BNL-104852-2014-TECH

AGS/AD/Tech Note No. 436;BNL-104852-2014-IR

BNL MAD Program Notes Fmatch: Enhanced Matching Commands

J. Niederer

June 1996

Collider Accelerator Department

Brookhaven National Laboratory

U.S. Department of Energy

USDOE Office of Science (SC)

DISCLAIMER

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. 436

BNL MAD Program Notes

Fmatch: Enhanced Matching Commands

J. Niederer

June 3, 1996

BNL MAD Program Notes

Fmatch: Enhanced Matching Commands

J. Niederer

AGS Department
Brookhaven National Laboratory

June 3, 1996

1. Introduction

The group of Match commands of the older Version 7 of MAD in use at the BNL AGS has been enhanced to support a wider range of calculations based upon machine measurements, and to take advantage of the more capable data base and faster BNL FastMad tracking features. All of the older features have been retained, with the same attribute nomenclature. The material in Version 7 manuals has been retained without change, and these notes supplement the original material. Closed orbit, off momentum, and second order tracking features have been added, generally in the spirit of more recent additions in the CERN Version 8. New constraint features include fitting to values of first and second order matrix elements, before and after difference forms, local orbit values, and more specialized forms that can be readily modified for custom use. A wider range of parameters, including magnet offsets and field errors, can be varied. Silicon Graphics GL based graphics and the data base Object Inspector couple to this new version. Keywords for the newer features are named by adding the letter "F" to the original keyword. In general, new and old forms of Match keywords should not be mixed within a Match group of commands.

2. FMatch and FCell

These commands open a Match group, and correspond to the ordinary *Match* and *Cell* commands. **FMatch** has additional attributes X, Px, Y, Py for orbit initial values, and off momentum options. Optionally, other tracked quantities can be named as variables and their initial values given.

Certain results are also stored in named cells in the command module after calculations. Names of these temporary cells filled by the program are usually prefixed with the slash ("/") character. Their contents may be accessed by parameter references, or by any of the data logging and viewing utilities of MAD. In recent practice, storage cells have been included in the various data base objects of the program, to permit viewing and access to the information by the various data and debugging services.

In the following, command attributes already in the older version are listed without comment. New entries *X, Px, Y, Py, Deltap,* and *Orbit* are consistent with MAD Version 8.

2.1. Fmatch Command

Line	Same
Betx	Same
Alfx	Same
Mux	Same
Bety	Same
Alfy	Same
Muy	Same

X	Orbit Starting X, decimal in meters. (In Version 8)
Px	Orbit Starting Px, decimal in radians
Y	Orbit Starting Y, decimal in meters.
Py	Orbit Starting Py, decimal in radians.
Dx	Same
Dpx	Same
Dy	Same
Dpy	Same
Deltap	Off momentum, as fraction of central momentum. Decimal. (In Version 8)
Orbit	Logical Flag: If True, orbits are tracked. (In Version 8)
Others(6)	Names of extra variables to be set to starting values. MAD Variable format: Element or Command_name [Attribute_name (Index)]
Otherv(6)	Starting values of named extra variables. Decimal.
Second	Logical Flag: If True, full second order tracking is done.
Clear	Logical Flag: If True, the collection of Dconstraints will be marked as not initialized. Needed if multiple fitting runs are done within a Match group.
Testmode	Logical Flag: If True, a trial matching setup will be run even if an X or Y motion stability condition is violated. Using this option with a debugging trace may help in diagnosing the cause of the problem.
/lineptr(2)	Internal storage - pointer to expanded line module if *Line* used.
/cx	Internal storage - receives orbit X at end of match.
/cpx	Receives orbit Px at end of match.
/cy	Receives orbit Y at end of match.
/cpy	Receives orbit Py at end of match.

2.2. FCell Command

Deltap	Off momentum, decimal, as fraction of central momentum. (In Version 8)
Orbit	Logical Flag: If True, orbits are tracked. (In Version 8)
Second	Logical Flag: If True, full second order tracking is done.
Clear	Logical Flag: If True, the collection of Dconstraints will be marked as not initialized. Needed if multiple fitting runs are done within a Match group.
Testmode	Logical Flag: If True, a trial matching setup will be run even if an X or Y motion stability condition is violated. Using this option with a debugging trace may help in diagnosing the cause of the problem.
/cx, /cpx, /cy, /cpy	Same as for FMatch

3. Variables to Be Fitted

3.1. FVary Command

FVary corresponds to the regular MAD Vary command used to note which parameters are to be varied during the matching. Usage corresponds to the examples in the MAD manuals. Storage cells have been added for data logging during Showmatch runs.

