

BNL MAD Program Notes: BTA Lattice Matching

J. Niederer

March 1996

Collider Accelerator Department
Brookhaven National Laboratory

U.S. Department of Energy

USDOE Office of Science (SC)

Notice: This technical note has been authored by employees of Brookhaven Science Associates, LLC under Contract No.DE-AC02-76CH00016 with the U.S. Department of Energy. The publisher by accepting the technical note for publication acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this technical note, or allow others to do so, for United States Government purposes.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or any third party's use or the results of such use of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or its contractors or subcontractors. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

For Internal Distribution Only

Accelerator Division
Alternating Gradient Synchrotron Department
BROOKHAVEN NATIONAL LABORATORY
Upton, New York 11973

Accelerator Division
Technical Note

AGS/AD/Tech. Note No. 431

BNL MAD Program Notes

BTA Lattice Matching

J. Niederer

March 18, 1996

BNL MAD Program Notes

BTA Lattice Matching

J. Niederer

AGS Department
Brookhaven National Laboratory

1. Overview

Another of the quaint tribal customs at the AGS is the annual application of the MAD matching utilities to the BTA transfer line joining the Booster to the AGS. Over the years, numerous people have tried their hand with various talents, techniques, and programs. In general this work has not yielded particularly successful solutions, and so the subject is revisited here. This note reviews the MAD program's treatment of parameter matching, and gives a practical example of matching criteria, along with some folklore, and limitations. Given the amount of work that has gone into this problem by others, it has seemed appropriate to look for more convincing ways to present requirements to the program, to improve the program as needed, and to examine some of the conventional assumptions about inputs and outputs. Parts of the internal searching utilities have been rewritten to accommodate techniques that were either unavailable or not particularly satisfactory in previous versions of the program.

2. Matching

Matching is a set of techniques for adjusting a set of variables to satisfy a set of conditions. Conditions, or constraints, are statements that at some point or points in a machine or beam line, one or more dependent variables should have a definite value, or lie within some limits. Such constrained variables are typically tunes, closed orbit coordinates, or the amplitude functions α and β . The constraints are met by varying independent variables (parameters), such as magnet strengths, and computing orbits until some acceptable combination is identified. The mechanics of fitting usually involve some multi dimensional search over ranges of the parameters to minimize a penalty function. In MAD, the penalty functions are weighted least squares differences between some present state and what is expected by each of the constraints, summed over the constraints. In a well posed matching problem, the individual penalty contributions at each constraint are more or less parabolic functions of the independent variables, and ideally have minima near some common value for each variable. Conventional minimizing techniques presume that this approach to a minimum is quadratic. This ideal condition, that of similar, well defined minima, is indeed often reached, particularly in studies *before* a beam line is actually built. However, even if not ideal, minor items like beam transfer

lines tend to be considered readily malleable, that is they can be fudged later after the primary machines have been placed to someone else's tastes, and are often treated as student exercises. There are places that have paid dearly for discounting the proper design and matching flexibility of such transfer lines.

Matching can become considerably more complicated in modern machines such as colliders, as designs must produce extremely small beams in local regions, without arousing all sorts of problems in other parts of the rings. Mismatches among feeder machines aggravate later problems in these larger machines. Accordingly the matching functions may acquire a number of conflicting minima as various criteria compete for attention, and frequently a set of conditions does not optimize to a common set of parameter values. Furthermore, some ranges of variables do not all lead to stable orbits, and must be excluded, which can further confuse the functional landscapes. Weighting may be applied blindly, with little concern for its consequences. Different kinds of particles usually require different beam line properties, but produced by the same sets of magnets. The presentation of the matching problem is thus very important. Various mathematical techniques have been developed to find "best" solutions based upon minima, and programs try to offer some information about the relative merits of the fits obtained. Sometimes these techniques will succeed even if the constraints are inconsistent, or correlated, or even if the various independent parameters may not be the most useful ones. More frequently the resulting minima will be broad or even missed, and a number of iterations may be needed to sharpen the definition of the matching problem with more realistic constraints and more convincing parameters. Conventional matching programs may not be particularly helpful in sorting out causes for failures to match, and people have been known to plod along with the programs without questioning the results. Amidst all of this possible complication and perhaps a remaining lack of effective diagnostics, matching can become something of an occult art.

3. The Booster to AGS Line

The problem of matching the BTA beam optics to that of the source and target machines has often had much of the character of twiddling 15 magnet currents, given fixed source conditions, to avoid hitting the pipe and to obtain given exit conditions, give or take some play in the machines involved. This surely resembles the case in both the models and operations. While this approach usually get most of the beam through the pipe, it becomes increasingly dubious as rising beam currents have radiation consequences for losses in transit. Operationally, tuning is often abnormally delicate, and transit losses can be appreciable.

To help in understanding BTA matching mechanics and results, graphics displays were introduced to show the penalty functions of each parameter at each constraint. (The Showmatch command, and plotfile). While most of the individual constraint penalty contributions more or less resembled parabolas, seldom did their minima fall at the same value of any independent variable, and some didn't even fit on the same scale. As might be expected, with fifteen quad current variables and perhaps six exit constraints other than rough beam line aperture limits, fitting seldom gave unique answers, nor necessarily credible ones. Criteria for amplitude functions conflicted with criteria for dispersion functions, and usually neither won. The BTA lattice as built mainly transferred a given dispersion or amplitude function as given; it offered little useful adjustment of one kind

of function relative to the other, except perhaps for dispersion near the entrance to the AGS. The character of the line is such that improving amplitudes tended to ruin dispersion, and conversely.

With our newer Runtwiss graphics, orbits and orbit functions could be traced and adjusted visually, and consequently believable constraints could be posed more realistically. Moreover, the graphics techniques suggested that there might be a range of input conditions better suited to the lattice geometry, which tended to be given as fixed. Perhaps some of the BTA troubles could be reduced by a different kind of orbit in the Booster before ejection, in particular by lowering the dispersion in the ejection region, a rather conventional approach. When initial values of the lattice functions were now treated as free parameters, the graphics indicated that there was a range of input values leading to reasonable trajectories with rather small envelopes in both betas and dispersion. Manually searching through a 21 parameter space to meet competing dispersion and amplitude criteria (ie, magnet twiddling) soon grew frustrating, so the matching program was asked to take over this search. The graphics based studies did suggest alternative ways to guide the search program, which this writer had found to be not particularly effective for the BTA in the past.

