

BNL MAD Program Notes: Field Tracking Commands

J. Niederer

May 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. 434

BNL MAD Program Notes

Field Tracking Commands

J. Niederer

May 22, 1996

BNL MAD Program Notes

Field Tracking Commands

J. Niederer

AGS Department
Brookhaven National Laboratory

May 22, 1996

1. Introduction

Commands have been added to the elderly MAD Versions in use at the BNL AGS for detailed tracking of charged particles through magnetic fields for which vertical field map measurements are available. The first section describes the commands added for tracking calculations, and changes to existing tracking commands. The second section describes vector commands for entering lists of decimal numbers, and a vector command for manipulating contents of vectors.

The results of a program of this kind depend somewhat on the quality of the measurements in the field maps. For working with new magnets, it has seemed worthwhile to provide alternate fitting methods with differing degrees of accuracy to help in evaluating the combination of tracking and measurements. Least squares techniques in effect smooth a set of values over local grids within a magnet. Several kinds of approximation and various interpolation grids have been explored during the development of the field tracking features. Most of these variants have been retained as options in the tracking codes, and can be enabled with appropriate attributes on the tracking command statements.

The tracking features use a modified predictor corrector integration algorithm adapted by A. Luccio.

2. Choices of Methods.

The tracking computations integrate the equations of motion of a charged particle moving in a nominally vertical magnetic field, in both horizontal and vertical planes. During the development of these tracking procedures, there was concern about the relative accuracy of both the predictor corrector integration and the field interpolation methods. Errors can accumulate in this kind of computation. For example, the results of tracking through both halves of a measured field map should be symmetric with respect to both entrance and exit displacement and direction for a centered reference orbit; any discrepancy reflects on the methods. Earlier experience with similar kinds of calculations suggested that results would be affected at some level by the smoothness of the field data. There are tradeoffs between tracking speed and thoroughness of method. It is also well known that there is an optimum step size with respect to the accuracy of integration. Accordingly, this set of field tracking procedures was designed with a number of options to allow comparisons of interpolation methods, fine tuning, and step by step observation of the integration process.

The tracking can be customized for specific applications, and classes of magnets. While considerable effort has been given to make the tracking calculations quite general, there are undoubtedly prejudices due to our own needs and experiences solely based on AGS combined function magnets. Our experience in tracking through an AGS 90 inch magnet profile is presented in a separate report.

The **Fieldmap** attribute list provides details for operating three kinds of field map interpolation. At each integration step along the path, a local grid of nearest field measurements is identified: X, Z coordinates and corresponding field values. The grid should be rectangular, although spacing may differ among the grid lines of both x and z directions. This local grid is nominally centered at a grid point just before or

after the current path point, (x,z), but will be adjusted as needed at the ends of the field to supply valid measurements for the higher order options. Grid line and track coordinates are translated to this local grid center to reduce computational roundoff as the distances from the local center are raised to various powers in fitting field coefficients.

The kinds of interpolation offered are:

1. Simple, fast, linear interpolation among the four nearest field points, $B(x,y) = B(1,1), B(2,1), B(1,2), B(2,2)$. For the first two points at z_1 , $B(x,1)$ falls between $B(1,1)$ and $B(2,1)$; $B(x,2)$ falls between $B(2,1)$ and $B(2,2)$. Then $B(x,z)$ is similarly interpolated between $B(x,1)$ and $B(x,2)$.
2. Simple, relatively fast, quadratic interpolation from nine nearby field points. On each of three bands with the same Z value, a $B(x,i)$ is computed from three adjacent field points. These three $B(x,i)$ values are then combined quadratically to find the field $B(x,z)$. X measurements are presumed to be equally spaced. Z measurements need not be equally spaced.
3. More complicated, relatively slow, least squares fit in two dimensions among the points in an (N+1) square grid surrounding the current path point (x,y). X and Z measurements need not be equally spaced.

The pattern of the fitted function is:

$$N = 1. \quad B_1 = C_0 + C_1 * x + C_2 * z$$

$$N = 2. \quad B_2 = B_1 + C_3 * x^2 + C_4 * x * z + C_5 * z^2$$

$$N = 3. \quad B_3 = B_2 + C_6 * x^3 + C_7 * x^2 * z + C_8 * x * z^2 + C_9 * z^3$$

$$N = 4. \quad B_4 = B_3 + O(x^4, z^4))$$

$$N = 5. \quad B_5 = B_4 + O(x^5, z^5))$$

In practice, the two higher order least squares fits may extend the local grid of points over a region which becomes large compared with the kinds of gradient involved, defeating any gains from the larger number of measurement points.

In the AGS magnet cases, fields were measured along only half of the length, from the outside to the center. Use of these field maps has to presume mirror symmetry along the Z axis. As full field grids can be very large, there may be computational advantages in working with half maps instead of messaging the data into a full map. Some program gymnastics are included to select field points so that the symmetric half field calculation produces the same result as the full field data one. Otherwise even small differences build up as they are magnified over the hundreds of remaining steps in the integration. Ideally, the C_2 (z derivative) coefficient should vanish at the middle, which can be checked by selectively printing local results using the trace feature attributes on the tracking element command.

In each method there is some sensitivity to the step size in z, which stabilizes for steps less than 20% or so of the z spacing in the case of AGS magnets. If the grid spacing is reduced, as is usually the case near the magnet edges, the integration step should probably be reduced accordingly. This dynamic step adjustment, which is complicated for the integration method used, may be introduced in future versions.

3. Field Tracking Commands

3.1. Fieldmap

The **Fieldmap** Command treats the passage of a particle through a magnetic field described by a detailed map of its vertical component at points along a rectangular X, Z grid. It is a kind of lattice element, which appears in a Beam Line definition. At present, **Fieldmap** elements are available only for particle tracking. In Twiss optics and matching calculations they are presently treated as simple drifts.

Fieldmap attributes denote the input vectors that contain the field data, the translation of the field coordinate system into that of the reference orbit system, and certain details of the particle tracking calculation. The description follows the field measuring coordinate system in which X is horizontal, Y is vertical, and Z is along the direction of the magnet. The field data consist of values of the Y component of the field measured at successive X points beginning with an initial Z, repeated at intervals in Z. [$B_y(x,z)$] The program presumes input information expressed in mks units; invariably the fields are measured in some other

units, which may be converted using the *Vserve* service command noted in a later section. The sign convention for the field data must be appropriate for the particle being tracked.

