, but we were not aware of that. However, once this needed term is properly introduced, the code crashes on the RHIC lattice ARH3NEW for δ exceeding a modest 0.49%, and fails to find a periodic solution for much smaller δ values. Its absence seems to have been vaguely defended by some symmetry arguments, but with its presence the code nevertheless should not have crashed or gone berserk in attempting to find a periodic solution. We tried to understand how the closed orbit finding routine functions but, similarly to G. F. Dell, we got stuck with something totally incomprehensible. After that, one of us (J. M.) did something similar to what the other (A. G. R.) had done many years ago when PATRIS was born, i.e. scrapping PATRICIA's closed orbit finder and installing a brand new closed orbit finder, based on the Newton root finding method [F3] for the four dimensional space of (X, X', Y, Y'). After that PATRICIA's predictions for the closed orbit started agreeing to 4-5 significant digits with PATRIS and FASTRAC. Further improvements in agreement between the three codes were obtained by disabling PATRICIA's compulsory sextupole fitting, by making equal the matrix for the bend in all three codes and by making sure that the edge focusing is being handled in exactly the same manner. After that PATRICIA, PATRIS and FASTRAC agreed to 9-10 significant digits, and this modified PATRICIA was then retested. The results that will be reported in the second part of this note are those obtained after the problem had been fixed. Interestingly enough, the results of computations with the deficient closed orbit finder did not set PATRICIA much apart from the other codes. The tunes went along fine with the majority of codes (Fig. U1 and U2), and some differences in beta functions are mainly seen in the vicinity of $\delta = +1\%$ (Fig. U7 and U8 are such characteristic examples).

PATRIS displayed three discrepancies, two of them fictitious and one real. The two fictitious discrepancies were the closed orbit function in the vicinity of $\delta = 0$ (Fig. U5a), and the momentum dispersion differing significantly from the other codes' predictions for $\delta \neq 0$ values. The former problem was due to insufficient number of printed digits and was dealt with simply by increasing the length of the appropriate printout. The latter problem turned out to be an outcome of comparing apples and oranges. By inspecting the code it was determined that PATRIS's momentum dispersion function was in fact the closed orbit function $X_{co} = X_{fp}/\delta$, the same as in its ancestor PATRICIA, and not the slope $\eta = dX_{fp}(\delta)/d\delta$, present in the other codes. Only different scales used for plotting prevented us from inferring this fact directly by looking at the graphs. To ensure that PATRIS delivers the differential quantity (i.e. the slope) if desired, we have installed the appropriate routines in the code, which now finds η by numerical differentiation of the closed orbit. This has brought the code into a perfect agreement with the remaining codes that do supply η, with the exception of SYNCH. The only real problem that plagued PATRIS was the

discrepancy in predicting tunes versus momentum deviation (Fig. U1 and U2), which was quadratic in δ when compared with the majority of other, mutually agreeing codes. By inspecting the source code, we found that PATRIS did not have part of the necessary momentum dependence built in its bend transfer matrix. That bug had remained from the days of PATRIS's birth. That was fixed up by reintroducing the necessary momentum dependence, based on the solution to the equations of motion given by R. Ruth,[18] who consistently derived them from the Hamiltonian [F4]. In this respect, PATRIS now slightly differs from ORBIT, PATRICIA and old RACETRACK, and is the same as FASTRAC which recently also moved into this direction. These differences will be explained in Appendix. We conclude this paragraph by mentioning that when PATRIS works in a momentum scanning mode it actually computes eta and beta functions at the beginning of the lattice, for the whole range of momenta, while at the same time supplying interpolation formulas to be used by hand if the user is interested in finding momentum dependence of these functions at other lattice locations. Our tests have disproved the reliability of these interpolation formulas, since the code does not come up with correct values of coefficients entering these formulas. However, the problem is minuscule, since the code supplies on-momentum quantities correctly through the whole lattice and the user interested in a momentum scan at a specific lattice location can get reliable off-momentum values simply by shifting the beginning point of observation to the desired location.

As we previously mentioned, SYNCH disagreed to some extent with FASTRAC, MAD and PATRIS in predicting the momentum dispersion function η. This discrepancy, which is linear in δ, is still puzzling us [F5]. The most perplexing aspect of the discrepancy is a sort of accompanying internal inconsistency between predictions of closed orbit functions and momentum dispersion functions, produced either by SYNCH or by the other three codes. To understand this puzzle, the reader should compare Figures C3 and C4. The reader will notice (Fig. C4) that η(SYNCH) > η(Other codes) for all values $\delta > 0$. Therefore, since η is the slope of the curve representing the closed orbit displacement at the observing location, and since the closed orbit displacements given by all of our tested codes are equal at $\delta = 0$, one expects that the coordinate of the fixed point at $\delta = +1\%$ behaves as

$$X_{fp}^{SYNCH}\Big|_{\delta=0.01} = \int_{o}^{0.01} \eta^{SYNCH}(\delta)d\delta > \int_{o}^{0.01} \eta^{Others}(\delta)d\delta = X_{fp}^{Others}\Big|_{\delta=0.01} .$$

In short, one expects that the SYNCH fixed point displacement at $\delta = +1\%$ exceeds that of the other three codes and this also implies that the closed orbit function $X_{co} = X_{fp}/\delta$ of SYNCH, at $\delta = +1\%$, exceeds the other codes' predictions. Figure C3 (and Table C3), however, shows that this is not true, i.e. SYNCH's value for X_{co} is the lowest for $\delta = +1\%$ among all the codes!

Finally, we address TEAPOT and its closed orbit discrepancy ($\sim 10^{-7}$ mm) at $\delta = 0$. We have mentioned that the code employs the Newton method with a stringent exiting criterion and with the number of iterations interactively chosen by the user. The code reaches $\sim 10^{-7}$ in two or three steps and stays there in the subsequent iterations; this is a stable fixed point of the

code's closed orbit finder, at $\delta = 0$. Apparently, at $\delta = 0$ and on a computer such as CRAY X-MP, this cannot be a result of a truncation error committed in executing the Newton fixed point searching algorithm. E. Forest has offered an explanation of this peculiar behavior. Due to finite precision of lattice representation in the computer, the ring fails to close on itself by the amount $\sim 10^{-7}$, and the residual closed orbit displacement ($\sim 10^{-7}$) at $\delta = 0$ is simply a measure of the code's failure to close the lattice better.

Reliability of Code Predictions. Error Bars after the Implementation of Corrective Measures

As already mentioned, there are two kinds of quantities we compared. First group is composed of the so-called global characteristics of the machine, such as for instance on-momentum tunes, transition gamma, on-momentum path length over the ring, bare and corrected chromaticities, etc. We present these general characteristics in Table G1. Some of them, like on-momentum tunes on the one hand, tell us about the ability of the code to make realistic predictions, in addition to serving as code and/or lattice debugging tools. On the other hand, a quantity like the total on-momentum path length serves solely as a debugging tool since we know that it must be equal to the physical length of the machine. Unfortunately, not all the codes supplied all these quantities, thus many places in the table remain filled with asterisks. For the available quantities, we notice excellent agreements, with the exception of ORBIT's on-momentum tunes, whose deviation from the other codes' predictions is, however, well understood.[20]

Second group of items that we have compared consists of various quantities whose momentum dependence is being evaluated and compared. With the exception of tunes, whose momentum dependence we include in this group, all the quantities are local, i.e. they explicitly depend on the location in the lattice where they are being observed. As mentioned in introductory chapters, we have selected three locations. The first one is where we begin with our code lattice inputs, i.e. the middle of an inner arc, labeled as SYM. All quantities are being evaluated here. They are the closed orbit function $X_{co} = X_{fp}/\delta$, the momentum dispersion function $\eta = dX_{fp}(\delta)/d\delta$, if available, and finally the beta functions in both planes. The other two locations are the first two crossings that follow the middle of the arc downstream. They are labeled as CRIO and CROI (for CRossing In-Out and CRossing Out-In, respectively). The closed orbit function has not been evaluated there, while the remaining available quantities have. Note that ORBIT and PATRICIA did not supply the momentum dispersion function, nor had PATRIS been doing it until we accommodated an extra algorithm for that purpose. If a particular quantity was not available, the fact is indicated in the appropriate column. With this in mind, we move on to present the numerical results of our comparisons.

First we present the error bars in Table G2. They are given in absolute quantities for $\delta = -1\%$ and $\delta = +1\%$, and are evaluated as $|Q_{max} - Q_{min}|$ for the particular quantity Q. They refer to the results obtained after the corrective measures had been taken. There are two important exceptions to this rule. The first one refers to ORBIT's tune dependence on momentum deviation, where we "renormalized" the quantities, by subtracting

out the on-momentum discrepancy, which is well understood, and used this renormalized tune values when determining the appropriate error bars. The second one refers to the same quantities produced by MAD, which were simply dismissed from the tune error bar determination, being plainly wrong.

After this, we display momentum dependence of the various quantities, that we have evaluated and compared, in Tables C1 through C12. Momentum deviation δ in all cases runs from -1% to +1%. Note that for SYNCH and ORBIT the only available values are for -1.0, -0.5, 0.0, +0.5 and +1.0 per cent, whereas the other codes cover the interval [-1.0, +1.0] more densely, but skip -0.5% and +0.5%. This is a simple consequence of the fact that the two codes were run by people other than us, and reflect their taste as to how densely to perform a momentum scan. Specific comments, if necessary, are given in the section called Tables.

As the final step, we display graphs of momentum dependence of the same quantities as those given in Tables C1 through C12. The graphs are denoted in the same manner, i.e. by Figure C1 through C12. As in the case of tables, "C" here stands for "Corrected." Specific comments, if necessary, are given in the section called Figure Captions.

Appendix

Here we discuss some of the facts that have given rise to discrepancies between code predictions. To be specific, by discrepancies we here primarily mean the differences in predicting on and off-momentum tunes. Also, we can discuss here only things that we understand well. There will be no account for the problem with the off-momentum tunes given by MAD, or less dramatic differences in prediction of η between SYNCH and most of other codes (linear in δ). The two well-understood code ingredients that had a direct impact on the tune versus momentum behavior are the transfer matrix for the bend and the edge focusing matrix.

In the linearized theory, the (X, X′) change through a magnet is given by

$$\begin{pmatrix} X \\ X' \end{pmatrix}_{fin} = M \begin{pmatrix} X \\ X' \end{pmatrix}_{in} + D\delta , \qquad (A.1)$$

where M is the so-called transfer matrix, while D is the so-called dispersion vector. Alternatively, in a 3 x 3 matrix formalism for one degree of freedom this is equivalently given by

$$\begin{pmatrix} X \\ X' \\ \delta \end{pmatrix}_{fin} = \begin{pmatrix} M & D \\ 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ X' \\ \delta \end{pmatrix}_{in} . \qquad (A.2)$$

The transfer matrix for the bend is obtained by integrating the linearized equations of motion. Once the bug from PATRIS has been removed, there are three different expressions that are present in the codes we compare (except, of course, SYNCH and TEAPOT). The oldest is probably the matrix derived from the equation

$$X'' + \left(\frac{K}{1+\delta} - \frac{1}{\rho_o^2 (1+\delta)^2} \right) X = \frac{1}{\rho_o (1+\delta)} \delta . \qquad (A.3)$$

This is given in many standard references, for instance in K. Steffen's book.[21] Its derivation seems to have been based on starting from the equation of motion for the on-momentum case, i.e. (A.3) with $\delta = 0$, $\rho = \rho_o$ and then by replacing $\rho_o \rightarrow \rho = \rho_o (1+\delta)$. Apparently, this procedure is not highly satisfactory. In the absence of a field gradient (i.e. K=0), the matrix and the dispersion vector are given by

$$M = \begin{pmatrix} \cos\left[\frac{\ell}{\rho_o(1+\delta)}\right] & \rho_o(1+\delta) \sin\left[\frac{\ell}{\rho_o(1+\delta)}\right] \\ -\frac{1}{\rho_o(1+\delta)} \sin\left[\frac{\ell}{\rho_o(1+\delta)}\right] & \cos\left[\frac{\ell}{\rho_o(1+\delta)}\right] \end{pmatrix} , \qquad (A.4a)$$

$$D = -\begin{bmatrix} \rho_o(1+\delta)\left\{1 - \cos\left[\frac{\ell}{\rho_o(1+\delta)}\right]\right\} \\ \\ \sin\left[\frac{\ell}{\rho_o(1+\delta)}\right] \end{bmatrix} . \qquad (A.4b)$$

This sort of matrix exists in the standard PATRICIA and RACETRACK. Until recent times it also existed in FASTRAC.

The next kind of matrix is derived from

$$X'' + \left[\frac{K}{1+\delta} - \frac{1}{\rho_o^2}\frac{(1-\delta)}{(1+\delta)}\right] X = \frac{1}{\rho_o(1+\delta)}\ \delta \ . \qquad (A.5)$$

This expression was derived many years ago by G. Parzen, who installed the resulting (K=0) transfer matrix

$$M = \begin{bmatrix} \cos\left[\frac{\ell}{\rho_o}\sqrt{\frac{1-\delta}{1+\delta}}\right] & \rho_o\sqrt{\frac{1+\delta}{1-\delta}}\ \sin\left[\frac{\ell}{\rho_o}\sqrt{\frac{1-\delta}{1+\delta}}\right] \\ \\ -\frac{1}{\rho_o}\sqrt{\frac{1-\delta}{1+\delta}}\ \sin\left[\frac{\ell}{\rho_o}\sqrt{\frac{1-\delta}{1+\delta}}\right] & \cos\left[\frac{\ell}{\rho_o}\sqrt{\frac{1-\delta}{1+\delta}}\right] - \end{bmatrix} \qquad (A.6)$$

in his code ORBIT. The expression is derived from the Newton equation of motion and from the Lorentz force expression for a particle in electromagnetic field. This expression is more justifiable than (A.4a).

The third expression is derived from the linearized equations of motion given by R. Ruth[18]

$$X'' + \left[\frac{K}{1+\delta} - \frac{1}{\rho_o^2\,(1+\delta)}\right] X = \frac{1}{\rho_o(1+\delta)}\ \delta \ . \qquad (A.7)$$

This results in the (K=0) transfer matrix

$$M = \begin{bmatrix} \cos\left[\frac{\ell}{\rho_o\sqrt{1+\delta}}\right] & \rho_o\sqrt{1+\delta}\ \sin\left[\frac{\ell}{\rho_o\sqrt{1+\delta}}\right] \\ \\ -\frac{1}{\rho_o\sqrt{1+\delta}}\ \sin\left[\frac{\ell}{\rho_o\sqrt{1+\delta}}\right] & \cos\left[\frac{\ell}{\rho_o\sqrt{1+\delta}}\right] \end{bmatrix} , \qquad (A.8)$$

which is currently installed in PATRIS, FASTRAC and a version of PATRICIA that we finally tested.

Ruth started from the Hamiltonian of a particle in electromagnetic field and retained only the terms up to and including the second order in coordinates and momenta. This is logically highly consistent, but differs from (A.5) because of the presence of first order terms in H that give rise to some cross terms, which are lost when H is truncated at the second order but which nevertheless contribute to linearized equations of motion. As a result, (A.5) is more accurate even though the derivation of (A.7) may be logically somewhat more consistent. Potential problems with (A.5) may arise only when path length computation is involved.

All these differences are more academic than practical in their importance, for a machine like the RHIC [F6]. The reason is that to the lowest order in δ all these matrices are equivalent. However, to ensure code agreement in the 5th or 6th significant digit and beyond, one has to remove even so slightly different transfer matrices and resort to a single transfer matrix. An example of this was the testing of PATRICIA's new closed orbit finder.

The second code ingredient that gives rise to tune differences is the edge focusing matrix. All the codes treat this problem in the so-called hard edge approximation, except MAD that has an option of simulating the edge field fall-off; an option we never resorted to. However, even though the same kind of approximation is being used, there are differences. The edge focusing matrix for the quadruplet (X, X', Y, Y') of transverse coordinates is given by

$$M_{edge} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{-\tan\epsilon}{\rho} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{\tan\epsilon}{\rho} & 1 \end{pmatrix} . \qquad (A.9)$$

Here ρ is the bending radius, while ϵ is the angle between the incoming particle direction and the normal to the magnet cross section at the place of incidence. One such matrix precedes the bend matrix, another one follows. If the magnet is rectangular with equal entrance and exit angles, then for the design particle ϵ is one half of the total bending angle ϕ, and the three matrices A_{entr} M and A_{ex} can be combined into a single transfer matrix (K. Steffen)[22] that describes the behavior of the design particle, with nondesign particles described by the same kind of edge focusing matrix with ϵ being now the deviation from $\phi/2$. This is done in this manner in RACETRACK and presumably in ORBIT. However, these expressions do not have to be combined into a single analytic matrix expression for a rectangular bend; instead they can simply be left to the computer to multiply them. This is being done in this manner in FASTRAC, PATRICIA and PATRIS.

There is no ambiguity for an on-momentum particle. Some confusion arises, however, for the off-momentum case. First of all, one will notice that (A.9) is reminiscent of a thin lens quad matrix. Indeed, the on-momentum case is actually simulated by a thin lens quad in FASTRAC, PATRICIA and PATRIS, and quite explicitly in TEAPOT. However, for the off-momentum case this picture breaks down since in fact edge focusing is not a quad,[22] owing to different origins of the focusing in the two planes (i.e. a

geometric vs. a Maxwellian effect). In the absence of a uniquely accepted extension to the off-momentum case, these codes contain some arbitrary assumptions about the off-momentum effects, depending on their designers/developers' taste. As an outcome, PATRICIA simulates the edge effects for a rectangular bend by (A.9) with a δ-dependent tan ϵ/ρ, where $\rho=\rho_o(1+\delta)$ and

$$\epsilon = \frac{\ell}{2\rho_o(1+\delta)} \quad , \tag{A.10}$$

for the off-momentum incidence angle of a particle whose entry coordinates are (0, 0, 0, 0). Therefore, PATRICIA's edge focusing matrix for a rectangular bend is in fact given by

$$M_{edge} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{\rho_o(1+\delta)}\tan\left[\frac{\ell}{2\rho_o(1+\delta)}\right] & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{\rho_o(1+\delta)}\tan\left[\frac{\ell}{2\rho_o(1+\delta)}\right] & 1 \end{pmatrix} . \tag{A.11}$$

PATRIS, on the other hand takes ϵ explicitly from the input, hence it is frozen to $\ell/2\rho_o$ for both on and off-momentum cases. FASTRAC currently does the same except that the code evaluates the same ϵ as PATRIS, from the values of ℓ and ρ_o supplied at the input. But once this ϵ has been specified, both codes in their current versions that we finally tested had the following matrix

$$M_{edge} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{\rho_o(1+\delta)}\tan\left[\frac{\ell}{2\rho_o}\right] & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{\rho_o(1+\delta)}\tan\left[\frac{\ell}{2\rho_o}\right] & 1 \end{pmatrix} , \tag{A.12}$$

for the edge focusing of a rectangular bend. The same matrix was also installed in a version of PATRICIA when we tested its new closed orbit finder.

All this may change at any future time, once a consensus is reached. On a lattice such as that of the RHIC, the effects of different off-momentum edge focusing are small and they show up only under a stringent testing. However, for a small ring they will be much more important.

