SLAC–285
January 1990
(A)

USERS GUIDE TO THE PROGRAM DIMAD*

ROGER V. SERVRANCKX
University of Saskatchewan, Saskatoon, Canada
and
Stanford Linear Accelerator Center
Stanford University, Stanford, CA 94309 USA

KARL L. BROWN
Stanford Linear Accelerator Center
Stanford University, Stanford, CA 94309 USA

LINDSAY SCHACHINGER
SSC CDG, Lawrence Berkeley Lab
Berkeley, CA 94720 USA

DAVID DOUGLAS
CEBAF Project, SURA
Newport News, VA 23606 USA

* Work supported by Department of Energy contract DE–AC03–76SF00515, and the National Science and Engineering Research Council of Canada.

TABLE OF CONTENTS

ABSTRACT**

The program DIMAD studies particle behaviour in circular machines and in beam lines. The trajectories of the particles are computed according to the second-order matrix formalism.[1] It does not provide synchrotron motion analysis, but can simulate it. The program provides the user with the possibility of defining arbitrary elements to tailor the program to specific uses. The present version of DIMAD is not fully debugged. Please inform one of the following persons about any anomalies observed:

David Douglas at CEBAF: (804) 249-7512.
Douglas at CEBAF–VAX.

Roger Servranckx at Saskatoon: (604) 222-1047.
Servranc at TRIUMFCL.

** The present guide corresponds to the program version dated FEBRUARY 7, 1989.

INTRODUCTION

DIMAD, like its predecessor DIMAT, is the result of many years of experimenting with several different charged-particle computer codes.

In 1970, the first author had the good fortune of discovering the program OSECO (Optique du SECond Ordre) written by J. L. Laclare at SACLAY. Basically, OSECO was a second-order tracking program. It was based on the second-order matrix formalism of TRANSPORT and was originally written for a CDC computer. Its usefulness in the simulation of the extraction procedure of the Beam Stretcher ALIS and later of EROS led to the desire for a program that would have more analysis power. The first attempt to develop a new program resulted in the program DEPART which was written as a pure differential equation ray tracing program, but it soon became clear that DEPART was very awkward to use because of the cumbersome way in which bending magnets were defined in the code. An evolutionary process then took place over a period of several years, finally resulting in the present program called DIMAD.

Many people contributed in various ways to the development of DIMAT. Dr. Leon Katz, while he was director of the Linear Accelerator Laboratory at Saskatoon, provided strong support for the work. Sheila Flory, Dean Jones, Edward Pokraka, Jim Morrison, and Jean Mary Miketinac provided programming support at different times during the initial program development. Ideas were borrowed freely from the program OSECO and Jean Louis Laclare helped formulate some of the early developments. Karl L. Brown of SLAC became influential during the later development phases. He helped formulate the more recent contributions to the program (geometric aberrations, linear analysis of motion around arbitrary reference orbits, and magnet misalignment simulations).

The authors wish to thank the many DIMAT users of other laboratories for their comments and assistance in locating the many programming errors that have occurred during the evolution of DIMAD.

In 1984, it became clear that tracking codes should operate in a canonical environment, should provide options for symplectic tracking, and should conform to the input STANDARD.[2]

Adapting the input code of MAD,[3] Lindsay Schachinger transformed the program so it would enjoy a common input with MAD, thereby conforming to the input STANDARD.

With ideas developed originally by E. Forest,[4] David Douglas introduced the symplectic tracking options and the canonical variables.

ELEMENT AND MACHINE DATA INPUT

The input format to DIMAD now conforms quite closely to the standard format, as laid out in Ref. 2. This conversion of DIMAD to standard input was accomplished by taking the input subroutines from the program MAD[3] and making from these routines (with modifications) an input interface for DIMAD.

One exception to the standard format is the units conventions.

Input to DIMAD can be in either transport units (indicated by the keyword **utransport**), or in standard units (indicated by the keyword **ustandard**). For more information on units, see the next section. The second difference between DIMAD and the standard is the addition of several keywords for DIMAD. The added keywords are **quadsext**, **gkick**, and **mtwiss**. These elements are described more fully later. Also, in DIMAD, the solenoid can have a quadrupole field. Elements which are described in Ref. 2, but which are not implemented in DIMAD are *separator* and *rbend.*

The job title is entered on a line following one with the keyword **title**. This should be followed with a units keyword. If no units keyword is found, the units are assumed to be the standard units.

UNITS

The keyword **utransport** indicates that the input has the following units:

- angles in degrees, except for dx' and dy' for the kicks, which are always in radians.
- lengths in meters.
- energy in GeV.
- accelerating voltage in kV.
- frequency in Hz.
- field expansion is $B(x,0) = (B\rho \sum K_n x^n)$.
- positive K_1 is horizontally focusing.

Also, the **utransport** keyword has implications for the field expansion coefficients in the *sbend* element.

The keyword **ustandard** indicates that the input has the following units:

- angles in radians.
- lengths in meters.
- energy in GeV.
- accelerating voltage in MV.
- frequency in MHz.
- field expansion is $B(x,0) = (B\rho \sum K_n (x^n/n!))$
- positive K_1 is horizontally focusing.

GENERAL SYNTAX

When describing the machine, a statement can be continued on the next line by ending the current line with an "&". A comment line begins with an "!". Any line containing one of the characters "!", "*", "(", or "@" in the first column is treated as a comment.

A ";" is used to separate statements on the same line. At most, eight letters in a keyword are checked. The keyword NOECHO can be used to suppress transmission of the input data stream defining the lattice and its elements to the output files. The keyword ECHO reinstates the stream of the input data to the output files.

PARAMETERS

Parameters are defined with a statement like

$$pname = value$$

where *pname* is any parameter name. Parameters can then be used in element definitions. Also, *value* can be an arithmetic expression involving other parameters. Throughout the element definitions, a parameter value can also be an arithmetic expression. Note that in DIMAD, the relationships between parameters are lost, but during the machine definition phase, they are treated correctly.

Examples:

$$lslot = 100$$
$$lb = lslot/8$$
$$lh = sqrt(lslot)$$

Note: The value of PI, if needed, must be defined as an input parameter. The value half-turn must be understood as either π radians or as 180°, depending on the units chosen.

ELEMENT DEFINITIONS

To define an element,

$$label: \ type \ [, pkeyw = value, \ldots]$$

where *label* is the name of the element, *type* is an element type (see below) and *pkeyw* is a parameter keyword appropriate for the element type (see below). Again, *value* can be a parameter name, or an expression involving parameters.

Examples:

$$b: \ sbend, \ l = lb, \ angle = lb/rho$$

$$d0: \ drift$$

ELEMENTS

A list of all element types and the relevant parameter keywords follows. Unless otherwise noted, all values default to zero except the aperture, which defaults to 1 meter.

drift

l is the length.

sbend

l is the length.

angle is the bend angle.

k1 if the standard convention is being used, $k1$ is given by the field expansion in the UNITS section. If transport conventions are being used, $k1$ is $n = -(\rho/B_0)\ (dB_0/dx)$.

e1 is the entrance edge angle.

e2 is the exit edge angle.

tilt is the tilt angle. If tilt is entered with no value, half-turn/2 is assumed.

k2 if the standard convention is being used, $k2$ is given by the field expansion in the UNITS section. If transport conventions are being used, $k2$ is $\beta = (\rho^2/2B_0)\ (d^2B_0/dx^2)$.

h1 is the entrance pole face curvature.

h2 is the exit pole face curvature.

hgap is the entrance half-gap size.

fint is the entrance fringe field integral, which defaults to 0.5.

hgapx is the exit half-gap size. If *hgapx* is not given a value, it defaults to the value of *hgap*. During fitting, however, both *hgap* and *hgapx* must be varied together.

fintx is the exit fringe field integral. If *fintx* is not given a value, it defaults to the value of *fint*. During fitting, however, both *fint* and *fintx* must be varied together.

rbend

The *rbend* is a parallel-faced dipole magnet. Its parameters are the same as those of the *sbend*. Parameters *e*1 and *e*2 are not provided by the user and are set by the program to half the value of the bend angle.

quadrupole

l	is the length.
*k*1	is the strength.
tilt	is the tilt angle. If *tilt* is entered with no value, half-turn/4 is assumed.
aperture	is the magnet aperture for the hardware operation.

sextupole

l	is the length.
*k*2	is the strength.
tilt	is the tilt angle. If *tilt* is entered with no value, half-turn/6 is assumed.
aperture	is the magnet aperture for the hardware operation.

quadsext

l	is the length.
*k*1	is the quadrupole strength.
*k*2	is the sextupole strength.
tilt	is the tilt angle. If *tilt* is entered with no value, half-turn/4 is assumed.
aperture	is the magnet aperture for the hardware operation.

octupole

l	is the length.
*k*3	is the strength.
tilt	is the tilt angle. If *tilt* is entered with no value, half-turn/8 is assumed.
aperture	is the magnet aperture for the hardware operation.

multipole

l	is the length. If the length is zero, the strengths are interpreted as integrated strengths.
*k*0–*k*20	are the strengths.
*t*0–*t*20	are the tilt angles. If *tn* is entered without a value, half-turn/2$(n+1)$ is assumed.
scalefac	is a dimensionless strength factor, used to scale all the strengths together.
tilt	is the overall tilt angle.
aperture	is the magnet aperture for the hardware operation.

Note (1): Only the components with nonzero amplitude are stored! If zero components need be kept for the purpose of generating errors via the ERROR definition, then enter components with small amplitudes.

Note (2): When a quadrupole or a sextupole component is present the matrix of this component is computed for half the length of the multipole. This does not change the value of the total second-order matrix. During tracking operations, the particles are tracked through half the element as quadrupole or sextupole; then the higher-order multipole kicks are applied and the particles are tracked through the second half of the quadrupole or sextupole component. This feature is important in computing misalignment effects with multipole components present.

solenoid

l	is the length.
ks	is the solenoid strength. $ks = (0.5/B\rho)\ Bs$.
$k1$	is the quadrupole strength.
tilt	is the tilt angle. If *tilt* is entered with no value, half-turn/4 is assumed.
aperture	is the magnet aperture for the hardware operation.

rfcavity

l	is the length.
volt	is the cavity voltage (kV for **Utransport**).
lag	is the phase lag of the cavity with respect to a nominal particle (0,0,0,0,0,0) at the start of the machine (degrees for **Utransport**).
$freq$	is the frequency of the cavity (Hz for **Utransport**).
energy	is the energy (GeV).

roll

This element performs a rotation of the coordinate system about the longitudinal axis.

angle	is the rotation angle. A positive angle means the new coordinate system is rotated clockwise about the s-axis with respect to the old system.

zrot

This element performs a rotation of the coordinate system about the vertical axis. The angle must be small.

angle	is the rotation angle. A positive angle means the new coordinate system is rotated clockwise about the local z-axis with respect to the old system.

hkick, vkick

These elements are translated by the program into general kicks (*gkick*).

kick a horizontal (vertical) kick of size *kick*.

angle angle of rotation about the longitudinal axis.

gkick

This element is a general kick.

l is the length.

dx is the change in x.

dxp is the change in x'.

dy is the change in y.

dyp is the change in y'.

dl is the change in path length.

dp is the change in (dp/p). $\delta = (\Delta p/p)$.

angle is the angle through which the coordinates are rotated about the longitudinal axis.

dz is the longitudinal displacement.

v is the entrance-exit parameter of the kick. v is positive for an entrance kick, and negative for an exit kick. The absolute value of v is used to force the kick to be applied every $abs(v)$ turns. The default value of v is 1.

t is the momentum-dependence parameter. The kicks dx' and dy' can be thought of as misalignment errors or as angle kicks of orbit correctors. In the first case ($t = 0$), they are momentum independent. When $t = 1$, the kicks dx' and dy' vary inversely with momentum. When t is set to a negative integer value $-n$, the kick is applied every turn and the momentum of a particle with initial momentum p will oscillate around the nominal momentum $p0$ with amplitude $(p - p0)$ and a period equal to n turns.