Input attributes admit the field data to be considered as a full field profile, or as a combination among an entrance section, a center section, and an exit section. The exit section is treated as a reflection of the entrance section half field map. The program has attributes to adjust for cases where the origins of the measurement grid do not correspond to the nominal axis of the magnetic system being described.

Fieldmap attributes are:

- | | |
|---------|--|
| L | The complete length of the tracked section of field in meters, which MAD expects as the first attribute of a lattice element. Fieldmap routines compute this length from the <i>FieldL</i> , <i>CenterL</i> , and <i>Halfmap</i> attributes below. If the field map extends beyond the dimensions of the magnet itself, the submitted length extends similarly. This used part of the length may differ from that of the full length of the actual field map presented. |
| Angle | The total bend angle in radians over the length of the reference orbit through the magnet. Decimal. Used only in the <i>Calibration</i> mode, entrance and exit transformations between the magnetic field map coordinate system and the external reference orbit will be computed by fitting to this angle. |
| FieldL | The length of the part of the presented field map to be used for tracking. Decimal, in meters. This entry should be consistent with the value of <i>Z0</i> below, and the coordinate system of the map. |
| For | at a point <i>Z0</i> = -.13 meters out beyond the magnet, and continue for a distance of 1. meter from the edge to the magnet center. The entries should be:
$\text{FieldL} = .13 + 1.$
$Z0 = -.13$
In the event of an apparently unwarranted diagnostic about one of the various lengths lying outside of the map limits, adjust <i>FieldL</i> by some small number to deal with a possible round off condition among the inputs. |
| CenterL | The length of a central section of field, decimal in meters. Field values for this section will be the same as those of the X band of the last Z value presented in the field map vector described below. If no <i>CenterL</i> value is given, no center section will be tracked. |
| Xvector | The name of the pool vector containing the decimal values in meters of the X coordinates at the points on an X, Z grid at which the Y (vertical) component of the Field was measured. |
| Zvector | The name of the pool vector containing the decimal values in meters of the Z coordinates at the points on an X, Z grid at which the Y (vertical) component of the Field was measured. |
| Fvector | The name of the pool vector containing the decimal values of the vertical Field as measured at the points on an X, Z grid. |
| PCRatio | The decimal ratio of effective momentum between that for which field was measured and tracking calculation. Necessary to match bend angle to $B * D1$ integral. Field data are multiplied by this ratio. Default is 1.0

<i>NOTE: The limiting value of PCR = 0. for tracking is refused by the program, and reset to PCR = 1. Values vanishingly close to zero may be used, but may encounter some lingering very small residues in ds if a different PCR value was used for calibration.</i> |
| X0 | Decimal value in meters to be added to X, the horizontal coordinate, to enter grid on corresponding measurement center line. Normally map is centered, and <i>X0</i> can be ignored. |

Px0	Decimal value in radians to be added to Px, the horizontal coordinate direction, for orbit to pass along measurement center line. Should normally be 0., and ignorable.
Y0	Decimal value in meters to be added to Y, the vertical coordinate, to enter grid corresponding to X, Z plane of field grid. (Reference Orbit)
Py0	Like Px0.
Z0	Decimal value in meters of Z in measurement grid to be used for entrance to field. Z0 can be any point within the grid; it need not correspond to the beginning of field information. A Z0 outside of the measured or requested grid causes an error condition.
Norder	<p>An integer describing the order of field interpolation calculation to be followed. Normally <i>Norder</i> applies to the detailed least squares options, where neighborhoods of field points about an advancing track are considered. For N = 1, 4 grid points are used to fit three coefficients with a linear fit. For N = 2, 9 grid points are used to fit six coefficients with a quadratic fit. For N = 3, 16 grid points are used to fit ten coefficients with a cubic fit. <i>Norder</i> can take values of 1 through 5. The default is 1.</p> <p>In the ordinary nine point interpolation, <i>Norder</i> = 1 causes a simple linear interpolation among four points to be applied.</p>
Zstep	<p>Decimal value in meters of increment (Step) in Z for integration. The calculation may adjust this step locally if it is too coarse to match the field grid. The step size is roughly inverse to the speed of computation. When <i>Zstep</i> is too large, the calculation may tend to spend more time adjusting the size downward, and accuracy may suffer. When too small, the calculation may just take more time while encountering accuracy problems with round off. A value of about .0254 m has been found adequate for a three meter Ags bend magnet. Making the step size an exact fraction of the measurement grid spacing helps with the problem of getting similar results from the two halves of a calculation. A (temporary?) programming idiosyncrasy limits the step size to less than that of the measurement grid spacing.</p> <p>Performance depends rather directly on the step size. For a z step size of .0254 meter, the rate is about 16 tracks per second for four point fits through our 90 inch field example, using rather slow debugging level code on an Iris 4000.</p>
Calibrate	<p>Logical Flag: If True, the calibration option is carried out to obtain the entering offset in the X coordinate that matches the specified reference orbit bend angle. Calibration produces the two quantities <i>TrX0</i> and <i>AngX0</i>, which represent the entering X offset and angle for which a particle follows the reference orbit. These values are passed to the tracking part of the program. Calibration runs are normally done separately, before actual tracking, as they can be slow. It is wise to review the calibration results before committing their use.</p> <p>If calibration is not requested, and the two calibration results are not given, for tracking the missing one(s) will be taken as zero.</p>
TrX0	<p>For a bend magnet in a circular machine, normally the translation in the X coordinate relative to the magnet axis needed by the reference orbit to yield the specified bend angle. Decimal in meters. This and the following four attributes are provided so that lengthy calibration calculations need not be repeated. These attributes should be the results of a detailed calibration calculation run previously.</p> <p><i>TrX0</i> can also be set for calculating absolute displacements through a given field.</p>
AngX0	For a bend magnet in a circular machine, normally the entering angle to be used in transforming between the particle orbit and the magnet coordinate system. Decimal in radians. In the case of the reference orbit, this entry will be the half angle of the bend. For a simple transit of a particle entering along the field axis, this angle is zero.
TrY0	This is similar to <i>TrX0</i> , but in the Y direction, and nominally is zero. Decimal in meters.