Finally, we would like to mention for the sake of completeness that FASTRAC, PATRICIA and PATRIS handle ideal drifts and quads in the same manner. In the noncanonical (X, X',Y, Y') coordinates, the drift matrix is given as

$$M_{drift} = \begin{pmatrix} 1 & \ell & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \ell \\ 0 & 0 & 0 & 1 \end{pmatrix} , \tag{A.13}$$

whereas the matrix for a horizontally focusing quad is described by

$$M_{\overline{quad}} = \begin{bmatrix} \cos(\ell\sqrt{|K|}) & \frac{1}{\sqrt{|K|}}\sin(\ell\sqrt{|K|}) & 0 & 0 \\ -\sqrt{|K|}\sin(\ell\sqrt{|K|}) & \cos(\ell\sqrt{|K|}) & 0 & 0 \\ 0 & 0 & \cosh(\ell\sqrt{|K|}) & \frac{1}{\sqrt{|K|}}\sinh(\ell\sqrt{|K|}) \\ 0 & 0 & \sqrt{|K|}\sinh(\ell\sqrt{|K|}) & \cosh(\ell\sqrt{|K|}) \end{bmatrix} . \quad (A.14)$$

In the latter expression, $K=B'/B\rho_o(1+\delta)<0$.

Footnotes

1. More precisely, among all of the codes (code versions) we have tested so far. MARYLIE, a major tracking/analysis code based on Lie algebraic methods for transfer map representations, employs a more accurate description of hard-edge focusing (Ref.22). We plan to run MARYLIE on the RHIC lattice and report the results in a separate technical note.

2. It was found out later that E. Forest had been aware of this and had a code version with the problem fixed.

3. This method will be described in a separate note.

4. More precisely, it is a second order expansion in transverse variables, of the exact Hamiltonian arising from the adopted physical model.

5. It seems that this problem has been solved in one of the recently modified VAX versions of SYNCH, presented to us by A. Garren at LBL.

6. These differences, however, could be quite crucial for small machines, with large bends, where the quantity ℓ/ρ_o could be quite large.

References

1. B. T. Leemann and E. Forest, Brief Description of the Tracking Codes "FASTRAC" and "THINTRAC," SSC-133, June 1987

2. F. Christoph Iselin and James Niederer, The MAD Program - Version 6 User's Reference Manual, CERN/LEP-TH/87-33, April 1987

3. G. Parzen, private communications

4. H. Wiedemann, USER'S GUIDE FOR PATRICIA VERSION 84.9, ESRP-IRM-72/84 SSRL-ACD-NOTE #22 (1984)

5. A. G. Ruggiero, PATRIS.D. On Line Manual for PATRIS (mfe Cray X-MP)

6. A. A. Garren, A. S. Kenney, E. D. Courant and M. J. Syphers, A USER'S GUIDE TO SYNCH, (FN-420/0170.000), June 1985

7. L. Schachinger and R. Talman, TEAPOT. A Thin Element Accelerator Program for Optics and Tracking. SSC-52, December 1985.

8. A. Wrulich, RACETRACK. A Computer Code for the Simulation of Nonlinear Particle Motion in Accelerators, ISSN 0418-9833, DESY 84-026, March 1984

9. R. Raja, A. Russell & C. Ankembrandt, Nucl. Inst. & Meth. A242, 15 (1985)

10. A. J. Dragt, R. D. Ryne, L. M. Healy, F. Neri, D. R. Douglas and E. Forest, MARYLIE 3.0. A Program for Charged Particle Beam Transport Based on Lie Algebraic Methods, University of Maryland, 1985

11. E. Forest, The Synch Fix (in preparation)

12. E. Forest, private communication

13. Z. Parsa, "ACCELERATOR PHYSICS CODE COMPARISON," AD/AP/Tech. Note No. 8, October 1987

14. Booster Design Manual, ADD/Brookhaven Natl. Lab., October 1986

15. Conceptual Design of the Relativistic Heavy Ion Collider - RHIC, BNL 51932, May 1986

16. E. Forest, private communication

17. E. Courant and S. Y. Lee, private communications

18. R. D. Ruth, Single-Particle Dynamics in Circular Accelerators, AIP Conference Proceedings 153 (1987), p. 152

19. E. Forest, private communication

20. G. Parzen, private communication

21. K. G. Steffen, High Energy Beam Optics (Ch. 2), Interscience Publish., 1965

22. E. Forest, Healy's Modular Approach to the Computation of the General Bending Magnet Map Applied to the Quadratic Part of the Hamiltonian Which is Exact in $\Delta p/p_o$, SSC-141, October 1987.

Tables

(G stands for General, U for Uncorrected, C for Corrected)

G1. This table represents some general on-momentum lattice characteristics. All of the quantities in this table have been generated by the corrected versions of the codes. The tunes supplied by ORBIT have been redefined by subtracting the on-momenum discrepancies. Presence of asterisks indicates that relevant numbers have not been supplied for various reasons.

G2. This table represents errors at dp/p = -1% and at dp/p = +1% . The error in the particular quantity Q is evaluated as $|Q_{max} - Q_{min}|$. These quantities refer to the results obtained from the codes after the various corrective measures have been implemented.

* * * * * * * * .

U1. This table displays the horizontal tune dependence on momentum deviation δ. The tunes were generated by the original (Uncorrected) versions of the codes. MAD and PATRIS deviate from SYNCH quadratically in δ, ORBIT disagrees on momentum but otherwise deviates linearly in δ, all the other codes deviate linearly.

U2. This table displays the vertical tune dependence on momentum deviation. The tunes were generated by the original (Uncorrected) versions of the codes. MAD and PATRIS deviate from SYNCH quadratically in δ, ORBIT disagrees on momentum but otherwise deviates linearly in δ, all the other codes deviate linearly.

U3. This table displays the horizontal tune dependence on momentum deviation δ. The tunes were here generated by ORBIT and the two versions of PATRIS. Both versions of PATRIS are corrected in their description of the body of the bend but have different δ dependences in the edge focusing matrix. Both versions of PATRIS were run on a fake lattice with symmetrized entry and exit angles on the BC1 magnets, to mimic what ORBIT is actually doing. On-momentum agreement is now perfect, while for off-momentum values the version of PATRIS without δ dependence in the edge focusing matrix comes closer to ORBIT's predictions. The residual difference is due to slightly different transfer matrices for the body of the magnet. In the vertical plane, where the magnet acts driftlike, this difference is negligible (see also U4 and the corresponding figures).

U4. This table displays the vertical tune dependence on momentum deviation. All the comments going with U3 apply here. Note that PATRIS run on the fake lattice agrees extremely well with ORBIT if no δ dependence is introduced in its edge focusing matrix.

U5. This table represents the closed orbit function $X_{co} = X_{fp}/\delta$, generated by the original versions of the codes. PATRICIA significantly disagrees with the other codes, while PATRIS displays some disagreement for small values of δ, because of its very short printout format. This disagreement can be best seen on the graph (see Fig. U5, U5a, U5b).

U6. This table displays the same as U5, except that the PATRIS values are obtained with more significant digits printed out. The agreement is now excellent.

U7. This table displays the dependence of horizontal beta function on momentum deviation δ, at the first crossing point CRIO. All uncorrected codes agree very well with each other except that PATRICIA starts deviating from the rest at $\delta > 0.2\%$, as a result of its deficient closed orbit finder and, to a much lesser extent, its compulsory sextupole fitting.

U8. This table displays the dependence of vertical beta function on momentum deviation δ, at the first crossing point CRIO. All the comments going with Table U7. also apply here.

* * * * * * * *

C1. This table displays the horizontal tune dependence on momentum deviation δ. The tunes were generated by the corrected versions of the codes, if applicable. ORBIT's tune has been adjusted by subtracting out the on-momentum difference, PATRIS runs with corrected bend, while PATRICIA runs with a new closed orbit finder and with sextupole fit suppressed. As a result of the various corrective measures all of the codes deviate from SYNCH by small amounts linear in δ, except MAD whose deviation is more significant and quadratic in δ.

C2. This table displays the vertical tune dependence on momentum deviation. The tunes were generated by the corrected versions of codes, if applicable. All the comments going with Table C1. apply also here.

C3. This table displays the dependence of the closed orbit function X_{co} on momentum deviation δ, at the middle of an inner arc (SYM). With PATRICIA's closed orbit finder being corrected and with a more accurate printout from PATRIS, all of the codes agree pretty well.

C4. This table displays the dependence of the momentum dispersion function η on momentum deviation δ, at the middle of an inner arc (SYM). Only FASTRAC, MAD, PATRIS and SYNCH supplied this quantity. Apparently, the agreement between SYNCH and the other three codes is not very good.

C5. This table displays the dependence of beta horizontal on momentum deviation δ, at the middle of an inner arc (SYM)). The corrected versions of codes have been used here, if applicable.

C6. This table displays the dependence of beta vertical on momentum deviation δ, at the middle of an inner arc (SYM)). The corrected versions of codes have been used here, if applicable.

C7. This table displays the dependence of the momentum dispersion function η on momentum deviation δ, at the first crossing point (CRIO). Only FASTRAC, MAD, PATRIS and SYNCH supplied this quantity. Apparently, the agreement between SYNCH and the other three codes is not very good.

C8. This table displays the dependence of beta horizontal on momentum deviation δ, at the first crossing point (CRIO). The corrected versions of codes have been used here, if applicable. Comparing this table with U7. shows the impact of PATRICIA's deficient closed orbit finder on its predictive accuracy for beta function, at this lattice location.

C9. This table displays the dependence of beta vertical on momentum deviation δ, at the first crossing point (CRIO). The corrected versions of codes have been used here, if applicable. Comparing this table with U8. shows the impact of PATRICIA's deficient closed orbit finder on its predictive accuracy for beta function, at this lattice location.

C10. This table displays the dependence of the momentum dispersion function η on momentum deviation δ, at the second crossing point (CROI). Only FASTRAC, MAD, PATRIS and SYNCH supplied this quantity. Apparently, the agreement between SYNCH and the other three codes is not very good. In addition, the agreement between MAD on the one hand and FASTRAC and PATRIS on the other, seems somewhat degraded at δ = +1%, when compared with C7. However, MAD printed only one significant digit in this region of η values, which does not justify making strong conclusions.

C11. This table displays the dependence of beta horizontal on momentum deviation δ, at the second crossing point (CROI). The corrected versions of the codes have been used here, if applicable. The agreement between the codes is good. The uncorrected codes produced disagreements similar to those in U7. (not shown in this note).

C12. This table displays the dependence of beta vertical on momentum deviation δ, at the second crossing point (CROI). The corrected versions of the codes have been used here, if applicable. The agreement between the codes is good. The uncorrected codes produced disagreements similar to those in U8. (not shown in this note).

Figure Captions

(U stands for Uncorrected, C for Corrected)

1. This figure displays the RHIC Layout taken from the RHIC Design Manual.

2. This figure represents a crossing point within a RHIC insertion, together with the two adjacent BC1 magnets.

* * * * * * * *

U1. This figure displays the horizontal tune dependence on momentum deviation δ. The tunes were generated by the original (Uncorrected) versions of the codes. MAD and PATRIS deviate from SYNCH quadratically in δ, ORBIT disagrees on momentum but otherwise deviates linearly in δ, all the other codes deviate linearly.

U2. This figure displays the vertical tune dependence on momentum deviation. The tunes were generated by the original (Uncorrected) versions of the codes. MAD and PATRIS deviate from SYNCH quadratically in δ, ORBIT disagrees on momentum but otherwise deviates linearly in δ, all the other codes deviate linearly.

U3. This figure displays the horizontal tune dependence on momentum deviation . The tunes were here generated by ORBIT and the two versions of PATRIS. Both versions of PATRIS are corrected in their description of the body of the bend but have different δ dependences in the edge focusing matrix. Both versions of PATRIS were run on a fake lattice with symmetrized entry and exit angles on the BC1 magnets, to mimic what ORBIT is actually doing. On-momentum agreement is now perfect, while for off-momentum values the version of PATRIS without δ dependence in the edge focusing matrix comes closer to ORBIT's predictions. The residual difference is due to slightly different transfer matrices for the body of the magnet. In the vertical plane, where the magnet acts driftlike, this difference is negligible (see also figures U3a., U4., U4a.).

U3a. This figure is the same as U3., but on a smaller scale to display the details better. The scale of U3. is the same as U1., for the sake of easier comparisons. Apparently, PATRIS with no δ dependence in its edge focusing matrix agrees better with ORBIT.

U4. This table displays the vertical tune dependence on momentum deviation. All the comments going with U3 apply here. Note that PATRIS run on the fake lattice agrees extremely well with ORBIT if no δ dependence is introduced in its edge focusing matrix.

U4a. This figure is the same as U4., but on a smaller scale to display the details better. The scale of U4. is the same as U2., for the sake of easier comparisons. Apparently, PATRIS with no δ dependence in its edge focusing matrix agrees perfectly with ORBIT.

U5. This figure represents the closed orbit function $X_{co} = X_{fp}/\delta$,

generated by the original versions of the codes. PATRICIA significantly disagrees with the other codes, while PATRIS displays some disagreement for small values of δ, because of its very short printout format (see also Fig. U5a.).

U5a. This figure represents the same as U5., but on a smaller scale. PATRICIA even cannot be completely described on this scale, while PATRIS shows the difficulties due to its too short printout more clearly that on U5.

U5b. This figure represents the same as U5. or U5a. (closed orbit function X_{co}), but with the result from PATRICIA replaced by the closed orbit function supplied by ORBIT (PATRICIA and ORBIT could not appear simultaneously on the same drawing due to limitations of our graphics package).

U6. This figure displays the same as U5a., except that the PATRIS values are obtained with more significant digits printed out. The agreement is now excellent.

U7. This figure displays the dependence of horizontal beta function on momentum deviation δ, at the first crossing point CRIO. All uncorrected codes agree very well with each other except that PATRICIA starts deviating from the rest at $\delta > 0.2\%$, as a result of its deficient closed orbit finder and, to a much lesser extent, its compulsory sextupole fitting.

U8. This figure displays the dependence of vertical beta function on momentum deviation δ, at the first crossing point CRIO. All the comments going with Figure U7. also apply here.

* * * * * * * *

C1. This figure displays the horizontal tune dependence on momentum deviation δ. The tunes were generated by the corrected versions of the codes, if applicable. ORBIT's tune has been adjusted by subtracting out the on-momentum difference, PATRIS runs with corrected bend, while PATRICIA runs with a new closed orbit finder and with sextupole fit suppressed. As a result of the various corrective measures all of the codes deviate from SYNCH by small amounts linear in δ, except MAD whose deviation is more significant and quadratic in δ.

C2. This figure displays the vertical tune dependence on momentum deviation. The tunes were generated by the corrected versions of the codes, if applicable. All the comments going with Figure C1. apply also here.

C3. This figure displays the dependence of the closed orbit function X_{co} on momentum deviation δ, at the middle of an inner arc (SYM). With PATRICIA's closed orbit finder being corrected and with a more accurate printout from PATRIS, all of the codes agree pretty well.

C4. This figure displays the dependence of the momentum dispersion function η on momentum deviation δ, at the middle of an inner arc (SYM). Only FASTRAC, MAD, PATRIS and SYNCH supplied this quantity. Apparently, the agreement between SYNCH and the other three codes is not very good.

C5. This figure displays the dependence of beta horizontal on momentum deviation δ, at the middle of an inner arc (SYM)). The corrected versions of the codes have been used here, if applicable. The agreement between different codes is very good.

C6. This figure displays the dependence of beta vertical on momentum deviation δ, at the middle of an inner arc (SYM)). The corrected versions of the codes have been used here, if applicable. The agreement between different codes is very good.

C7. This figure displays the dependence of the momentum dispersion function η on momentum deviation δ, at the first crossing point (CRIO). Only FASTRAC, MAD, PATRIS and SYNCH supplied this quantity. Apparently, the agreement between SYNCH and the other three codes is not very good.

C8. This figure displays the dependence of beta horizontal on momentum deviation δ, at the first crossing point (CRIO). The corrected versions of the codes have been used here, if applicable. Comparing this figure with U7. shows the impact of PATRICIA's deficient closed orbit finder on its predictive accuracy for beta function, at this lattice location. After correction, PATRICIA's agreement with the other codes is excellent.

C9. This figure displays the dependence of beta vertical on momentum deviation δ, at the first crossing point (CRIO). The corrected versions of the codes have been used here, if applicable. Comparing this figure with U8. shows the impact of PATRICIA's deficient closed orbit finder on its predictive accuracy for beta function, at this lattice location. After correction, PATRICIA's agreement with the other codes is excellent.

C10. This figure displays the dependence of the momentum dispersion function η on momentum deviation δ, at the second crossing point (CROI). Only FASTRAC, MAD, PATRIS and SYNCH supplied this quantity. Apparently, the agreement between SYNCH and the other three codes is not very good. In addition, the agreement between MAD on the one hand and FASTRAC and PATRIS on the other, seems somewhat degraded at δ = +1%, when compared with C7. However, MAD printed only one significant digit in this region of η values, which does not justify making strong conclusions.

C11. This figure displays the dependence of beta horizontal on momentum deviation δ, at the second crossing point (CROI). The corrected versions of the codes have been used here, if applicable. The agreement between the codes is good. The uncorrected codes produced disagreements similar to those in U7. (not shown in this note).

C12. This figure displays the dependence of beta vertical on momentum deviation δ, at the second crossing point (CROI). The corrected versions of the codes have been used here, if applicable. The agreement between the codes is good. The uncorrected codes produced disagreements similar to those in U8. (not shown in this note).

Table G1

Some General On-Momentum Lattice Characteristics

Quantity	SYNCH	PATRIS	PATRICIA	FASTRAC	MAD	TEAPOT	ORBIT
2*Pi*R	3833.8677	3833.8677	3833.8677	3833.868	3833.8677	****	****
GAMMAt	25.44196	25.44196	25.43872	****	25.441958	****	****
Qh	28.82850	28.82850	28.82850	28.82850	28.82850	28.82850	28.82850
Qv	28.82903	28.82903	28.82903	28.82903	28.82904	28.82910	28.82903
Cx bare	-56.39246	-56.605	-56.532	-56.5355	****	-56.3911	****
Cy bare	-56.39131	-56.344	-56.464	-56.4724	****	-56.4058	****
Cx sext	0.09193	-0.120	-0.048	-0.05106	0.093425	0.09401	****
Cy sext	0.05025	0.097	-0.022	-0.03094	0.046681	0.0365	****
Bth(SYM)	50.2645	50.26448	50.2645	50.2645	50.264	****	50.08
Btv(SYM)	8.4948	8.49482	8.4948	8.4948	8.495	****	8.48
ETA(SYM)	1.5010	1.50100	1.50100	1.50101	1.501	****	1.501
Bth(CRIO)	3.1314	3.13145	3.1314	3.1314	3.131	****	3.14
Btv(CRIO)	3.1331	3.13312	3.1331	3.1331	3.133	****	3.13
ETA(CRIO)	0.0079	0.00786	0.00786	0.00786	0.008	****	****
Bth(CROI)	3.1314	3.13145	3.1314	3.1314	3.131	****	3.14
Btv(CROI)	3.1329	3.13292	3.1329	3.1329	3.133	****	3.13
ETA(CROI)	0.0091	0.00912	0.00912	0.00912	0.009	****	****

Remarks: The quantities in this table have been generated by the corrected versions of the codes. The tunes supplied by ORBIT have been redefined by subtracting the on-momentum discrepancies. Presence of asterisks indicates that relevant numbers have not been supplied for various reasons.