More than one such kick may be put in the line (all identical, though); the phase of the cosine oscillation is proportional to the pathlength of the reference trajectory up to the location of its kick.

hmon, vmon, monitor

These elements are horizontal, vertical, and horizontal and vertical monitors, respectively.

l is the monitor length.

xserr, *yserr*, *xrerr*, *yrerr*
are the x and y systematic and random errors.

Note: The errors are not used in this form presently. Errors are introduced via the misalignment operations.

marker

A marker is a drift element of zero length. It has no parameters.

ecollimator, rcollimator

An *ecollimator* is elliptic, and an *rcollimator* is rectangular. The particles are checked at the entrance and at the exit of the collimator.

l is the length.

xsize, *ysize* are the x and y collimator apertures. The default apertures are 1 meter.

arbitelm

This is the arbitrary element. Its parameters are used in the user-supplied routine TRAFCT, which contains the transfer function describing the effect of arbitrary elements on the individual particles. All arbitrary elements use the same subroutine. Distinct arbitrary elements can only be recognized by the program through the use of one parameter as a flag.

l is the length.

*p*1–*p*20 are the parameters.

mtwiss

l is the length.

mux, *betax*, *alphax*, *muy*, *betay*, *alphay*
are the Twiss parameters for this transfer matrix. *betax* and *betay* have default values of 1.

matrix

This element is a general transfer matrix.

rij, *tijk* are the matrix elements. i, j, and k range from 1 to 6, but j is always less than or equal to k.

BEAMLINE DEFINITIONS

A beamline is a list of elements, which can include other beamlines.

$$label: \ line = (member1, \ member2, \ member3, \ldots)$$

denotes a beamline called *label*. The members can be elements, other beamlines, sequences of members, or any of the above preceeded by a repetition count and/or a minus sign for reflection. Examples are:

$$df: line \ = \ (dq, \ oo, \ b, \ oo \ qf)$$

$$fdstar: line \ = \ (qf, \ sf, \ b, \ sd, \ qd)$$

$$arc: line \ = \ (df, \ 64 * (fdstar, df))$$

Beamlines can also have formal arguments. An example is:

$$fdstar(sf, sd): line \ = \ (qf, \ sf, \ b, \ sd, \ qd)$$

where sf and sd are not defined elements, but variables. So,

$$super: line \ = \ (fdstar(sd1, sf1), df, fdstar(sd2, sf2))$$

is a line in which the elements $sd1$, $sd2$, $sf1$, and $sf2$ are substituted for the variables sd and sf in the original definition.

CONTROL FLOW

Beamline definitions are followed by a use statement in the form

use, beamlinename.

This causes the beamline *beamlinename* to be the current machine for DIMAD.

Next comes the statement

dimat

which passes control to DIMAD, after translating the machine into the correct data structures for DIMAD. Any DIMAD command can then be issued.

A ";" will cause DIMAD to stop and return control to the input interface. Now the user can define a new machine and then go back to DIMAD and do a new calculation, or stop execution with the command STOP. The use command causes the old machine to be replaced by a new one. This new machine can be a previously defined beamline. For debugging purposes, the dump command from MAD has been retained. This command produces a dump of the MAD-type data structure describing the machine.

After a ";" and return to MAD control, one can specify the use of a new line, keeping the previously defined (and perhaps modified by DIMAD) elements. To do so, one uses the commands:

use, newlinename

newbeam

The last command, *newbeam*, passes control back to DIMAD.

Observe that with *dimat* instead of *newbeam*, all the element parameters are redefined to their initial input values.

A new MAD command is introduced: EXPLODE. Its purpose is to provide an explicit description of the beamline used.

OPERATION LIST DESCRIPTION

Each array specifying an operation starts with a title line of 80 characters or less. The first four nonblank characters (capitalized in the following presentation) specify the operation and MAY NOT BE ALTERED.

The lines following the title may have 72 characters.

Any line containing one of the characters "!", "*", "(", or "@" in the first column is treated as a comment line.

Each array terminates with a "," or a ";". In the first case, another operation is expected; in the second case, control is returned to the interface program.If the user desires to stop the run at this point, the line following the ";" must contain the MAD command STOP.

In all tracking operations, the particle coordinates are checked at the entrance of some element. Particles are lost when the square of the radial excursion is greater than the expulsion factor. It is set at the default value of 1. Its value can be changed via the constant definition operation.

GENERAL OPERATIONS

ADIABATIC VARIATIONS

BEAM MATRIX TRACKING

CONSTANT DEFINITION

DETAILED CHROMATIC ANALYSIS

GENERATION OF PARTICLES

GEOMETRIC ABERRATIONS

HARDWARE VALUES LISTING OF MACHINE

INTERACTIVE MANIPULATION OF LATTICE

LEAST-SQUARE FIT

LINE GEOMETRIC ABERRATIONS

MACHINE AND BEAM PARAMETERS COMPUTATIONS

MATRIX COMPUTATION

MODIFICATION OF ELEMENT DATA

MOVEMENT ANALYSIS

OUTPUT CONTROL

PARTICLE DISTRIBUTION ANALYSIS

PRINT SELECTION

PROGRAM GENERATION

RMATRIX COMPUTATION (6×6)

SEISMIC PERTURBATION SIMULATION

SET FIT POINT

SET SYMPLECTIC OPTION ON

SHO VALUES OF CONSTANTS

SIMPLE FITTING

TRACKING OF PARTICLES

ADIABATIC VARIATIONS

This operation enables the user to vary parameters of elements during particle tracking operations. At the present stage, this operation destroys the original value of the parameters varied and so cannot be used in fitting or repeatedly in the same job. Two options are available: linear and sinusoidal variation.

Input format:

ADIAbatic variations of some parameters (up to 80 char)

name pkeyw nopt p_1 p_2 val_1 val_2 val_3 val_4

...

name pkeyw nopt p_1 p_2 val_1 val_2 val_3 val_4

99,

Parameters:

name name of element having a parameter to be varied.

pkeyw keyword of parameter to be varied (*i.e.*, $k1$ for a quad).

nopt option number.

1 means variation will be linear according to the following rule: the parameter remains constant at value val_1 until turn p_1, then varies linearly to achieve the value val_2 at turn p_2. In this case, only two parameter p and only two values val are present in the input format.

2 means the variation will be sinusoidal between turn p_1 and turn p_2. The variation is done according to the formula:

$$value = val_1 + val_2 * sin((2\pi * turn/val_3) + val_4)$$

where *turn* is the current turn at which *value* is applied. Outside turns p_1 and p_2, the original value is applied.

BEAM MATRIX TRACKING

Computes beam matrices at selected points of the machine from the initial beam matrix defined in the input of the operation.

If sig_i and sig_0 denote the beam sigma matrices at the entrance and exit of a beam line section, then

$$sig_0 = R * sig_i * R^t$$

where R and R^t are the transformation matrix of the section and its transpose.

Input format:

BEAM matrix tracking computations ... (up to 80 char)

$\sigma_x\ r_{xx'}\ r_{xy}\ r_{xy'}\ r_{xl}\ r_{xp}$

$\sigma_{x'}\ r_{x'y}\ r_{x'y'}\ r_{x'l}\ r_{x'p}$

$\sigma_y\ r_{yy'}\ r_{yl}\ r_{yp}$

$\sigma_{y'}\ r_{y'l}\ r_{y'p}$

$\sigma_l\ r_{lp}$

σ_p

mprint [list]

or

0

$\beta_x\ \alpha_x\ \eta_x\ \eta'_x\ c_x$

$\beta_y\ \alpha_y\ \eta_y\ \eta'_y\ \epsilon_y$

$\sigma_l\ \sigma_p$

mprint [list]

or

0

0 0 0 0 ϵ_x

0 0 0 0 ϵ_y

$\sigma_l\ \sigma_p$

mprint [mlist]

Parameters:

σ_i σ extension of the beam [as defined in Refs. (1) and (5)].

r_{ij} correlation cosines as defined in Ref. (1).

β_x, α_x, η_x, η'_x, β_y, α_y, η_y, η'_y
initial values of Twiss parameters used to define an uncoupled beam.

ϵ_x, ϵ_y emittances in x and y of the input beam.

Note: When β_x etc. are zero, the values are obtained from a previous movement analysis calculation made within a MATRix operation or a Fit operation that generates a matrix calculation.

mprint

-2 no computation is done. The operation serves only to define a beam as needed in the operations BEAM tracing and DETAiled analysis.

-1 print final result only.

0 print all intermediate and final results.

n $n > 0$ used with list. There are n intervals in which printing will occur.

mprint $+1000$
when 1000 is added to the value of *mprint*, the printing occurs in the same fashion as above, but a table of beam envelopes is printed instead of the full beam matrix.

list contains the beginning and end of all intervals in which printing is done. *list* is a set of pairs of numbers. They are positions in the order list of machine elements). *list* may contain up to *mxlist* numbers (set at 40 initially). See also the PRINT operation.

CONSTANT DEFINITION

This operation allows the user to redefine basic constants. The purpose of this operation is to enable comparison of the computation results with other programs or to update the values as their accuracies increase.The constants accessible to the user are: π, the velocity of light (in m/sec), the electron mass (or particle mass) (in GeV), the electron (or particle) radius, and the electron (or particle) charge. The reference relative momentum ($\Delta p/p$) that is used in some Taylor expansion with δ as independent variable. Two parameters used in the least-square minimizer routine are also accessible to the user as well as the expulsion factor. The scale factors ETAFAC and SIGFAC are also accessible via this operation.

Use the operation SHO Constant to examine the constants.