AngY0	This is similar to <i>AngX0</i> , but in the Y direction, and again is nominally zero. Decimal in radians.
Ds	<p>The FULL arc length of the reference orbit through the magnet, which is available from calibration runs. Decimal in meters. This quantity is subtracted from the arc length of particle tracks to obtain the fifth component of tracking vectors.</p> <p>If <i>Ds</i> is omitted in the input, the value of <i>L</i> will be substituted, along with a warning message. If an excessively large value of the <i>ds</i> component appears in tracking, it will most likely be due to a mistake in this attribute.</p>
Weights(36)	<p>Decimal numbers that weight each field point when the least squares fitting options are used to interpolate field values. Weights apply to positions in the local grid of field values in use at a given step in the tracking.</p> <p>If no weights are given, the default is to use field values at all points of the local grid with equal weights.</p> <p>If weights are given, there should be enough of them to describe the full $(N + 1) * (N + 1)$ points of the local grid for the order being computed. Weights are entered in the order of the points of the local x, z grid: $W(x,z)$ points for a dimension of $W(N+1, N+1)$. The number of non zero values times weights in the local grid should equal or exceed the number of coefficients being fitted: 3 for $N = 1$, 6 for $N = 2$, etc.</p> <p>In our AGS trials, weighting had only a minor effect, somewhat less than the differences among methods which sampled grid points differently.</p>
Niter	An integer giving the maximum number of iterations to be used for computing the apparent reference orbit in the X, Z plane. Used for calibration only. Default is 10, which is excessive.
Tol	Decimal value of a tolerance figure for adjusting z step size downwards, when projected value does not match computed value for an interval. Calibration only. Default is 1.d-6.
Tolh	Decimal value of a tolerance figure for adjusting a reduced z step size upwards. Calibration only. Default is 1.d-8.
Csymm	A Logical Flag: if True, the program tracks through the given field map enforcing Z symmetry about the magnet center in the use of grid points in the two halves of the magnet.
Halfmap	Logical Flag: If True, the program regards the field map as an entrance section, to be followed by an exit section which is treated as a reflection of the entrance section. The effective tracking length is then twice that of <i>FieldL</i> , plus <i>CenterL</i> if given.
Leastsq	Logical Flag: if True, the program will use the slower but more thorough least squares methods for interpolating field points using supplied weights.
Xmode	Logical Flag: If True, calibration will be done by assuming the given bend angle, and adjusting the entrance position <i>X0</i> until the exit angle exactly matches the entrance angle. This is the normal mode. The alternative, the angle mode, in effect measures the bend angle corresponding to an entrance position <i>X0</i> that exits at the same value of X.
Dogrid	<p>Logical Flag: If True, the program will use coefficients computed from the edge region of the field map to extrapolate to regions which are off the map in the X coordinate. The <i>Npoint</i> attribute can be used to extend the reach of the sampling grid in X, for which <i>Npoint</i> is taken as the spacing of X points used from the field grid. The default is to use the <i>Taper</i> form of extrapolation.</p> <p>For example, for <i>Npoint</i> = 3, <i>Norder</i> = 3, on the left edge grid points $X = 1$, $X = 4$, $X = 7$, and $X = 10$ would be used for fitting field coefficients.</p>
Taper(2)	Positive decimal values in meters for the distance in X for which the field is to be extended from the last values given in the field map. Values are for the negative (or

smaller) and positive (or larger) X edges respectively. The field is extrapolated linearly from the last field value given to vanish at the given *Taper* values. Ignored if *Doslope* flag is set in input.

Doslope A Logical Flag: If True, field values beyond the edges of the map are treated by computing the slope of a linear fall off in the X direction from the last *Npoint* field points available. The program expects the map to have at least 80 points in the X dimension, or this option is suppressed.

Npoint An integer giving the number of points between the edge point and an inside point for carrying out a linear approximation of the field fall off beyond the map edges. Field values beyond an edge are taken from this line joining the two edge points. The default is 9 points, about a .9 inch separation for an AGS magnet.

Alternatively, in the *Dogrid* mode, *Npoint* gives the X spacing between the field map values near the edge of the field.

In the case of AGS field data, the field values at the edge of the map range from about half that of the values at the center in the fringe area, to about a quarter on one side and a third on the other in the main field region. AGS field maps range from about -9 inches to about +7 inches in the X dimension. Problems which have to track through regions beyond these measured ones can be expected to be rather sensitive to the edge approximations. This particular attribute is provided as a crude way to test the sensitivity of results to these edge of field approximations. Again, in the AGS case, the fall off is not quite linear with position X, so a run or two with different point separations will give some idea of the sensitivities involved.

Spline Logical Flag: if True a spline process will be used to smooth the given field map. (Not implemented)

Offlimit(4) Integer limits on the number of diagnostic messages to be printed for the conditions:

1. Failure to locate an X value within the given X grid. Indicates a bad value in the X grid, or a possible bad tracking X due to bad field point or program failure.
2. Failure to locate a Z value within the given Z grid. Indicates a bad value in the Z grid, or a possible bad tracking Z due to bad field point or program failure. Cases of a bad match of field lengths on the input statement are supposed to be rejected by the program before tracking.
3. Message about one or more points off the X grid, merely noting the condition. Tracking is handled anyhow using the fitted grid coefficients or the *Taper* information.
- 4.

Dumps(10) Optional set of one to ten z points along trajectory for which detailed dumps are to be written. These dumps display the course of the integration and field interpolation calculations, and can be helpful for diagnosing erratic results. The internal dump counters are reset whenever the local count of intervals exceeds *Dumpct*. Needed program flags for these dumps are noted below.

Dumpct An integer defining the number of dumps for each request in *Dumps* above. Default is 100.

/iweights An internal pointer (integer) to the weight array in this structure in the common pool.

/offct(4) Internal integers counting diagnostic messages.

/status An internal cell to mark status of command handling.

/stamp An internal cell to mark time of handling.

The detailed dump features should be used when first applying the tracking procedure to a class of magnet. A useful test setup involves a beam line with a single fieldmap element. The field tracking part of the program, found in source files *TrackF.B* and *MatrixF.B.*, should be compiled with the *Debug* program

library switch set, which may not be the usual case for the distributed production copies. The tracking section should be operated with the F trace option set (**Setdumps** command) and optionally with Y and or Z as well. The kinds of dump record produced are:

- F A short record of the initialization step that iterates the fit to the reference orbit.
- Y A detailed record of each interpolation step in the intervals given on the *Dumps* and *Dumpct* attributes, performed on the first iteration pass.
In addition, details of fitting the coefficients of the x, z polynomials for each local grid change are supplied. The amount of information can be very large if not managed with the attributes supplied.
- Z The same detailed record as for Y, but from the final iteration pass. This can be helpful in searching for causes of lack of symmetry between incoming and outgoing sections, and residual fitting quirks in the program.