4. One More Try at BTA Matching

As noted, both the graphics and countless matching efforts of the past, including those of this writer, showed that the transfer line could not adequately handle both the dispersion and the amplitude functions as received from the source. Hence it appears worthwhile to learn what ranges of entry values of these functions could lead to better transfer line orbits, and then whether the Booster can accommodate these values as well. Graphics studies hinted that the BTA seemed to prefer low dispersion and somewhat larger initial amplitude functions for its inputs. Trial runs on a "typical" Booster lattice showed that one way to get negligible dispersion and adequately increased β_x in the F6 ejection region would be to increase focussing by up to about 20% at the F2 quad in the Booster, granting the adequacy of the trial lattice. In fact, the region just following F3 has this combination of favorable exit conditions, whereas that after F6 is the worst of the entire ring. Other runs treating all Booster quads as free parameters, with a 5% swing, not the way they are wired in practice, could obtain similar results, with no effort made to optimize. These results are to be regarded merely as suggestions about possible feasibility. For our purposes here, such paper changes are considered upper limits, and are open for discussion among our Booster experts as to what may be more practical ways to adjust dispersion at F6.

Next the BTA feasibility was explored by fitting entry values of the amplitude functions β_{x0} , α_{x0} , β_{y0} , and α_{y0} , and dispx_0 and disppx_0 , along with the first seven BTA quads to satisfy BTA constraints of both betas less than 20 and dispx less than 5 from the beginning through the shielding wall, with β_x and $\beta_y < 15$, and $\text{dispx} < 1$ at the entrance to Q8. This trial produced sensible families of solutions, which were then continued through Q12, and finally through Q15, all without major changes in the character of the previously fitted orbit entry parameters and the quad currents. The general idea here is to get the beam through the first part of the line with small dimensions and reasonably low dispersion. Then the tuning available in later quads can be used to match specific AGS requirements. Thus finally, these orbits were further

matched to a small range of values about a particular entrance to the AGS favored by W. Glenn, again without disturbing the earlier sets of fitted parameters significantly. In effect, in only several runs, the BTA could be walked through a set of conditions placed upon amplitude and dispersion, ones somewhat tighter than applied in the past. Table 1 shows the results of one of these fits, based upon the matching information supplied in the input file, presented in condensed form for Table 2. A representative BTA lattice circulating within the last year or so was used, with BTA quad currents held to a range of 0. to 550. amps.

The sensitivity of this tentative class of fit to beam entry parameters is easily checked with our graphics displays of orbits with iterated parameters. Figures 1 - 4 illustrate the technique and some results for the six entry parameters. Solutions seem rather stable with respect to the amplitude and dispersion functions at BTA entry. This kind of study was continued to explore the range of orbit entry values for which the resulting orbits would satisfy our other boundary constraints. In particular, the initial fitted α_0 was rather high (4.4) relative to that of the Booster trials, and could be reduced to more natural values of < 3.4 . The first cut β_{y0} was high and close to $\beta_{x0}(21)$; β_{y0} also could be reduced to a value consistent with a trial Booster result for modifying dispersion at F6 (< 11). The magnet changes among these amplitude function changes were modest, surely a necessary property of a meaningful solution.

Without further laboring the details, this set of techniques appears to have identified a family of initial conditions which lead to satisfactory orbits in the BTA (on paper). The important point is that there is a rather wide range of entry values lying well outside of the conventional Booster ejection ones for which the BTA can be tuned to handle narrow orbits. With low dispersion at entry, the early BTA quads can be tuned to favor lower orbit amplitude functions without having to compromise markedly to keep dispersion manageable. Wider apertures in the BTA can still be traded for lower exit β_{y0} in the Booster, and other tweaking is available. Relative to some of the older fitting arrangements, these newer ones appear to produce very clean solutions, with reasonable magnet currents and narrow orbit envelopes, and are rather insensitive to the selected entrance conditions.

Figure 1, which displays the effect of a range of the BTA entry parameters β_{x0} and β_{y0} , indicates a more subtle property of this class of solutions. For these curves, magnet currents are held constant. For a rather wide range of entry values, the transfer amplitudes stay within narrow envelopes. Some of the fits themselves did show a fair amount of adjustment around magnet Q9 where one of the amplitude functions grows larger, approaching one of our limits. This appears to be something of an artifact of fitting, indicating an edge of range value, in the sense that for a particular step in the value of a parameter all constraints were satisfied. The curves show that the constraints may well be satisfied for further steps; for now the program is content with finding the first configuration that satisfies the constraints. These solutions are occurring rather early in the searches, usually for relatively large values of the magnet current steps, which is a further suggestion of the stability of these solutions. In principle, some of our imposed amplitude limits can also be tailored better to actual dimensions of the beam pipe.

5. Using Simplex

These studies have used the Simplex method for locating minima, or preferably, full agreement with limit constraints. The alternative Migrad method was not very useful for this kind of constraint landscape. Rather than demanding exact values for the BTA exit constraints, which can artificially prejudice fitting, we posed these constraints in the form of limiting values, which were narrowed over time. Simplex searches invariably found magnet settings that produced orbits within the required range of values, usually quite early while step sizes were rather large. Perhaps this reliance on limits is a somewhat different way to present this kind of matching problem, as several bugs in the Simplex section of the program hindered or prevented its use. A solution that lies within all the boundary constraints achieves a zero penalty function, but is hardly likely to be unique. It is more likely a result of the particular iterating pattern of the search program. The program can certainly be improved to explore the width of these parameter ranges which satisfy the bounding form of constraints. Naturally the glitches have been fixed, but a few other related problems also surfaced. As an aside, the elderly MAD Simplex does not finish when it achieves a zero penalty result, and does not always deliver the proper final result for listing.

Because of the chancy nature of the first zero penalty fit delivered, it is very useful to explore the nearby regions of parameter space, so the tentative first fits can be tailored for more practical input values or tighter exit limits. In the CERN / MAD adaptation of the Simplex kind of fit, the initial values of variables have to lie within the narrower ranges of succeeding searches. This restriction is inherent in the non - linear transformation used to sample parameter space in the MAD and presumably other versions of Simplex. Yet it is probably better practice to begin further searches with a proved solution, whose fitted parameters may lie outside of the narrower limits, as Simplex searches can be influenced by the starting values of the parameters. This is one kind of problem that might be handled better by the matching program.

A somewhat more technical problem can be encountered in exploring the range of plausible entry parameters when using a set of such parameters that have produced a zero penalty fit, perhaps achieved under more generous limits. These starting parameters will produce a zero penalty result, with which all further iterations will be compared. The searches are influenced at several early stages by these comparisons with the zero penalty result, possibly closing out some other promising search direction, and the zero penalty may not be reproduced within the changed set of limits. The printing of results can be confusing as well, as the original zero result is reported as best penalty, but the perhaps compromised result obtained within the present search limits may have some non-zero constraint contributions.

Rather than modifying the familiar Simplex to deal with these cases, a simpler linear search version of Simplex was added as an option to aid in this kind of study. This alternate version will also serve as a basis for future schemes to map out the actual range of parameter values that can be accommodated within the zero penalty fits. The use of this "revised" Simplex is noted below. A detailed walk through of the MAD version of the Simplex method is given in the Appendix.