Input format:

CONStant definition... (*up to* 80 *characters*)

$n_1, val_1, \ldots\ n_p, val_p$

Parameters:

n_i is the order number of the ith constant to be redefined according to the following order: π, velocity of light, particle mass, particle radius, particle charge, reference energy used in Taylor series expansions, least-square fit initial tolerance, factor for maximum function calls in least-square fit, the expulsion factor, $etafac$, $sigfac$.

val_i new value of the constant with order number n_i.

DETAILED CHROMATIC ANALYSIS

Traces particles (2 per plane, per momentum) to determine the linearized transfer matrix from the initial point to any related point in the lattice. A Twiss function computation is then done at these points. The initial central particle position is assumed to be $x_0\ x_{0'}\ y_0\ y_{0'}$.

Note: No kicks simulating synchrotron oscillation (parameter $T < 0$) may exist in the lattice. Results are meaningless in the presence of such kicks and of cavities.

For each energy e_i, particles are generated around the point P_0 ($x_0\ x_{0'}\ y_0\ y_{0'}$) to compute the elements R_{ij} of the matrix describing the linear motion around the trajectory defined by the point P_0.

Input format:

DETAiled chromatic analysis ... (*up to* 80 *characters*)

$nh\ nv\ nhv$

$x_0\ x_{0'}\ y_0\ y_{0'}$

$dx\ dx'\ dy\ dy'$

$\beta_x\ \alpha_x\ \eta_x\ \eta'_x$

$\beta_y\ \alpha_y\ \eta_y\ \eta'_y$

$Nener\ Ncoef$

$e_1\ e_2 \ldots\ e_{nener}$

mlocat [*list*]

Parameters:

nh	1	only xx' motion is traced and computed. β_x, α_x, and ν_x are computed.
	0	xx' motion alone is not analyzed.
nv	1	only yy' motion is traced and computed. β_y, α_y, and ν_y are computed.
	0	yy' motion alone is not analyzed.
nhv	1	coupled motion xx', yy' is traced. The full beam matrix is computed.
	2	coupled motion is computed. A short print lists $x\ x'\ y\ y'\ \beta_x\ \alpha_x\ \beta_y\ \alpha_y\ \nu_x\ \nu_y$ for all energies requested.

3 in this case, three energies have to be defined: δ_0, $\delta_0 - \epsilon$, $\delta_0 + \epsilon$. This enables computing the basic machine parameters associated with energy δ_0. Choose ϵ comfortably close to zero for accurate computation of the η functions. A short print of the machine parameters is provided. The values of η and η' are affected by a scale factor $etafac$ which can be set via the constant definition operation. Its default value is 1.0. When set to 10^3, the printed values are in mm and mrad.

4 the beam matrix values (as defined in the BEAM operation) computed for one energy are printed in a convenient table format. The values of the matrix values for the beam σs (not the correlation coefficients) are affected by the scale factor $sigfac$ which can be set via the constant definition operation.

5 same as 3 (above), with the code of the element added in the first column to facilitate some plotting work.

Note: The input beam must have been defined previously in a BEAM MATRIX TRACKING operation.

0 coupled xx', yy' motion is not analyzed.

When nh, nv, and nhv are all zero, the program prints the centroid positions only.

dx, dx', dy, dy'
increments at which the off-orbit particles are placed to compute the s_x, c_x, s_y, and c_y functions [see Ref. (1)].

β_x, α_x, η_x, η'_x, β_y, α_y, η_y, η'_y
initial values used in the Twiss function computations.

$Nener$ number of energies for which the analysis is done (maximum 15).

$Ncoef$ number of coefficients used in the Taylor series expansion as a function of momentum (maximum 6).

$e_1 \ldots e_{nener}$ momentum values in the form

$$\frac{\Delta p}{p_0} = \frac{p - p_0}{p_0}$$

mlocat indicates the number of intervals in which printing is to occur. If $mlocat = -1$, then printing occurs at end of lattice only and no number is in list.

list a set of pairs of numbers each of which indicate the beginning and end position (in the order list of machine elements) of each of *mlocat* intervals in which printing takes place. *list* may contain up to *mxlist* numbers (initially set at 40). See also the PRINT operation.

GENERATION OF PARTICLES

This operation generates a set of particles to be used subsequently in one or more particle tracking operations. Presently, only Gaussian distributions can be generated. This operation MUST ALWAYS be preceded by a BEAM definition operation and by a SEED operation.

Input format:

GENEration of particles

nopt $\sigma_1 \ldots \sigma_6$ *scale npart*

$x_0, x_0', y_0, y_0', al_0, \delta_0$

Parameters:

nopt: 1 the particles are randomly generated on the surface of a six-dimensional ellipsoid (defined previously by a BEAM operation. In this case, the values of σ_i are not operational (but for computational efficiency, they should be set to 1).

3 the particles generated have coordinates that satisfy a six-dimensional Gaussian distribution.

Note: The same value for the option parameter must be used in the PARTicle analysis operation (if used) following the tracking of such particles.

σ_i the number of σs above which the Gaussian distribution is truncated for each of the six variables x, x', y, y', al, δ. The beam is defined by a previous BEAM operation which is assumed to define the one σ distribution.

scale scales the beam size by the given factor.

npart number of particles to be generated. Maximum number is *mxpart* (initially set at 1000).

x_0, x_0', y_0, y_0', al_0, δ_0
centroid coordinates around which the beam is generated.

GEOMETRIC ABERRATIONS IN MULTIPLE-TURN OPERATION

This operation traces particles that are placed on ellipses with nominal emittances ϵ_{xi}, ϵ_{yi} for many turns. It then fits an ellipse to the output points obtained. From this fitted ellipse, it determines the average values for β_x, α_x, β_y, α_y, ν_x, ν_y, ϵ_x, and ϵ_y. It also computes the maximum and minimum emittances which inform about the diffusion pattern of the motion. Variances of the tunes are also computed. A fast Fournier analysis can also be performed.

Input format:

GEOMetric aberrations... (*up to* 80 *characters*)

β_x, α_x, β_y, α_y

x_{co}, x'_{co}, y_{co}, y'_{co}, δ

ncase, *nturn*, *njob*

nplot, *nprint*

ϵ_{x_1}, ϵ_{y_1}

...

ϵ_{xncase}, ϵ_{yncase}

anplprt

Parameters:

β_x, α_x, β_y, α_y

input values of the Twiss parameters at the entrance of the lattice. When $\beta_x = 0$, the Twiss parameter values are obtained from a previously run movement analysis with *nanal* not zero. The values corresponding to the first energy are used. This includes the parameters x_{co} to δ.

x_{co}, x'_{co}, y_{co}, y'_{co}, δ

coordinates and momentum of the closed orbit around which the aberrations are to be computed. When $\beta_x = 0$, these parameters are obtained from a previous movement analysis operation. Values corresponding to the first energy are used.

ncase number of cases analyzed. MXGACA originally set at 10.

nturn number of turns for tracing. MXGATR originally set at 1030.

njob

1 coupled motion analysis is wanted.

2 uncoupled motion analysis is wanted.

nplot

1 plotting of the resulting particles. The operation always accumulates the particles at every *nplot* turns, but plots the accumulation at the end of the job. It also computes its own plotting windows.

−1 no plotting.

nprint

−2 no printing.

−1 printing at end of lattice only.

0 printing after every element.

n printing after every n turns. Normally *nprint* should be set equal to *nturn*.

ϵ_{x_i}, ϵ_{y_i} values for the chosen nominal emittances in x and y, using the unit mm-mrad (E–06 mrad).

anplprt parameter selecting the fast Fourier transform options. When 0, no Fourier transform is performed. When 1, the Fourier transform components are printed. When 10, the amplitude of the Fourier transform is printer-plotted. When 100, an analysis of the peaks is provided. A combination of those values is allowed; *e.g.*, when 111, all three are done.

It is advised to trace for at least 500 turns, preferably 1000. The number of turns should have as many low-valued factors as possible to benefit from the speed of the fast Fourier transform.

Only the first case of the geometric aberration run is Fourier analyzed.

HARDWARE VALUES LISTING OF MACHINE

Computes the geometry of the lattice and parameters related to the strengths and fields of the magnetic elements.

Note: Presently, this operation works only with transport units. The run must have started with the command UTRANSPORT.

Input format:

hardware layout and element parameters ... (up to 80 char)

E s x y z θ φ ψ conv mprint [list]

Parameters:

E momentum (GeV/c) used for computation of field values.

s x y z coordinates of starting point in some absolute reference coordinate system. The coordinate s is the length of arc along the reference trajectory. To justify the choice of the angles θ, ϕ, and ψ, the z-axis should coincide more or less with the longitudinal axis of the beam. The angles θ, ϕ, and ψ describe the motion needed to bring the absolute system of reference in coincidence with the local system of coordinates. The local system of coordinates is the system used by the program. Its z-axis is tangent to the reference trajectory. The x-axis (uniquely defined by the bends) is in the midplane of symmetry and points outwards of the bend. The y-axis completes the local right-handed system of reference. To bring the absolute system in coincidence with the local reference system, one executes the following rotations (strictly in the order indicated):

A rotation θ around the y-axis (positive when the z-axis turns towards the x-axis).

A rotation ϕ around the x-axis (positive when the z-axis turns towards the y-axis: *i.e.*, points upwards for a bend deflecting the beam to the right).

A rotation ψ (called sometimes the roll) around the z-axis (positive when the x-axis turns towards the y-axis).

conv conversion factor to enable the printout in various practical units. For feet, the conversion factor is, for example 0.3048 (the length of a foot in

meters). The program recognizes yards, feet, inches, cm, mm, microns. However, any conversion factor is accepted even if not recognized.

mprint

-2	no printing of results.
-1	printing final result only.
0	print all intermediate and final results.
n	$n > 0$ used with *list*, there are n intervals in which printing will occur.
list	contains the beginning and the end of all intervals in which printing is done. *list* is a set of pairs of numbers. *list* may contain up to *mxlist* numbers (set at 40 initially). See also the PRINT operation.

INTERACTIVE CONTROL OF LATTICE

This operation enables to vary parameters of chosen elements while observing the beam at an end point. The beam has to be defined in a previous BEAM operation and a previous GENERATION of particles. The beam can be observed in a printer-plot or by its statistical parameters. The particles are tracked individually in each element. This operation is not fully developed and debugged! It is NOT machine independent!

Input format:

INTEractive control of ... (*up to* 80 *char*)

niopt nivar

name keyword (*repeated nivar times*)

Parameters:

niopt option parameter not used now.

nivar number of parameters to be varied (maximum 8).

At run time, follow the instructions of the program. This CANNOT work if at implementation of the program the output channel 9 has NOT been assigned to the terminal.