Name	Same, name of variable (parameter) in MAD Variable format: Element [Attribute (Index)]

Step	Same
Lower	Same
Upper	Same
/value	Receives decimal value of variable at start of fitting.
/vsave	Receives decimal value of variable at a stage in fitting.
/vbest	Receives decimal value from best fit to constraints.
/pbest	Receives decimal penalty value for best fit to constraints.
/status	An internal integer noting status of constraint (and other) command module: state of initialization, error conditions, open / closed, etc.
/stamp	Marker that notes whether constraint is a member of current Match group. Has value of input line counter when command was read or enabled.

A much broader selection of variables can be reached with the Fmatch package than with the older versions. Magnet position and angle misalignments may be of particular interest, and can be modeled against constraints of measured orbit data, along with survey data. Within the program, an internal table of misalignment quantities is generated with an entry for each lattice element marked for error treatment when the lattice map is expanded. The language of the MAD Error command group also allows for more than one occurrence of a given element. Thus another language feature may be needed to pin down the occurrence and the table location of a particular misalignment variable. Where secondary tables are involved, the more general variable name format becomes:

Element_name (Occurrence_index : Table_marker) [Attribute_name (Index)]

The *occurrence index* has the usual meaning of noting which of possibly multiple copies of a generic lattice element is to be selected within a beam line. If this index is skipped, the first occurrence of the named element is assumed.

The *table marker* notes the search chain to be followed in seeking the attribute name of the variable. It is set negative to distinguish it from the occurrence index in cases where only a single index may be given. The notation *Element1 (-4) [Adx]* is accepted as noting the attribute *Adx* of *Element1* located in the fourth kind of data table given below.

-1 Optional. Presumes attribute belongs to ordinary element module, not a secondary table. Default.

-2 Causes search for variable in one of the secondary tables attached to the current lattice map. Secondary tables include alignment and field errors, preserved matrices and factors, and optical functions.

-3 Causes search for variable/attribute name in a secondary structure attached to command or element module, such as the snapshot of orbit results appended to an FTwiss command. This search is carried out automatically by the program, so it need not be specified.

-4 Search for variable/attribute name in one of the secondary tables attached to the current copy of the lattice map used in matching. This form is to be used when trying to model magnet misalignments via matching.

Names of alignment errors which can be accessed for matching are:

Adx	Displacement of X {Y, S} coordinate of lattice element in meters.
Ady	
Ads	
Adphi	Misalignment of tilt angle Phi {Theta, Psi} in radians. (See Error Section of MAD Manual for particulars)
Adtheta	
Adpsi	

Example: Vary X displacement of a Quad during Matching -
V.hf8 FVary Qhf8(1:-4)[adx] Step = .0005, Lower = -.0050, Upper = .005

The full complement of names of quantities in secondary tables is given in the dictionary file called **FMDict.call.** Many entries are terms and factors which are retained to speed up tracking passes and hence may not be of general interest.

3.2. FFix Command

FFix has the same parameter as the regular MAD Fix command.

Name Name of the variable.

4. Constraints

Constraints are numerical conditions on an orbit that are to be satisfied by some combination of the parameters being matched to these conditions. A typical constraint is keyed to a particular lattice element, but may also cover a range of elements often with some restriction about upper and lower limits of a particular constraint, such as a beta function. Each parameter is matched to a set of constraints by varying the parameter until it has a minimal effect on a penalty function which measures the success of the match. In MAD the penalty function is taken as the sum of the weighted squares of the differences between the stated value of each constrained quantity and its value computed from the varied parameters. In practice, a penalty function can become a complicated landscape with numerous minima. While the program will search mechanically for the lowest minimum or set of similar minima, a proper match usually depends more upon a realistic, properly weighted set of constraints. Judgments and compromises about the relative importance of conditions are expressed to the program by means of the several versions of constraint statements discussed in this section.