6. BNL MAD Program Matching Changes

The major physics sections of MAD have been rebuilt at BNL to operate much faster with considerably more capable internal data base services and graphics. The names of these "fast" version commands are formed by prefixing the letter **F** to the usual names given in the MAD manuals. These changes may have escaped the notice of readers still comfortable with the 1985 versions. While an older version based upon MAD 7 of about 1989 vintage is retained in the BNL distribution, all new features such as the following only appear in the "fast" version.

6.1. FSimplex

Our reworked Simplex search procedure is enabled with the *Revised* logical attribute added to the BNL MAD **FSimplex** command.

Fss FSimplex Tol = 1.e-7, Calls = 2000, Revised

The default is the regular MAD Simplex, modified to exit if it should achieve a zero penalty function.

6.2. FEndmatch

A minor nuisance in using boundary constraints over a large part of a lattice is the seemingly endless printing of constraints by **FEndmatch** at the end of a run. The *Verbose* logical attribute has been added to the **FEndmatch** command to deal with this printing.

To list all constraints:

FEndmatch Verbose

To list only constraints with non-zero penalty:

FEndmatch

Or

FEndmatch - Verbose

A similar *Verbose* attribute is attached to the **FSimplex** and **FMigrad** commands to produce a rather large amount of intermediate printing to help with debugging a matching run.

6.3. FCell and FMatch

Both the **FMatch** and **FCell** commands have been given two new attributes:

Conlist The name of a list of names of constraint commands to be used in the fit. Optional

Constraint commands can be offered from among the **FConstr**, **FCouple**, **FDConstr**, **FXconstr**, **FRmatrix**, **FTmatrix**, or **Fweights** classes.

Varlist The name of a list of names of **FVary** commands to be used in the fit. Optional

While these "list" features can be obtained by grouping constraint and vary statements within subroutines, they have been added to support a broader style of program use by which particular commands are selected from among a larger group archived in a

a command library. The idea is that selected library commands are activated by editing a list of their names, which is more convenient than editing entire commands in and out of matching command groupings. The list entries are initiated at the beginning of the fitting command (FMigrad or FSimplex). This is an "either or" feature; if a list name is used, loose constraint or vary statements within the Match group are decoded, but otherwise ignored. The protocol with **FWeight** statements is the same as before; each Fweight is in effect until the next Fweight is encountered, and only the weight fields actually given on the input statements are handled. If the list names are not given, the program will consider all constraints and varies lying between the starting FMatch or FCell command and the fitting command, as before.

7. Acknowledgements

These notes obviously benefit from the experiences and tribulations of AGS colleagues who have contributed to the BTA studies over the years. Indeed, this topic is affectionately known as professional suicide among its followers. M. Syphers and T. Roser have suggested related works which have altered dispersion properties of working machines, a probable prerequisite to the kinds of BTA entry conditions discussed here. C. Gardner has shared in some practical (paper) studies concerned with altering Booster conditions at extraction.

Appendix. The Simplex Method

The internal workings of the Simplex method may be of some interest. The quantities to be varied in search of a suitable match to a penalty function derived by computing orbits and applying orbit constraints are called parameters here. We describe our recent revision, which uses the parameters as given, applies range limits directly, and avoids the somewhat complicated set of transforms supplied with the usual CERN / MAD version of Simplex. There are three main stages:

- 1). Interpretation of starting parameter values, initial error assignments, and range limits from the given input data.
- 2). Formation of the simplex, a figure formed by joining a set of points in parameter space clustered about a trial "best" point.
- 3). Achieving a solution by reducing the simplex.

The Simplex method itself was first described in J. A. Nelder and R. Mead, 1965, *Computer Journal*, vol 7, pp 388 - 313, and has a history of numerous adaptations and improvements. A description more generic than the one given below may be found in the *Numerical Recipes* book of W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, 1993, pp 402 - 406 (Cambridge University Press).

Stage 1. Initial Values

The starting values of the N free parameters are assigned according to the upper and lower range limits given in the Fvary input statements for the search. For the case of both upper and lower limits given, if the parameter lies within the limits, the starting value is the the initial parameter value, and the starting step is the step value given. If the initial value of the parameter lies outside of the limits, or is at a limit, the starting value is set at the nearest limit, and the direction of stepping adjusted up or down accordingly. If the

upper or lower range limit(s) is (are) not given, the missing limit(s) is (are) set to a suitable combination of whatever limit(s) is (are) given and a small multiple of the step size.

The starting set of N parameter values is treated as the coordinates of a point, \mathbf{P}_0 , in an N dimensional parameter space. A "best" point \mathbf{P} and a corresponding "best" penalty value are maintained throughout the process. Similarly the steps given for the N parameters are formed into a corresponding vector \mathbf{Perr} of length N . Individual parameters are shifted about to look for a promising direction for improvement. In Stage 2, these shifts are based upon the steps of \mathbf{Perr} . Steps are continually adjusted, being expanded first while searching for a promising region, and then being reduced as the fit succeeds.

Stage 2. Building the Simplex

Stage 2 is a rough search for a promising direction or region, moving each parameter in turn. At each pass through this stage, a matrix is formed from $N + 1$ trial points (vectors), each derived from the current "best" point \mathbf{P} . When entering from Stage 1, $\mathbf{P} = \mathbf{P}_0$, the initial set of parameters. When reentered from Stage 3 below, after an unsuccessful probe of a previous configuration, $\mathbf{P} = \mathbf{P}_{low}$, the "best" result so far. The procedure iterates the $j = 1, N$ parameters as follows:

1. The current value of the step j , $\mathbf{Perr}[j]$, is added to the value of parameter j of the "best" point, $\mathbf{P}[j]$, yielding a new point \mathbf{P}_j , for which a penalty function $\mathbf{F}[j]$ is computed. Each new \mathbf{P}_j differs from \mathbf{P} by an increment j to the single parameter j .
2. If the penalty function for \mathbf{P}_j is less than the present "best" value, the step $\mathbf{Perr}[j]$ is increased, added again to parameter j of $\mathbf{P}_j[j]$, and the penalty recomputed. This stepping is continued until the penalty function rises, the parameter hits a range limit, or a small count is exceeded.
3. If the penalty function for \mathbf{P}_j from operation 1. is greater than the current minimum, the step for parameter j is decreased to 40%, and the step sign is reversed. The step is again added to $\mathbf{P}_j[j]$, and the penalty recomputed. This sequence may be repeated once more if the penalty remains larger. If there is no improvement, the present "best" value of the parameter is restored.
4. The point \mathbf{P}_j for which the j th parameter has just been stepped in search of an improved penalty is stored as column j in the point matrix. The penalty $\mathbf{F}[j]$ corresponding to this improved (or unchanged) point j is also recorded.
5. The original value of $\mathbf{P}[j]$ is restored, and operations 1 - 5 are repeated, cycling for each of the N parameters. The full iteration effectively explores the parameter space in the neighborhood of the current "best" point \mathbf{P} for a better penalty function.
6. After this set of iterations, the point \mathbf{P} at entry is stored as column $j = N + 1$ in the point matrix, with penalty $\mathbf{F}[N + 1]$. The column j_{high} with the worst penalty $\mathbf{F}[j_{high}]$ and the column j_{low} with the best (lowest) penalty $\mathbf{F}[j_{low}]$ are noted. This matrix of N parameters * $N + 1$ trial points will be fed to Stage 3 below.