LEAST-SQUARE FIT

This operation handles any fitting problem. Some care must be exercised in the choice of *nstep* and *nit*. Experience will show what choices are best suited to the problem. A safe choice is 2 2 (1 1 is faster but less accurate). If the program is very slow at finding a solution, or if an overflow condition is developed in the subroutine LMDIF, the solution sought is probably not a practical one. A new minimizer (LMDIF) was installed in December 1984. It has a default tolerance and default increments for the variables which seem adequate. As a consequence, the input parameters Δ_i have no influence. We have kept them to avoid changes in the input format until we are satisfied with the new minimizer.

Input format:

LEASt square fit of ... (up to 80 char)
nstep nit nvar ncond
β_x, α_x, η_x, η'_x, β_y, α_y, η_y, η'_y
$name_i$ $pkeyw_i$ Δ_i *for* $i = 1$ *to nvar*
$nval_j$ $valf_j$ $weight_j$ *for* $j = 1$ *to ncond*
nasp
repeat the following nasp times
$name_1$ *npas*
$name_k$ $pkeyw_k$ $coef_k$ *for* $k = 1$ *to npas*

Parameters:

nstep — number of steps taken to approach final fit.

nit — number of iterations used in final step of fit.

nvar — number of independent variables (max: MXLVAR set at 50).

ncond — number of conditions to be met (max: MXLCND set at 100).

β_x, α_x, η_x, η'_x, β_y, α_y, η_y, η'_y
initial values needed for the function computation. When β_x value is entered as zero, then the program uses the $\beta_x \ldots \eta'_y$ values computed in the last matrix operation preceding the present operation.

$name_i$ — name of element with an independent parameter to be varied.

$pkeyw_i$ variable element parameter keyword.

Δ_i this parameter is not used in the new minimizer implementation, but was kept in the input to avoid a major change in the input format.

$nval_j$ reference number of output value to be fitted. Numbers 1 to 20 are for the values of the stable motion analysis of the total matrix, in the same order as mentioned in the SIMP operation.

Numbers 21 to 30 refer to $\beta_x\ \alpha_x\ \eta_x\ \eta'_x\ \nu_x\ \beta_y\ \alpha_y\ \eta_y\ \eta'_y\ \nu_y$ at the end of the machine. These values are computed from the initial values present in the second line of the input format.

Numbers 31 to 40 refer to the values $\beta_x \ldots\ \nu_y$ computed at the first fit point defined by the preceding SET Fit point operation.

Numbers 1031 to 1040 refer to the difference between the values $\beta_x \ldots\ \nu_y$ computed at the first and second fit point defined by the preceding SET Fit point operation (*i.e.*, v_2–v_1).

Numbers 41 to 61 refer to the beam values $\sigma_x \ldots\ \sigma_p$ and the r_{ij} at the end of the machine. See operation BEAM for the meaning of these parameters and the order in which they appear. These values can only be fitted if a BEAM operation defining the beam values at the begining of the machine has preceded the fitting operation.

Numbers 71 to 91 refer to the same beam values computed at the first fit point defined by the operation SET Fit point.

Numbers 1071 to 1091 refer to the differences of the same beam values computed at the first and second fit point defined by the operation SET Fit point (*i.e.*, $v2$–$v1$).

Numbers 93 to 98 fit the average chromatic errors for β_x, α_x, β_y, α_y ν_x, ν_y as computed in the detailed chromatic analysis operation. A fit on these elements can only be done after a previous detailed chromatic analysis operation is done which serves to define the parameters needed for the computation.

Selected numbers 110 to 666 specify matrix elements in the following fashion:

- $ij0$ represents the first-order matrix element $R_{i,j}$.
- ijk represents the second-order matrix element $T_{i,j,k}$ (as in the TRANSPORT notation).

$valf_j$ value to be achieved.

$weight_j$ weight attached to the value $valf_j$ in the fit function.

$nasp$ number of associated parameters. If $nasp = 0$, then the following data is not to be entered.

$name_1$ name of the basic element to which the associated parameters are connected. It must be present in the list of basic elements varied.

$npas$ number of parameters to be associated to $name_1$ (max: 6).

$name_k$ name of one element having a parameter associated to $name_1$.

$pkeyw_k$ keyword of the parameter of $name_k$ associated with $name_1$.

$coef_k$ coefficient with which the BASE parameter (that of $name_1$) is to be multiplied to obtain the value of the parameter of $name_k$.

LINE GEOMETRIC ABERRATIONS (ONE-TURN COMPUTATION)

This operation traces *npart* particles, placed on ellipses with nominal emittances ϵ_{x_i}, ϵ_{y_i} for one turn. It then fits an ellipse to the output points obtained. From this fitted ellipse it determines the average values for β_x, α_x, β_y, α_y, ν_x, ν_y, ϵ_x, and ϵ_y.

Input format:

line geometric aberrations... (*up to* 80 *characters*)

β_x, α_x, β_y, α_y

x_{co}, x'_{co}, y_{co}, y'_{co}, δ

ncase, *npart*, *ncoup*

nplot, *nprint*, *mlocat*, [*list*]

ϵ_{x_i}, ϵ_{y_i} $i = 1$ *to ncase*

Parameters:

β_x, α_x, β_y, α_y

input values of the Twiss parameters at the entrance of the line.

x_{co}, x'_{co}, y_{co}, y'_{co}, δ

coordinates and the momentum of the trajectory around which the aberrations are to be computed.

ncase number of cases analyzed (maximum 10).

npart number of particles to be traced (maximum MXPART set at 1000).

ncoup not used presently, but a value must be inserted.

nplot

1 plot the resulting particles at the end of the job. It computes its own plotting windows.

−1 no plotting.

nprint

−2 no printing.

−1 printing at end of the line only.

0 printing after every element.

mlocat number of intervals in which printing is to occur; used in conjunction with *list*. This is not yet implemented.

list intervals in which printing occurs. *list* may contain up to *mxlist* numbers (set at 40 initially).

ϵ_{x_i}, ϵ_{y_i} values for the chosen nominal emittances in x and y using the unit mm-mrad (E–06 mrad).

MACHINE AND BEAM PARAMETERS COMPUTATIONS

Computes β, α, η, η', ν values at selected points around the machine. If requested, beam parameters are computed. In some cases, the optimum coupling values may be meaningless (if coupling is > 1).

Input format:

MACHine and beam parameters ... (*up to* 80 *char*)

E_1 E_2 *dE nlum dnu nint nbunch*

β_x α_x η_x η'_x

β_y α_y η_y η'_y *mprint* (*list*)

Note: If E_1 is zero, then *nlum* is assumed to be zero and the input Twiss parameters values are those obtained in a previous matrix analysis. If E_1 is nonzero but β_x is zero, the first line of parameters must be given and the initial Twiss parameters values will be those of the preceding matrix analysis.

Parameters:

E_1	start momentum for beam data and luminosity computations.
E_2	end momentum.
dE	momentum step.
nlum	
0	no beam size related computations are done.
1	synchrotron integrals and basic beam size computations are done.
2	full luminosity computations are made.
dnu	*dnu* value used for optimum luminosity computation.
nint	number of interaction regions.
nbunch	number of bunches.
$\beta_x \ldots \eta'_y$	function values at starting point of lattice.

mprint

-2	no printing of results.
-1	print final result only.
0	print all intermediary and final results.
n	$n > 0$ is used with *list*. There are n intervals in which printing will occur.
list	beginning and end positions of each interval in which printing is done. *list* is a set of pairs of numbers. *list* may contain up to *mxlist* numbers (set at 40 initially). See also the PRINT operation.

MATRIX COMPUTATION

Computes matrices and performs movement analysis on matrix obtained at end of lattice.

Input format:

MATRix computations ... (*up to* 80 *char*)

norder mprint [*list*]

Parameters:

norder

1	first-order matrix only is printed.
2	second-order terms are also printed.
< 0	computation is done to order *abs*(*norder*). *mprint* must be > 0 and the program will print matrices of the beam line situated between, and including, the element pairs defined in LIST.

When *norder* is -1 or -2, the format of the output is identical to the input format of the *Gxxxxxx* element.

When *norder* is -11 or -12, the computation is to order 1 and 2 and the format of the output the standard program output for matrices.

mprint

-2	no printing of matrix.
-1	print matrix at end of machine only.
0	print all intermediary matrices plus final matrix.
n	where $n > 0$, used with *list* and indicates the number of intervals in which printing is to occur. See also the PRINT operation.
list	set of pairs of numbers which indicate the beginning and end position (in the order list of machine elements) of each of *mlocat* intervals in which printing takes place. *list* may contain up to *mxlist* numbers (set at 40 initially).

MODIFICATION OF ELEMENT DATA

Enables user to change input parameters between successive operations. It is particularly useful in simulations of injection and extraction processes in conjunction with the kick elements and the TRACking operation.

Input format:

MODIfication of input parameters... (*up to* 80 *char*)

n

name pkeyw value

name pkeyw value

...

name pkeyw value

Parameters:

n	number of varies to be made (= number of *name pkeyw value* entries which follow).
name	name of the machine element which is to be varied.
pkeyw	keyword of the parameter in the given element which is to be changed.
value	value the parameter is to be changed to.

MOVEMENT ANALYSIS

This operation finds closed orbits and analyses both stable and unstable motions for up to 15 different momenta.

Note: No kicks simulating synchrotron oscillation (parameter $T < 0$) and no cavities may exist in the lattice. Results are meaningless in the presence of such kicks and cavities.

Input format:

MOVEment analysis ... (up to 80 char)

nprint nturn nanal nit nener ncoef dist

$x\ x'\ y\ y'\ l\ \delta_1 \ldots\ \delta_{nener}$

naplt delmin delmax dnumin dnumax dbmin dbmax ncol nline

Parameters:

nprint — print action for the closed-orbit information.

- 0 — action after every element.
- n — action after every n turn.

nturn — number of turns over which analysis is performed. It enables user to study higher-order resonances.

nanal

- 0 — no stability analysis is done; only the closed orbit is computed.
- 1 — stability analysis is performed (both stable and unstable).
- 2 — gives information about an order-two resonance (*nturn* must then be equal to 2).
- 3 — gives information about an order-three resonance (*nturn* must then be equal to 3).

Note: In both cases, where *nanal* is equal to 2 or 3, the resonance motion analyzed is supposed to occur in the horizontal phase plane. If the user wants to study resonance in the vertical plane, the machine should be set up so that its planes are exchanged.

In the versions subsequent to April 1, 1988, the coordinates of the particles close to the unstable fixed point of the first momentum are stored

for subsequent use in a tracking operation. See the **demo3** input file for its use.

nit number of iterations used. $ABS(nit)$ iterations are performed. If *nit* is negative, only the results of the last iteration are printed.

nener number of momenta for which the analysis is performed (max 15).

ncoef number of coefficients to be used in the Taylor expansion of the parameters *nu*, *beta*, *eta*, and *etap* versus momentum. The reference momentum in the expansion is set at 0.005 (0.5%) (max 6). The reference momentum can be changed by the CONStant definition operation.

dist indicates the *distance* (in phase space) from the estimated position of the closed orbit at which the particles needed for the computation are initially generated. A safe choice is 0.001 or some lower value, depending on the size of the phase space occupied by the beam.

x, x', y, y', 1 estimate of the coordinates of the closed orbit.