4.1. FConstraint Command

FConstraint corresponds to the regular MAD Constraint command. Orbit position and angle constraints, and an option to name additional constrained variables have been added. Up to a total of twenty different constrained items may be treated on each statement; fourteen predefined orbit items, and six optional items for the user to specify. Orbit quantities and details of the individual penalty calculations are recorded in the constraint module at each pass, and may be reached by any of the data base utilities for viewing, logging, and plotting. These storage cells may be reused; if a range of lattice elements is involved, only the values for the last location of the range are saved on each pass.

Place Same, location or range of locations of constraint.

Line Same, name of Beam Line if involved in constraint.

Betx Same, numerical value and kind of constraint.

Alfx Same, constraint.

Mux Same, constraint.

Bety Same, constraint.

Alfy Same, constraint.

Muy Same, constraint.

X Numerical value and kind (=, >, <) of orbit X constraint. (In Version 8)

Px Same for Px constraint.

Y Same for Y constraint.

Py Same for Py constraint.

Dx Same, Dx constraint.

Dpx Same, Dpx constraint.

Dy Same, Dy constraint.

Dpy	Same, Dpy constraint.
Others(6)	Names of other extra constraints in MAD Variable form. Data base variables. (Optional, user supplied)
Othercs(6)	Decimal value and kind (=, >, <) of extra named constraints.
/icons(4)	Internal pointers - give first entry in scanned constraint arrays. Used to reduce extent of loops in constraints. Entries: 1) index, 2) constraint, 3) input array, 4) internal array.
/lcons(4)	Internal pointers - give last entry in scanned constraint arrays.
/placptr	Internal pointer to Place data group.
/lineptr	Internal pointer to Line data group.
/cmin(20)	Internal minimum values of constraints.
/cmax(20)	Internal maximum values of constraints.
/cwgt(20)	Internal weight values of constraints.
/cdif(20)	Internal differences between stated and computed values of constrained quantities.
/cpen(20)	Internal penalty values of constraints. Showmatch demonstrations often track members of */cpen* arrays to show contributions of individual constraints to the total penalty function.
/cval(20)	Internal computed values of constrained quantities.
/ictype(20)	Internal integers - constraint types. (1, 2, 3)
/icvari(20)	Internal integers - index of constraint position in list.
/mapcon(6)	Internal pointers to expanded Map, if *Line* option is given. (Data vector form of m, i, j, k, l, n components)
/optics(16)	At each pass, the current optical functions at the position (last position) of the constraint are copied into this set of cells in the FConstraint and some of the other kinds of constraint module. This snapshot of orbit values at the constraint position may be referenced by other parts of the program, including current constraints. Decimal. Contents are: 1. Betax 2. Alfax 3. Mux * twopi 4. Betay 5. Alfay 6. Muy * twopi 7. Orbit X 8. Orbit Px 9. Orbit Y 10. Orbit Py 11. Disp X 12. Disp Px 13. Disp Y 14. Disp Py 15. Sum L 16. Delta P

/stamp, /status

4.2. FCouple Command

FCouple corresponds to the regular Couple constraint command. Up to twenty constrained items may be involved, of which six are user supplied options. Like the other "Fast" constraint forms, it has holding cells which can be sampled by the data utilities, or user composed parameters.

Range	Same, MAD Place form. Beginning and end of region for which eight optical functions are to be matched.
	(Betax, Betay, Alfx, Alfy, DispX, DispPx, DispY, DispPy)
Mux	Same, alternative X tune constraint.
Muy	Same, alternative Y tune constraint.
Others(6)	Names of additional orbit quantities to be constrained at ends. (Optional, user supplied)
Othercs(6)	Decimal values of additional named constraints.
/icons(4), /lcons(4)	Internal pointers - give first, last entry in scanned constraint arrays. As above for **FConstraint.**
/rangptr(2)	Internal integer - pointer to *Range* place group in input.
/cmin(20)	Internal minimum values of constraints. Same as for FConstraint.
/cmax(20)	Internal maximum values of constraints.
/cwgt(20)	Internal weight values of constraints.
/cdif(20)	Internal differences between stated and computed values of constrained quantities.
/cpen(20)	Internal penalty values of constraints.
/cval(20)	Internal computed values of constrained quantities.
/ictype(20)	Internal integers - constraint types.
/icvari(20)	Internal integers - index of constraint position in list.
/stamp, /status	

4.3. FRMatrix Command

FRMatrix is similar to the first order matrix constraints of later Versions 8 of MAD at CERN. The values of one or more of the elements of the local matrix formed between two lattice endpoints may be specified for constraining. An element value must be stated, and the corresponding weight must be non zero for the constraint to be enabled.