The N dimensional figure formed by joining the $N + 1$ points derived above is called a simplex. For the case of $N = 3$, the figure is a tetrahedron. By stepping outward from the initial point, the lines joining the resulting points or vertices do not cross each other, and a simple "volume" is obtained.

Considered in terms of the mechanics of this stage, these iterations produce a matrix $S[N, N+1]$ with $N + 1$ columns that are copies of the original point P , except for the diagonal elements $S[j, j]$ of the $S[N, N]$ submatrix, which are modified as just noted. The matrix notation is much less helpful while describing the reduction of the simplex, which is more a matter of points and shapes than matrix properties.

Stage 3. Reduction of the Simplex.

The Simplex method attempts to reduce the volume of the simplex figure by applying several operations to the set of parameter space points assembled above. These operations include *reflection* about an average point, P_{Bar} , *expansion* of the simplex figure in a promising direction, and eventual *contraction* of the figure as a solution is approached. These operations are applied to the simplex point with the largest penalty. The mechanics of these operations in MAD and BNL MAD are:

1. An average point, P_{Bar} , is found by summing each of the j coordinates of N of the $N + 1$ points, excluding the point P_{High} with the worst penalty.
2. An offset vector P_{off} is computed as the vector difference (residual) between the coordinates of the presently worst point P_{High} and the average point P_{Bar} . A trial point P_{new} is formed by adding P_{Bar} and the negative of P_{off} , reflecting the worst vector across the average one:

$$P_{\text{new}} = P_{\text{bar}} - P_{\text{off}}$$

The components of P_{off} now serve in effect as the previously used steps of P_{err} . This scheme insures that the "volume" and shape of the revised simplex is preserved, preventing crossings of surfaces at this first stage of reduction.

3. If the penalty of operation 2. is lower than the current "best", the offset is increased, added to P_{new} , and the penalty for P_{new} computed again. That trial point leading to the lowest penalty of these two trials replaces the current worst point P_{High} in the point matrix. The simplex is said to be expanded in the direction of the better penalty.
4. If the penalty for P_{new} in operation 2. is higher than for the "best" point P but is still better than that of the worst point, P_{High} , the reflected point P_{new} replaces P_{High} in the matrix. The step sizes of P_{err} are recomputed for each parameter as the spread between the components of the best and worst points of the simplex. P_{err} is now the length of the line joining the two points P_{low} and P_{High} .
5. If after operations 2. and 4. P_{High} remains the worst point, another trial point P_{new} is formed as the average point P_{bar} plus half of P_{off} , which lies inside of the current simplex. This operation contracts the simplex, as it reduces a principal dimension.
6. If the penalty resulting from operation 5. is better, P_{new} is stored in place of P_{High} , and the steps P_{err} are recomputed. This stage then repeats at operation 1. until an acceptable fit or other condition for finishing is achieved. As the simplex is reduced, $P_{\text{err}} = P_{\text{High}} - P_{\text{low}}$ contains the largest dimension of the figure, so its size is compared with the requested tolerance of the fit as a condition for finishing the iterations. The contents of P_{err} are printed at the end of the fit as a measure of the residual errors.

7. If the penalty result of operation 5. is worse than that with the worst point, control branches back to Stage 2, using the currently achieved "best" point **P** = **P_{low}** as input. This branch usually succeeds in avoiding local minima.

Documents

Unix Typesetter Format - nroff / psroff Files.

Host **rapt.ags.bnl.gov**

BTA Lattice Matching /usr/disc2/jn/Docum+/Match.notes

FMatch /usr/disc2/jn/Docum+/Fmatch.man

Showmatch /usr/disc2/jn/Docum+/Showmatch.man

Runtwiss /usr/disc2/jn/Docum+/Runtwiss.man

To Print from rapt: (To Room 218 AGS 2nd Floor)

alias it.'cat * | psroff -t -ms > ppp; lp ppp'