$\delta_1 \ldots \delta_{nener}$ momenta ($\Delta p/p$) for which the analysis is performed.

naplt

0 no plot of the Taylor expansion is required.

1 a plot is required.

delmin, *delmax*

min max of ($\Delta p/p$) for the plot.

dnumin, *dnumax*

min max for *dnu* (the first momentum serves as the reference to compute the tune difference *dnu*).

dbmin, *dbmax*

min max for relative difference in *betas*.

ncol, *nline* number of columns and lines desired for plot.

OUTPUT CONTROL

This operation provides control of the output printout. It affects only the *dimat* part of the output.

Input format:

OUTPut control

nopt

Parameters:

nopt

0	all output is supressed (except error messages).
1	the main results of the computation only are printed.
2	the main results and the input data are printed.
3	all output is printed.
4	used for short printing of tracking results when such printing can be used for input to plotting programs.
14	same as 4 (above). The fifth coordinate will then be the phase relative to an RF cavity instead of the path length.

Notes: Not all output has been affected by this option in the present version of the program. In the operation section of the input data, this option should only appear between two operation arrays and not inside one such array. This operation cannot appear in the standard format input section. In this section, the command NOECHO can be used to suppress printout; the NOECHO can be reversed by use of the command ECHO.

PARTICLE DISTRIBUTION ANALYSIS

This operation is destined to provide some analysis of particle distribution.

Input format:

PARTicle distribution analysis

nopt

Parameters:

nopt must be equal to the parameter chosen in the particle generation. When equal to 1, the values for the beam sizes are independent of the choice of the scale parameter of the GENEration operation. This facilitates comparison between similar beams with different scale factors (useful in nonlinearity studies).

PRINT SELECTION

This operation allows the user to determine points or intervals at which results should be printed. Whenever, in some operation, the printing option is set to -2, -1, or 0, it does superede the print selection of the present operation. If a print selection has been defined, the print option within any subsequent operation should be set positive.

Input format:

PRINt selection
keyword
name; as many as needed
99
end,

The different keywords are: **interval, name, type.**

interval: allows the user to define up to MXLIST intervals (default 40). The intervals are defined by pairs of names of elements present in the currently used beamline. The names must be in ascending order of position and must be unique. Preferably, one should use markers. "99" is the flag that terminates the sequence of names. "99" is followed by another keyword or by "end".

name: followed by names of elements at which printing is to occur. The maximum number is 10. The names may contain the wild character "*". AB* means all names starting with the characters *ab* will produce printing.

type: the types are: *drift*, *bend*, *quadrupole*, *multipole*, *gkick*, *collimator*, *rfcavity*, *sextupole*, *solenoid*, *monitor*, *quadsext*, *matrix*, *mtwiss*, and *arbitrary*. Two types may be defined.

PROGRAM GENERATION

Note: THIS OPERATION IS TEMPORARILY NOT AVAILABLE.

RMATRIX COMPUTATION (6×6)

Computes in chosen intervals the 6×6 transfer matrix of the beamline comprised in these intervals. The computation is done by the tracking of seven particles chosen around a given initial set of coordinates. This enables to determine the first-order behavior of beamlines affected by errors and misalignments. The program also provides the entrance and exit orbit displacements. They are needed to use the matrix correctly.

Input format:

RMATrix ... (up to 80 characters)
$x_0\ x'_0\ y_0\ y'_0\ l_0\ \delta_0$
$d_x\ d'_x\ d_y\ d'_y\ dl\ d\delta$
norder mprint

Parameters:

$x_0\ x'_0\ y_0\ y'_0\ l_0\ \delta_0$
initial coordinates of reference orbit.

When $\delta_0 = 1$, then the coordinates are the coordinates of the closed orbit computed in a previous movement analysis. The values corresponding to the first energy are used.

$d_x\ d'_x\ d_y\ d'_y\ d_l\ d\delta$
increments used to generate the six particles surrounding the reference orbit.

norder order of the computation: 1 or 11. The order of the computation is 1. When *norder* = 11, the output is in the standard input format.

mprint number of intervals wanted.

nlist *mprint* pairs of numbers defining the intervals for which the matrix will be computed. See also the PRINT operation.

SHO VALUES OF CONSTANTS

This operation displays the values of the basic constants used in the program.

Input format:

SHO Values of the basic constants

no parameters are used for this operation.

SIMPLE FITTING

This operation is used for easy fitting (tunes, chromaticity). It uses Newton's method involving an equal number of conditions and variables.

Input format:

SIMPle fitting ... (up to 80 char)
nstep nit nvar
$name_i$ $pkeyw_i$ δ_i $(i = 1$ *to* $nvar)$
$nval_i$ $valf_i$ $(i = 1$ *to* $nvar)$
nasp
repeat the following nasp times
name pkeyw npas
$name_i$ $pkeyw_k$ $mult_k$ add_k $k = 1$ *to* $npas$

Parameters:

nstep number of steps to reach the final stage.

nit number of iterations performed in the last step in order to refine the variable values.

nvar number of variables and conditions (max: 10).

name name of element containing a variable.

pkeyw keyword of the parameter to be varied in the element.

del increment by which the variable is to be varied. This number is divided by 5 in every iteration of the last step.

nval order number of value to be achieved. The order is given by the following list:

$$compf\ \nu_x\ \eta_x\ \eta'_x\ \alpha_x\ \beta_x\ (d\mu_x/d\delta)\ \chi_x\ (d\alpha_x/d\delta)\ (d\beta_x/d\delta)\ \mu_y\ \nu_y\ \eta_y\ \eta'_y\ \alpha_y\ \beta_y\ (d\mu_y/d\delta)\ \chi_y\ (d\alpha_y/d\delta)\ (d\beta_y/d\delta),$$

where *compf* stands for the compaction factor in x. These values are those computed in the stable motion analysis of the matrix of the complete machine. Note that the momentum dependence of η cannot be fitted.

$valf$	the values to be achieved in the final step.
$npas$	total number of parameters to be associated to the parameter $pkeyw$ of $name_1$.
$name_k$	name of the element which has a parameter to be associated with $name_1$.
$pkeyw_k$	parameter keyword to be associated.
$mult_k$, add_k	multiplicative and additive constants which define the value of the associated parameter according to the following formula $$parvalue_k = mult_k * parval + add_k$$ where $parval$ is the value of the parameter used in the element $name_1$ and to which $pkeyw_k$ is associated.

SEISMIC PERTURBATION SIMULATION

Sets transverse misalignments according to sinewaves of some chosen frequency and amplitude as a function of the longitudinal coordinate. The vertical oscillation may be different from the horizontal oscillation.

This operation only affects the tracking of particles and all operations that use tracking.

Input format:

SEISmic simulation ... (up to 80 characters)

λ_{xs} a_{xs} ϕ_{xs}

λ_{ys} a_{ys} ϕ_{ys}

beginname endname

Parameters:

λ_{xs}, λ_{ys} wavelength (in m) for x and y waves.

a_{xs}, a_{ys} amplitudes of the x and y waves.

ϕ_{xs}, ϕ_{ys} x and y phase shift of each wave.

beginname, *endname*

name of elements where the wave is to start and where it is to stop. Use unique names, preferably markers.

SET FIT POINT

Sets an intermediate fit point to be used with the least-square fit operation.

Input format:

SET Fit point... (up to 80 characters)

n Position$_1$ (*Position*$_2$)

Parameters:

n	number of fit points defined (maximum 2).
Position$_i$	name (must be unique in machine list) of the element after which the fitted values are applied. It is recommended to use a marker for this purpose.

SET SYMPLECTIC OPTION ON

Sets the symplectic option on. As soon as this operation is executed, all the matrices are transformed to the six dimensional space defined by the canonical variables x, p_x, y, p_y, $-\tau = -t * c = -al$, $(\Delta E/E)$.

Please note that in the present implementation, the approximation $v/c = 1$ was made. All the movement analyses performed in the program relate to the matrix, so the values will change when this option is on. Please note that this operation changes the sign of the fifth parameter. This may need to be taken into account in the definition of the lag parameter of cavities!

Note: In the present version of the program , this option cannot be followed by any fitting which changes matrices. Any fitting not changing matrices is allowed (*e.g.*, in alignment fitting, when steering only is involved).

Input format:

SET Symplectic option on ... (up to 80 characters)
Option Energy

Parameters:

Option determines the mode of tracking. This affects only the operations based on tracking and not those based on matrix analysis (MATRIX, BEAM MATRIX, MACHINE FUNCTIONS).

0 nonsymplectic ray trace is done using the canonical matrices.

1 fast version of ray trace is done with the variables x, x', y, y', al, and δ, using the canonical matrices.

2 fast version of ray trace is done with the variables x, p_x, y, p_y, $-\tau$, and $(\Delta E/E)$, using the canonical matrices.

3 slow version of ray trace is done with the variables x, x', y, y', al, and δ, using the canonical matrices.

4 slow version of ray trace is done with the variables x, p_x, y, p_y, $-\tau$, and $(\Delta E/E)$, using the canonical matrices.

Note: Options 3 and 4 (above) are not for the general user. They have been used and maintained for debugging purposes only.

Energy energy of nominal particle (in GeV).

TRACKING OF PARTICLES

Tracks up to *mxpart* (1000) particles around the machine. The initial values of the coordinates of the particles are lost in the process of tracking. They are replaced by the final values of the coordinates.

Input format:

TRACking of particles... (up to 80 char)
nplot nprint npart nturn
particle data ($x\ x'\ y\ y'\ l\ \delta$ – *for all particles*)
mlocat list ngraph xmin xmax xpmin xpmax ymin ymax
ypmin ypmax ncol nline (*almin almax delmin delmax*)

Parameters:

nplot

0	plot action after every element.
−1	no plot action.
n	action occurs after *n* turns (used in conjunction with *mlocat* and *list*) at *mlocat* locations specified by *list* elements.

nprint

0	print action after every element.
−1	printing at end only.
−2	no printing occurs.
n	action occurs after *n* turns (used in conjunction with *mlocat* and *list*) at *mlocat* locations specified by *list* elements.

Note: When *nplot* = −1 and *nprint* = −2, then *mlocat* and *list* do not appear. *mlocat* and *list* are the same for plot and print.

npart	number of particles traced.
< 0	*abs*(*npart*) particles are added to particles already present from previous operation.
0	existing particles kept—none added.