Range	MAD Place form, giving beginning and end of region for which a local transfer matrix is to be computed. Needed.
RM(6,6)	Constraints on matrix elements between two end points. Only those elements supplied will be constrained. Decimal.
Weight(6,6)	Decimal values of weights to be given to constrained matrix elements. A weight is squared when used. Defaults are 0.
Save	Name of Matrix form of internal table to receive copy of computed first order transfer matrix. Optional, used only if *Save* name given. Matrix can be printed with MAD Dump utility.
/R1(6,6)	Internal accumulative matrix, advanced at each lattice element of range. Alternates with */R2* to reduce recopying time.
/R2(6,6)	Internal accumulative matrix, alternates with */R1*.
/flag12	Internal Logical Flag to note which of two alternating */R1, /R2* matrices are current.

/rangptr Internal pointer to *Range* place data group.

/stamp, /status

Example:

```
RM.11  FRMatrix, RM(1,1) = 1.002, RM(3,3) = .980,    &
                 Weight(1,1) = 2., Weight(3,3) = 1., &
                 Save = RRM.11_
```

4.4. FTMatrix Command

FTMatrix is a second order version of the FRMatrix constraint described above. It should be applied judiciously, as each occurrence in a range takes about 2600 data base pool spaces.

Range MAD Place form, giving beginning and end of region for which transfer matrix is to be computed. Needed.

TM(6,6,6) Constraints on second order matrix elements between two end points. Only those elements supplied will be constrained. Decimal.

Weight(6,6,6) Decimal values of weights to be given to constrained matrix elements. A weight is squared when used.

Save Name of Matrix form of internal table to receive copy of computed first order transfer matrix. Optional, Matrix written only if *Save* name is given. Matrix can be printed with MAD Dump utility.

Tsave Name of Matrix form of internal table to receive copy of computed second order transfer matrix. Optional, Matrix written only if *Tsave* name is given.

/R1(6,6) Internal first order accumulative matrix, advanced at each lattice element of range. Alternates with */R2* to reduce recopying time.

/R2(6,6) Like */R1*.

/T1(6,6,6) Internal second order accumulative matrix, advanced at each lattice element of range. Alternates with */T2*.

/T2(6,6,6) Like */T1*.

/flag12 Internal Logical Flag to note which of two alternating */R1* - */T1*, */R2* - */T2* matrix sets are current.

/rangptr Internal pointer to *Range* place data group.

/stamp, /status

Example:

```
TM.11  FTMatrix, TM(1,1,1) = 2.002, TM(3,3,3) = -.990,    &
                 Weight(1,1,1) = 1., Weight(3,3,3) = 2., &
                 Save = TRM.11_, Tsave = TTM.11_
```

4.5. FJConstraint Command

This constraint handles related amplitude constraints, such as betax and alfax, jointly in a single function, following Syphers and Sen, and antecedents. [ref 1]. These are sometimes called the mismatch functions. Up to four such pairs, betax and alfax, betay and alfay, dispx and disppx, and dispy and disppy, may be treated at a lattice point. The joint constraint functions have the form:

The ratios here are of fitted - desired values relative to desired values.

The ratios in F2 are of given error values relative to desired values.

For the equality type of constraint:

For the less than form of constraint, with the F1 / F2 ratio > than the given constraint:

The denominator *F2* is composed of the desired values at the constraint plus given errors, which is compared with achieved values in the numerator *F1*. This arrangement produces a unit penalty when the fitting matches the given errors *dBetx,* etc.

FJConstraint attributes are:

Place — MAD place form, giving location or range of locations of constraint.

Betx — Decimal value wanted at constraint. Zero values are refused.

dBetx — Decimal value of permitted error in *Betx.* If omitted, a number of the order of .1 + 5% of *Betx* will be defaulted.