it. Manual Name

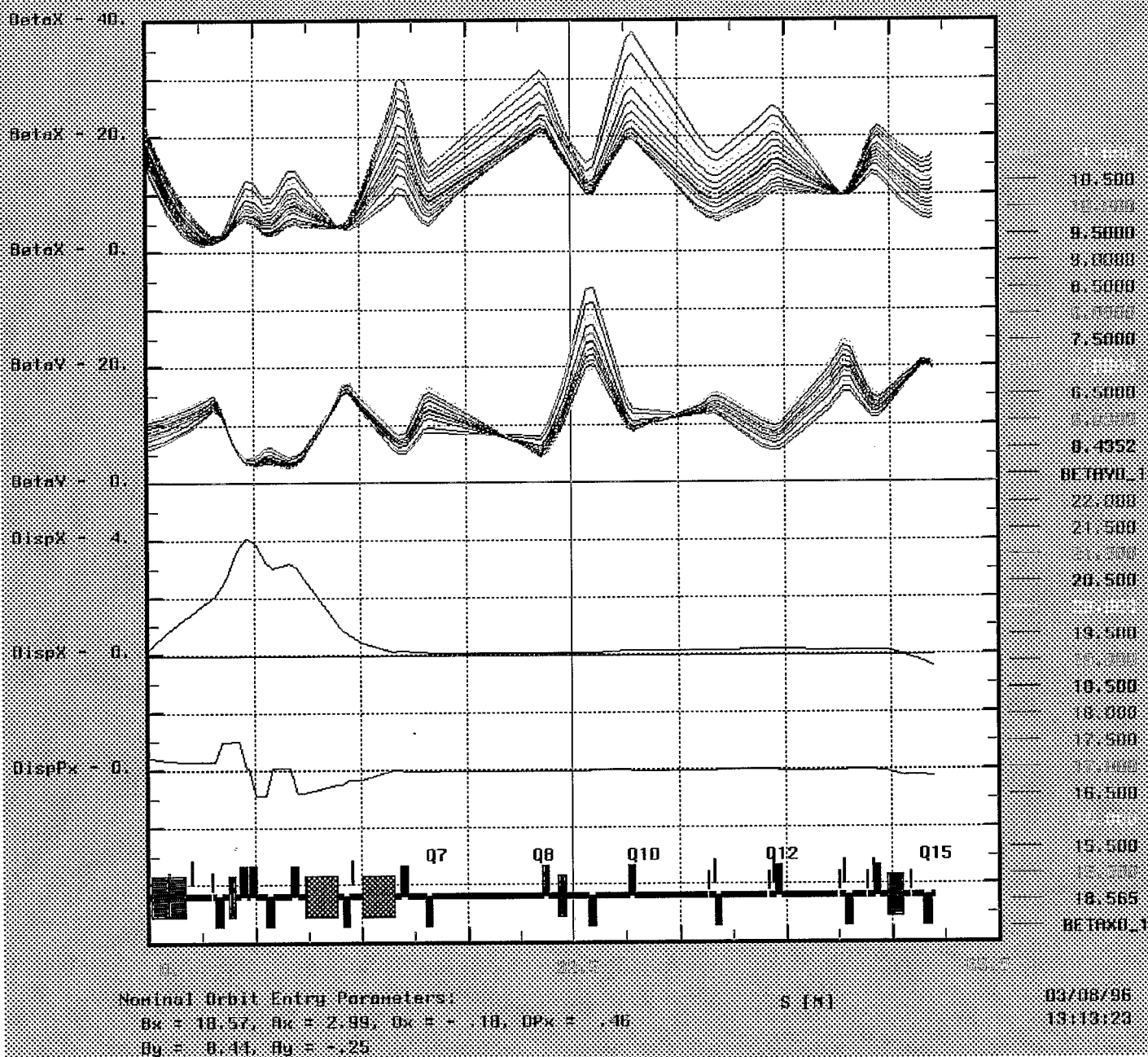


Figure 1. BTA Sensitivity to Entry Beta

For the fit shown in Table 1,
incoming beta X and beta Y are varied between limits of:

15. $< Bx0 < 22.$ stepped by .5

6. $< \text{By0} < 11$. stepped by .5

Magnet values are held at nominal fit values.
There is no coupling between X and Y.

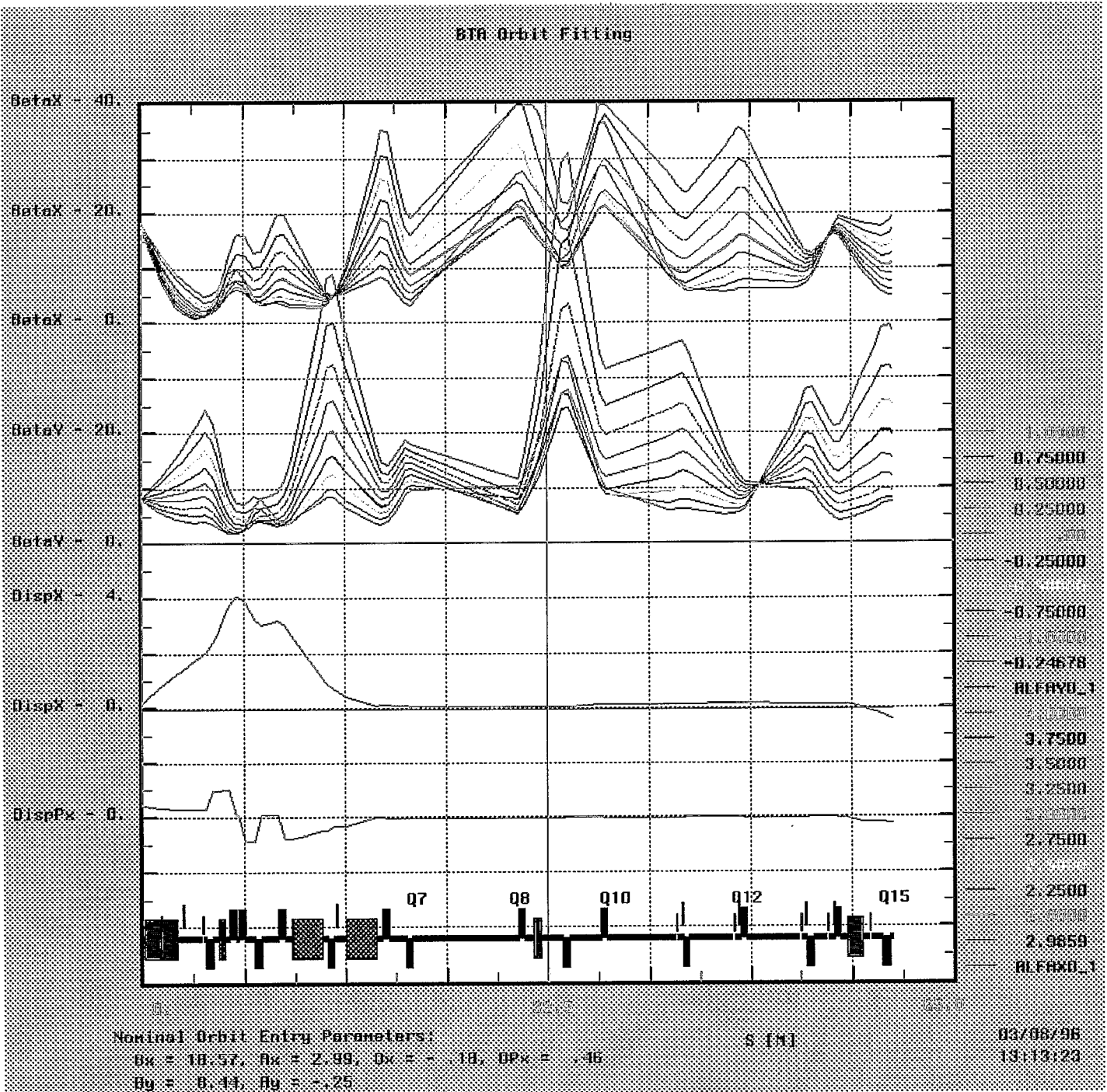


Figure 2. BTA Sensitivity to Entry Alpha.

Incoming alpha X and alpha Y are varied between limits of:

2. $< A_{x0} < 4$. stepped by .25

-1. $< A_{y0} < 1$. stepped by .25

Magnet values are held at nominal fit values.

There is no coupling between X and Y.