The dumps at each field interpolation step contain:

A line with the current z, effective z in the grid, track vector(x, px, y, py), local accelerations (px', py'), and grid indices.

A line with field components Bx, By, Bz, and dx and dz coordinates in the local grid reference. The dz should reflect the step spacing.

A set of 2 - 6 coordinates of the measurement grid, in the local reference, for X and Z. One of these, used as a pivot in fitting a polynomial to the field values, is always zero.

One or more lines of fitted field coefficients (Cjk), with 3 - 21 values depending on the order of the fits.

One or more lines of contributions of each polynomial term corresponding to the Cjk coefficients (Terms), again with 3 - 21 values. This entry is helpful in checking for ridiculous local field interpolations.

3.2. FTrack

The **FTrack** command has been modified to be used with field and beam line tracking. (**Ftrack** corresponds to the **Track** command of the MAD manual, but its code has been modified for the better performing Data Base and tracking routines of the BNL Versions) Optical Parameters (Betx, etc) should be submitted if the tracking will not make use of the conventional reference orbit and rf cavity initializations used for circular machine tracking.

Rfcavity	Name of cavity to be used for adjusting energy changes. (As before)
NoRef	Logical Flag: If true, reference orbit and rf adjustment steps are skipped in setting up tracking. Optical parameters should be supplied, although the ones that appear in denominators of computations will be set to 1.0 by default. (New attribute)
Betx	Decimal value of Betax, as alternative to one otherwise computed from reference orbit in <i>Reference</i> mode. (New attribute)
Bety	Betay, as above.
Alfx	Alfax, as above.
Alfy	Alfay, as above.
Qs	Historical. No longer used.
/status	
/stamp	

3.3. Ftrgen

The **Ftrgen** command generates a set of starting tracks following attributes supplied on its input statement. Only a simple subset of options is given here; more complicated ones are treated in the *Track-plots* Manual. This command takes the place of sets of **Start** commands in conventional tracking use, and appears after the initializing **FTrack** command noted above. Optical parameters for generating ellipses are

needed from the **FTrack** command above, either from a reference orbit calculation, or as attributes *Betx*, etc., in the *NoReference* mode. Emittances are taken from a **Beam** command, which must precede.

Random	Logical Flag: If True, tracks will be randomly generated within X and Y phase ellipses described by the initial optical parameters and the given emittances. If <i>Random</i> is not given, the program will generate tracks in equally spaced intervals on the boundaries of the X and Y phase ellipses.
N	Number of starting tracks to be generated. Integer. If not randomized, the number of tracks will be $N - 1$ from the boundary of the ellipses, plus one from the center. If more than one <i>Deltap</i> is given, the number of tracks will be the number of <i>Deltap</i> times $(N - 1) + 1$. Example: $Deltap = -.001, .0, +.001$, $N = 51$ Will lead to $3 * (51 - 1) + 1 = 151$ starting tracks.
Tracknam	Name of Track Bank, an internal structure, to contain generated tracks. This name may be matched later to a <i>Frun</i> command to conduct tracking. A default of <i>"*TRAC.xx"</i> , where xx is a serial integer, will be supplied.
X0	Bias in origin, phase ellipse, decimal in meters.
Y0	
Px0	Bias in origin, phase ellipse, decimal in radians.
Py0	
Deltap(5)	Deltap to be used for phase ellipse tracks. Decimal, as fraction of momentum pc. One group of <i>N</i> tracks will be generated for each <i>Deltap</i> supplied. Default is 0.
/status	
/stamp	

A more complete **Ftrgen** description, including newer geometric beam shape options, is given in the *Trackplots Manual*.

3.4. FMTrgen

The **Multiple Track Generate** command similarly generates sets of tracks by iterating a particular track coordinate or coordinates between two limits. This command also takes the place of sets of **Start** commands in conventional tracking use, and appears after the initializing **FTrack** command noted above.

Tracknam	Name of Track Bank, an internal structure, to contain generated tracks. This name may be matched later to a Frun command to conduct tracking. A default of <i>"*TRAC.xx"</i> , where xx is a serial integer, will be supplied.
N	Number of starting tracks to be generated. Integer.
X1, X2	Range in X over which N tracks are to be generated, from X1 through X2. Decimal in meters.
Px1, Px2	Range in Px over which N tracks are to be generated, from Px1 through Px2. Decimal in radians.
Y1, Y2	Range in Y over which N tracks are to be generated, from Y1 through Y2.
Py1, Py2	Range in X over which N tracks are to be generated, from Py1 through Py2.
Ds1, Ds2	Range in Ds over which N tracks are to be generated, from Ds1 through Ds2. Decimal in meters.
Dp1, Dp2	Range in Dp over which N tracks are to be generated, from Dp1 through Dp2. Decimal.
Deltap(5)	Deltap to be used for phase ellipse tracks. One group of <i>N</i> tracks will be generated for each <i>Deltap</i> supplied. Default is 0.

/status

/stamp

3.5. Ftrplot

The track plotting section, upgraded from the older **MAD PLOT** command, has been augmented to draw phase ellipses corresponding to the track groups prepared by the **Ftrgen** command presented above. Unlike the scatter plots produced by the other *Type* options, the *Ellipse* option produces closed curves by joining the tracks originally generated along an ellipse. Colors and line types are taken from a **Pendata** command in the plot description file; line characteristics are indexed to the ordered positions at which phase ellipses are recorded. The **Ftrplot** command has the additional *Type* option of "Ellipse", which causes the drawing of closed ellipses at selected points along a lattice. This option is keyed to the starting track ensemble generated by **Ftrgen** or **Fstart** commands for starting tracks.

To use these features, a **Print** command should be used to note all lattice points for which track data is to be recorded. The **Frun** command should usually be set for a single turn.

Additions to the **Plot** command as described in the MAD Manual are:

Type	The <i>Ellipse</i> option has been added to the list of types of Plot.
Plotdef	The <i>Name</i> of a Plotdef command. To obtain screen plotting, either a customized plot file or the default one must be available in the input data. A complete plot setup is available as a default, which requires the plot definition CALL file Tplots.call to be available in the input data.
Prplot	A Logical Flag: If True, an archaic form printer plot will be attempted. This feature is subject to benign neglect.