Alfx — Decimal value wanted at constraint. Paired with *Betx.* Both members of pair must be given. Zero values are refused.

dAlfx — Decimal value of permitted error in *Alfx.* Will be defaulted similarly to *dBetx.*

Bety, dBety, Alfy, dAlfy — Same meanings as for joint constraint in *Betx, Alfx.*

Dx — Decimal value wanted for X dispersion at constraint. Paired with *DPx.* Zero values are refused.

dDx — Decimal value of permitted error in *Dx.*

DPx — Decimal value wanted for X' dispersion at constraint. Zero values are refused.

dDpx — Decimal value of permitted error in *DPx.*

Dy, dDy, DPy, dDPy — Same meanings as for joint constraint in *Dx, DPx*.

Constr(4) — Decimal values for joint constraint functions as activated. Because of the form of the penalty function, for the equality (and default) case, the constraint value is zero. For the less than case, the constraint value(s) should nominally be ~ 1., which is the default, or less than some value similar in size. NOTE: these entries are compared with functions which contain the squares of error terms. The greater than case is meaningless, and defaulted to the equality case.

Weights(4) — Weights for joint constraints. Default is 1.

/icons(4), /lcons(4) — Pointers for minimizing loops through constraint list.

/placptr(2) — Internal pointers to *Place* data group.

/wbetx, /wdbetx, /walfx, /wdalfx

/wbety, /wdbety, /walfy, /wdalfy

/wdx, /wddx, /wdpx, /wddpx

/wdy, /wddy, /wdpy, /wddpy

Working values for corresponding inputs. Some of the error entries may have been defaulted or otherwise adjusted during checking of inputs.

/pen — Value of penalty function for all parts of constraint command.

/cmin(4), /cmax(4) — Internal minimum, maximum values for constraints.

/cwgtsq(4) — Working values of weights squared for joint constraints.

/cpen(4) Computed penalty values for joint constraints.

/cval(4) Given values of joint constraints.

/cf1(4) Computed values of square roots of numerators of joint constraints.

/cf2(4) Computed values of square roots of denominators of joint constraints.

/cdif(4) Difference of *cf2 - cf1*. Gives a vague measure of disagreement of fitted Beta, etc with desired value at constraint.

/ictype(4) Internal integers - constraint types.

/icvari(4) Internal integers - index of constraint position in list.

/optics(16) Usual snapshot of orbit functions at constraint.

/stamp, /status

/Open(4)

4.6. FXConstraint Command

FXConstraint is an extension of the FConstraint form that is basically a template for providing information to hard coded special forms of extra constraints, such as beam envelopes. A total of eight such possibilities are provided now, but the command is easily extended.

Place Place form of location or range of constraint. Same description as ordinary MAD Constraint.

Xenv Decimal value of X envelope constraint. (Sigma X?)

Yenv Decimal value of Y envelope constraint. (Sigma Y?)

Dxemit Decimal value of constraint coded as the area of an X dispersion ellipse:

Dyemit Decimal value of constraint coded as the area of a Y dispersion ellipse.

Gammatr Decimal value of transition gamma constraint. Meaningful only at end of orbit. Treated same as on "Twiss" summary page.

Alfa Decimal value of alfa constraint. Same comments as for *Gammatr*.

Chromx Decimal value of chromaticity constraint in X. Meaningful only at end of orbit. Treated same as Qx' on Twiss summary page, which is perhaps controversial, as it is explicitly second order.

Chromy Decimal value of chromaticity constraint in Y. (Qy' on "Twiss" summary page). Same comments as for *Chromx* above.

Cons9 Dummy names for additional constraint possibilities.

Cons10

Weights(10) Weights in decimal for the above constraints. Default is 1. Does not affect weights carried by **Fweight** command.

/icons(4), /lcons(4)

Internal pointers - give first and last entry in scanned constraint arrays.
By 1) index, 2) constraint, 3) input array, 4) internal array.

/placptr(2) Internal pointer to Place data group.

/cmin(10), /cmax(10)

Internal minimum, maximum values of constraints. Same as for FConstraint.

/cwgt(10) Internal weight values of constraints.

/cdif(10) Internal difference values between stated and computed values of constrained quantities.