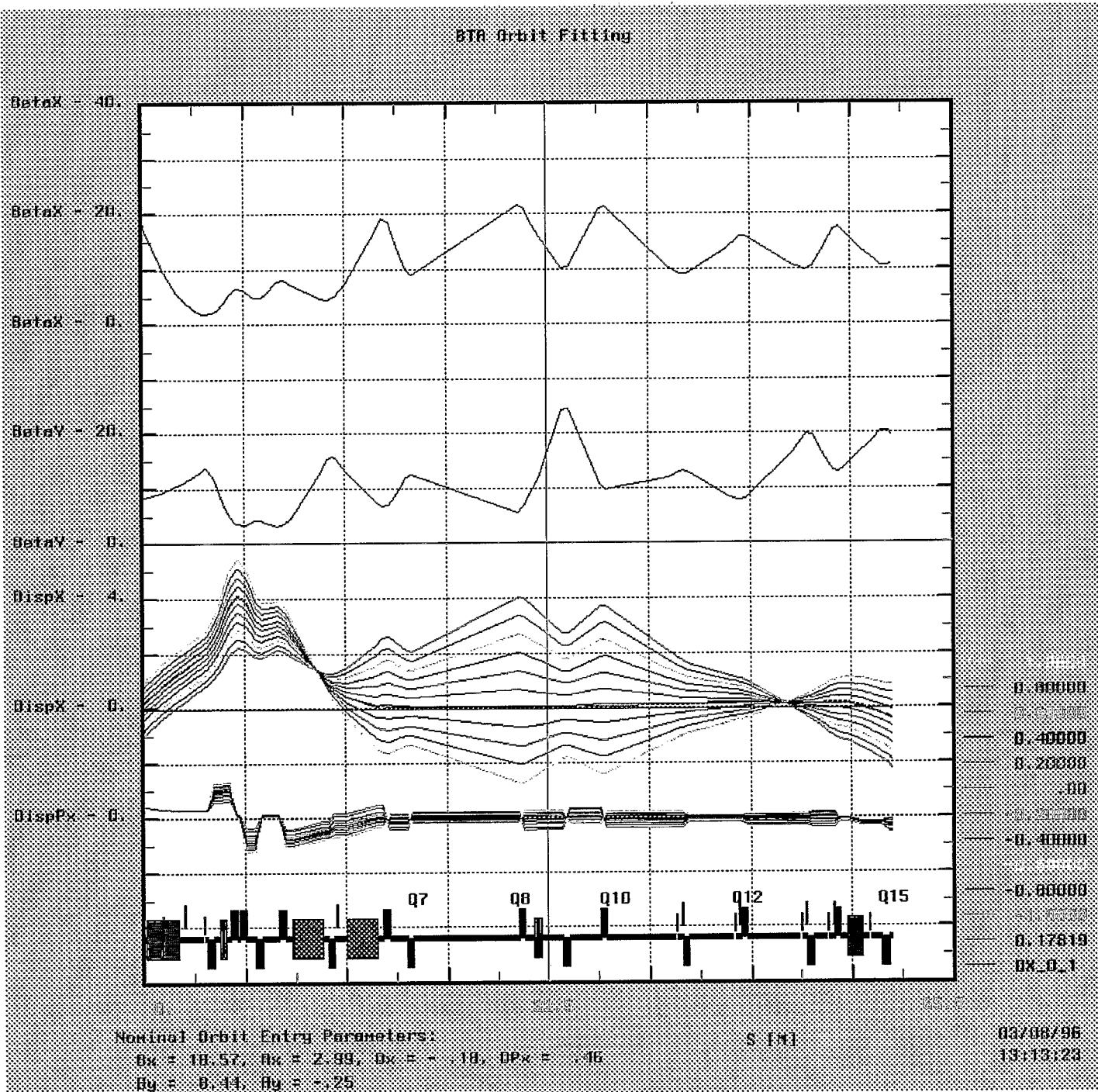


Figure 3. BTA Sensitivity to Entry Disp X.

Incoming disp X is varied between limits of:

$-1. < D_x < 1.$ stepped by .2

Magnet values are held at nominal fit values.

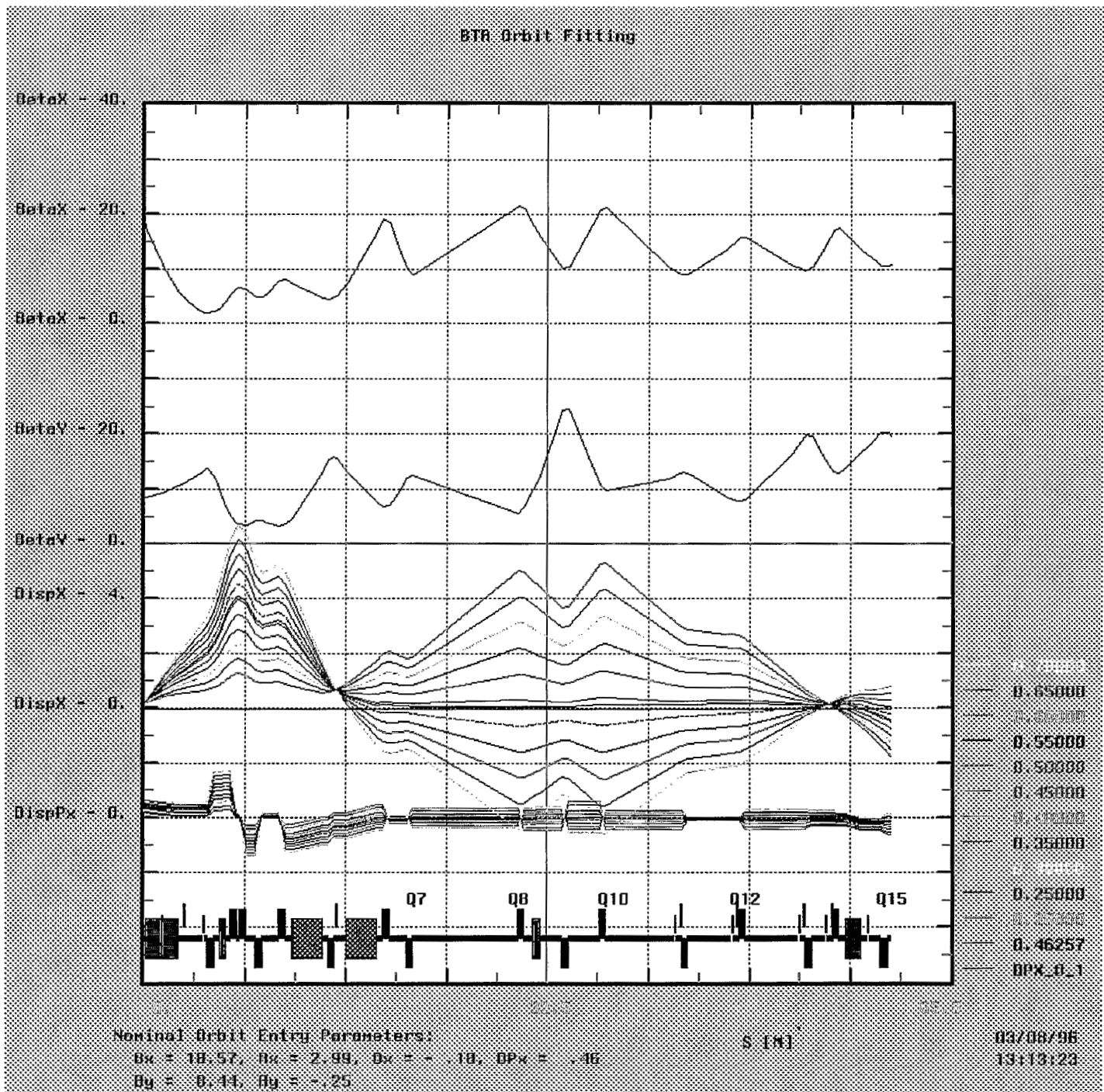


Figure 4. BTA Sensitivity to Entry Disp PX.