Table G2

Errors at Momentum Deviation -1% and +1%

Quantity	dp/p=-1% Average value	dp/p=-1% Absolute error	dp/p=-1% Error in per cent	dp/p=+1% Average value	dp/p=+1% Absolute error	dp/p=+1% Error in per cent
Qh	.825637	.002855	.346	.835174	.002735	.327
Qv	.827225	.000707	.085	.832623	.001010	.121
XCo(m)	1.48340	.00100	.067	1.49628	.00114	.076
ETA(SYM)	1.45834	.01150	.789	1.47771	.01369	.926
ETA(CRIO)	0.02073	.00190	9.165	0.00179	.00197	110.056
ETA(CROI)	0.01858	.00170	9.150	0.00712	.00188	26.404
Bth(SYM)	26.70090	.10870	.407	90.71510	1.35690	1.496
Btv(SYM)	9.45225	.03233	.342	7.11785	.07830	1.100
Bth(CRIO)	3.99329	.01160	.290	3.10780	.01440	.463
Btv(CRIO)	3.81953	.00696	.182	3.13665	.01410	.450
Bth(CROI)	3.95614	.01400	.354	3.28574	.01150	.350
Btv(CROI)	3.76993	.00570	.151	3.35948	.00780	.232

Table U1

Horizontal Tune Dependence on Momentum Deviation

dp/p	SYNCH	PATRIS (original)	PATRICIA (old,fits)	FASTRAC	MAD	TEAPOT	ORBIT
-1.0	.82437	.8319	.82433	.825968	.818626	.824345	.8284
-0.8		.8311	.82606	.827425	.822567	.826175	
-0.6		.8303	.82727	.828240	.825320	.827333	
-0.5	.82773						.8302
-0.4		.8295	.82800	.828578	.827094	.827989	
-0.2		.8288	.82836	.828604	.828089	.828317	
0.0	.82850	.82850	.82850	.82850	.82850	.828505	.8297
0.2		.8287	.82860	.828476	.828534	.828771	
0.4		.8297	.82889	.828780	.828419	.829371	
0.5	.82986						.8298
0.6		.8319	.82963	.829706	.828413	.830608	
0.8		.8356	.83117	.831597	.828809	.832830	
1.0	.83638	.8414	.83391	.834845	.829943	.836435	.8349

Table U2

Vertical Tune Dependence on Momentum Deviation

dp/p	SYNCH	PATRIS (original)	PATRICIA (old,fits)	FASTRAC	MAD	TEAPOT	ORBIT
-1.0	.82700	.8328	.82773	.827538	.820721	.826833	.8258
-0.8		.8316	.82841	.828406	.823993	.827878	
-0.6		.8305	.82882	.828877	.826309	.828511	
-0.5	.82869						.8276
-0.4		.8297	.82903	.829072	.827835	.828856	
-0.2		.8292	.82908	.829092	.828706	.829018	
0.0	.82903	.8290	.82903	.829034	.829035	.829097	.8279
0.2		.8294	.82899	.829007	.828928	.829203	
0.4		.8303	.82904	.829136	.828498	.829467	
0.5	.82973						.8284
0.6		.8321	.82934	.829581	.827871	.830050	
0.8		.8350	.83006	.830536	.827196	.831150	
1.0	.83324	.8392	.83140	.832234	.826650	.833000	.8317

Table U3

ORBIT vs. PATRIS on a Fake Lattice with Symmetric Entrance/Exit Angles
Horizontal Tune Dependence on Momentum Deviation

	PATRIS (ang.sym)	PATRIS (ang.sym)	ORBIT
Presence of dp/p in the edge focusing matrix:	No	Yes	
dp/p			
-1.0	.827757	.827045	.8284
-0.8	.829058	.828511	
-0.6	.829738	.829341	
-0.5			.8302
-0.4	.829956	.829698	
-0.2	.829876	.829749	
0.0	.829674	.829674	.8297
0.2	.829558	.829683	
0.4	.829772	.830024	
0.5			.8298
0.6	.830605	.830988	
0.8	.832398	.832917	
1.0	.835535	.836199	.8349

Table U4

ORBIT vs. PATRIS on a Fake Lattice with Symmetric Entrance/Exit Angles
Vertical Tune Dependence on Momentum Deviation

	PATRIS (ang.sym)	PATRIS (ang.sym)	ORBIT
Presence of dp/p in the edge focusing matrix:	No	Yes	
dp/p			
-1.0	.825765	.826418	.8258
-0.8	.826773	.827284	
-0.6	.827371	.827748	
-0.5			.8276
-0.4	.827682	.827932	
-0.2	.827811	.827936	
0.0	.827857	.827857	.8279
0.2	.827931	.827804	
0.4	.828164	.827904	
0.5			.8284
0.6	.828719	.828317	
0.8	.829794	.829238	
1.0	.831627	.830904	.8317

Table U5

Dependence of Closed Orbit Function on Momentum Deviation
at the Middle of an Inner Arc (SYM)

dp/p	SYNCH	PATRIS	PATRICIA (old,fits)	FASTRAC	MAD	TEAPOT	ORBIT
-1.0	1.48342	1.48300	1.52496	1.48332	1.48400	1.48345	1.483
-0.8		1.48750	1.50885	1.48777	1.48825	1.48790	
-0.6		1.49167	1.49956	1.49191	1.49217	1.49203	
-0.5	1.49388						
-0.4		1.49500	1.49398	1.49559	1.49575	1.49571	1.494
-0.2		1.50000	1.49424	1.49867	1.49850	1.49881	
0.0	1.5010	1.50100	1.50100	1.50100	1.501	1.50100	1.501
0.2		1.50000	1.51619	1.50244	1.50250	1.50233	
0.4		1.50250	1.54228	1.50282	1.50275	1.50272	
0.5	1.50246						
0.6		1.50167	1.58251	1.50200	1.50217	1.50185	1.502
0.8		1.50000	1.64157	1.49983	1.50013	1.49962	
1.0	1.49586	1.49600	1.72747	1.49615	1.49680	1.49586	1.497

Table U6

Dependence of Closed Orbit Function on Momentum Deviation
at the Middle of an Inner Arc (SYM)

dp/p	SYNCH	PATRIS (long prnt)	PATRICIA (old,fits)	FASTRAC	MAD	TEAPOT	ORBIT
-1.0	1.48342	1.48318	1.52496	1.48332	1.48400	1.48345	1.483
-0.8		1.48766	1.50885	1.48777	1.48825	1.48790	
-0.6		1.49182	1.49956	1.49191	1.49217	1.49203	
-0.5	1.49388						1.494
-0.4		1.49553	1.49398	1.49559	1.49575	1.49571	
-0.2		1.49865	1.49424	1.49867	1.49850	1.49881	
0.0	1.5010	1.50100	1.50100	1.50100	1.501	1.50100	1.501
0.2		1.50245	1.51619	1.50244	1.50250	1.50233	
0.4		1.50290	1.54228	1.50282	1.50275	1.50272	
0.5	1.50246						1.502
0.6		1.50212	1.58251	1.50200	1.50217	1.50185	
0.8		1.49999	1.64157	1.49983	1.50013	1.49962	
1.0	1.49586	1.49636	1.72747	1.49615	1.49680	1.49586	1.497

**** Remark: PATRIS's printout has been lengthened for better accuracy!

Table U7

Horizontal Beta Function Dependence on Momentum Deviation at the First Crossing Point (CRIO)

dp/p	SYNCH	PATRIS	PATRICIA (old,fits)	FASTRAC	MAD	TEAPOT	ORBIT
-1.0	3.9981	3.9876	3.9964	3.99000	4.0016	N	3.99
-0.8		3.7595	3.7636	3.76077	3.7646	o	
-0.6		3.5590	3.5608	3.55951	3.5605	t	
-0.5	3.4714						3.47
-0.4		3.3867	3.3882	3.38680	3.3864	a	
-0.2		3.2436	3.2452	3.24360	3.2435	v	
0.0	3.1314	3.13145	3.1314	3.13140	3.1310	a	3.14
0.2		3.0520	3.0469	3.05219	3.0521	i	
0.4		3.0076	2.9918	3.00798	3.0060	l	
0.5	3.0007					a	3.00
0.6		3.0006	2.9664	3.00130	2.9969	b	
0.8		3.0333	2.9712	3.03428	3.0270	l	
1.0	3.1121	3.1076	3.0061	3.10898	3.0977	e	3.11

Table U8

Vertical Beta Function Dependence on Momentum Deviation at the First Crossing Point (CRIO)

dp/p	SYNCH	PATRIS	PATRICIA (old,fits)	FASTRAC	MAD	TEAPOT	ORBIT
-1.0	3.8204	3.8194	3.8083	3.81744	3.8244	N	3.82
-0.8		3.6392	3.6340	3.63806	3.6406	o	
-0.6		3.4786	3.4768	3.47801	3.4780	t	
-0.5	3.4067						3.40
-0.4		3.3393	3.3399	3.33919	3.3386	a	
-0.2		3.2235	3.2246	3.22344	3.2235	v	
0.0	3.1331	3.13312	3.1331	3.13310	3.1330	a	3.13
0.2		3.0705	3.0669	3.07033	3.0701	i	
0.4		3.0377	3.0266	3.03730	3.0351	l	
0.5	3.0331					a	3.03
0.6		3.0370	3.0121	3.03631	3.0321	b	
0.8		3.0704	3.0205	3.06926	3.0613	l	
1.0	3.1401	3.1395	3.0452	3.13797	3.1260	e	3.14

Table C1

Horizontal Tune Dependence on Momentum Deviation

dp/p	SYNCH	PATRIS	PATRICIA (new,no fit)	FASTRAC	MAD	TEAPOT	ORBIT
-1.0	.82437	.825968	.82597	.825968	.818626	.824345	.8272
-0.8		.827424	.82742	.827425	.822567	.826175	
-0.6		.828240	.82824	.828240	.825320	.827333	
-0.5	.82773						.8290
-0.4		.828578	.82858	.828578	.827094	.827989	
-0.2		.828604	.82860	.828604	.828089	.828317	
0.0	.82850	.82850	.82850	.82850	.82850	.828505	.8285
0.2		.828475	.82848	.828476	.828534	.828771	
0.4		.828779	.82878	.828780	.828419	.829371	
0.5	.82986						.8286
0.6		.829705	.82971	.829706	.828413	.830608	
0.8		.831597	.83160	.831597	.828809	.832830	
1.0	.83638	.834844	.83484 (input=in5)	.834845	.829943	.836435	.8337

**** Remark: PATRICIA (new) runs on in5, which is debugged and suppresses fit!
PATRIS runs fully corrected!
ORBIT tunes have been adjusted by subtracting out
the on-momentum difference!

Table C2

Vertical Tune Dependence on Momentum Deviation

dp/p	SYNCH	PATRIS	PATRICIA (new,no fit)	FASTRAC	MAD	TEAPOT	ORBIT
-1.0	.82700	.827536	.82754	.827538	.820721	.826833	.8269
-0.8		.828405	.82841	.828406	.823993	.827878	
-0.6		.828876	.82888	.828877	.826309	.828511	
-0.5	.82869						.8287
-0.4		.829070	.82907	.829072	.827835	.828856	
-0.2		.829090	.82909	.829092	.828706	.829018	
0.0	.82903	.829033	.82903	.829034	.829035	.829097	.82903
0.2		.829005	.82901	.829007	.828928	.829203	
0.4		.829134	.82914	.829136	.828498	.829467	
0.5	.82973						.8295
0.6		.829579	.82958	.829581	.827871	.830050	
0.8		.830534	.83054	.830536	.827196	.831150	
1.0	.83324	.832233	.83223	.832234	.826650	.833000	.8328
			(input=in5)				

**** Remark: PATRICIA (new) runs on in5, which is debugged and suppresses fit!
PATRIS runs fully corrected!
ORBIT tunes have been adjusted by subtracting out
the on-momentum difference!

Table C3

Dependence of Closed Orbit Function on Momentum Deviation
at the Middle of an Inner Arc (SYM)

dp/p	SYNCH	PATRIS (strg.f.p.)	PATRICIA (new,nofit)	FASTRAC	MAD	TEAPOT	ORBIT
-1.0	1.48342	1.48332	1.48332	1.48332	1.48400	1.48345	1.483
-0.8		1.48778	1.48777	1.48777	1.48825	1.48790	
-0.6		1.49192	1.49191	1.49191	1.49217	1.49203	
-0.5	1.49388						1.494
-0.4		1.49560	1.49559	1.49559	1.49575	1.49571	
-0.2		1.49865	1.49867	1.49867	1.49850	1.49881	
0.0	1.5010	1.50100	1.50100	1.50100	1.501	1.50100	1.501
0.2		1.50245	1.50244	1.50244	1.50250	1.50233	
0.4		1.50283	1.50282	1.50282	1.50275	1.50272	
0.5	1.50246						1.502
0.6		1.50200	1.50200	1.50200	1.50217	1.50185	
0.8		1.49983	1.49983	1.49983	1.50013	1.49962	
-1.0	1.49586	1.49615	1.49615	1.49615	1.49680	1.49586	1.497
			(input=in5)				

**** Remark: PATRICIA (new) runs on in5, which is debugged and suppresses fit! PATRIS runs with its bend fully corrected, with a longer printout of its fixed point and with a more stringent exiting criterion (more iterations) for its Newton closed orbit finder.

Table C4

Dependence of Momentum Dispersion Function on Momentum Deviation at the Middle of an Inner Arc (SYM)

dp/p	SYNCH	PATRIS (eta cor.)	PATRICIA	FASTRAC	MAD	TEAPOT	ORBIT
-1.0	1.4505	1.46063	N	1.46044	1.462	N	N
-0.8		1.47069	o	1.47051	1.472	o	o
-0.6		1.48028	t	1.48013	1.481	t	t
-0.5	1.4793						
-0.4		1.48890	a	1.48879	1.489	a	a
-0.2		1.49601	v	1.49594	1.496	v	v
0.0	1.5010	1.50101	a	1.50101	1.501	a	a
0.2		1.50330	i	1.50337	1.503	i	i
0.4		1.50227	l	1.50243	1.503	l	l
0.5	1.5069		a			a	a
0.6		1.49732	b	1.49758	1.498	b	b
0.8		1.48784	l	1.48821	1.490	l	l
1.0	1.4874	1.47323	e	1.47371	1.476	e	e

**** Remark: Only SYNCH, FASTRAC, MAD and modified PATRIS supplied this quantity!

Table C5

Horizontal Beta Function Dependence on Momentum Deviation at the Middle of an Inner Arc (SYM)

dp/p	SYNCH	PATRIS (strg.f.p.)	PATRICIA (new,nofit)	FASTRAC	MAD	TEAPOT	ORBIT
-1.0	26.6678	26.70967	26.7096	26.70961	26.7587	N	26.65
-0.8		30.28438	30.2844	30.28437	30.3413	o	
-0.6		34.36720	34.3672	34.36715	34.4153	t	
-0.5	36.6022						36.50
-0.4		39.01969	39.0197	39.01970	39.0502	a	
-0.2		44.30133	44.3013	44.30132	44.3112	v	
0.0	50.2645	50.26448	50.2645	50.26450	50.2640	a	50.08
0.2		56.94731	56.9473	56.94727	56.9677	i	
0.4		64.36449	64.3645	64.36453	64.4638	l	
0.5	68.3297					a	68.11
0.6		72.49622	72.4962	72.49618	72.7710	b	
0.8		81.27703	81.2770	81.27706	81.8698	l	
1.0	90.4923	90.58717	90.5872	90.58720	91.6969	e	90.34
			(input=in5)				

**** Remark: PATRICIA (new) runs on in5, which is debugged and suppresses fit! PATRIS runs with its bend fully corrected, with a longer printout of its fixed point and with a more stringent exiting criterion (more iterations) for its Newton closed orbit finder.

Table C6

Vertical Beta Function Dependence on Momentum Deviation at the Middle of an Inner Arc (SYM)

dp/p	SYNCH	PATRIS (strg.f.p.)	PATRICIA (new,nofit)	FASTRAC	MAD	TEAPOT	ORBIT
-1.0	9.4512	9.44497	9.4450	9.44500	9.4773	N	9.45
-0.8		9.32864	9.3286	9.32867	9.3407	o	
-0.6		9.17138	9.1714	9.17134	9.1746	t	
-0.5	9.0807						9.07
-0.4		8.97622	8.9762	8.97625	8.9760	a	
-0.2		8.74840	8.7484	8.74837	8.7485	v	
0.0	8.4948	8.49482	8.4948	8.49480	8.4950	a	8.48
0.2		8.22346	8.2235	8.22341	8.2214	i	
0.4		7.94289	7.9429	7.94295	7.9346	l	
0.5	7.8025					a	7.78
0.6		7.66181	7.6618	7.66180	7.6406	b	
0.8		7.38858	7.3886	7.38854	7.3453	l	
1.0	7.1362	7.13100	7.1310	7.13100	7.0579	e	7.12

(input=in5)

**** Remark: PATRICIA (new) runs on in5, which is debugged and suppresses fit! PATRIS runs with its bend fully corrected, with a longer printout of its fixed point and with a more stringent exiting criterion (more iterations) for its Newton closed orbit finder.

Table C7

Dependence of Momentum Dispersion Function on Momentum Deviation at the First Crossing Point (CRIO)

dp/p	SYNCH	PATRIS (eta cor.)	PATRICIA	FASTRAC	MAD	TEAPOT	ORBIT
-1.0	0.0219	0.02050	N	0.02050	0.020	N	N
-0.8		0.01751	o	0.01751	0.017	o	o
-0.6		0.01478	t	0.01478	0.015	t	t
-0.5	0.0143						
-0.4		0.01228	a	0.01227	0.012	a	a
-0.2		0.00997	v	0.00997	0.010	v	v
0.0	0.0079	0.00786	a	0.00786	0.008	a	a
0.2		0.00599	i	0.00599	0.006	i	i
0.4		0.00441	l	0.00441	0.004	l	l
0.5	0.0029		a			a	a
0.6		0.00320	b	0.00320	0.003	b	b
0.8		0.00247	l	0.00247	0.002	l	l
1.0	0.0004	0.00237	e	0.00237	0.002	e	e

**** Remark: Only SYNCH, FASTRAC, MAD and modified PATRIS supplied this quantity!

Table C8

Horizontal Beta Function Dependence on Momentum Deviation at the First Crossing Point (CRIO)

dp/p	SYNCH	PATRIS (strg.f.p.)	PATRICIA (new,nofit)	FASTRAC	MAD	TEAPOT	ORBIT
-1.0	3.9981	3.99002	3.9900	3.99000	4.0016	N	3.99
-0.8		3.76076	3.7608	3.76077	3.7646	o	
-0.6		3.55956	3.5596	3.55951	3.5605	t	
-0.5	3.4714						3.47
-0.4		3.38684	3.3868	3.38680	3.3864	a	
-0.2		3.24364	3.2436	3.24360	3.2435	v	
0.0	3.1314	3.13145	3.1314	3.13140	3.1310	a	3.14
0.2		3.05218	3.0522	3.05219	3.0521	i	
0.4		3.00802	3.0080	3.00798	3.0060	l	
0.5	3.0007					a	3.00
0.6		3.00131	3.0013	3.00130	2.9969	b	
0.8		3.03431	3.0343	3.03428	3.0270	l	
1.0	3.1121	3.10901	3.1090	3.10898	3.0977	e	3.11

(input=in5)

**** Remark: PATRICIA (new) runs on in5, which is debugged and suppresses fit! PATRIS runs with its bend fully corrected, with a longer printout of its fixed point and with a more stringent exiting criterion (more iterations) for its Newton closed orbit finder.