For use with modern graphics workstations, and customized plot descriptions, the **Ftrplot** command has been designed to work with a plotting package based on the object oriented MAD data base, and the Silicon Graphics dedicated Graphics Library (GL) screen display primitives. In time this will hopefully become more portable, beyond Silicon Graphics screens. For the present, plotting requires the *Plotdef* attribute reference on the **Ftrplot** command.

The track plotting utility has been substantially rewritten, with screen menu driven features. A complete description is given in the *Trackplots* Manual.

3.6. Frun

This command has been augmented to record track vectors on the track history direct access file at the beginning of a run, and at positions flagged by the use of **Print** command(s). The recorded tracks may be displayed by appropriate listing and **Ftrplot** commands.

In addition, two new attributes have been added for regulating the maximum values of particle X and Y offsets:

Xmax	Decimal number giving maximum X in meters to be used for cutting off a particle in tracking. ("Overflow" condition) Default is one meter.
Ymax	Decimal number giving maximum Y in meters to be used for cutting off a particle in tracking.

4. Vector Handling Commands

A set of commands to handle vectors has been included to help with introducing machine gathered data into MAD calculations. Examples of useful vectors are field maps, and beam position monitoring data. Some vector service commands can also filter (exclude) out of range elements, and perform conversions among data with dissimilar systems of units. This Section highlights some vector commands useful for field tracking. The full description is in the manual document **Matrix** and **Vector** Commands.

4.1. Vector

The new **Vector** Command offers a way to package a set of decimal numbers for MAD use.

Examples:

```
V..1 Vector = 1., 2., 3., &  
              4., 5., 6.  
V..2 Vector = 14 * 1.
```

4.2. FiletoV

The new **File to Vector** Command offers a simple way to form a MAD pool vector from a set of numbers on an external ASCII file.

Example:

```
! Field Values - Input ASCII File to MAD Vector Form.  
FFvv. FiletoV Filename = "Field.agbf", Vname = "Agb.F0"
```

This creates the vector *Agb.F0* from the material on the file: *Field.agbf*.

4.3. Vserve

The new **Vector Serve** Command performs various operations among one or more vectors of equal length. Up to ten input vectors can be added, subtracted, etc, to form a new output vector. This is also a convenient command for performing units conversions on vectors, such as field grids, often copied as files from some other use.

Example

```
! Conversions for Above Vector. Gauss to Tesla.  
K.ff Vserve Vectors = "Agb.F0", Operation = CM, Factor = .00010, &  
Result = "Agb.F0"
```

This operation (Constant - Multiply) multiplies all of the elements of the input vector by the Gauss to Tesla conversion factor, and returns the result to the original vector.

5. Examples

At the end of this report are appended a magnet data vector preparation section (1), a field tracking definition section (2) for AGS magnets, a tracking command example section (3), and a partial command dictionary section. The tracking section is a test example of tracking through a few AGS entrance magnets with prescribed fields, as used to develop this group of features.

In the data preparation example, the X,Z position grid information of the field measurements are arranged in the form of two vector definitions. This grid coordinate data was edited from the first part of an AGS field measurement ASCII file, inserting the two command lines and end of line characters (&). The field measurement file is further edited to be read by a **FiletoV** (File to Vector) command and placed in a third vector. This particular final field data file consists of numbers separated by end of line characters. Other material on the file, such as a label at each new Z coordinate, was marked as comments in the editing pass. Further commands illustrate the use of the **Vserve** command to perform necessary units conversions.

Once these field information vectors are created, they are written (archived) in binary format into the local file directory. Hence these creation commands can be ignored on successive passes. Retrieval of these archived vector files is for practical purposes instant.

In the third example, a band of tracks are generated across the X direction and followed through a small group of magnets. For an alternative to the third example, a set of tracks generated in a phase ellipse pattern of starting position and direction could be followed through these magnets described by field maps. Running conditions are set up in the BEAM and FTRACK commands. The results appear in standard

MAD tracking history output tables, where they can be viewed by the various printing and plotting utilities.

The last pages give the current format of the commands above as they appear on the supplementary MAD dictionary file **FMDict.callx**.

6. Routines Involved

In the following, a routine is noted as new if it has appeared within the last three years.

6.1. MatrixF.B Source File

Fieldmap_F

This new routine performs necessary setups and data checking during the initialization / update stage when working matrices are created for fast tracking. It calls *FI_Hpcg* while computing the reference orbit centering transformations. It is called from the lattice element branch *Matrix_F*.

6.2. TrackF.B Source File

FI_TrackF

This new element tracking routine appears in the Tracking group, and is called by the element tracking branch manager *Traelm*. **FI_TrackF** sets up the information for the field tracking routine *FI_Hpcg*, and calls *FI_Hpcg* for each track in the current track bank.

FI_Hpcg

This new routine manages the tracking of one particle through a field map lattice element.

FI_Fct

This new routine, called by *FI_Hpcg*, interpolates field values from the submitted field map grid.

FI_Fctg

This new routine, called by *FI_Fct*, interpolates field values by least squares fitting values from the submitted field map grid. On option, it calls *FI_Gdumps* for detailed listings of the field computations.

FI_Gdumps

This new routine, called on option by *FI_Fctg*, writes diagnostic dumps of the least squares fitting process at each step of the integration.

FI_Rkstart

This new routine, called by *FI_Hpcg*, obtains starting values for the integration process.

FI_Xoff

This new routine, called by *FI_Fct* and *FI_Fctg*, extrapolates field values for X coordinates off the field map using simple linear techniques.

Runtrk_F

Initial values of tracks are now recorded on the history file.

Runtrn_F

Values of tracks are now recorded at positions marked by *Print* commands.

Trkgen_F

This new routine generates starting tracks.

Track_F

This older routine has been modified to support an optional mode that does not expect an initial reference orbit, and has additional attributes.

Trkplt_F

A phase ellipse drawing section has been added. In addition, the routine has been rebuilt to work with screen menus and rather elaborate track patterns.

6.3. TwissF.B Source File

Xbeam

This older routine has been augmented to make the working e / m ratio available to the rest of the program. It has also been rewritten in the more modern object format.

7. Comments

This study begins with an early tracking and field fitting program of A. Luccio. Both he and E. Bleser have helped check the resulting program with various BNL applications, and comparisons with conventional matrix representations. Other colleagues have shared in the development, which often involved searching for apparently minor problems involving the tenth or so digit, which would often propagate into the fourth or fifth digit in this kind of problem. R. Thern and colleagues obtained the field measurements.