Incoming disp Px is varied between limits of:

$$.2 < D_{px} < .7 \quad \text{stepped by } .05$$

Magnet values are held at nominal fit values.

Table 1. BTA Matching

Summary of a Fit with Zero Penalty.

Simplex. March 7, 1996

PENALTY FCT. CALL
0.000000E+00 73

PARAMETER NAME	PAR. VALUE	PAR. ERROR
BETAX0_1	1.856525E+01	3.800000E-01
BETAY0_1	8.435169E+00	5.000000E-01
ALFAX0_1	2.985861E+00	2.413900E-02
ALFAY0_1	-2.467764E-01	1.000000E-02
DX_0_1	1.781935E-01	1.000000E-02
DPX_0_1	4.625680E-01	7.600000E-02
IQV1	2.353544E+02	3.800000E+01
IQH2	1.764249E+02	3.800000E+01
IQV3	2.307495E+02	3.800000E+01
IQH4	2.068573E+02	3.800000E+01
IQV5	4.374057E+02	3.800000E+01
IQH6	5.482443E+02	1.855700E+00
IQV7	4.931418E+02	3.800000E+01
IQH8	2.808920E+02	3.800000E+01
IQV9	4.950551E+02	3.800000E+01
IQH10	3.281063E+02	3.800000E+01
IQV11	2.043485E+02	3.800000E+01
IQH12	2.259319E+02	3.800000E+01
IQV13	3.344793E+02	3.800000E+01
IQH14	3.532868E+02	3.800000E+01
IQV15	9.265637E+01	3.800000E+01

Orbit Functions at End Of L6, S = 60.14m.

	Fitted Result	Constraint:		Expected
		Lower	Upper	
BETX	1.092719E+01	10.	11.	10.44
ALFX	-6.109313E-01	- .68	- .58	- .63
MUX	1.059006E+00			
BETY	1.986402E+01	19.8	20.8	20.27
ALFY	1.228955E+00	1.15	1.25	1.20
MUY	1.105312E+00			
X	0.000000E+00			
PX	0.000000E+00			
Y	0.000000E+00			
PY	0.000000E+00			
DX	-3.918136E-01	- .45	- .35	- .4085
DPX	-2.092445E-01	- .235	- .135	- .1845
DY	0.000000E+00			
DPY	0.000000E+00			

Table 2. Example Input Data for BTA Matching.

! Command Library for BTA Runs

March 7, 1996

! File = ".usr/disc2/jn/BTA+/Wruns.bta"

! Starting Simplex "Fit" through Q15. Narrow Limits after Q15.

BETAX0_1	Param =	1.856525E+01
BETAY0_1	Param =	8.435169E+00
ALFAX0_1	Param =	2.985861E+00
ALFAY0_1	Param =	-2.467764E-01
DX_0_1	Param =	1.781935E-01
DPX_0_1	Param =	4.625680E-01
IQV1	Param =	2.353544E+02
IQH2	Param =	1.764249E+02
IQV3	Param =	2.307495E+02
IQH4	Param =	2.068573E+02
IQV5	Param =	4.374057E+02
IQH6	Param =	5.482443E+02
IQV7	Param =	4.931418E+02
IQH8	Param =	2.808920E+02
IQV9	Param =	4.950551E+02
IQH10	Param =	3.281063E+02
IQV11	Param =	2.043485E+02
IQH12	Param =	2.259319E+02
IQV13	Param =	3.344793E+02
IQH14	Param =	3.532868E+02
IQV15	Param =	9.265637E+01

Store

! Varys for BTA Fitting.

Vvvpd	Fvary	Pdelta, Lower	Step = - .020, Upper = .02001	
VIq1	Fvary,	IQv1,	STEP = 50. LOWER = 0. UPPER = 550.1	
VIq2	Fvary,	IQh2,	STEP = 50. LOWER = 0. UPPER = 550.1	
VIq3	Fvary,	IQv3,	STEP = 50. LOWER = 0. UPPER = 550.1	
VIq4	Fvary,	IQh4,	STEP = 50. LOWER = 0. UPPER = 550.1	
VIq5	Fvary,	IQv5,	STEP = 50. LOWER = 0. UPPER = 550.1	
VIq6	Fvary,	IQh6,	STEP = 50. LOWER = 0. UPPER = 550.1	
VIq7	Fvary,	IQv7,	STEP = 50. LOWER = 0. UPPER = 550.1	
VIq8	Fvary,	IQh8,	STEP = 50. LOWER = 0. UPPER = 550.1	
VIq9	Fvary,	IQv9,	STEP = 50. LOWER = 0. UPPER = 550.1	
VIq10	Fvary,	IQh10,	STEP = 50. LOWER = 0. UPPER = 550.1	
VIq11	Fvary,	IQv11,	STEP = 50. LOWER = 0. UPPER = 550.1	
VIq12	Fvary,	IQh12,	STEP = 50. LOWER = 0. UPPER = 550.1	
VIq13	Fvary,	IQv13,	STEP = 50. LOWER = 0. UPPER = 550.1	
VIq14	Fvary,	IQh14,	STEP = 50. LOWER = 0. UPPER = 550.1	
VIq15	Fvary,	IQv15,	STEP = 50. LOWER = 0. UPPER = 550.1	

&

! Match Command for Varying Input Optics.

FMA_3	Fmatch,	deltap = pdelta,	Second,	
	Varlist = FV_list3,	Conlist = FC_list3,		
	BETX = Betax0_1,	ALFX = Alfax0_1,		
	BETY = Betay0_1,	ALFY = Alfay0_1,		
	DX = Dx_0_1,	DPX = Dpx_0_1		
FTW_3	Ftwiss,	deltap = pdelta,		
	BETX = Betax0_1,	ALFX = Alfax0_1,		

&
&
&
&
&
&
&

BETY = Betay0_1, ALFY = Alfay0_1, &
 DX = Dx_0_1, DPX = Dpx_0_1

Endstore

! Constraint Limits for BTA Matching.

Bmax Param = 25.
 Dmax Param = 4.5
 Dmin Param = - 4.5

Store

! Constraint List for BTA.