/cpen(10) Internal penalty values of constraints.

/cval(10) Internal computed values constrained quantities.

/ictype(10) Internal integers - constraint types.

/icvari(10) Internal integers - index of constraint position in list.

/optics(16) Tracked optical parameters. Listed in FConstraint above.

/stamp, /status

4.7. FDConstr Command

The difference constraint is designed to help fit model parameters to experimental data. Measurements are taken before and after some change in running conditions is made, and conditions in the model are changed similarly. In one kind of application, orbit measurements are made, and model parameters are computed by expecting their results to match differences in model orbits similar to those of the measurements. For various reasons, models may not usually reproduce measured orbits very well in an absolute sense, but often can do quite well with differences between two sets of conditions.

Constraints are expressed as before and after conditions, using either the name of a program variable or its value. Up to four sets of difference constraints can be applied at the same point with the same difference command.

Place Same description as ordinary MAD Constraint - Place form of location or range of constraint. Typically an orbit position monitor.

V0(4) Name(s) of first model value(s) in MAD variable form. (Before) These are initial values, treated as fixed in the fitting.

V1(4) Name(s) of second model value(s) in MAD variable form. (After) These variables are presumably dependent on the parameters being varied.

Cv0(4) Name(s) of corresponding first measured value(s) in MAD variable form. (Constraint)

Cv1(4) Name(s) of second measured value(s) in MAD variable form. (Constraint)

Val0(4) Decimal value(s) of first model variable(s), alternative to naming *V0* variable(s).

Val1(4) Decimal value(s) of second model variable(s), alternative to naming *V1* variable(s).

Cval0(4) Decimal value(s) of first constrained variable(s), alternative to naming *Cv0* variable(s).

Cval1(4) Decimal value(s) of second constrained variable(s), alternative to naming *Cv1* variable(s).

CVDiff(4), Decimal value(s) of difference between two sets of measurements. (Constraint) Alternative to supplying both members of difference constraint.

Weight(4) Decimal values of weights to be applied to each stated constraint. Squared as used. Constraint is active only if weight is non zero. These weights apply only to the constraints given on this command.

/placptr(2) Internal pointer to Place data group.

/icons(4), /lcons(4)

Internal pointers - give last entry in scanned constraint arrays.
By 1) index, 2) constraint, 3) input array, 4) internal array.

/values(4,4) Internal array holding values for each of four (before/after, model/constraint) cases, and up to four active entries, gleaned from input opportunities.

/diffs1(4) Internal array of model before / after difference(s).

/diffs2(4) Internal array of constraint before / after difference(s).

/cmin(4), /cmax(4)

Internal minimum, maximum values of constraints. Same as for FConstraint.

/cwgt(4) Internal weight values of constraints.

/cdif(4) Internal difference values between stated and computed values of constrained quantities.

/cpen(4) Internal penalty values of constraints.

/cval(4) Internal computed values constrained quantities.

/ictype(4) Internal integers - constraint types.

/icvari(4) Internal integers - index of constraint position in list.

/optics(16) Array holding copy of optical functions at constraint place. Same as for FConstraint.

/status, /stamp

Example:

```
XD.C2    FDConstr,     PLACE = PUEHC2, Weight = 1.,    &
            V0  = TwX_0[value(9)],          V1 = PUEHC2["/X"]    &
           CV0 = MeasX_0[value(9)],         CV1 = MeasX_1[value(9)]
```

Here a difference constraint is built from *Cv0* "before" and *Cv1* "after" measured monitor values located in data base vectors MeasX_0 and MeasX_1. The comparison difference *V0* "before" results from a Twiss model run, located in data base vector TwX_0. (Monitor PUEHC2 is the 9th entry in these vectors) The *V1* "after" value will be read from the PUEHC2 monitor /X cell each time the constraint is evaluated.

4.8. FWeight Command

This command provides weights for the ordinary **FConstraint** and **FCouple** constraints. It has additional entries for closed orbit position and angle coordinates, and so is not interchangeable with the ordinary Weight command. Values given are squared when used.