The program computes transformations between the field map and reference orbit by adjusting the entrance X so that the specified bending angle is split equally between entrance and exit directions. This condition does not necessarily result in the same value for the exiting x for all choices of the z step size. Results for the various methods may have some exit x jitter as the step size is explored. Our experience with selecting and tuning parameters are given in an accompanying *Field Map Results* paper.

In general, tracking should be done with the same set of attributes as the corresponding calibration. For a single AGS magnet, differences among the various point sampling methods are rather small and are generally consistent with the AGS magnet measurements. However, these differences rapidly propagate, so alternatives and their sensitivities should be explored. Least squares tracking in the third order has proved to be the most suitable method for the particular grids of measurements performed on the AGS magnets.

Details of much of this material may be expected to change somewhat as further experience dictates. The relevant MAD dictionaries should be followed for up to date input command descriptions.

Documentation Files

Manual sources are in Unix psroff (troff) format.

Host:	rapt.ags.bnl.gov
This Report:	/usr/disc2/jn/Docum+/Fields.man
Results:	/usr/disc2/jn/Docum+/Fields.res
Vector/Matrix:	/usr/disc2/jn/Docum+/Matrix.man
Plotting:	/usr/disc2/jn/Docum+/Trackplots.man (Also an AGS Technical Note)

!** Section 1. Prepare Magnet Field Maps from Measurement Data.

!! File = "/usr/disc2/jn/Docum+/Fields.data.1"

! May 21, 1996

!!*****

!! Template for Building BNL MAD Files for Field Integration.
! Three Edited ASCII Files, "a0360a.x", | .z | .f , in each set,
! from Original "a0360a.mpa", etc with X, Z, F Arrays.

!!.....

! Input File to Internal Vector, with Metric Conversions.
! X Grid Values, Z Grid Values, X - Z Grid Field Values.

Fax	FiletoV	File	=	"/usr/disc2/jn/BTA+/Magnets.orig+/a0360a.x",	&
		Length	=	128,	&
		Vname	=	"X.a0360",	Convert = .0254
Faz	FiletoV	File	=	"/usr/disc2/jn/BTA+/Magnets.orig+/a0360a.z",	&
		Length	=	128,	&
		Vname	=	"Z.a0360",	Convert = .0254
Faf	FiletoM	File	=	"/usr/disc2/jn/BTA+/Magnets.orig+/a0360a.f",	&
		Length	=	128,	N1 = 161, &
		Vname	=	"F.af0360",	Convert = .0001

! Archive Vectors to File System for Further Use.
Archive "X.a0360", "Z.a0360", "F.af0360", &
Path = "/usr/disc2/jn/BTA+/Magnets+" &

!!.....

Fbx	FiletoV	File	=	"/usr/disc2/jn/BTA+/Magnets.orig+/b0360a.x",	&
		Length	=	128,	&
		Vname	=	"X.b0360",	Convert = .0254
Fbz	FiletoV	File	=	"/usr/disc2/jn/BTA+/Magnets.orig+/b0360a.z",	&
		Length	=	128,	&
		Vname	=	"Z.b0360",	Convert = .0254
Fbf	FiletoM	File	=	"/usr/disc2/jn/BTA+/Magnets.orig+/b0360a.f",	&
		Length	=	128,	N1 = 161, &
		Vname	=	"F.bf0360",	Convert = .0001

Archive "X.b0360", "Z.b0360", "F.bf0360", &
Path = "/usr/disc2/jn/BTA+/Magnets+" &

!!.....

! Reverse Rows for BF to BD Magnet Type.
Fbd Mserve Matrix = "F.bf0360", &
Result = "F.bd0360" Oper = Rows

Archive "F.bd0360", &
Path = "/usr/disc2/jn/BTA+/Magnets+" &

!!.....

Fcx	FiletoV	File	=	"/usr/disc2/jn/BTA+/Magnets.orig+/c0360a.x",	&
		Length	=	128,	&
		Vname	=	"X.c0360",	Convert = .0254
Fcz	FiletoV	File	=	"/usr/disc2/jn/BTA+/Magnets.orig+/c0360a.z",	&
		Length	=	128,	&


```
Fcd      FiletoM      Vname   = "Z.c0360",          Convert = .0254
File     = "/usr/disc2/jn/BTA+/Magnets.orig+/c0360a.f", &
Length   = 128,          N1       = 161,          &
Vname    = "F.cd0360",    Convert = .0001
Dump     "X.c0360", "Z.c0360"
Archive  "X.c0360", "Z.c0360", "F.cd0360",          &
Path     = "/usr/disc2/jn/BTA+/Magnets+"
!!.....
```

! Reverse Rows for CD to CF Magnet Type.

```
Fcf      Mserve      Matrix = "F.cd0360",          &
Result   = "F.cf0360"          Oper      = Rows          &
Archive  "F.cf0360",
Path     = "/usr/disc2/jn/BTA+/Magnets+"

```

Return

Section 2. Field Tracking Definitions for AGS Magnets.

AgS Lattice Region Receiving BTA.

File = "/usr/disc2/jn/Docum+/Fields.data.2"

May 20, 1996

PCF Param = 1.

!AGS dipole AFA5

! Retrieve Force, "X.bf0360", "Z.b0360", "F.bf0360", &
Path = "/usr/disc2/jn/BTA+/Magnets+" &

!! Type Af and Bf Use Same Field Measurements, but with Different
! Center Section Length.

FI.af Fieldmap, &
FieldL = (45. + 12.) * .0254, &
Z0 = - 12. * .0254, &
Angle = 0.0279649 Niter = 6, &
Norder = 3, Leastsq, &
Pcratio = PCR, &
Zstep = .0254 * .1, Halfmap, Csymm, &
Fvector = "F.bf0360", Xvector = "X.bf0360", &
Zvector = "Z.b0360", Tol = .000002, &
Trx0 = -.0092706, AngX0 = .01398245, &
Dogrid, &
Taper = .2, .2, &
Doslope, Npoint = 2, &
Ds = 2.89572607 &

!!.....
! AGS dipole BFA1

! FI.bf1 Fieldmap, &
FieldL = (37.5 + 12.) * .0254, &
Z0 = - 12. * .0254, &
Angle = 0.0235024, Niter = 6, &
Norder = 3, Leastsq, &
Pcratio = PCR, &
Zstep = .0254 * .1, Halfmap, Csymm, &
Fvector = "F.bf0360", Xvector = "X.bf0360", &
Zvector = "Z.b0360", Tol = .000002, &
Trx0 = -.0072743, AngX0 = .0117512, &
Dogrid, &
Taper = .2, .2, &
Doslope, Npoint = 4, &
Ds = 2.51467997 &