Table C9

Vertical Beta Function Dependence on Momentum Deviation at the First Crossing Point (CRIO)

dp/p	SYNCH	PATRIS (strg.f.p.)	PATRICIA (new,nofit)	FASTRAC	MAD	TEAPOT	ORBIT
-1.0	3.8204	3.81746	3.8175	3.81744	3.8244	N	3.82
-0.8		3.63805	3.6380	3.6D806	3.6406	o	
-0.6		3.47803	3.4780	3.47801	3.4780	t	
-0.5	3.4067						3.40
-0.4		3.33916	3.3392	3.33919	3.3386	a	
-0.2		3.22347	3.2235	3.22344	3.2235	v	
0.0	3.1331	3.13312	3.1331	3.13310	3.1330	a	3.13
0.2		3.07034	3.0703	3.07033	3.0701	i	
0.4		3.03734	3.0373	3.03730	3.0351	l	
0.5	3.0331					a	3.03
0.6		3.03628	3.0363	3.03631	3.0321	b	
0.8		3.06921	3.0692	3.06926	3.0613	l	
1.0	3.1401	3.13793	3.1379	3.13797	3.1260	e	3.14
			(input=in5)				

**** Remark: PATRICIA (new) runs on in5, which is debugged and suppresses fit! PATRIS runs with its bend fully corrected, with a longer printout of its fixed point and with a more stringent exiting criterion (more iterations) for its Newton closed orbit finder.

Table C10

Dependence of Momentum Dispersion Function on Momentum Deviation at the Second Crossing Point (CROI)

dp/p	SYNCH	PATRIS (eta cor.)	PATRICIA	FASTRAC	MAD	TEAPOT	ORBIT
-1.0	0.0197	0.01831	N	0.01831	0.018	N	N
-0.8		0.01654	o	0.01654	0.016	o	o
-0.6		0.01462	t	0.01462	0.015	t	t
-0.5	0.0144						
-0.4		0.01267	a	0.01267	0.013	a	a
-0.2		0.01081	v	0.01080	0.011	v	v
0.0	0.0091	0.00912	a	0.00912	0.009	a	a
0.2		0.00774	i	0.00774	0.008	i	i
0.4		0.00677	l	0.00677	0.007	l	l
0.5	0.0056		a			a	a
0.6		0.00634	b	0.00634	0.006	b	b
0.8		0.00662	l	0.00662	0.006	l	l
1.0	0.0059	0.00778	e	0.00778	0.007	e	e

**** Remark: Only SYNCH, FASTRAC, MAD and modified PATRIS supplied this quantity!

Table C11

Horizontal Beta Function Dependence on Momentum Deviation at the Second Crossing Point (CROI)

dp/p	SYNCH	PATRIS (strg.f.p.)	PATRICIA (new,nofit)	FASTRAC	MAD	TEAPOT	ORBIT
-1.0	3.9615	3.95378	3.9538	3.95376	3.9640	N	3.95
-0.8		3.73568	3.7357	3.73567	3.7388	o	
-0.6		3.54126	3.5413	3.54122	3.5416	t	
-0.5	3.4555						3.46
-0.4		3.37335	3.3734	3.37335	3.3735	a	
-0.2		3.23541	3.2354	3.23542	3.2355	v	
0.0	3.1314	3.13145	3.1314	3.13140	3.1310	a	3.14
0.2		3.06592	3.0659	3.06592	3.0651	i	
0.4		3.04365	3.0436	3.04363	3.0421	l	
0.5	3.0511					a	3.05
0.6		3.06964	3.0696	3.06961	3.0663	b	
0.8		3.14885	3.1488	3.14889	3.1429	l	
1.0	3.2885	3.28580	3.2858	3.28583	3.2785	e	3.29
			(input=in5)				

**** Remark: PATRICIA (new) runs on in5, which is debugged and suppresses fit! PATRIS runs with its bend fully corrected, with a longer printout of its fixed point and with a more stringent exiting criterion (more iterations) for its Newton closed orbit finder.

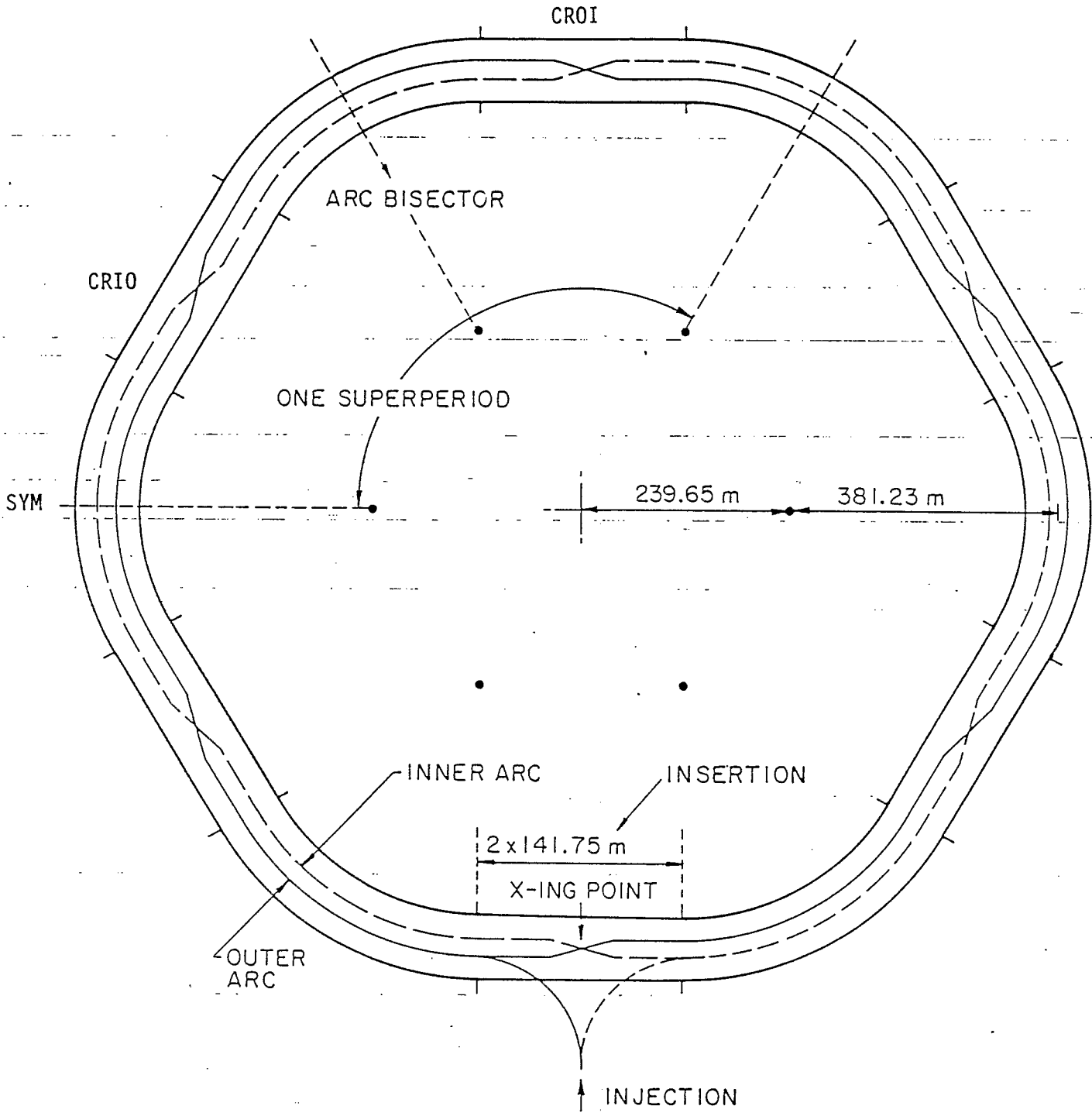
Table C12

Vertical Beta Function Dependence on Momentum
Deviation at the Second Crossing Point (CROI)

dp/p	SYNCH	PATRIS (strg.f.p.)	PATRICIA (new,nofit)	FASTRAC	MAD	TEAPOT	ORBIT
-1.0	3.7710	3.76822	3.7682	3.76824	3.7739	N	3.77
-0.8		3.59885	3.5989	3.59888	3.6000	o	
-0.6		3.44642	3.4464	3.44640	3.4462	t	
-0.5	3.3787						3.37
-0.4		3.31501	3.3150	3.31499	3.3147	a	
-0.2		3.20900	3.2090	3.20897	3.2086	v	
0.0	3.1329	3.13292	3.1329	3.13290	3.1330	a	3.13
0.2		3.09136	3.0914	3.09137	3.0912	i	
0.4		3.08892	3.0889	3.08891	3.0873	l	
0.5	3.1042					a	3.10
0.6		3.13011	3.1301	3.13007	3.1266	b	
0.8		3.21924	3.2192	3.21925	3.2145	l	
1.0	3.3620	3.36022	3.3602	3.36027	3.3542	e	3.36
			(input=in5)				

**** Remark: PATRICIA (new) runs on in5, which is debugged and suppresses fit! PATRIS runs with its bend fully corrected, with a longer printout of its fixed point and with a more stringent exiting criterion (more iterations) for its Newton closed orbit finder.

Fig. 1



Layout of the Collider.

Fig. 2

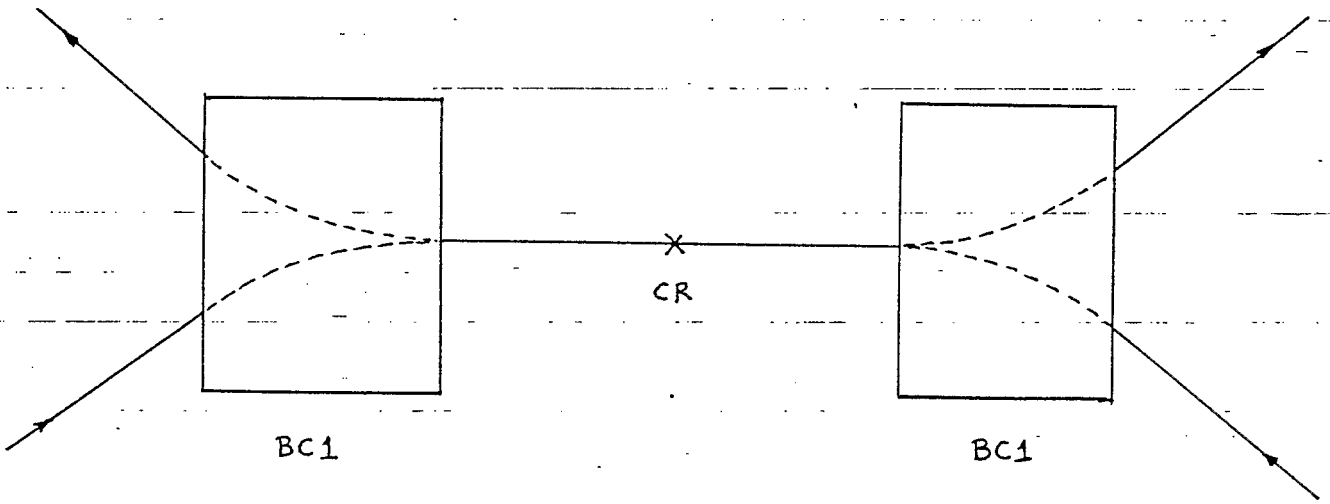


Fig. U1

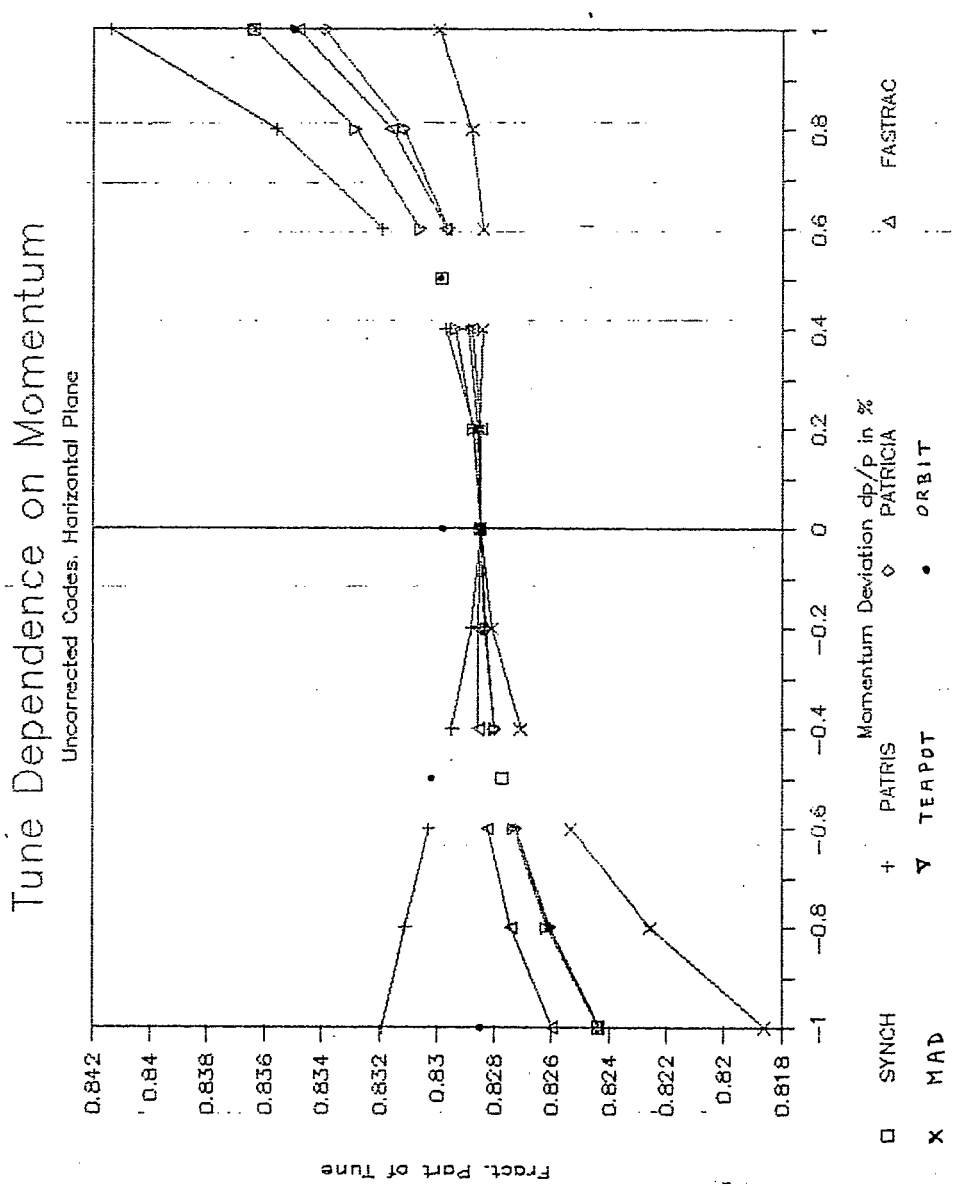

Tune Dependence on Momentum
Uncorrected Codes, Horizontal Plane
Fract. Part of Tune
0.842
0.84
0.838
0.836
0.834
0.832
0.83
0.828
0.826
0.824
0.822
0.82
0.818
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
Momentum Deviation dp/p in %
SYNCH
PATRIS
PATRICIA
FASTRAC
MAD
TEAPOT
ORBIT

Fig U2

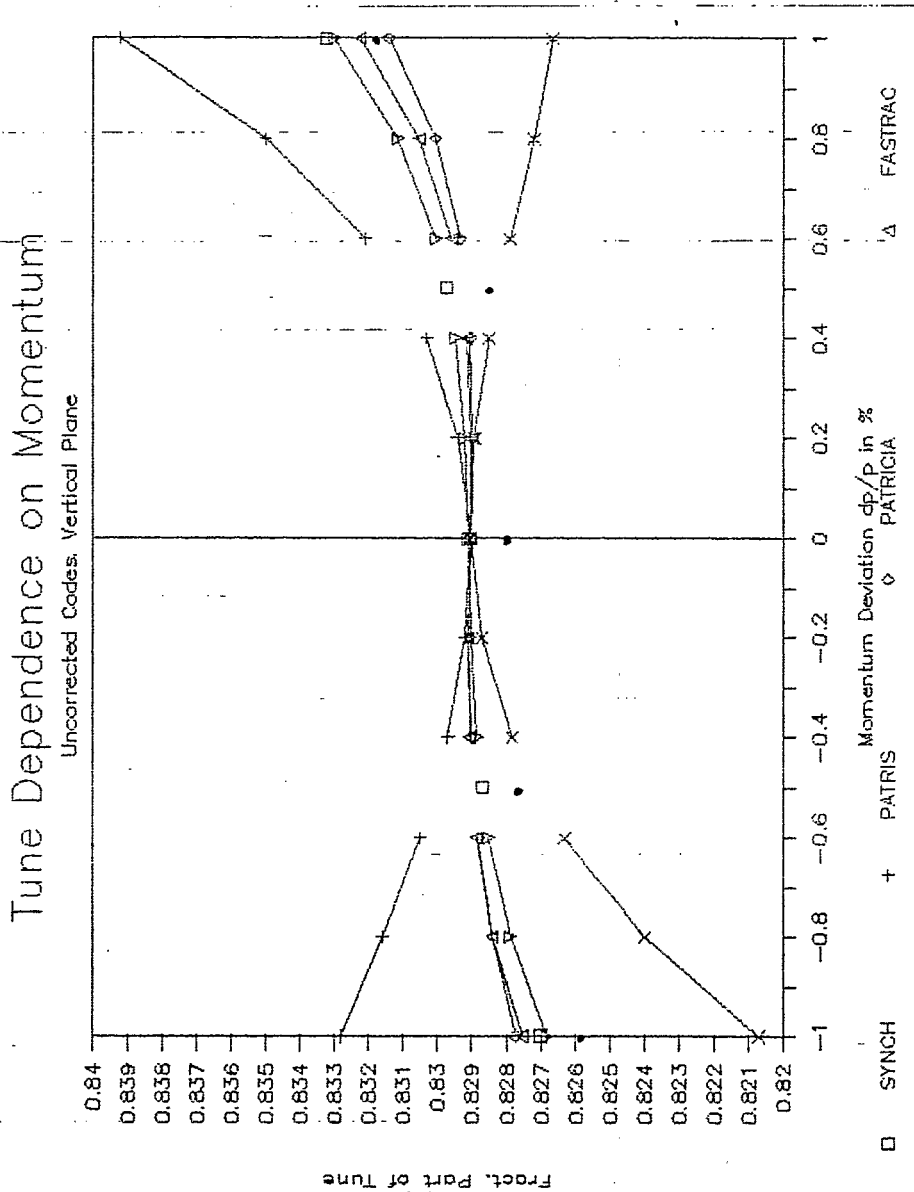

Tune Dependence on Momentum
Uncorrected Codes, Vertical Plane
Fract. Part of Tune
0.84
0.839
0.838
0.837
0.836
0.835
0.834
0.833
0.832
0.831
0.83
0.829
0.828
0.827
0.826
0.825
0.824
0.823
0.822
0.821
0.82
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
Momentum Deviation dp/p in %
□ SYNCH
+ PATRIS
◇ PATRICIA
△ FASTRAC
× MAD
▽ TEAPOT
• ORBIT