> 0 previously used particles deleted. *npart* new particles introduced.

nturn number of turns to be traced.

x, x', y, y', 1, δ

particle data for *npart* particles.

mlocat indicates the number of intervals in which printing is to occur. If *mlocat* is equal to zero, then printing occurs at end of lattice only. When *mlocat* is zero, no number is in list.

list set of pairs of numbers, each of which indicates the begining and end position (in the order *list* of machine elements) of each of *mlocat* intervals in which printing takes place. *list* may contain up to *mxlist* numbers (set at 40 initially). See also the PRINT operation.

ngraph

1 plot x, x' plane.

2 plot y, y' plane.

3 plot x, y plane.

4 plot all planes.

11, 12, 13, 14 as above, but the graphs are accumulated and the plot is printed at the end.

15, 16 accumulates the *al*, δ or the ϕ, δ plots, where *al* is the pathlength coordinate of the particles, and ϕ is the phaseshift with respect to the frequency of the cavities (they must be present for this graph to be meaningful; the cavities need not be in phase with the total length of the machine). δ is the sixth coordinate of the particles.

Note that 15 will present a correct plot of the longitudinal phase-space only if the frequency of the cavity and the length of the machine match perfectly (8 digits usually!). Using the value 16 garantees a plot which uses the RF phase instead of path length differences and is always readable.

17 is equivalent to 14 (above) as regards the xx', yy' and xy plots and at the same time will produce an E ϕ plot identical to that of produced by the *ngraph* value of 16 (above).

xmin, *xmax*, *xpmin*, *xpmax*, *ymin*, *ymax*, *ypmin*, *ypmax*
limits for the plotting windows.

almin, *almax*, *delmin*, *delmax*
limits for the plotting windows for the cases *ngraph* = 15, 16 or 17. They are not present for the other values of *ngraph*. For *ngraph* = 16 or 17, *al* is to be interpreted as ϕ.

ncol, *nline* number of columns and number of lines to be used in plot matrix.

OPERATIONS ASSOCIATED WITH MISALIGNMENTS AND ERRORS

ALIGNMENT FITTING

BASELINE DEFINITION

BLOCK MISALIGNMENT

CORRECTOR DATA DEFINITION

ERRORS DATA DEFINITION

MISALIGNMENT DATA DEFINITION

REFERENCE ORBIT DISPLAY

SEED

SET CORRECTOR VALUES

SET ERRORS OF ELEMENTS

SET MISALIGNMENT OF ELEMENTS

SHO MISALIGNMENTS

SHO ERRORS

SYNCHROTRON RADIATION DATA DEFINITION

General note of caution: random generators produce different sequences on different computers, even when using the same initial seed. So results provided in the demos using such random generation may vary in detail, though the trends will be similar.

ALIGNMENT FITTING

This operation allows the user to fit values read in monitors (see their definition in the machine list). Any parameter can be used as variable. Successive use of this operation can simulate progressive alignment correction of a beamline. A new minimizer is installed since December 1, 1984. It has a default tolerance and default increments for the variables which seem adequate. As a consequence, the input parameters del_i have no influence. We have kept them to avoid changes in the input format until we are satisfied with the new minimizer.

Input format:

ALIGnment fitting... (*maximum* 80 *characters*)

nstep nit nvar ncond nfit nopter

$\beta_x\ \alpha_x\ \eta_x\ \eta'_x$

$\beta_y\ \alpha_y\ \eta_y\ \eta'_y$

$x_0\ x'_0\ y_0\ y'_0$

$dx\ dx'\ dy\ dy'$

$nener\ ener_1 \ldots\ ener_{nener}$

Origin

$name_i\ keywd_i\ del_i\ i = 1$ *to npar*

When nfit equals 1 *or* 2, *the following group applies* :

corr

$name_i\ pos_i\ opt_i\ param_i\ del_i\ i = 1$ *to ncor*

Note : $ncor + npar = nval.$

$mon_i\ pos_i\ val\#_i\ value_i\ weight_i\ error_i\ i = 1$ *to ncond*

End of the group for nfit 1 *or* 2.

If nfit equals 3, *the following group applies* :

cprr

mcorr

$name_i\ opt_i\ param_i$ *for* $i = 1$ *to mcorr*

nmon nskip

$name_i\ val\#_i\ value_i\ weight_i\ error_i$ *for* $i = 1$ *to nmon*

End of group for nfit 3.

nasp

repeat the following nasp times :

name keywd npas

$name_k$ $keywd_k$ $mult_k$ add_k *for* $k = 1$ *to npas*

Parameters:

nstep	number of steps taken to approach final fit.
nit	number of iterations used in final step.
nvar	number of variables used (maximum: MXLVAR set at 50).
ncond	number of conditions fitted (maximum: MXLCND set at 100).
nfit	selects the fitting procedure.
1	Newton's method is used. In this case, $ncond = nvar$.
2, 3	a least-square fit is used.
nopter	error option parameter for the reading of the monitors; this is a noise error of the reading.
0	the monitors have no errors.
1	the monitor error is the value given in the error parameter of the monitor (see below) multiplied randomly by + and − signs.
2	the monitor error is a uniform random distribution with a sigma equal to the error value.
3	the monitor error is a Gaussian distribution cutoff at two *sigmas*.
4	the monitor error is a Gaussian distribution cutoff at six *sigmas*.
11, 12, 13, 14	the random error is the same as for 1, 2, 3 or 4 with a fast random generation of the random sequence. This random could, in some cases, be affected by unwanted correlations. In case of doubt, check the STATISTICAL validity of your results with a family of runs using the options 1, 2, 3 or 4.

The initial seed used is the same as that defined by the operation SEED. The generation of the random errors for the monitors is INDEPENDENT of that of the misalignments and of the field errors.

β_x, α_x, η_x, η'_x, β_y, α_y, η_y, η'_y
: input parameters used in the computing the beam line function values.

x_0, x'_0, y_0, y'_0 : initial values of nominal orbit.

dx_0, dx'_0, dy_0, dy'_0
: increments used in the computation of the c_x s_x c_y s_y functions needed to generate the transfer matrices around the nominal orbit.

nener : number of momenta traced (maximum 3).

ener : values of the momenta $(p - p_0)/p_0$.

origin : position used as current origin to position the correctors used later. This position is be specified by the name of an element.

$name_i$, $keywd_i$
: names of the elements having parameters to be varied. $npar \leq nvar$ such elements can be used.

δ_i : not used in the present version, but must be present in the input.

corr : flag to signal that the correctors are going to be used.

mcor : for fit 3: number of corrector names. The program picks the first *ncor* available correctors whose name are any of $name_i$. Remember that $ncor = nvar - npar$.

$name_i$: name of corrector.

pos_i : relative position ($origin + pos_i$ is the absolute position of the corrector).

opt_i : option defining the type of corrector (see SETCorrector operation).

$param_i$: parameter number of parameter to be varied.

dcl_i : increment used in the fitting routine to vary the parameter.

mon_i : monitor name as present in machine list.

nmon : for $nfit$ 3: names of distinct monitors. The program picks *ncond* monitors whose name fits $name_i$ AFTER SKIPPING *nskip* monitors!

pos_i : relative position of the monitor with respect to the origin point.

$Val\#_i$: value number:
$= (iener - 1) * 4 + 1$ x value as read by monitor.
$= (iener - 1) * 4 + 2$ y value as read by monitor.

$= (iener - 1) * 4 + 3$ σ_x value as read by monitor.
$= (iener - 1) * 4 + 4$ σ_y value as read by monitor.

$value_i$ values read are those corresponding to momentum *iener* (1 to 3 maximum).

$weight_i$ used in conjuction with the least-square fit. This parameter enables the user to put more weight on certain values to be fitted. The higher the $weight_i$ the stronger the constraint to fit the $value_i$.

$error_i$ used in conjunction with the parameter *nopter*. If *nopter* is zero, no error affects the monitors. If *nopter* is > 0, the monitor mon_i is affected by the error $error_i$.

nasp number of associated parameters.

$name_1$ name of element to which some parameters are to be associated.

keywd parameter keyword of element $name_1$ to which some parameters are to be associated.

npas total number of parameters to be associated to the parameter $keywd_k$ of $name_1$.

$name_k$ name of the element which has a parameter to be associated with $name_1$.

$keywd_k$ keyword of parameter to be associated.

$mult_k$, add_k multiplicative and additive constants which define the value of the associated parameter according to the following formula

$$parvalue_k = mult_k * parval + add_k$$

where *parval* is the value of the parameter used in the element $name_1$ and to which $keywd_k$ is associated.

BASELINE DEFINITION

This operation defines a baseline resulting from surveying errors. The baseline must be considered as being like a new reference orbit. It is obtained by two successive operations. In the first, a few points on the original reference orbit are chosen as main surveying points. In tunnel construction, they could be associated with the surveyor's penetration points. They are accompanied by random x, y, z coordinate errors (usually rather big: say 5 to 10 mm) The second operation defines, between the preceding basepoints, intermediate points which are obtained by successive aiming from the current point to the next basepoint. This aiming is accompanied by a systematic aiming error (varying from segment to segment) to which is added a random aim error (usually smaller than the systematic error). The origin of the systematic error can be due not only to the instruments used but also to ambient conditions under which the surveying is performed. This operation MUST BE PRECEDED by a SEED operation. This operation is still being tested and developed. Use at OWN RISK.

Input format:

BASEline definition

npen $Kxxxxxx$ $Kxxxxxx \ldots$ $Kxxxxxx$

nsub σ

$Kyyy$ $Kyyy \ldots$ $Kyyy$

σ_1 σ_2

Parameters:

npen	number of penetration points.
$Kxxxxxx$	a set of *npen* elements of the *gkick* type. There MUST be one such kick both at the beginning and at the end of the lattice.
nsub	number of subdivision points.
σ	the displacement σ to be used in the generation of the coordinate displacements of the *npen* penetration points.
$Kyyy$	*nsub gkick* elements defining the intermediate points. They may not coincide with any of the basepoints.
σ_1	angular σ (in radians) of the systematic aim error.
σ_2	angular σ (in radians) of the random aim error.

BLOCK MISALIGNMENT

This operation sets up misalignment condition for subsets of a beamline. The whole subset is treated as if it were one element. However, any misalignment defined in the misalignment data definition will be superimposed. The block misalignment is implemented via *gkick* elements and uses a sequence of random numbers that is distinct from the other random numbers used in the program.

Input format:

BLOCk misalignment... (*maximum* 80 *characters*)

$Name_1$ $name_2$ dx dy dz dz_r $d\delta$

...

$Name_1$ $name_2$ dx dy dz dz_r $d\delta$

99,

Parameters:

$name_1$, $name_2$

name of two *gkick* elements whose names uniquely define the beamline interval to be misaligned as a block.

dx, dy, dz, dz_r, $d\delta$

one σ value of the random generation of the x, y, z offsets of the roll around the longitudinal axis and the relative field offset.

CORRECTOR DATA DEFINITION

This operation determines which elements in the machine are correctors. By corrector, we mean one element of a family (with the same name) whose position may be changed and/or whose setting may be changed to achieve corrections of closed orbits and/or beam size at the monitor locations. At the moment, only dipoles with small bend angles can be used as correctors.