!!.....
! AGS dipole BFA2

! FI.Bf2 Fieldmap, &
FieldL = (37.5 + 12.) * .0254, &
Z0 = - 12. * .0254, &
Angle = 0.0235024, Niter = 6, &
Norder = 3, Leastsq, &
Pcratio = PCR, &

```

Zstep      = .0254 * .1, Halfmap,    Csymm,      &
Fvector    = "F.bf0360", Xvector = "X.bf0360",    &
Zvector    = "Z.b0360", Tol      = .000002,      &
Trx0       = -.0072743, AngX0    = .0117512,      &
Dogrid,    &
Taper      = .2, .2,                &
Doslope,   &
Ds         = 2.51467997             Npoint = 2,    &

```

!!.....
! ACS dipole L20
!

```

Retrieve    Force,    "X.bd0360", "Z.b0360", "F.bd0360",    &
            Path = "/usr/disc2/jn/BTA+/Magnets+"
FI.Ed1      Fieldmap,    &
            FieldL      = (37.5 + 12.) * .0254,    &
            Z0          =      - 12. * .0254,    &
            Angle       = 0.0235024, Niter   = 6,    &
            Norder      = 3,           Leastsq,    &
            Pcratio     = PCR,         &
            Zstep       = .0254 * .1, Halfmap,    Csymm,    &
            Fvector     = "F.bd0360", Xvector = "X.bd0360",    &
            Zvector     = "Z.b0360", Tol      = .000002,      &
            Trx0        = -.0065727, AngX0    = .0117512,      &
            Dogrid,     &
            Taper       = .2, .2,                &
            Doslope,    &
            Ds          = 2.51467505             Npoint = 2,    &

```

!! EdBleser
!! Trx0 = -.006914
!!.....

! ACS dipole CDA3

```

Retrieve    Force,    "X.cd0360", "Z.c0360", "F.cd0360",    &
            Path = "/usr/disc2/jn/BTA+/Magnets+"
FI.Cd1      Fieldmap,    &
            FieldL      = (45. + 12.) * .0254,    &
            Z0          =      - 12. * .0254,    &
            Angle       = 0.0279649, Niter   = 6,    &
            Norder      = 3,           Leastsq,    &
            Pcratio     = PCR,         &
            Zstep       = .0254 * .1, Halfmap,    Csymm,    &
            Fvector     = "F.cd0360", Xvector = "X.cd0360",    &
            Zvector     = "Z.c0360", Tol      = .000002,      &
            Trx0        = -.0095826, AngX0    = .01398245,      &
            Dogrid,     &
            Taper       = .2, .2,                &
            Doslope,    &
            Ds          = 2.89572485             Npoint = 2,    &

```

!!.....
! ACS dipole CDA4
!

```

FI.Cd2      Fieldmap,    &
            FieldL      = (45. + 12.) * .0254,    &
            Z0          =      - 12. * .0254,    &
            Angle       = 0.0279649, Niter   = 6,    &
            Norder      = 3,           Leastsq,    &
            Pcratio     = PCR,         &
            Zstep       = .0254 * .1, Halfmap,    Csymm,    &
            Fvector     = "F.cd0360", Xvector = "X.cd0360",    &
            Zvector     = "Z.c0360", Tol      = .000002,      &

```

```

Trx0      = -.0095826,   AngX0   = .01398245,   &
Dogrid,   &
Taper     = .2, .2,      &
Doslope,  &
Ds        = 2.89572485   Npoint  = 2,          &

```

```

.....
Retrieve   Force, "X.cf0360", "Z.c0360", "F.cf0360", &
          Path = "/usr/disc2/jn/BTA+/Magnets+"
FI.Cf1     Fieldmap, &
          FieldL  = (45. + 12.) * .0254, &
          Z0      = - 12. * .0254, &
          Angle   = 0.0279649   Niter   = 6, &
          Norder  = 3,         Leastsq, &
          Pcratio = PCR, &
          Zstep   = .0254 * .1, Halfmap, Csymm, &
          Fvector = "F.cf0360", Xvector = "X.cf0360", &
          Zvector = "Z.c0360", Tol     = .000002, &
          Trx0    = -.009583,   AngX0   = .01398245, &
          Dogrid, &
          Taper   = .2, .2,      &
          Doslope, &
          Ds      = 2.89572      Npoint  = 2,          &

```

```

!!.....
FI.Cf2     Fieldmap, &
          FieldL  = (45. + 12.) * .0254, &
          Z0      = - 12. * .0254, &
          Angle   = 0.0279649   Niter   = 6, &
          Norder  = 3,         Leastsq, &
          Pcratio = PCR, &
          Zstep   = .0254 * .1, Halfmap, Csymm, &
          Fvector = "F.cf0360", Xvector = "X.cf0360", &
          Zvector = "Z.c0360", Tol     = .000002, &
          Trx0    = -.009117,   AngX0   = .01398245, &
          Dogrid, &
          Taper   = .2, .2,      &
          Doslope, &
          Ds      = 2.89572      Npoint  = 2,          &

```

Section 3. Sample Tracking Run Using AGS Field Maps.

BTA Work with Field Maps.

File = "/usr/disc2/jn/Docum+/Fields.data.3"

May 20, 1996

Beam Particle= Proton, PC= 2.095633, Ex= 0.00001, Ey= .00001

use AGSIN

Print, AGSIN

!-----Tracking through the 5 AGS A magnets-----

Ftrack NoRef,

Betx = 13.101, Alfx = 1.913, Bety = 3.870, Alfy = -0.596

&

!-----Pattern in X across Face of Magnet-----

FMtrgen Track= "X_scan"

&

N = 21, X1 = -.001, X2 = .001, Px1 = 0.

Frun Turns = 1, Track = "X_scan", Short, Bank = "X_Bank"

Print all

Frun Turns = 1, Short, Fprint = 1

Endtrack

Quit

Section 4. Dictionary Definitions for Field Tracking

File = "/usr/disc2/jn/Docum+/Fields.dict"

Exerpted from "FMDict.callx"

May 16, 1996

! Label = Eqfieldmap, Header = Field, Mode = Tags

!*----- Field Element - Mapped Field.