Fig. U3

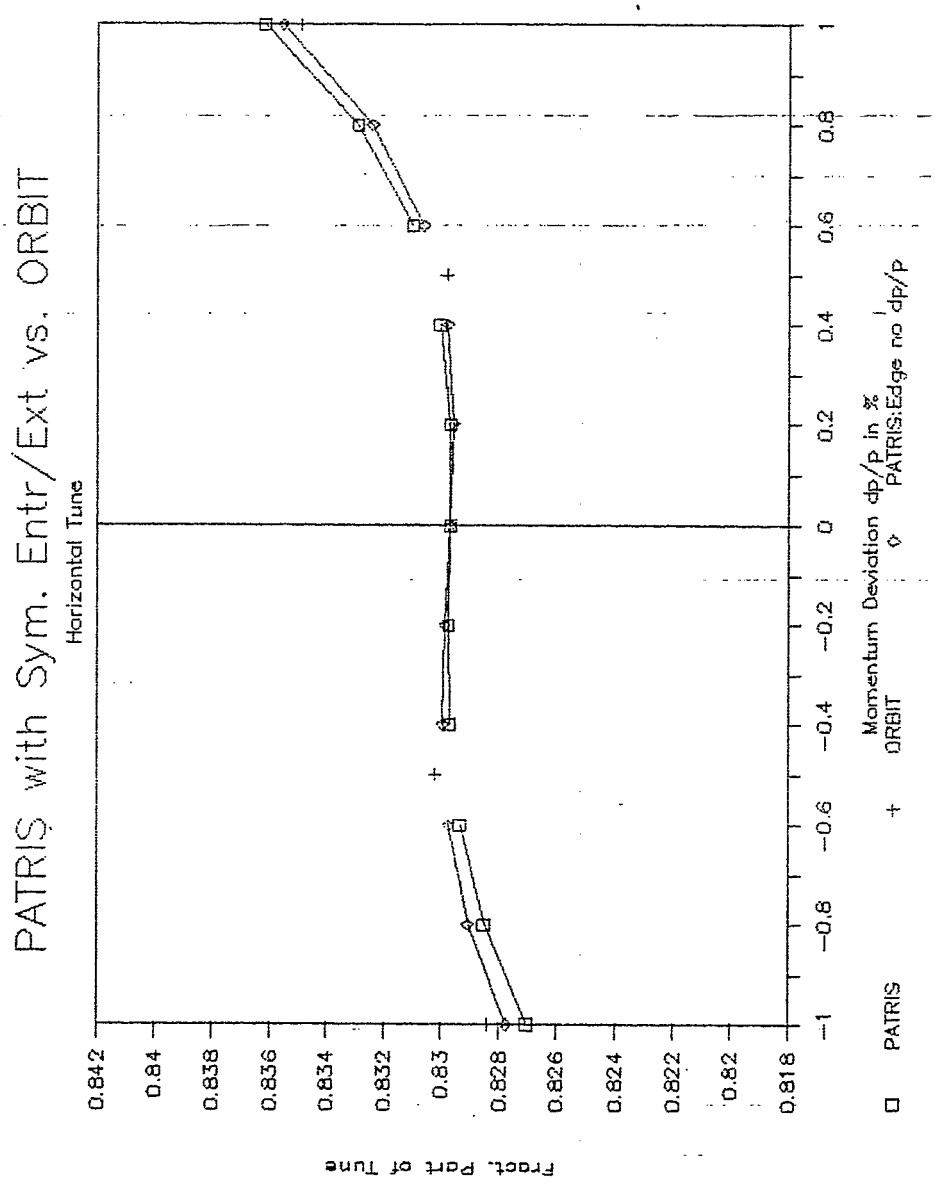

PATRIS with Sym. Entr/Ext vs. ORBIT
Horizontal Tune
Fract. Part of Tune
0.842
0.84
0.838
0.836
0.834
0.832
0.83
0.828
0.826
0.824
0.822
0.82
0.818
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
Momentum Deviation dp/P in %
PATRIS
ORBIT
PATRIS:Edge no dp/p

Fig. U3a

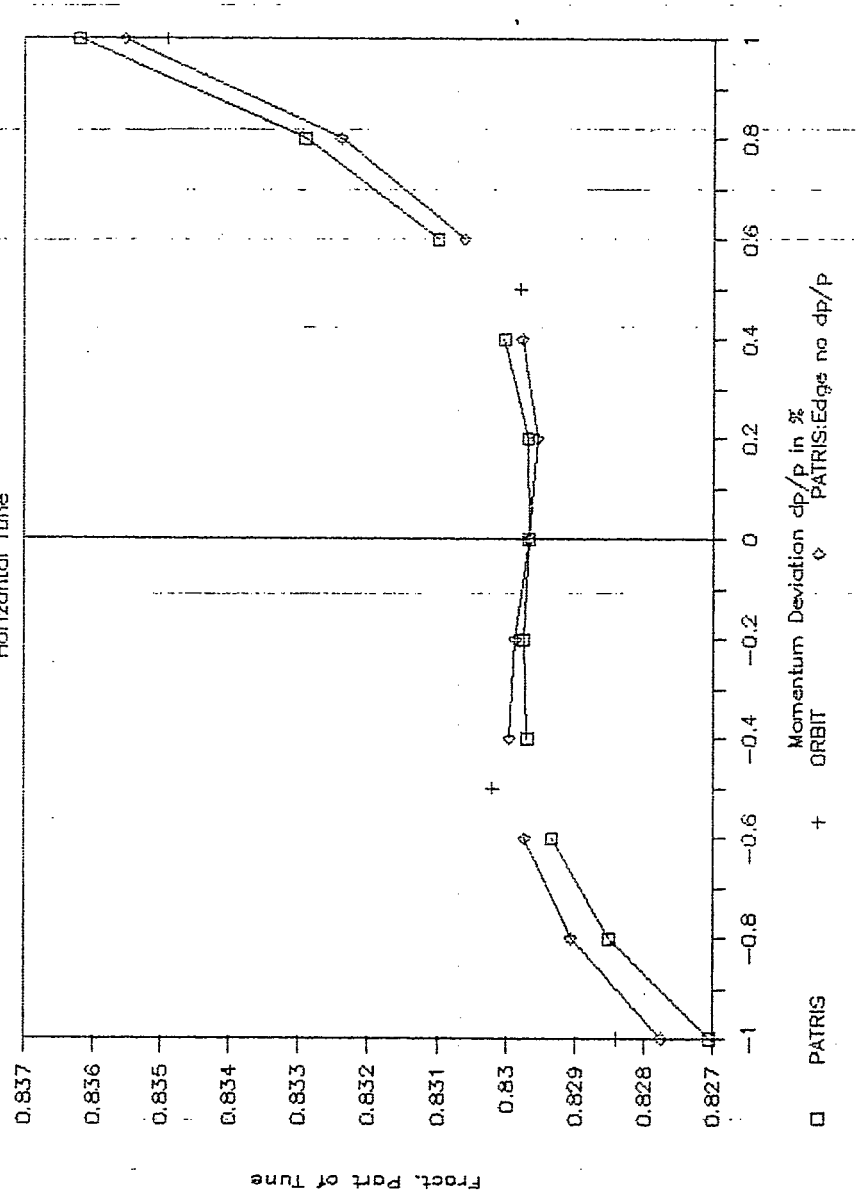

PATRIS with Sym. Entr/Ext vs. ORBIT
Horizontal Tune
Fract. Part of Tune
0.837
0.836
0.835
0.834
0.833
0.832
0.831
0.83
0.829
0.828
0.827
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
Momentum Deviation dp/p in %
PATRIS
ORBIT
PATRIS:Edge no dp/p

Fig U4

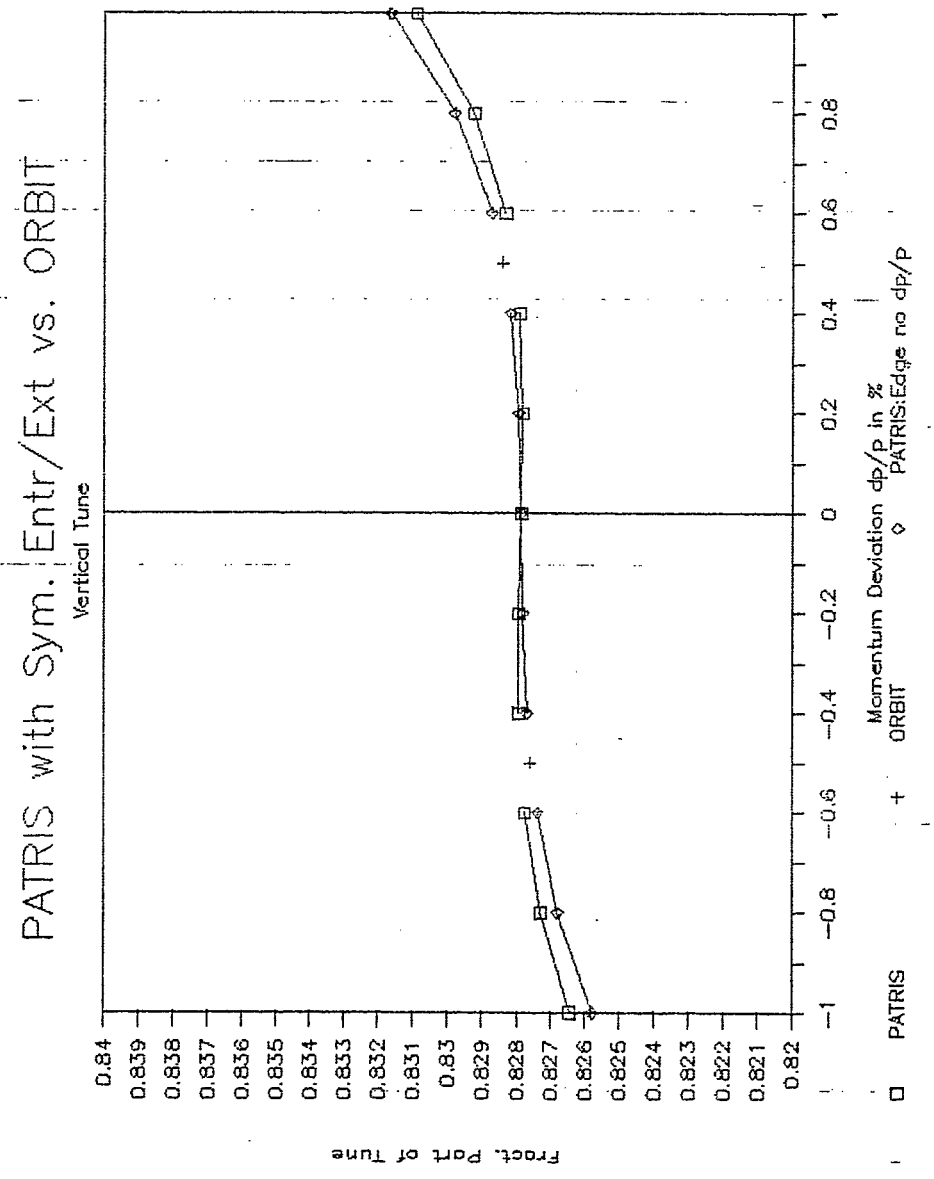

PATRIS with Sym. Entr/Ext vs. ORBIT
Vertical Tune
Fract. Part of Tune
0.84
0.839
0.838
0.837
0.836
0.835
0.834
0.833
0.832
0.831
0.83
0.829
0.828
0.827
0.826
0.825
0.824
0.823
0.822
0.821
0.82
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
Momentum Deviation dp/p in %
PATRIS
ORBIT
PATRIS:Edge no dp/p

Fig. U4a

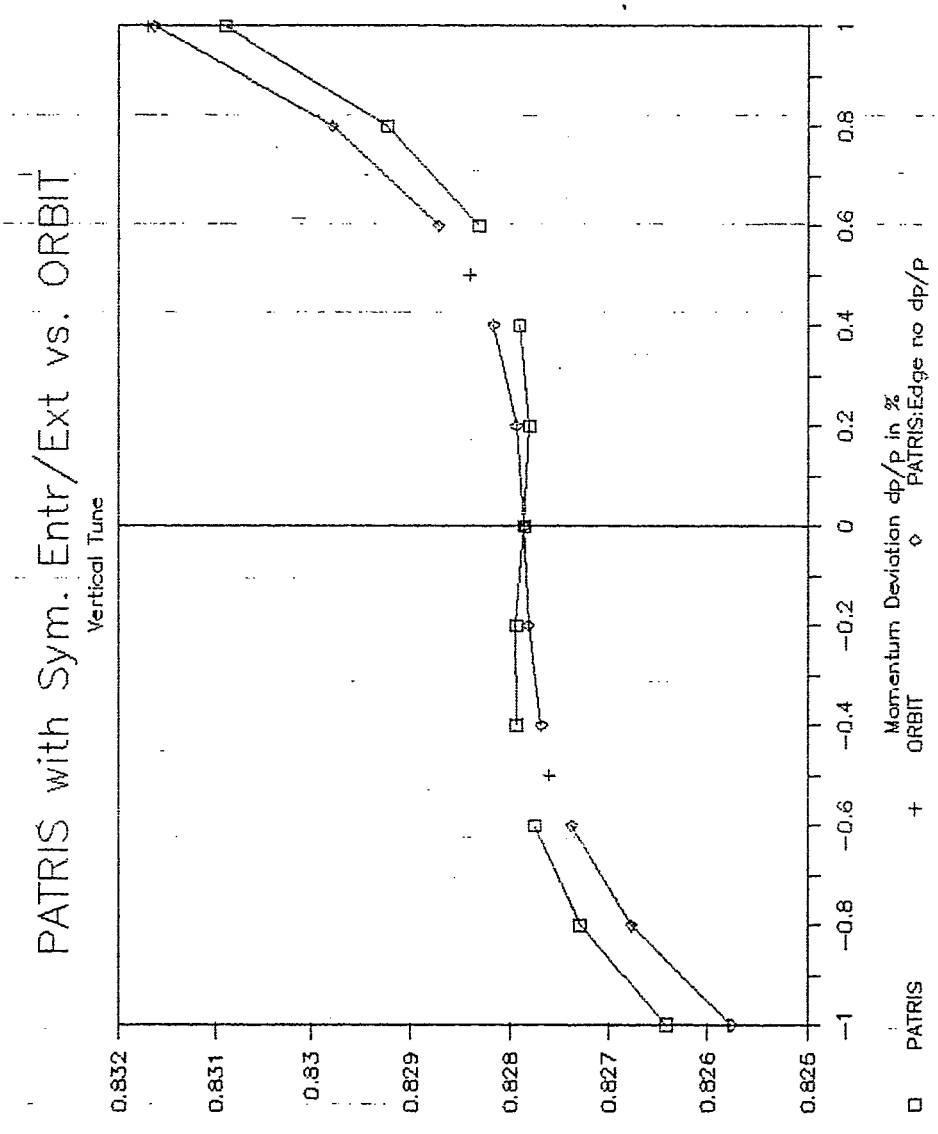

PATRIS with Sym. Entr/Ext vs. ORBIT
Vertical Tune
Fract. Part of Tune
0.832
0.831
0.83
0.829
0.828
0.827
0.826
0.825
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
Momentum Deviation dp/p in %
PATRIS
ORBIT
PATRIS:Edge no dp/p

Fig. U5

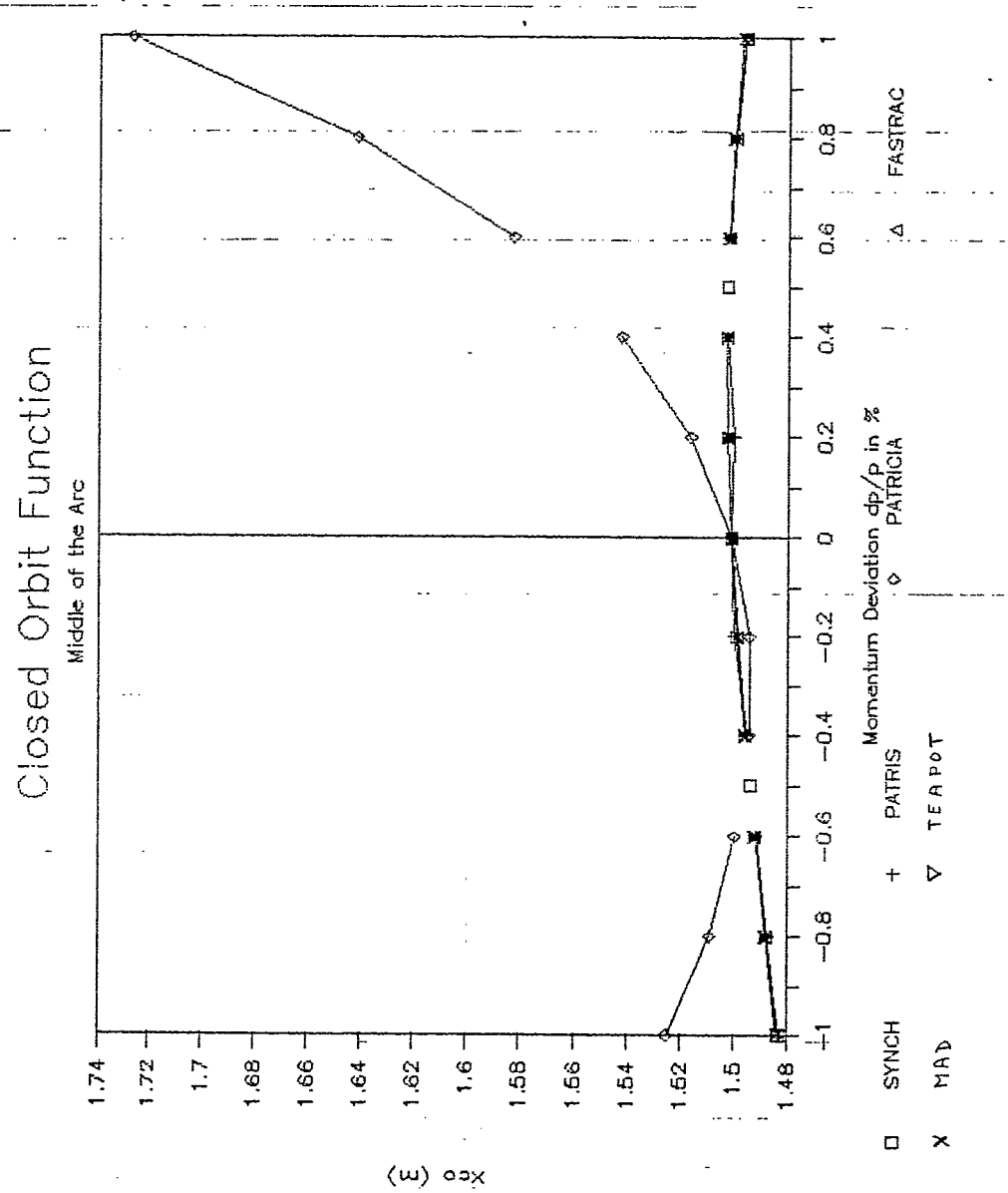

Closed Orbit Function
Middle of the Arc
Momentum Deviation dp/p in %
Xco (m)
SYNCH
PATRIS
TEAPOT
MAD
PATRICIA
FASTRAC