Active entries of weight data are read into a default weight table. **FWeight** values are in effect for all affected constraints which follow a command, until another FWeight is given. Nominal defaults are:

1. Betx, Bety

10. Alfx, Alfy, Mux, Muy, X, Y, Dx, Dy, Others

100. Px, Py, Dpx, Dpy

Weight attributes are keyed to constraint variables according to:

Betx

Alfx

Mux

Bety

Alfy

Muy

X Orbit X, Px, Y, Py weights.

Px

Y

Py

Dx

Dpx

Dy

Dpy

Others(6) These weights apply to optional variables added to **FConstraint** and similar commands.

The newer D, J, and X constraint commands have their own sets of *Weights* attributes, and are not affected by the **FWeight** values.

5. Formal Fitting Commands

5.1. Fmigrad Command

The first two attributes listed here for reference here are the same as for the regular Migrad command. **FMigrad** works only with the newer data base features.

Toleranc

Calls

Verbose Logical Flag: If True, causes additional printing to help with diagnostics.

5.2. Fsimplex Command

Like FMigrad, **FSimplex** retains its original attributes, and works only with the newer data base features.

Toleranc

Calls

Verbose Logical Flag: If True, causes additional printing to help with diagnostics.

This command has been considerably extended to deal with limiting constraints which may achieve zero penalties. For details, see the *Match.note* reference.

5.3. Showmatch

Showmatch produces a visual display of the results of iterations which try to minimize constraint based penalty functions. The command appears in the Match sequence in the same role as **FMigrad** and **FSimplex.** Showmatch is coupled to data logging utilities, to a relatively complicated graphical service, and to the optional data base Inspector, for which setup files are available. Showmatch has its own manual, but its attributes are listed here for reference.

Steps

Trace

Cycles(2)

Verbose

Vmode

Vary(10)

Plotdef

Sbutton

Inspector

/value

/penalty0

/penalty

/stamp, /status

6. Miscellaneous

6.1. FMBetween Command

The **FMBetween** command uses the matrix element constraint mechanism to evaluate one or more transfer matrices between specified pair(s) of lattice points. These points are specified on the **FRMatrix** and **FTMatrix** commands described above. All Match printing and execution is bypassed if no other

matching is done.

An **FMBetween** command group contains:

1. An **FMatch** or **FCell** command to begin.
2. One or more **FRMatrix** and / or **FTMatrix** commands to give end points.
3. An **FMBetween** command to operate the calculation.
4. An **FMCleanup** command to end.

If other matching commands are used within the group, the group should end with the usual **FEndmatch** command.

Print	Logical Flag; If True, print all first order (R) matrices computed in Match group.
Save	Logical Flag; If True, save all first order (R) matrices computed in Match group. Matrices are written to Matrix modules, using the names given on the **FRMatrix** commands involved.
Tprint	Logical Flag; If True, print all second order (T) matrices computed in Match group.
Tsave	Logical Flag; If True, save all second order (T) matrices computed in Match group. Matrices are written to Matrix modules, using the names given on the **FTMatrix** commands involved.

6.2. FLevel Command

FLevel is the same as the regular MAD Level command.

Level	A small integer governing volume of printing.

7. Printing, End of Run Commands

7.1. FCovar Command

FCovar is a new command that causes the printing of the covariance matrix, which expresses the dependence of each variable Vi on the penalty function at each constraint Cj.

(Second derivative matrix of penalty function vs Vi, Cj)

7.2. FEndmatch Command

FEndmatch closes the run, and prints results. Same function as the regular MAD Endmatch command, but internally different to deal with enhanced constraint features.

7.3. FMCleanup Command

FMCleanup is a new command that finishes runs such as **FMBetween** that make use of Matching commands just for tracking purposes.

Reference

M. Syphers, T Sen., "Notes on Amplitude Function Mismatch", SSCL Report SSCL-604, October 1992.

Documentation

Unix troff typeset format

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

This Report:	/usr/disc2/jn/Docum+/FMatch.man
Related Reports:	/usr/disc2/jn/Docum+/Showmatch.man /usr/disc2/jn/Docum+/Matrix.man /usr/disc2/jn/Docum+/Primer.boos /usr/disc2/jn/Docum+/Match.note
Dictionary File:	/usr/disc2/jn/Newmadd+/FMDict.call
Command File:	/usr/disc2/tmp/jn/FMatch.call