C_L0	Fconstraint, L0,	BETX > .5,	BETX < Bmax,	&
		BETY > .5,	BETY < Bmax,	&
		DX > Dmin,	DX < Dmax	
C_L1	Fconstraint, L1,	BETX > .5,	BETX < Bmax,	&
		BETY > .5,	BETY < Bmax,	&
		DX > Dmin,	DX < Dmax	
C_L2A	Fconstraint, L2A,	BETX > .5,	BETX < Bmax,	&
		BETY > .5,	BETY < Bmax,	&
		DX > Dmin,	Dx < Dmax	
C_L2B	Fconstraint, L2B,	BETX > .5,	BETX < Bmax,	&
		BETY > .5,	BETY < Bmax,	&
		DX > Dmin,	DX < Dmax	
C_L3	Fconstraint, L3,	BETX > .5,	BETX < Bmax,	&
		BETY > .5,	BETY < Bmax,	&
		DX > Dmin,	DX < Dmax	
C_L4	Fconstraint, L4,	BETX > .5,	BETX < Bmax,	&
		BETY > .5,	BETY < Bmax,	&
		DX > Dmin,	DX < Dmax	
C_L4S	Fconstraint, L4S,	BETX > .5,	BETX < Bmax,	&
		BETY > .5,	BETY < Bmax,	&
		DX > Dmin,	DX < Dmax	
C_L5	Fconstraint, L5,	BETX > .5,	BETX < Bmax,	&
		BETY > .5,	BETY < Bmax,	&
		DX > Dmin,	DX < Dmax	
C_L5A	Fconstraint, L5A,	BETX > .5,	BETX < Bmax,	&
		BETY > .5,	BETY < Bmax,	&
		DX > Dmin,	DX < Dmax	
C_L5B	Fconstraint, L5B,	BETX > .5,	BETX < Bmax,	&
		BETY > .5,	BETY < Bmax,	&
		DX > Dmin,	DX < Dmax	
C_L6	Fconstraint, L6,	BETX > .5,	BETX < Bmax,	&
		BETY > .5,	BETY < Bmax,	&
		DX > Dmin,	DX < Dmax	

! W Glenn, Feb 22.96. Entrance to AGS. Narrow Limits.

! Nominal:

! Bx = 10.44, Ax = - .63, Dx = -.4085, Dpx = - .1845

! By = 20.27, Ay = 1.20

C_L6enda	Fconstraint, #E,	Betx > 10.0,	Betx < 11.0,	&
		Alfx > - .68,	Alfx < - .58,	&
		Bety > 19.8,	Bety < 20.8,	&
		Alfy > 1.15,	Alfy < 1.25,	&
		Dx > - .45,	Dx < - .35,	&
		Dpx > - .235,	DPx < - .135	

C_L7	Fconstraint, L7,	BETX > .5,	BETX < Bmax,	&
		BETY > .5,	BETY < Bmax,	&
		DX > Dmin,	DX < Dmax	

Endstore

!*****

!!** Ignore

Match Section for BTA. Vary Input Optics, from Booster.

Vv_betx FVary	Betax0_1,	Step = .5,	&
	Lower = 5.000,	Upper = 21.01	
Vv_bety FVary	Betay0_1,	Step = .5,	&
	Lower = 5.000,	Upper = 9.01	
Vv_alfx FVary	Alfax0_1,	Step = .20,	&
	Lower = -0.0,	Upper = 3.41	
Vv_alfy FVary	Alfay0_1,	Step = .10,	&
	Lower = -1.0,	Upper = 2.01	
Vv_Dx0 FVary	Dx_0_1,	Step = .1,	&
	Lower = -2.0,	Upper = 2.01	
Vv_Dpx0 FVary	Dpx_0_1,	Step = .1,	&
	Lower = -1.,	Upper = 1.01	

! Match Section for BTA_3. Active Constraint and Vary Lists.

! Nominally to End of L6, Q1 - Q15. Match to W. Glenn "Matrix" for Ags.

Use BTA_3

FC_list3 Xmenu	= C_L0,	C_L1,	C_L2A,	C_L2B,	C_L3,	&
	C_L4,	C_L5A,	C_L5B,	C_L6,	C_L6enda	
FV_list3 Blist	= Vv_Betx,	Vv_Bety,				&
	Vv_alfx,	Vv_alfy,				&
	Vv_dx0,	Vv_Dpx0,				&
	VIq1,	VIq2,	VIq3,			&
	VIq4,	VIq5,	VIq6,	VIq7,		&
	VIq8,	VIq9,	VIq10,	VIq11,		&
	VIq12,	VIq13,	VIq14,	VIq15		

FMA_3

Fweight Dx = 2., Dpx = 5., Alfx = 5., Alfy = 5.

FMI Fsimplex, Calls = 2000, Toleranc = 1.E-5, Grid

Fendmatch -Verbose

Print BTA_3

Ftw.5

!*****

Ignore

! Twissdriver Rundata and Sample Vary of Pdelta.

!! Graphics Setup for BTA.

Vvvbetx FVary	Betax0_1,	Step = .50,	&
	Lower = 15.000,	Upper = 22.01	
Vvvbety FVary	Betay0_1,	Step = .50,	&
	Lower = 6.000,	Upper = 11.01	
Vvvvalfx FVary	Alfax0_1,	Step = .25,	&
	Lower = 2.0,	Upper = 4.01	
Vvvvalfy FVary	Alfay0_1,	Step = .25,	&
	Lower = -1.0,	Upper = 1.01	
VvvDx0 FVary	Dx_0_1,	Step = .2,	&
	Lower = -1.0,	Upper = 1.01	
VvvDpx0 FVary	Dpx_0_1,	Step = .05,	&
	Lower = .2,	Upper = .71	

!! Runtwiss Section.

Vvv_list	Blist	= Vvvpd,	VvvBetx,	VvvBety,	&
		Vvvvalfx,	Vvvvalfy,		&

VvvdX0,	VvvdPx0,		&
VIq1,	VIq2,	VIq3,	&
VIq4,	VIq5,	VIq6,	&
VIq7,	VIq8,	VIq9,	&
VIq10,	VIq11,	VIq12,	&
VIq13,	VIq14,	VIq15	

Use BTA_3

Rdtw	Rundata	Model	= MAD,	Machine	= BTA,	&
		Vlist	= Vvv_list,			&
		Draws	= DrawC5,	T.schema,		&
			T1.orbx,	T2.orbx,		&
		Ftwiss	= Ftw.5,	Ftw_1,	Ftw_2,	&
		Commands	= S.tww1,	S.tww2,		&
		Sliders	= SL_Pdelt			

Rtw Runtwiss Menu, Plotdef = "Pdef.ax", Tempv = C5_vals
Endignore

!*****

return

STOP