Fig. U5a

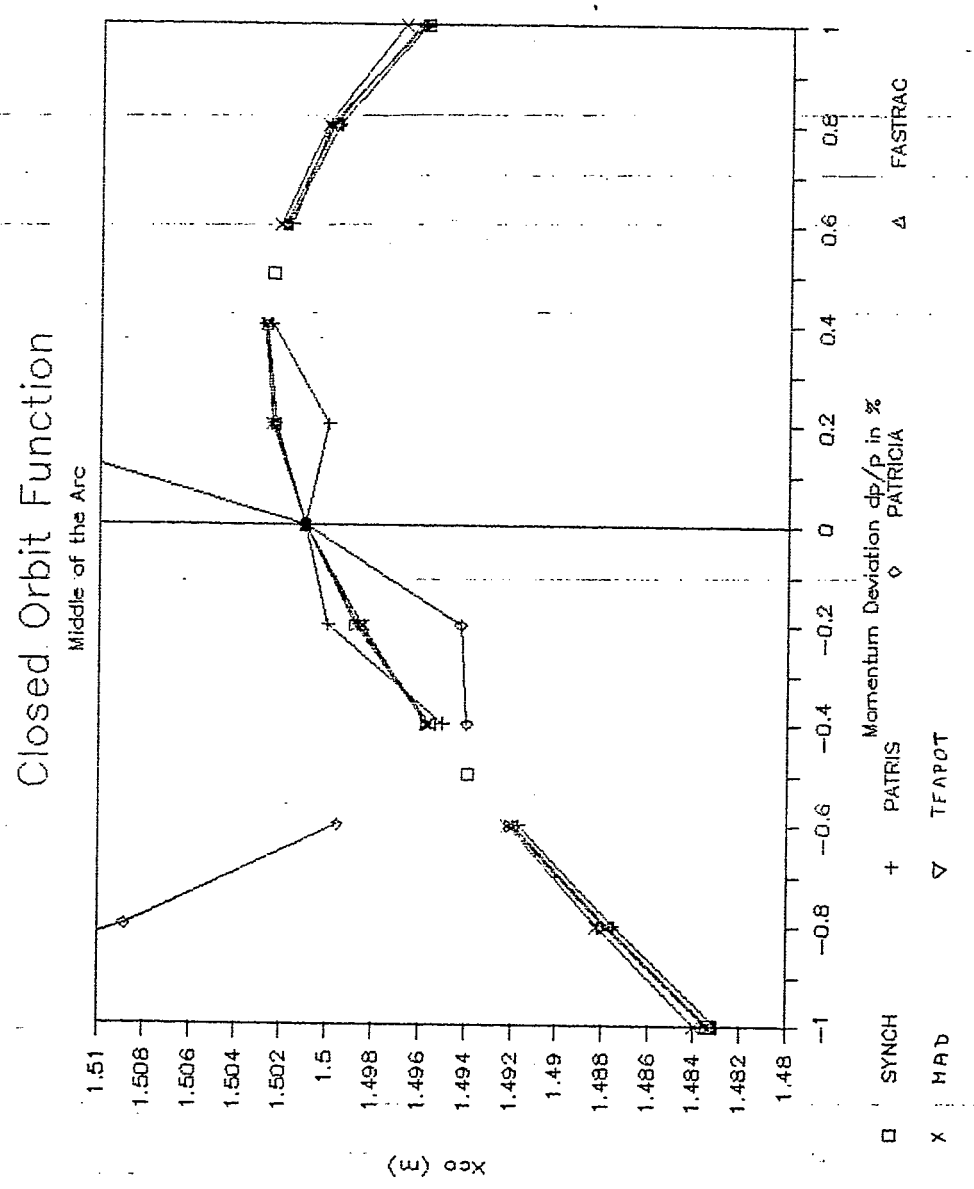

Closed Orbit Function
Middle of the Arc
Xco (m)
1.51
1.508
1.506
1.504
1.502
1.5
1.498
1.496
1.494
1.492
1.49
1.488
1.486
1.484
1.482
1.48
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
Momentum Deviation dp/p in %
SYNCH
PATRIS
TFAPOT
MAD
PATRICIA
FASTRAC

Fig. U5b

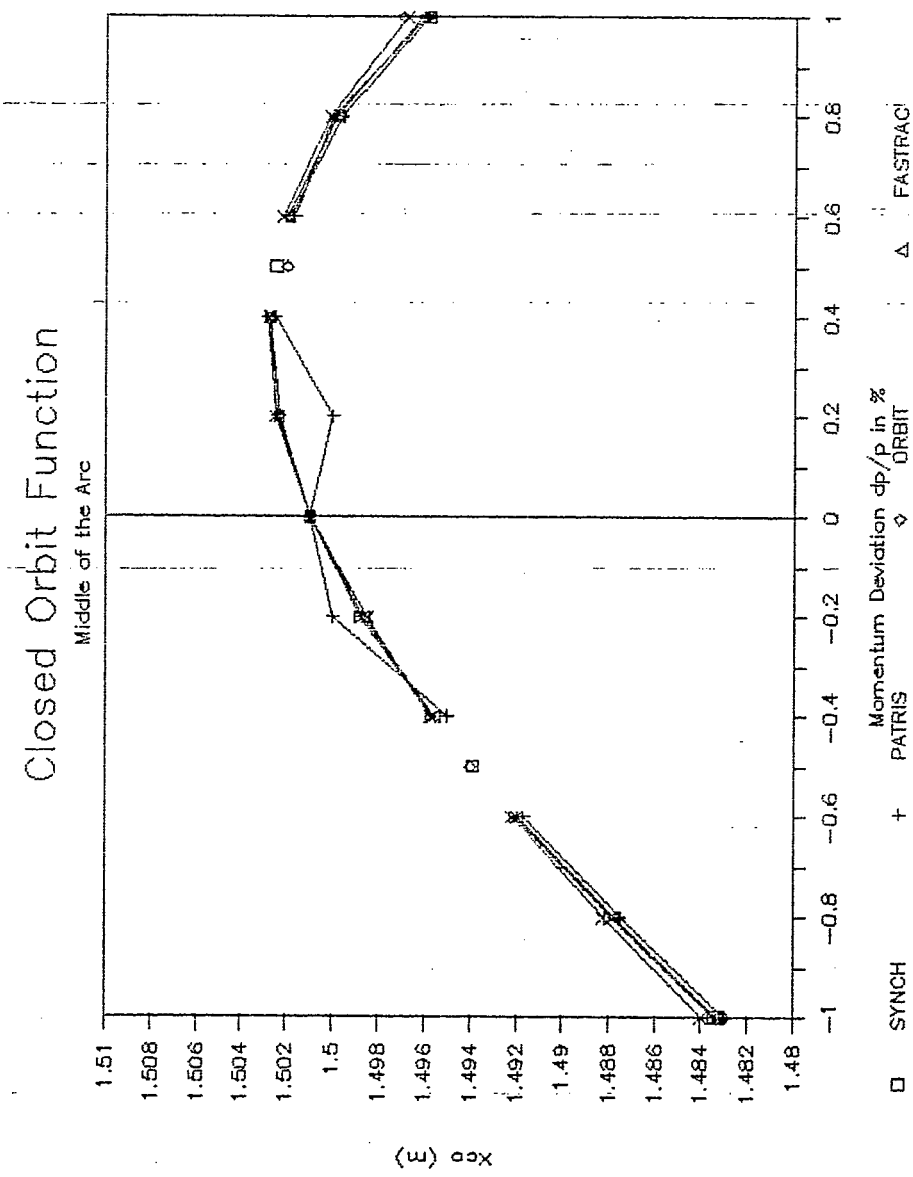

Closed Orbit Function
Middle of the Arc
Momentum Deviation dp/p in %
Xco (m)
SYNCH
PATRIS
MAD
TEAPOT
ORBIT
FASTRAC

Fig U6

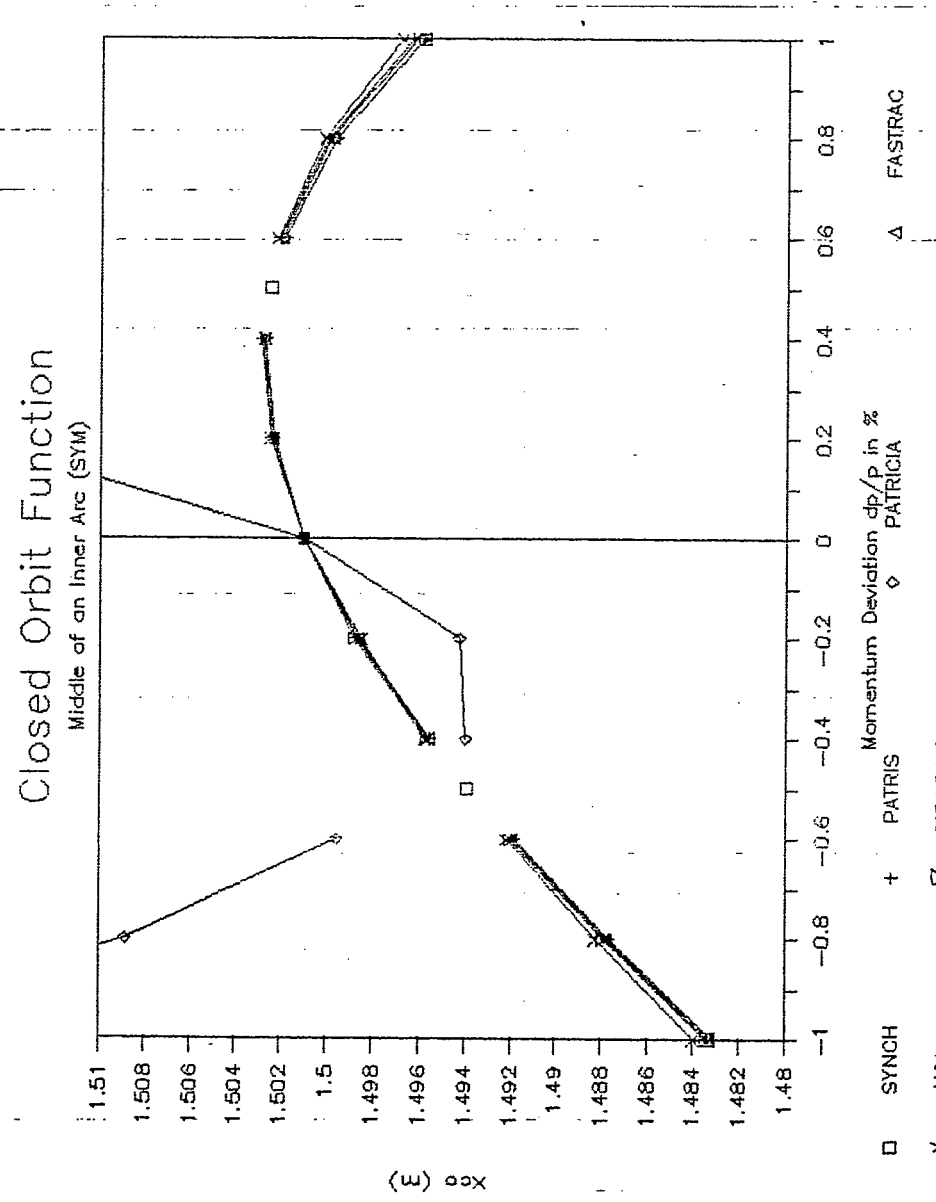

Closed Orbit Function
Middle of an Inner Arc (SYM)
Xco (m)
1.51
1.508
1.506
1.504
1.502
1.5
1.498
1.496
1.494
1.492
1.49
1.488
1.486
1.484
1.482
1.48
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
Momentum Deviation dp/p in %
SYNCH
PATRIS
PATRICIA
FASTRAC
MAD
TEAPOT

Fig. U7

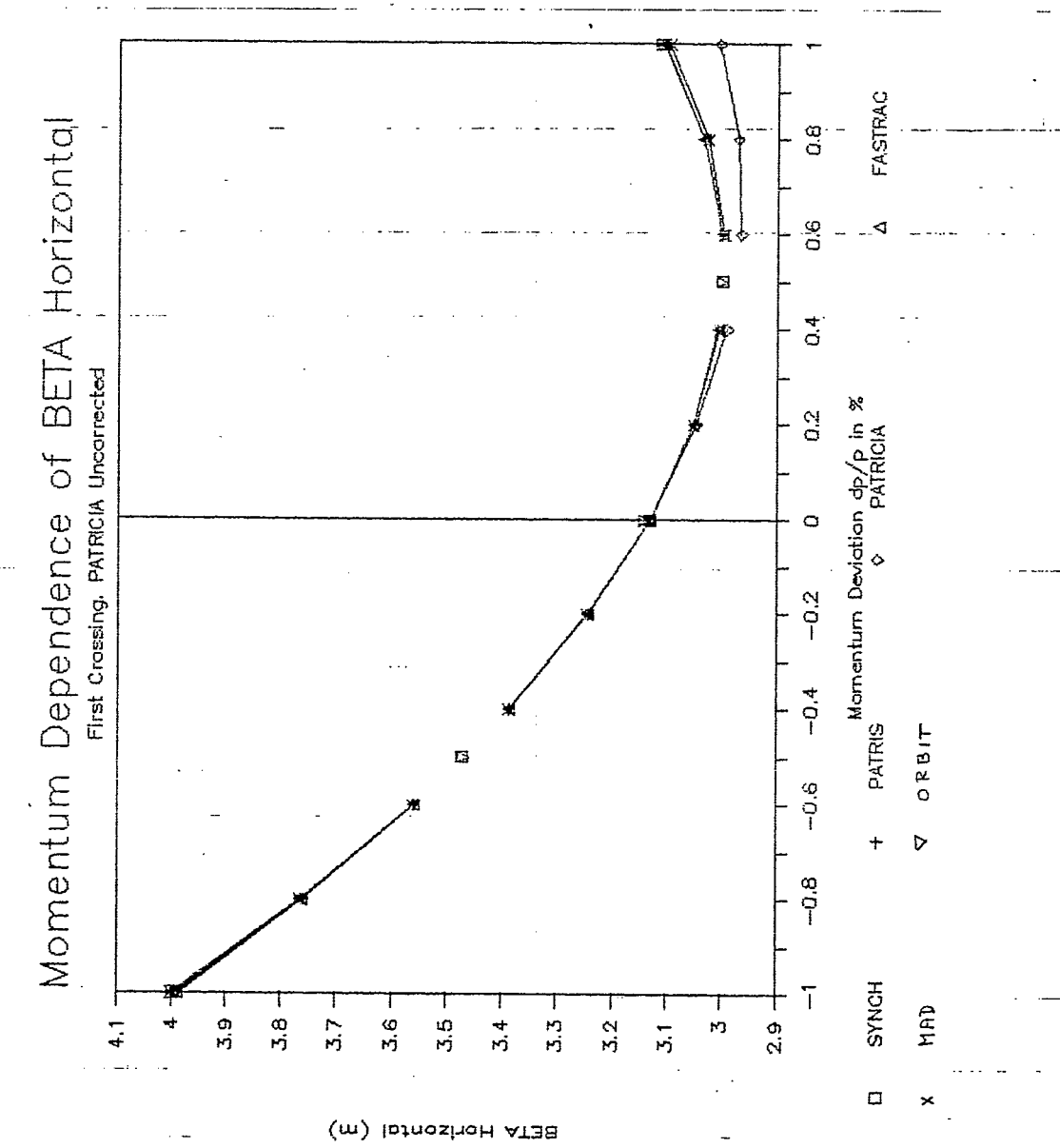
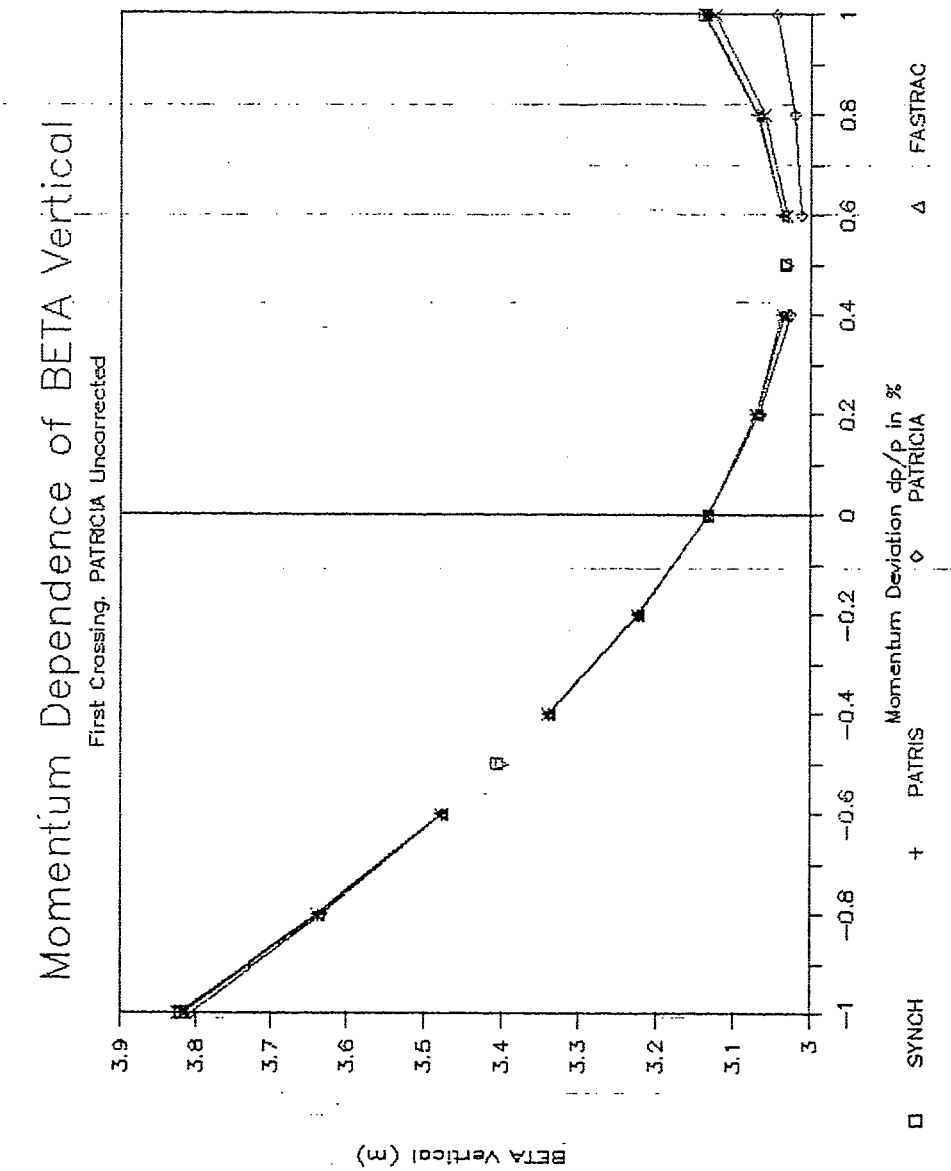

Momentum Dependence of BETA Horizontal
First Crossing, PATRICIA Uncorrected
BETA Horizontal (m)
4.1
4
3.9
3.8
3.7
3.6
3.5
3.4
3.3
3.2
3.1
3
2.9
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
Momentum Deviation dp/p in %
SYNCH
PATRIS
ORBIT
MAD
PATRICIA
FASTRAC

Fig. U8


Momentum Dependence of BETA Vertical
First Crossing, PATRICIA Uncorrected
BETA Vertical (m)
3.9
3.8
3.7
3.6
3.5
3.4
3.3
3.2
3.1
3
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
Momentum Deviation dp/p in %
SYNCH
PATRIS
ORBIT
MAD
PATRICIA
FASTRAC