Input format:

CORRector data definition ... (*maximum* 80 *characters*)

Name i_1 $f_1 \ldots$ i_n f_n,

...

Name i_1 $f_1 \ldots$ i_n f_n,

99,

Note: i_1, f_1 and i_n, f_n are pairs of numbers defining the intervals in which the elements *name* are to serve as correctors. MAXCOR(600) distinct elements can be used as correctors. Note the "," ending each line.

ERRORS DATA DEFINITION

This operation defines the errors that can affect certain parameters of elements.

Input format:

ERROrs data definition... (*maximum* 80 *characters*)

Name Parameter value... parameter value;

...

Name Parameter value... parameter value;

99,

Note: The semicolon ending each line defining errors for one element, and the line containing 99, ending the input. *parameter* is the parameter keyword of the element called *name* that is affected by the error. *value* is the value of the error.

MISALIGNMENT DATA DEFINITION

This operation defines the misalignments of different elements of the lattice. Up to 50 distinct elements can be misaligned.

Input format:

MISAlignment data definition... (*maximum* 80 *characters*)

Name dm_1 dm_2 dm_3 dm_4 dz dz_r $d\delta$ *option*

...

Name dm_1 dm_2 dm_3 dm_4 dz dz_r $d\delta$ *option*

99,

Note: The comma ending each line defining the misalignments. The list is terminated with 99.

Parameters:

For all values of the *option* parameter, the parameters dz and dz_r are the values of the longitudinal displacement and the rotation angle (in radians!) around the longitudinal axis. The values dm_1, dm_2, dm_3 and dm_4 are assumed to be small (either in displaments or angles). The program uses approximate formulae to set up the misaligned element.

The parameter *option* can take the three values 1, 2 or 3, which determines the nature of the misalignment.

option = 1 the element is misaligned around the tangent to the central trajectory at the entrance of the elements. In this case, the parameters dm_1, dm_2, dm_3 and dm_4 are, respectively, dx, dx_r, dy, dy_r; where dx and dy are the displacements along the axes x and y, and dx_r, dy_r are rotation angles (in radians) around the axes x and y, respectively.

option = 2 the element is misaligned around the chord defined by the two extreme points of the central trajectory. In this case, the parameters dm_1, dm_2, dm_3 and dm_4 are dx_1, dx_2, dy_1, dy_2 where dx_1, dy_1 are the displacements at the entrance of the element along the axes x and y. The parameters dx_2, dy_2 are the displacements at the exit of the element along the axes x and y.

option = 3 the element is misaligned around the tangent to the central trajectory at the midpoint of the element. The parameters dm_1, dm_2, dm_3 and dm_4 have the same meaning as in the case of the option value 1.

option = 4 this does not apply to dipole elements (bends). In this case, the parameters dm_1 and dm_3 represent displacements in x and y, respectively. The particle is also subjected at the entrance and the exit to an angle kick of $dm_2/2$ and $dm_4/2$ in x and y, respectively (same sign at exit as at entrance). This enables the user to simulate baseline excursion in a similar way as that defined under baseline operation. Parameters dz, dz_τ are in effect, but not $d\delta$. This should be mainly used on monitors.

REFERENCE ORBIT DISPLAY

This operation computes the orbit defined by the initial coordinates $x\ x'\ y\ y'\ al\ \delta$ ($\delta = (p - p_0/p_0)$), and provides either a printout or a printer-plot display.

Input format:

REFErence... (*maximum* 80 *characters*)

nprint $size_x$ $size_y$

$x_0\ x'_0\ y_0\ y'_0\ al\ \delta$

npos $pos_1 \ldots\ pos_{npos}$

Parameters:

nprint	controls display.
1	a printout is provided.
2	a printer-plot is provided.
11 or 12	same as above; the orbit is a four-dimensional closed orbit defined by the variables x, x', y, y'.
21 or 22	same as above; the orbit is a six-dimensional closed orbit on the variables x, x', y, y', al, δ.
$size_x$, $size_y$	always needed. They define the boundaries between which the coodinates x and y are plotted.
$x_0 \ldots$ *delta*	six coordinates of the input particle traced to determine the orbit.
npos	number of positions selected for interval computation; maximum 4.
pos_i	the rms values are computed individually in all the intervals defined by the values 0, pos_i, and *end of lattice*.

SEED

Using the clock of the computer, this operation generates and prints a seed to be used in the random generators.

Input format:

SEED... (*maximum* 80 *characters*)

n_s,

Parameters:

n_s 0	a seed is generated by the program.
$\neq 0$	n_s must be positive. The program insures that n_s is an odd number and prints the number used as the seed.

SET CORRECTOR VALUES

This operation is used to manually set correctors to some predetermined values.

Input format:

SET Corrector values

Name pos opt $p_1 \ldots p_4$,

...

Name pos opt $p_1 \ldots p_4$,

99,

Parameters:

Name	name of corrector element whose value is to be set.
pos	position of corrector.
opt	option number defining the kind of corrector.
$p_1 \ldots p_4$	the four parameters used to define the corrector.
$if\ opt = 0$	p_1 is dx, p_2 is dy, p_3 is dy' and p_4 is $d\delta$. The corrector element is displaced uniformly by dx and dy. It is preceded and followed by a momentum dependent kick of dy' (this simulates crudely the effect of backleg windings providing a B_x induction). The energy of the particle is changed by $d\delta$ (this simulates the effect of backleg windings providing a change in B_y).
$opt = 1$	the parameters have the same definition as above. The displacement dx and dy are imposed at the entrance of the magnet. The exit point of the magnet is assumed fixed. The operation of dy' and $d\delta$ remain the same as above.
$opt = 2$	as for option 1; the parameters keep their definition. This time the entrance is fixed and the displacements dx and dy are imposed at the exit of the magnet. In both cases 1 and 2, a momentum-independent slope is computed and imposed on the magnet.

$opt = 3$ in this option, the corrector acts as a pure dipole steering magnet. Parameter 1 is dx' and parameter 2 is dy'. Parameters 3 and 4 are not used. The angle kicks dx' and dy' are inversely proportional to the momentum of the partical (*i.e.*, dx' and dy' are divided by $1 + \delta$, where δ is the relative momentum of the particle traced).

SET ERRORS OF ELEMENTS

This operation specifies the random generation mode, the elements that should actually be affected by the errors, and the location intervals in the beam line where they lie.

Input format:

SETErrors... (*maximum* 80 *characters*)

nopt

nerr

name nint nb_1 $nf_1 \ldots$ nb_{nint} nf_{nint},

...

name nint nb_1 $nf_1 \ldots$ nb_{nint} nf_{nint},

Parameters:

The meaning of the parameters are the same as those of the SET MIS... operation. See next operation.

SET MISALIGNMENT OF ELEMENTS

Input format:

SETMisalignment... (*maximum* 80 *characters*)

nopt

nmis

name nint nb_1 $nf_1 \ldots$ nb_{nint} nf_{nint},

...

name nint nb_1 $nf_1 \ldots$ nb_{nint} nf_{nint},

This operation defines the random generation mode, the elements that should be actually misaligned, and the location intervals in the beam line where they must be misaligned.

Parameters:

nopt choice option for the random generators.

0 the elements are misaligned by the fixed values given in the MISA... operation. No randomness is introduced.

1 the misalignment values are obtained by multiplying the values given in the MISA... operation by +1 or −1 randomly generated.

2 a uniform distribution is generated having the rms values defined by the MISA... operation.

3 a Gaussian distribution truncated above two standard deviations is generated with the rms value defined by the MISA... operation.

4 a Gaussian distribution truncated above six standard deviations is generated with the rms value defined by the MISA... operation.

11, 12, 13, 14 the random error is the same as for 1, 2, 3 or 4 with a fast random generation of the random sequence. This random could, in some cases, be affected by unwanted correlations. In case of doubt, check the STATISTICAL validity of your results with a family of runs using the options 1, 2, 3 or 4.

nmis number of misaligned element type (names).

name name of the family of misaligned element.

nint	number of intervals in which element is misaligned.
0	all elements with that name are misaligned. In this case, no interval range is given.
−1	no elements with the name are misaligned. In this case, no interval range is given.
nb, *nf*	beginning and end of range of misaligned elements. These numbers correspond to the order number in the machine list.

SHO CORRECTORS

This operation displays the values of the correctors and gives an elementary analysis of their values.

Input format:

SHO Correctors

option ...

Parameters:

option determines the information to be printed out.

1 the rms, maxima and minima of the values are displayed.

2 *name*. This option prints the values 1 and 2 of the correctors with the label *name*. These values are multiplied by the scale factor $sigfac$ as defined in Constant definition.

3 under this option, the values of the correctors are printed out in the format accepted as input by the SET Correctors operation. This can be used to set up an input file for a misaligned and corrected machine which can be then studied without executing the alignement correction procedures.

SHO ERRORS (not implemented yet)

SHO MISALIGNMENT

This operation displays the actual values of the displacement generated in previous operations and provides manipulation of the misalignment features. See **demo6** for examples.

Input format:

SHO Misalignment

nrange...

Parameters:

nrange

0 then all misalignments are printed.

> 0 then *nrange* intervals *ni*, *mi* are used. The misalignments are printed in these intervals.

-1 then the operation sets up arrays containing all the misalignment data. Tracking execution proceeds faster.

-2

$name_1\ name_{1j1}\ name_{1k1} \ldots\ name_{1j5}\ name_{1k5}$,

$name_2\ name_{2j1}\ name_{2k1} \ldots\ name_{2j5}\ name_{2k5}$,

...

99,

$misfac$

Up to five names $name_i$ can be given and up to 5 intervals $name_{ij}$, $name_{ik}$. The names defining the intervals must be unique (use markers). The operation adds to the elements $name_i$ a random misalignment as defined in the misalignment data multiplied by the factor $misfac$ but using a different random sequence from that used in the main misalignment procedure. This operation must be preceded by a SHO MIS operation with option -1 to be successful.

-10 *name* this operation provides information of the average lateral displacement of the element labeled *name*, using the scale factor $sigfac$.

$nrange$	number of intervals in which the printout will occur. If $nrange = -1$, the operation sets up arrays containing all the misalignment data. Tracking execution will then proceed faster (more space is needed).
n_i, m_i	beginning and end of interval in which printout occurs.

SYNCHROTRON RADIATION DATA DEFINITION

This operation computes the energy losses due to synchrotron radiation in a deterministic way. It only affects operation using particle tracing.