Fieldmap	Keyword,	MT = 2, PR = 1, SP = 29, KL = 8,	&
L	= (R*1),		&
Angle	= (R*1),		&
FieldL	= (R*1),		&
Centerl	= (R*1),		&
Xvector	= (N*1, / ' ', = ' '),		&
Zvector	= (N*1, / ' ', = ' '),		&
Fvector	= (N*1, / ' ', = ' '),		&
Pcratio	= (R*1),		&
X0	= (R*1),		&
Y0	= (R*1),		&
Z0	= (R*1),		&
Px0	= (R*1),		&
Py0	= (R*1),		&
Norder	= (I*1),		&
Zstep	= (R*1),		&
Niter	= (I*1),		&
Tol	= (R*1),		&
Tolh	= (R*1),		&
Csymm	= (L*1),		&
Halfmap	= (L*1),		&
Leastsq	= (L*1),		&
Spline	= (L*1),		&
Calibrate	= (L*1),		&
Xmode	= (L*1),		&
Dogrid	= (L*1),		&
Doslope	= (L*1),		&
Npoint	= (I*1),		&
Trx0	= (R*1),		&
Angx0	= (R*1),		&
Try0	= (R*1),		&
Angy0	= (R*1),		&
Ds	= (R*1),		&
Weights	= (R*36),		&
Taper	= (R*2),		&
Offflimit	= (I*4),		&
Dumps	= (R*10),		&
Dumpct	= (I*1),		&
"/iweights"	= (M*1),		&
"/wknpt"	= (M*1),		&
"/wdoslope"	= (G*1),		&
"/offct"	= (M*4),		&
"/status"	= (M*1),		&
"/stamp"	= (M*1)		&

! Label = Eqfielddata, Header = Fld

!*----- Field Map - Vectors

"Fielddata"	Structure,	MT = 9, PR = -5, SP = 29,	&
	"/Sptrs"	= (M*8),	&

Reptr	= (K*1),	Xepr	= (K*1),	&
Matttype	= (K*1),			&
Ffact	= (F*1),	Ds0	= (F*1),	&
Xa0	= (F*1),	Xt0	= (F*1),	&
Ya0	= (F*1),	Yt0	= (F*1),	&
Sinhang	= (F*1),	Coshang	= (F*1),	&
Mxvect	= (M*6),	Mzvect	= (M*6),	&
Mfvect	= (M*6)			

```

!.      Label = Eqtrack_F, Header = Trak, Mode = Tags
FTrack  Keyword, MT = 6, PR = 6, SP = 14,
        Qs      = (R*1, / 0.0, = 0.0),
        Rfcavity = (N*1, / " ", = " "),
        Noref    = (L*1),
        Betx     = (R*1),
        Bety     = (R*1),
        Alfx     = (R*1),
        Alfz     = (R*1),
        "/status" = (M*1),
        "/stamp" = (M*1)

```

```

!.      Label = Eqtrgen_F, Header = Trgen, Mode = Tags
FTngen  Keyword, MT = 6, PR = 6, SP = 8,
        Tracknam = (N*1),
        Starts   = (N*1),
        Box      = (L*1),
        Cross    = (L*1),
        Diamond  = (L*1),
        Ellipse  = (L*1),
        Ring     = (L*1),
        Xaxis    = (L*1),
        Yaxis    = (L*1),
        Dw       = (R*1, = .01),
        Dh       = (R*1, = .01),
        Npoints  = (I*1, = 1),
        Random   = (L*1),
        Deltap   = (R*5),
        Ngtracks = (I*1, = 1),
        X0       = (R*1, = 0.),
        Y0       = (R*1, = 0.),
        Px0      = (R*1, = 0.),
        Py0      = (R*1, = 0.),
        "/betax" = (F*1),
        "/alfax" = (F*1),
        "/ex"    = (F*1),
        "/ntracks" = (M*1),
        "Adefault" = (N*2),
        "/status" = (M*1),
        "/betay" = (F*1),
        "/alfay" = (F*1),
        "/ey"    = (F*1),
        "/ngroups" = (M*1),
        "/stamp" = (M*1)

```

```

!.      Label = Eqtrgen_M, Header = Trgen, Mode = Tags
FMTrgen Keyword, MT = 6, PR = 6, SP = 18,
        Tracknam = (N*1),
        N        = (I*1),
        X1       = (R*1),
        X2       = (R*1),
        Y1       = (R*1),
        Y2       = (R*1),
        Px1      = (R*1),
        Px2      = (R*1),
        Py1      = (R*1),
        Py2      = (R*1),

```

```

Ds1      = (R*1),
Ds2      = (R*1),
Dp1      = (R*1),
Dp2      = (R*1),
Deltap   = (R*5),
"/status"= (M*1),
"/stamp" = (M*1)

```

```

!.      Label = Eqtplots_F, Header = Tplot, Mode = Tags
FTrplot Keyword, MT = 6, PR = 6, SP = 16,
Bank     = (N*1, / '*STOR.01', = '*STOR.01'),
Haxis    = (N*1, / 'X', = 'X'),
Vaxis    = (N*1, / 'Px', = 'Px'),
Hscale   = (R*2, / 0.0, = 0.0, 0.0),
Vscale   = (R*2, / 0.0, = 0.0, 0.0),
Type     = (N*1, / 'MULTIPLE', = 'SINGLE'),
Freq     = (I*1, / 1, = 1, >1),
Plotdef  = (N*1, = "#TrkPdef", / "#TrkPdef"),
Prplot   = (L*1, / .F., = .F.),
Menu     = (L*1),
Join     = (L*1),
Plsymbol = (N*1),
Symbols  = (N*10),
"/index" = (K*1),
"/status"= (M*1),
"/stamp" = (M*1)
PLTrplot Keywordcopy(FTrplot), PR = -4, SP = 8

```

```

!.      Label = Eqtrun_F, Header = Trun, Mode = Tags
Frun:   Keyword, MT = 6, PR = 6, SP = 15,
Mode    = (N*1, / 'SYNCHROT', = 'SYNCHROT'),
Method  = (N*1, / 'TRANSPOR', = 'TRANSPOR'),
Turns   = (I*1, / 1, = 1),
Fprint  = (I*1, / 1, = 0),
Sample  = (I*1, / 1, = 1),
Buffer  = (N*1, / "$TBANK.$", = "$TBANK.$"),
Track   = (N*1, / ' ', = ' '),
Bank    = (N*1, / ' ', = ' '),
Xmax    = (R*1, / 1., = 1.),
Ymax    = (R*1, / 1., = 1.),
Short   = (L*1),
"/status"= (M*1),
"/stamp" = (M*1)

```