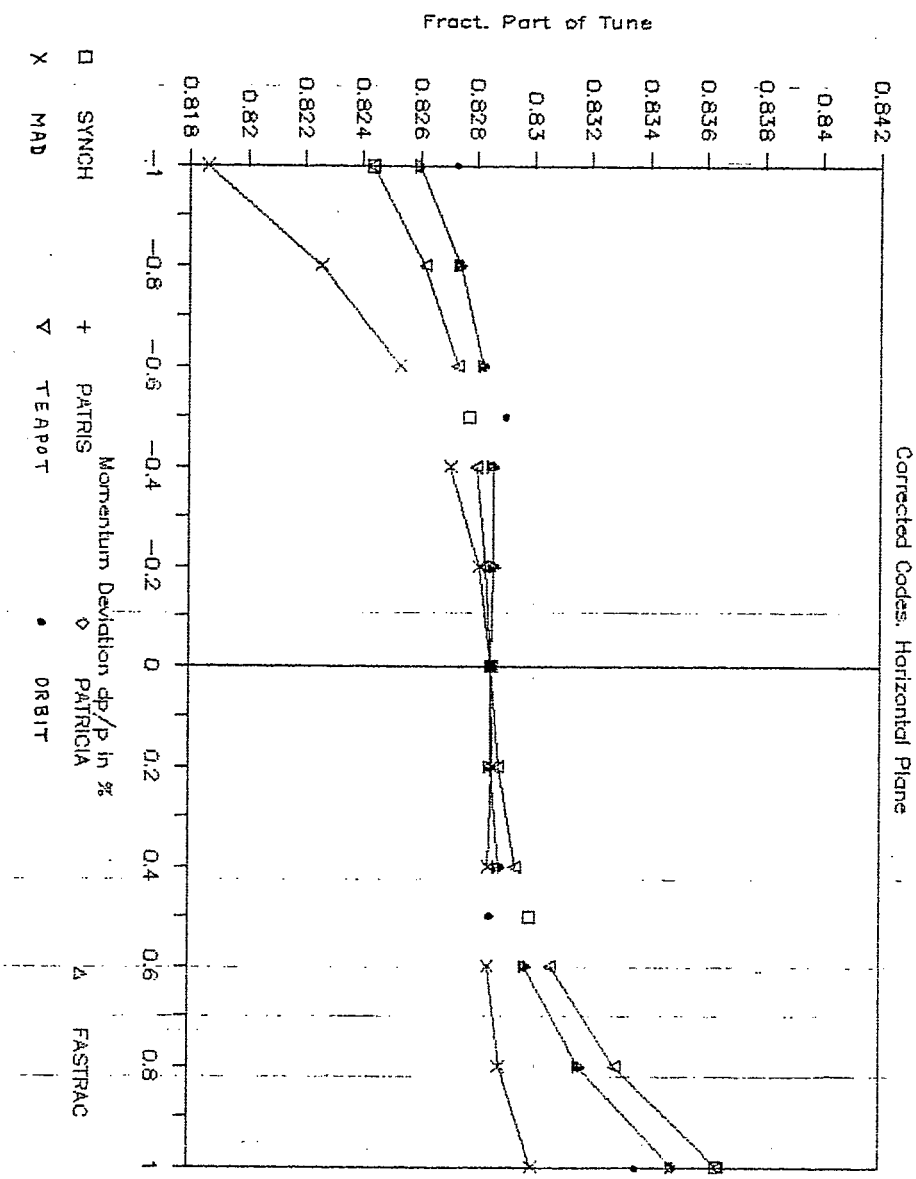

Tune Dependence on Momentum
Corrected Codes, Horizontal Plane
Fract. Part of Tune
0.842
0.84
0.838
0.836
0.834
0.832
0.83
0.828
0.826
0.824
0.822
0.82
0.818
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Momentum Deviation dp/p in %
□ SYNCH
+ PATRIS
◇ PATRICIA
△ FASTRAC
X MAD
▽ TEAPOT
• ORBIT

Fig. C1

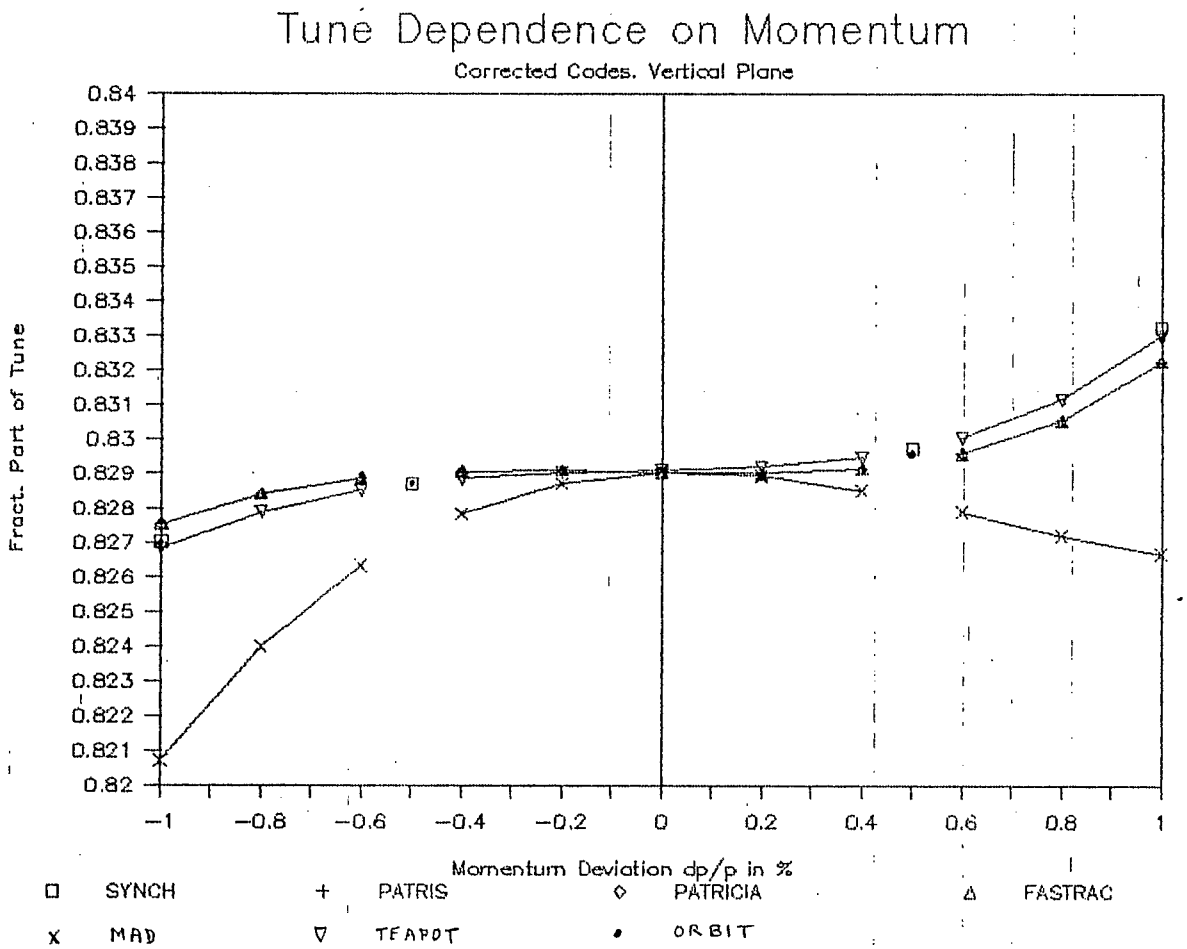

Tune Dependence on Momentum
Corrected Codes. Vertical Plane
Fract. Part of Tune
0.84
0.839
0.838
0.837
0.836
0.835
0.834
0.833
0.832
0.831
0.83
0.829
0.828
0.827
0.826
0.825
0.824
0.823
0.822
0.821
0.82
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
Momentum Deviation dp/p in %
SYNCH
PATRIS
PATRICIA
FASTRAC
MAD
TEAPOT
ORBIT

Fig C2

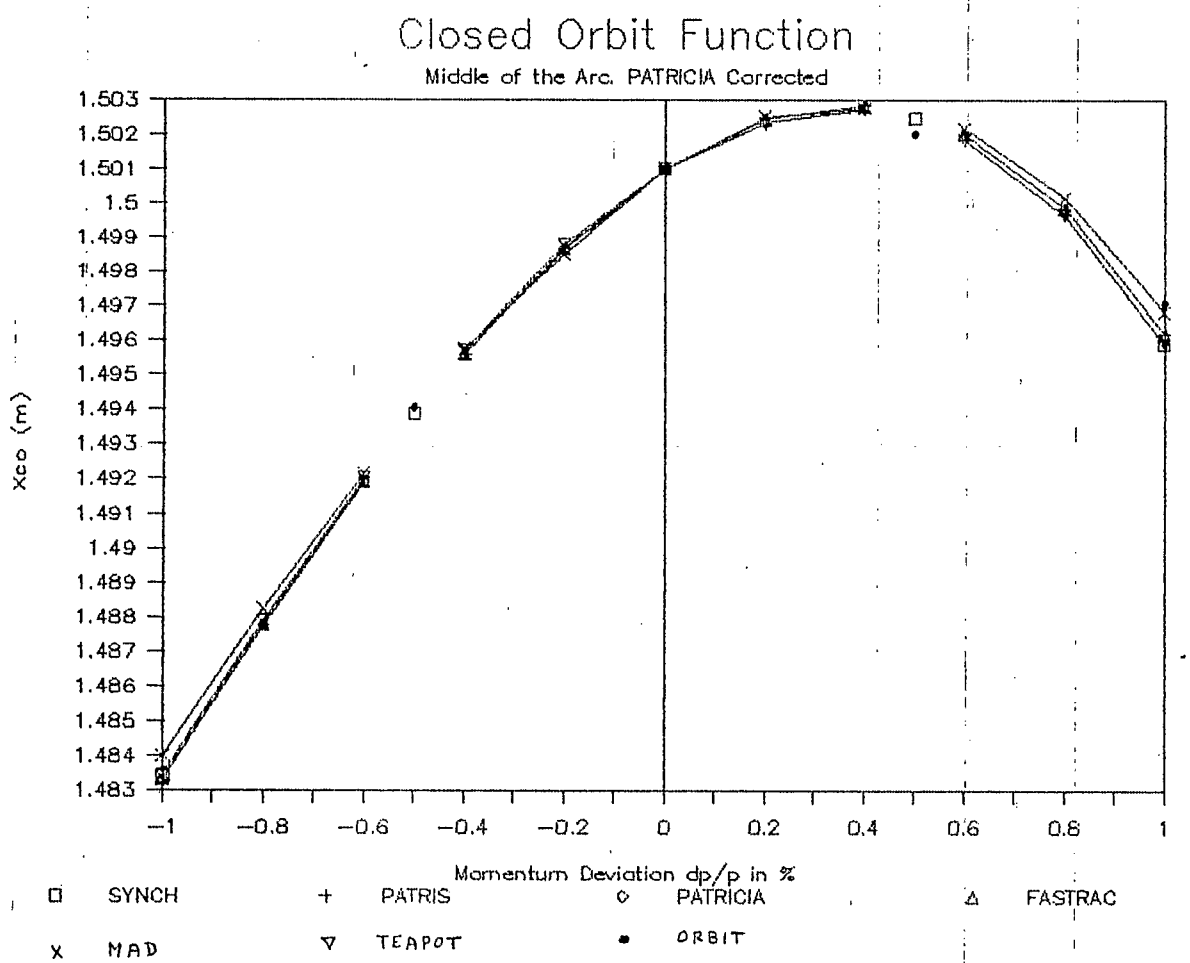

Closed Orbit Function
Middle of the Arc. PATRICIA Corrected
Xco (m)
1.503
1.502
1.501
1.5
1.499
1.498
1.497
1.496
1.495
1.494
1.493
1.492
1.491
1.49
1.489
1.488
1.487
1.486
1.485
1.484
1.483
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
Momentum Deviation dp/p in %
SYNCH
PATRIS
PATRICIA
FASTRAC
MAD
TEAPOT
ORBIT

Fig. C3

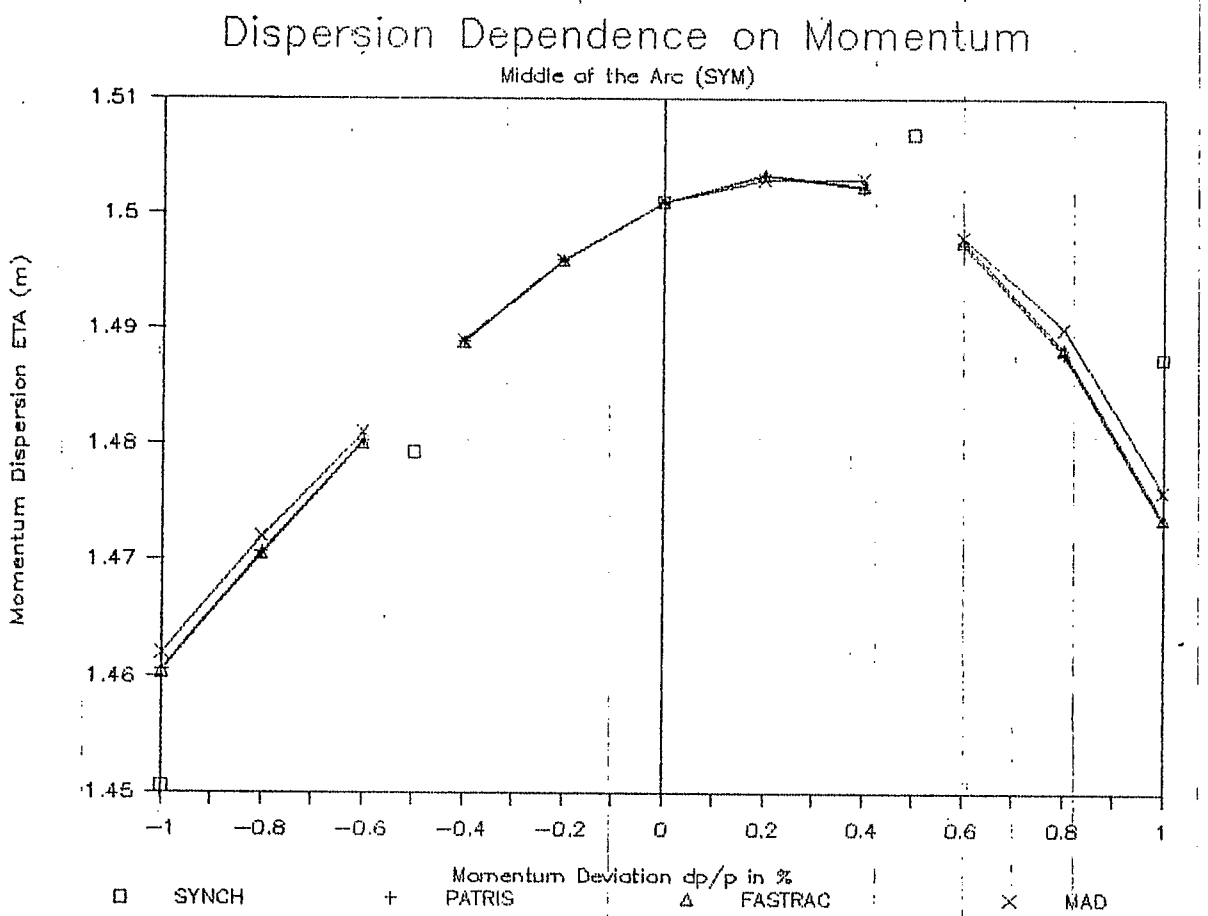

Dispersion Dependence on Momentum
Middle of the Arc (SYM)
Momentum Dispersion ETA (m)
1.51
1.5
1.49
1.48
1.47
1.46
1.45
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
Momentum Deviation dp/p in %
SYNCH
PATRIS
FASTRAC
MAD

Fig. C4

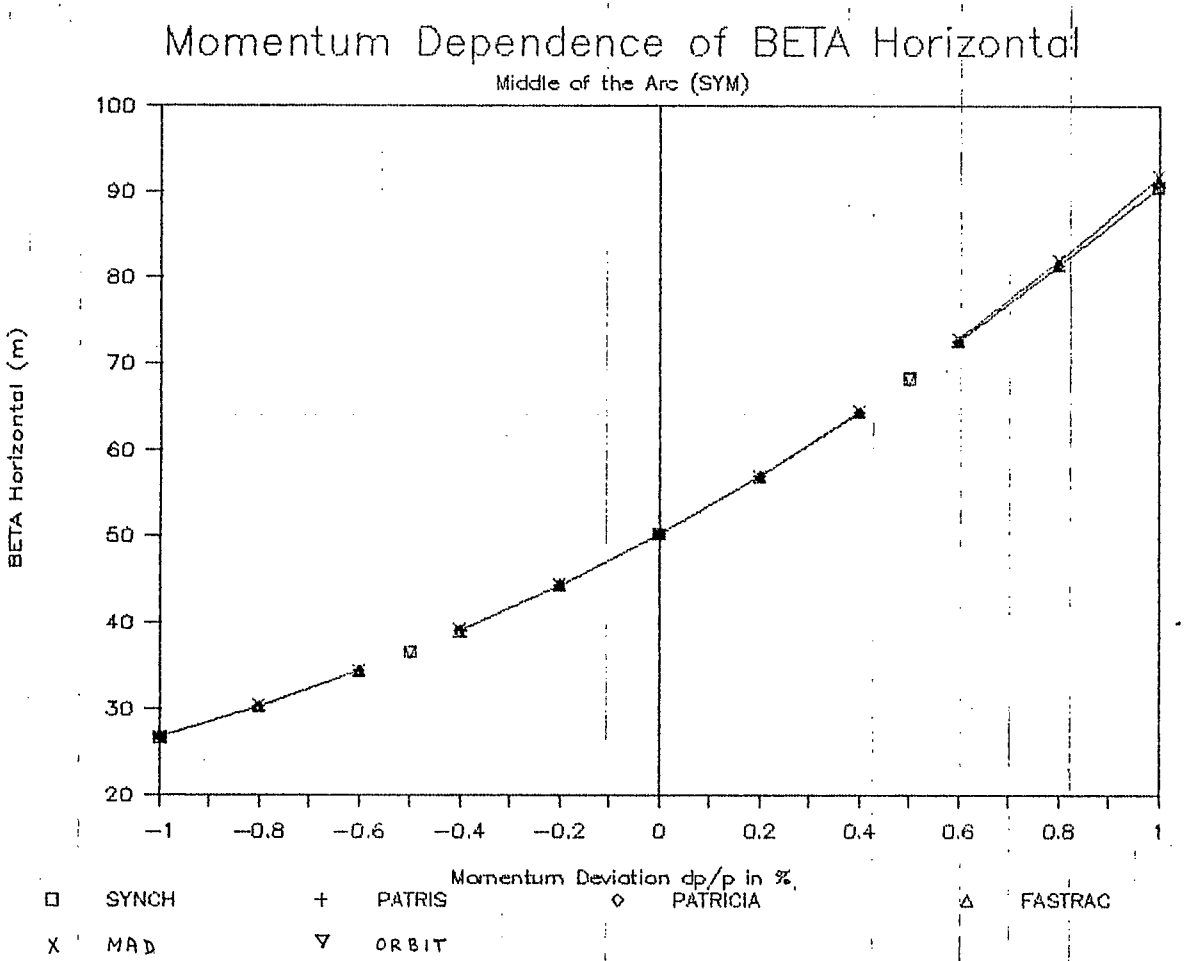

Momentum Dependence of BETA Horizontal
Middle of the Arc (SYM)
BETA Horizontal (m)
100
90
80
70
60
50
40
30
20
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
Momentum Deviation dp/p in %
SYNCH
PATRIS
PATRICIA
FASTRAC
MAD
ORBIT

Fig. C5

Momentum Dependence of BETA Vertical
Middle of the Arc (SYM)
BETA Vertical (m)
9.6
9.4
9.2
9
8.8
8.6
8.4
8.2
8
7.8
7.6
7.4
7.2
7
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Momentum Deviation dp/p in %
□ SYNCH
+ PATRIS
◇ PATRICIA
△ FASTRAC
× MAD
▽ ORBIT