Input format:

SYNChrotron radiation... (*maximum* 80 *characters*)

Energy option randomoption,

Parameters:

Energy	initial nominal energy in GeV.
option	
0	no synchrotron radiation effect is simulated.
1	synchrotron radiation loss is computed in every magnet particles lose that energy at the entrance and exit of the magnet.
2	synchrotron emittance growth is simulated randomly for each particle. This growth is due to the spread in the energy loss of the particles.
3	both the radiation loss and the emitttance growth are simulated.
randomoption	choice of the random generator; applies to above options 2 and 3 only.
1	the random generator is binary + and − randomly affecting the energy spread creating the emittance growth.
2	the random generator is uniform.
3	the random generator is Gaussian: Note that here the execution time will be considerably greater than with choice 2 for the options 2 and 3 (above) which enable the emittance growth calculations.
11, 12, 13	the fast random generator is used to produce the sequence of random numbers. This sequence may be affected by some unwanted correlations. In case of doubt, use the options 1, 2 or 3.

TABLE OF ERRORS FOR DIMAD

No.	Message	Explanation and Action
0	ARR MXLVAR MXLCOND	Too many variables or conditions requested in a fitting procedure. Reduce the number requested or increase the value of the corresponding parameter everywhere the parameter definition occurs.
1	ARRSIZE MAXCOR	Too many correctors requested. Reduce the number requested or change the value of parameter MAXCOR everywhere it is defined.
2	ARRSIZE MXLIST	Too many interval points requested for printing. Reduce number requested or change parameter MXLIST value everywhere it is defined.
3	ARRSIZE MAXMAT	Too many matrices requested. Reduce number requested or change parameter MAXMAT everywhere it is defined.
4	ARRSIZE MAXPOS	Too many elements in beam line. Reduce size of beam line or increase parameter MAXPOS everywhere it is defined.
5	ARRSIZE MAXELM	Too many distinct elements. Reduce number requested or change parameter MAXELM everywhere it is defined.
6	ARRSIZE MAXDAT	Too many elements in data. Reduce number requested or change parameter MAXDAT everywhere it is defined.
7	ARRSIZE MXELMD	Too many elements in DIMAD data. Reduce number requested or change parameter MXELMD everywhere it is defined.
9	ARRSIZE MAXPAR	Too many parameters in MAD input section. Reduce input deck size or change parameter MAXPAR everywhere it is defined.
10	ARRSIZE MAXLST	Too many elements in beam line in MAD input section. Reduce number requested or increase parameter MAXLST everywhere it is defined.
11	ARRSIZE SEQUENCE	Increase parameter MAXPOS everywhere it is defined.
12	ARRSIZE HARMON	Not applicable in this version.
13	ARRSIZE ADIAPAR	Too many adiabatic parameters. Not parameterized. Change number allowed: 5 everywhere NEEDED.
14	ARR SHOMIS INTVAL #	Too many intervals for listing of misalignment. Not parameterized. 5 maximum.

15	ARR SHOMIS ELM #	Too many distinct elements for misalignment printing. Not parameterized. 5 maximum.
16	ARRSIZE MXMTR	Too many misaligned elements in beam line. Reduce number requested or increase parameter MXMTR everywhere it is defined.
100	MAD INP MAJOR ERR	Various major MAD format violations. See the echo file (output unit 9) for information.
101	MADIN NORMAL END	Normal exit from program without error.
102	MADIN READ ERROR	Read error encountered. Look in the echo file.
103	MADIN SCAN STOP	Various errors were encountered. The program continued to scan the machine definition data for possible subsequent errors. The program stops before executing any computation.
200	ADIA:OPTION ERR	The adiabatic option number is wrong. Consult the user's guide.
201	ALIG:NOCORR FOUND	The element found at the indicated position is not a corrector. Check position in the machine list.
202	ALIG:MORE MONIT	Not enough monitors were found in the machine list. Either decrease the number of conditions or add some monitors in the machine list.
203	ALIG:NO MON FOUND	The element found at the indicated position is not the selected monitor. Check position in the machine list.
204	ALIG:MON OPT ERR	Wrong error option number for the monitor. Consult the user's guide.
205	ALIG:BAD ELM PAR	Element and parameter not in the element list. Add element or consult the user's guide.
206	ALIG:MORE CORR	Not enough correctors were found in the machine list. Either decrease the number of variables or add some correctors to the machine list.
210	BASE:NO BEGIN KICK	For this operation, the line must begin with a general kick element.
211	BASE:NO END KICK	For this operation, the line must end with a general kick element.
215	CAV:NOT A CAVITY	The element is not a cavity. Modify input. This error is in relation to either the adiabatic variation or the use of *Lcavity*.
220	CHNGCOR:NOT A COR	The element is not a corrector. Modify input.

221	CHNGCOR:NOTSET	This corrector has not been set by a previous SET Corrector operation.
222	CORDAT:NOT EVEN #	The number of data must be even. Correct the input accordingly.
223	SETCOR:NO MATCH	No such element was found in the machine list. Check spelling, or add element in machine list, or suppress element in the present operation.
230	CONDF:# ERROR	Error in number of constant variables. Consult the user's guide.
235	INTER:TERM ID ERR	Terminal identification number is wrong. Consult the user's guide.
236	INTER:NORMAL STOP	Normal termination of the interactive mode.
240	NOT A PARM NAME	This name is not a valid parameter name for the element. Consult the user's guide.
241	USE A GKICK	A general kick element must be used here instead of ***hkick*** or ***vkick***.
245	ELID:NO MATCH	No such element was found. Check the spelling and the element definition list.
246	ELEM NOT IN MACH	No such element was found in the machine list. Check the spelling and the machine list.
250	ERR:RAND OPT ERR	Error in the random option number. Consult the user's guide.
251	ERR:BAD NUMBER	Too many error values requested.
252	ERR:ELEM NO ERR	This element is not affected by errors. Check spelling and error definition list.
255	FIT:NO 2 FITPT	For this fitting condition, 2 fit points are required. Correct the machine list accordingly, or change the condition.
256	FIT:NO FITPT MATCH	No such name exists in the machine list. Check spelling or add name to machine list.
257	FIT:WRONG ELEM PARM	This parameter name is not valid for this element. Consult the user's guide.
258	FIT:UNSTABLE MOVMT	An unstable movement was encountered during the fitting procedure. Change the starting point, or use a beam line fitting procedure instead of a matrix movement analysis fitting procedure.

259	FIT:WRONG ELEM	Element is not used in the fit base list.
260	GENER:NO BEAM DEF	No beam definition operation preceded this particle generation operation. Introduce one.
261	GENER:BAD OPT #	Wrong particle generation option number. Consult the user's guide.
262	GENER:DIAGON FAIL	The beam matrix diagonalization failed. The previously defined beam has a bad aspect ratio at the beginning of this beam line. Choose another starting point to improve the aspect ratio, or check the beam definition.
265	ELEM:NOT EVEN #	An even number of numerical data entries is required. Check the input data.
266	ELEM:BAD INDEX	Bad index in defining general matrix. Consult the user's guide.
270	INPUT:BAD OUPUT OPT	A wrong option for the output control was chosen. Consult the user's guide.
271	INPUT:MORE INP #	Not enough numbers were given to fill the data array. Check the input and consult the user's guide.
272	INPUT:BAD FORMAT	Bad format was encountered reading a number. Check the input format. This error can also occur when data are missing. Consult the user's guide section for the item in which this error occurred.
273	LESS INPUT #	Too many numbers were given for the data array. Check the input and consult the user's guide.
275	MIS:BAD RAN OPT	Wrong option number for the random generation. Consult the user's guide.
276	MIS:BAD MIS OPT	Wrong misalignment option number. Consult the user's guide.
277	MIS:REPORT ERROR	When this error is encountered, please report it to one of the authors of the program.
278	MIS:ELEM NOT MISAL	This element is not affected by misalignment errors. Check spelling or add element to the misalignment data definition. Consult the user's guide.
280	OPER:BAD OPER NAME	This operation does not exist. Check spelling. This error can occur if a comma is used instead of a semicolon when exchanging beam line uses with USE, and NEWBEAM commands.

281	PARTAN:DIAGON FAIL	Matrix diagonalization failed in the particle distribution analysis. The beam shape is either a very elongated ellipsoid, or very distorted. In the first case, changing the point of observation may help cure the problem. In the second case, obvious by looking at the plots, the beam cannot be represented by an ellipsoidal distribution.
282	PLOT:NGRAPH ERR	Error in NGRAPH number. Consult the user's guide.
285	RAND:BAD OPT #	Bad option number for the random generation. Consult the user's guide.
286	RAND:BAD SIGMA #	In some options of the random generation, only a few sigma numbers are allowed. Consult the user's guide.
288	RMAT:PART LOST	All particles were lost during the tracking computations for the Rmatrix.
290	SHOCOR:BAD OPT #	Wrong option number in the SHO Corrector operation. Consult user's guide.
300	DETAIL:PART LOST	All particles were lost during the tracking computations for the Detail operation.
301	DETAIL:NO BEAM DEF	No beam definition was done before the current Detail operation. Introduce one.
302	TRACK:PART LOST	All particles were lost during tracking.
304	MOVMT:PART LOST	All particles were lost during tracking computations for the Movement Analysis operation.
305	MOVMT:BAD DETERM	A bad value for the determinant of the transfer matrix (very different from 1). This may be corrected by increasing the number of iterations, choosing a better starting point, or reducing the $dx\ dx'\ dy\ dy'$ used initially in the computation.
310	LMDIF:INFORM	Bad input conditions were met in the minimizer LMDIF. Follow the instructions of the general message. If the message is that there are less conditions than variables, and no input error exists, then the minimizer requirement can be met by duplicating one or more conditions.
320	SIMEQ:SING SET OF EQ	A singular set of linear equations was encountered while computing a matrix movement analysis. Generally, this means an error in setting up the beam line.

REFERENCES

(1) K. L. Brown, D. C. Carey, Ch. Iselin, and F. Rothacker, *TRANSPORT, A Computer Program for Designing Charged Particle Beam Transport Systems,* SLAC 91 (1973 Rev.), NAL 91 and CERN 80–04.

(2) D. C. Carey and F. C. Iselin, "A Standard Input Language for Particle Beam and Accelerator Computer Programs," *Proceedings of the 1984 Summer Study on the Design and Utilization of the Superconducting Super Collider,* Snowmass, Colorado, June 1984.

D. Douglas, L. Healy, F. C. Iselin, and R. Ryne, "Report of the Group on a Common Input Format," *SSC Aperture Workshop Summary,* SSC–TR–2001, Appendix 7, November 1984.

(3) F. C. Iselin, *The MAD Program Reference Manual,* CERN, LEP Division, November 1, 1984.

(4) D. Douglas, E. Forest, and R. Servranckx, "A Method to Render Second-Order Beam Optics Programs Symplectic," LBL Note SSC 28 LBL–18528. Also in the *Proceedings of the 1985 Particle Accelerator Conference,* Vancouver.

(5) K. L. Brown and R. V. Servranckx, "First- and Second-Order Charged Particle Optics," SLAC–PUB–3381 (July 1984), Stanford Linear Accelerator Center.