Fig. C6

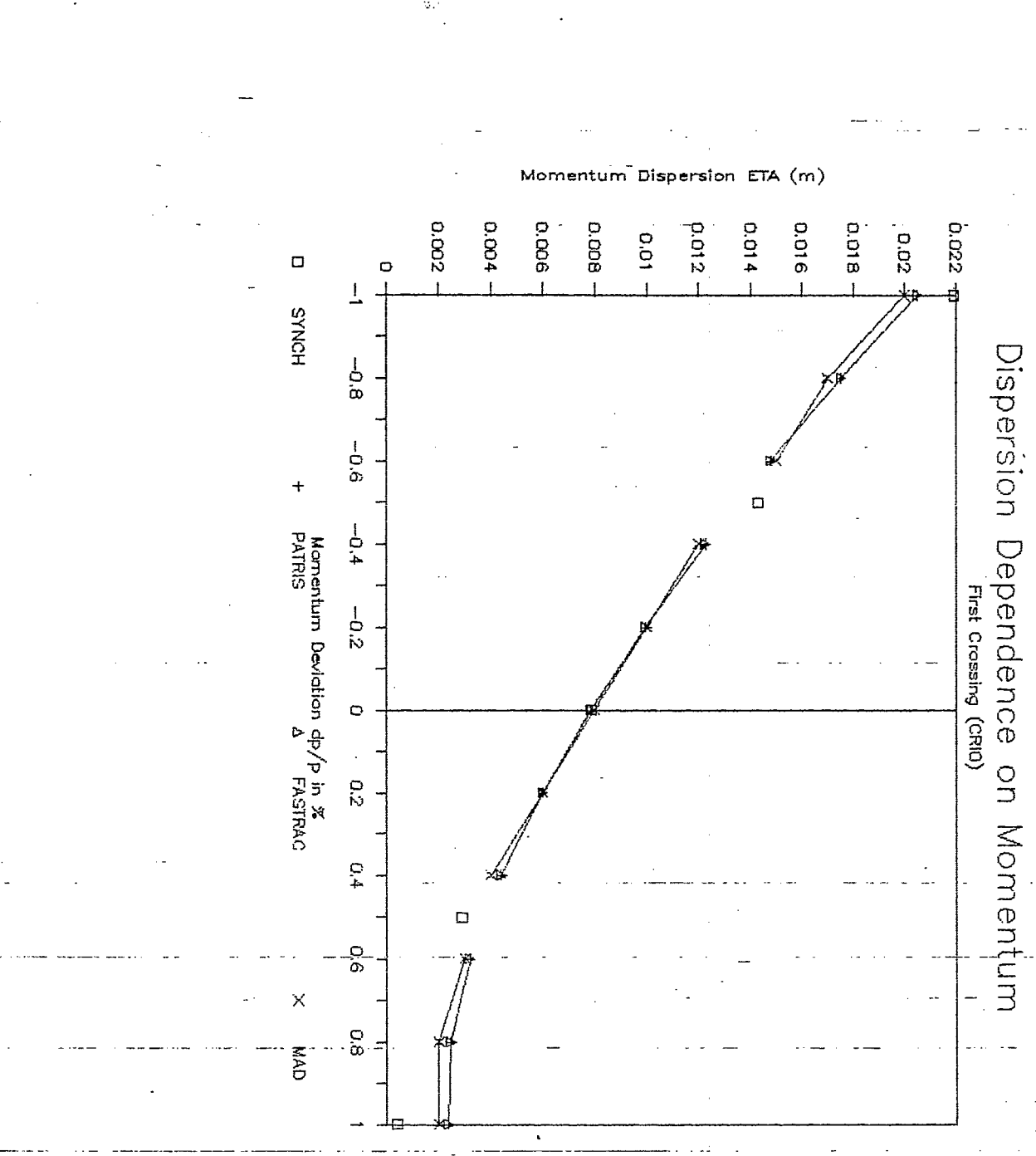

Dispersion Dependence on Momentum
Momentum Dispersion ETA (m)
0
0.002
0.004
0.006
0.008
0.01
0.012
0.014
0.016
0.018
0.02
0.022
First Crossing (CRIO)
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
Momentum Deviation dp/p in %
□ SYNCH
+ PATRIS
Δ FASTRAC
× MAD

Fig. C7

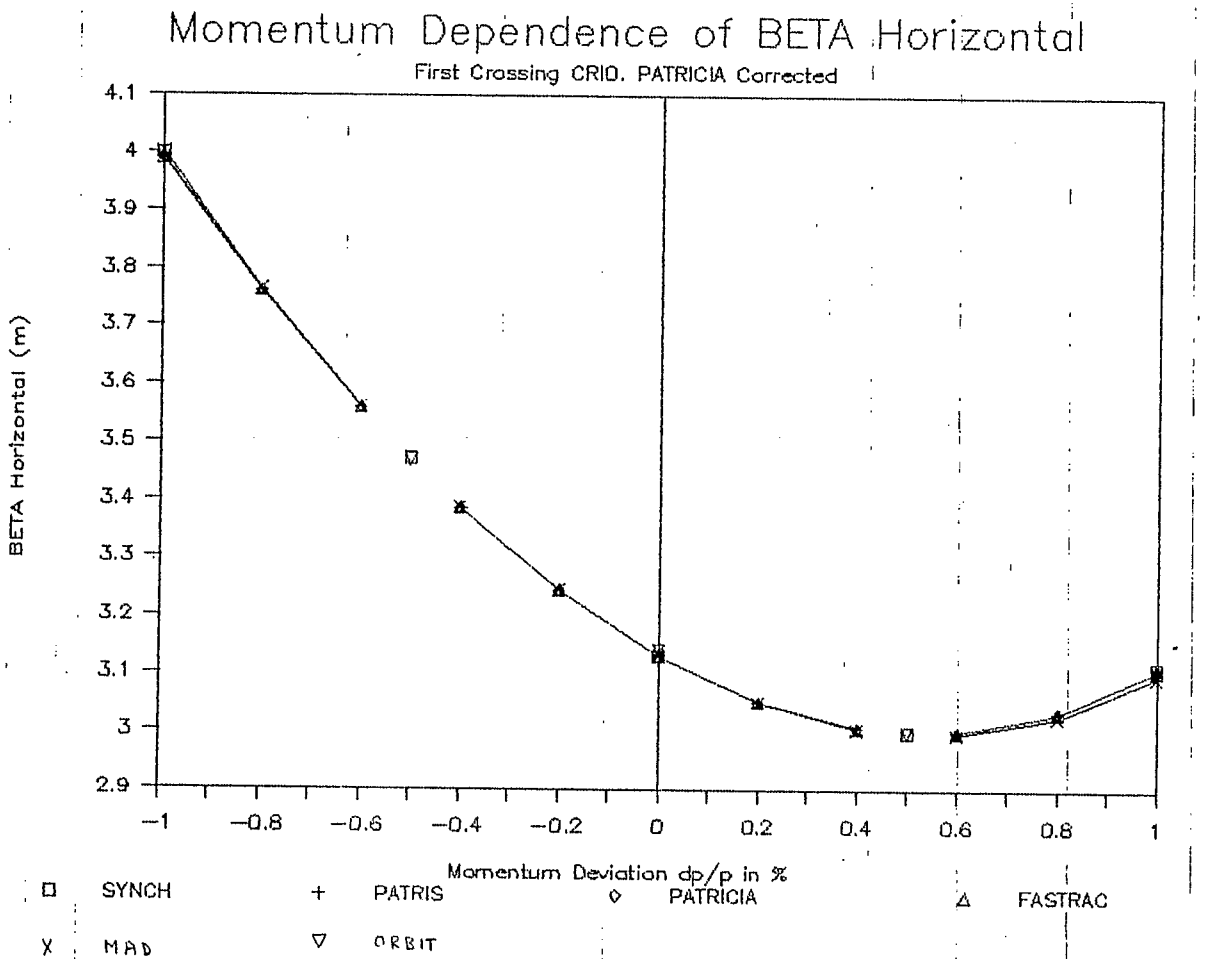

Momentum Dependence of BETA Horizontal
First Crossing CRIO. PATRICIA Corrected
BETA Horizontal (m)
4.1
4
3.9
3.8
3.7
3.6
3.5
3.4
3.3
3.2
3.1
3
2.9
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
Momentum Deviation dp/p in %
SYNCH
PATRIS
PATRICIA
FASTRAC
MAD
ORBIT

Fig. C8

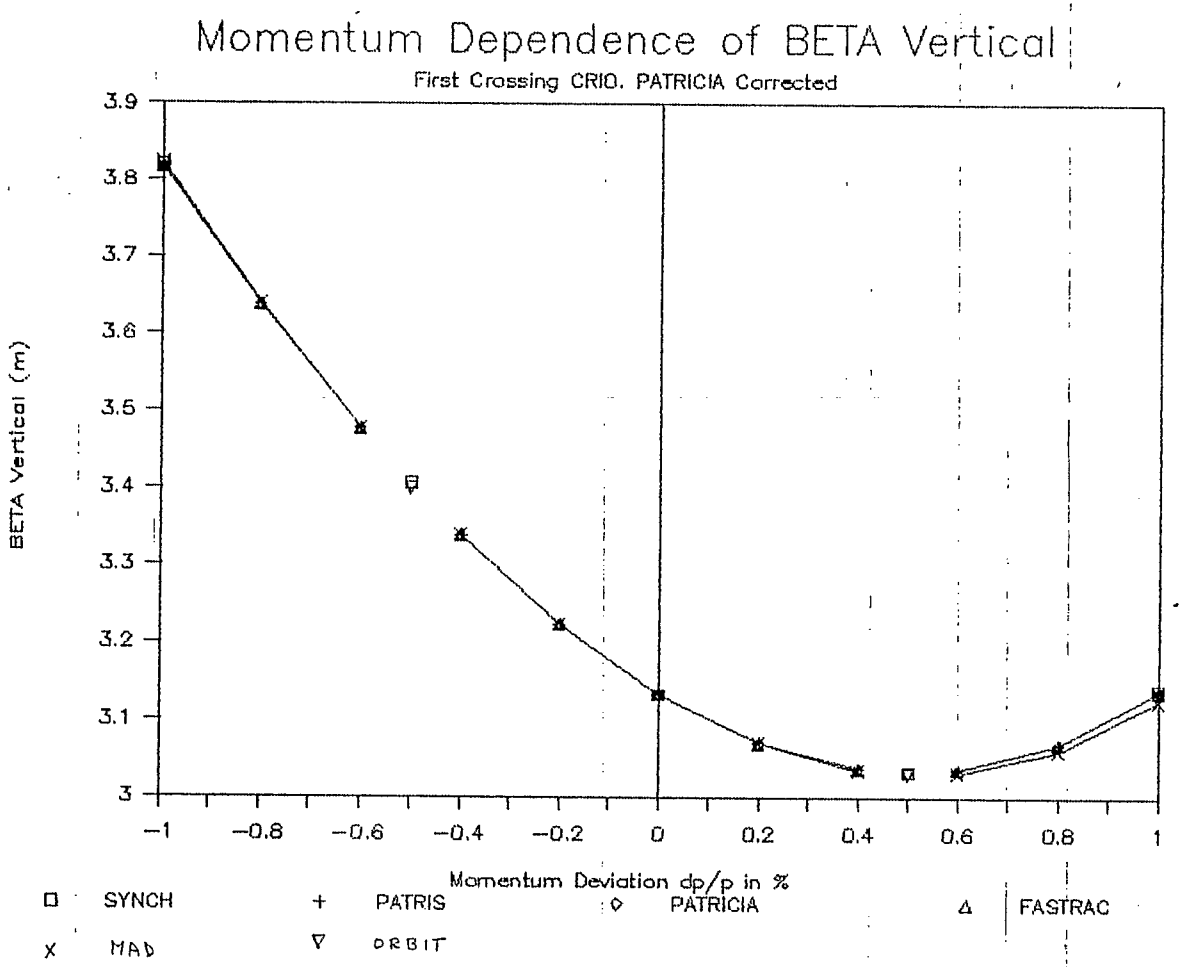

Momentum Dependence of BETA Vertical
First Crossing CRIO. PATRICIA Corrected
BETA Vertical (m)
3.9
3.8
3.7
3.6
3.5
3.4
3.3
3.2
3.1
3
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
Momentum Deviation dp/p in %
SYNCH
PATRIS
PATRICIA
FASTRAC
MAD
ORBIT

Fig. C9

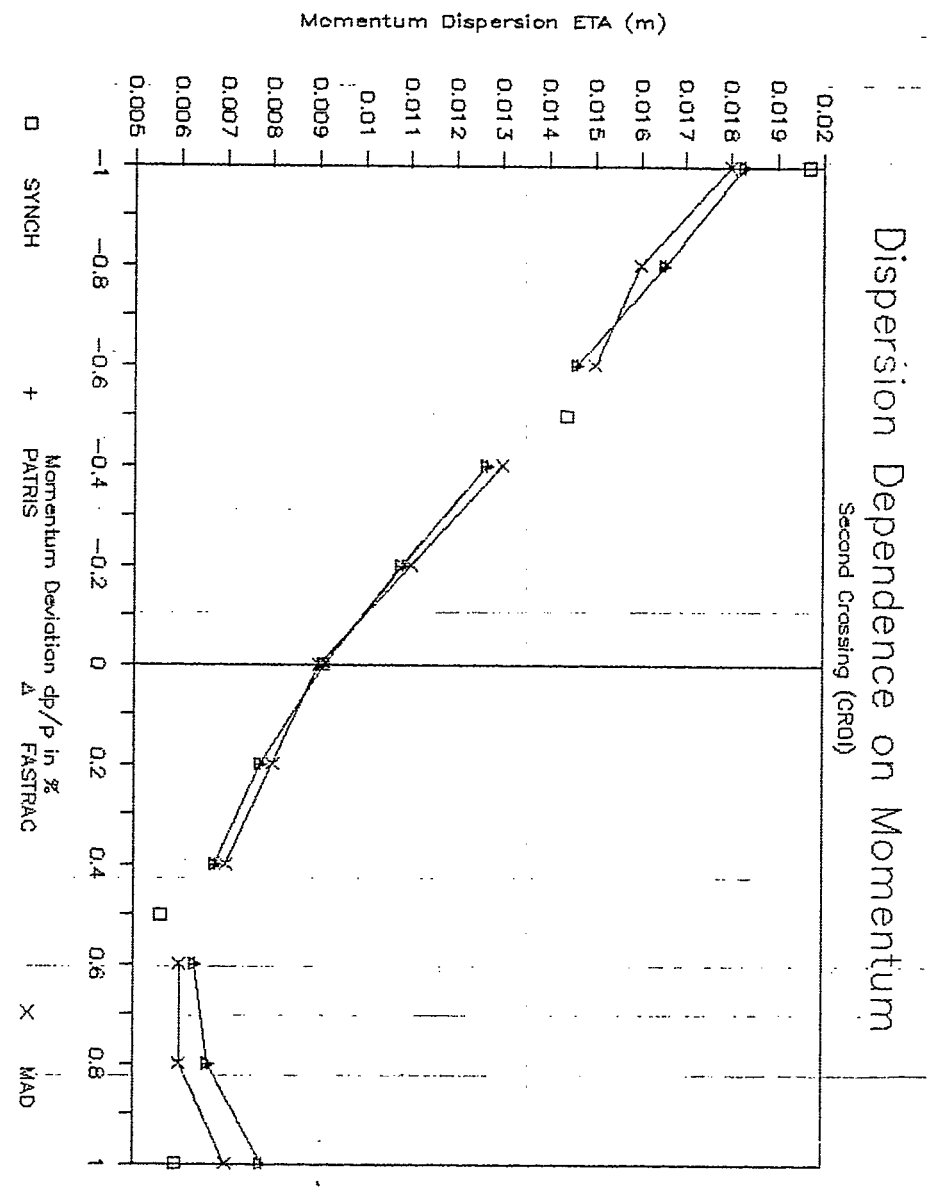

Dispersion Dependence on Momentum
Second Crossing (CR01)
Momentum Dispersion ETA (m)
0.005
0.006
0.007
0.008
0.009
0.01
0.011
0.012
0.013
0.014
0.015
0.016
0.017
0.018
0.019
0.02
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
Momentum Deviation dp/p in %
SYNCH
PATRIS
FASTRAC
MAD

Fig. C10

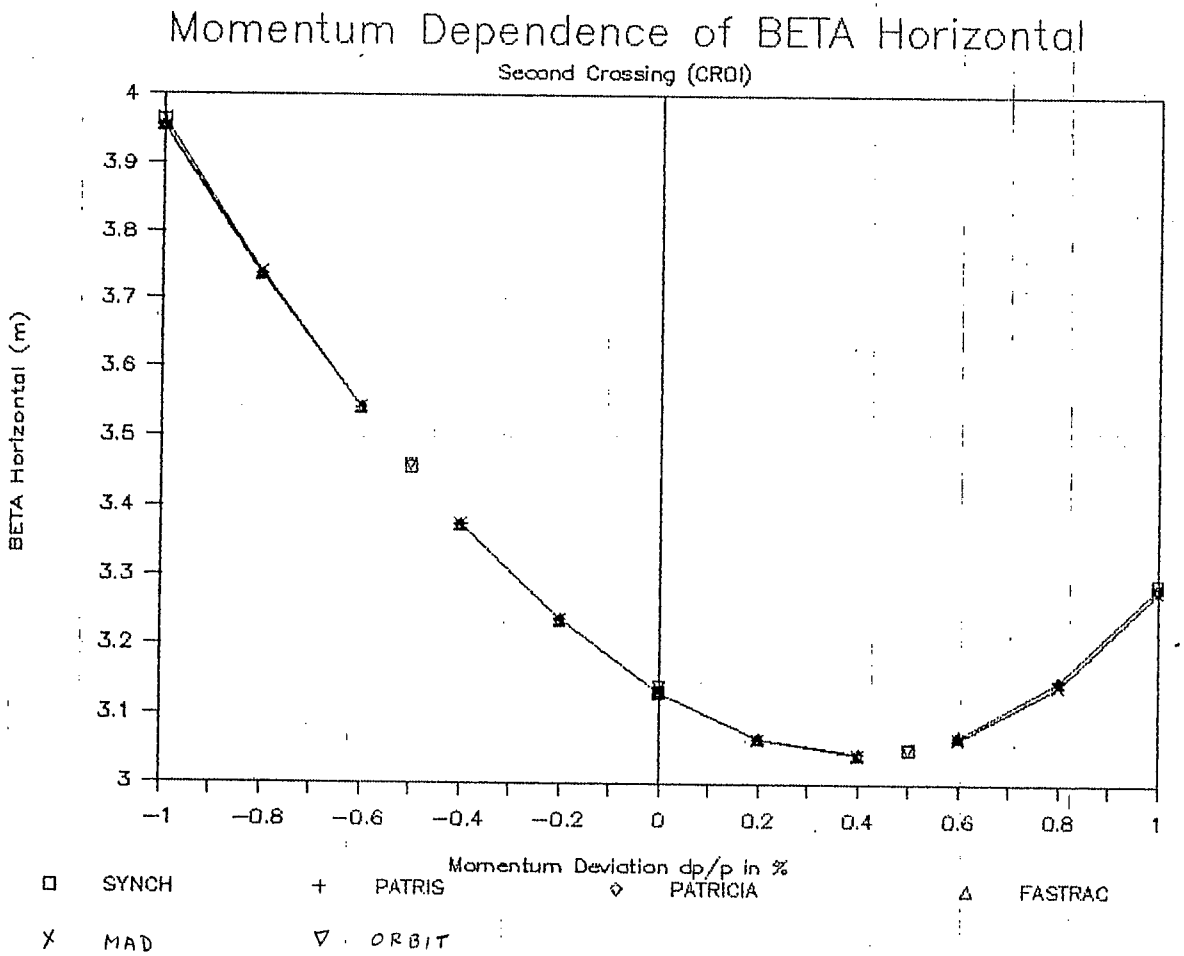

Momentum Dependence of BETA Horizontal
Second Crossing (CRO1)
BETA Horizontal (m)
4
3.9
3.8
3.7
3.6
3.5
3.4
3.3
3.2
3.1
3
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
Momentum Deviation dp/p in %
SYNCH
PATRIS
PATRICIA
FASTRAC
MAD
ORBIT

Fig. C11

Momentum Dependence of BETA Vertical
Second Crossing (ORBI)
BETA Vertical (m)
3.8
3.7
3.6
3.5
3.4
3.3
3.2
3.1
3
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
Momentum Deviation dp/p in %
□ SYNCH
+ PATRIS
◇ PATRICIA
△ FASTRAC
× MAD
▽ ORBIT

Fig. C12