1 Introduction

The present guide corresponds to the program version dated JUNE 01 1993.

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
Roger Servranckx at Gabriola 604/247-9567

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). It is for these reasons that he has become a coauthor of the present manual.

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] [3]. Adapting the input code of MAD [4], 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[5], David Douglas introduced the symplectic tracking options and the canonical variables.

The following persons, listed in alphabetical order, were actively involved in the development of DIMAD :

Karl.L.BROWN (Stanford Linear Accelerator Center)
David DOUGLAS (Continuous Electron Beam Facility)
Lindsay SCHACHINGER (Superconducting Super Collider)
Roger V. SERVRANCKX (University of Saskatchewan)
Uli WIENANDS (TRIUMF Vancouver)

The affiliation listed is the one when most of the contribution occured.

2 Element and Machine Data Input

The input format to dimad now conforms quite closely to the standard format, as laid out in [2][3]

This conversion of dimad to standard input was accomplished by taking the input subroutines from the program MAD [4] 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 [2] [3] but which are not implemented in dimad are separator and rbend.

The job title entered on a line following one with the keyword "title."

This can be followed with a units keyword. If no units keyword is found, the units are assumed to be the standard units.

2.1 Units and field expansion

The keyword "utransport" indicates that the input has the following units

angles in degrees except for $\Delta x'$ and $\Delta y'$ for the kicks
lengths in meters
energy in GeV

momentum in GeV/c
electromotive force in kiloVolts
frequency in Hz
field expansion is $B(x,0) = B\rho\ \Sigma_n\ K_n x^n$
positive K_1 is horizontally focussing

Also, the "utransport" keyword has implications for the field expansion coefficients.

The keyword ustandard indicates that the input has the following units

angles in radians
lengths in meters
energy in GeV
momentum in GeV/c
electromotive force in MegaVolts
frequency in MegaHertz
field expansion is $B(x,0) = B\rho\ \Sigma_n\ K_n \frac{x^n}{n!}$
positive K_1 is horizontally focussing

3 General Syntax

When describing the machine, a statement can be continued on the next line by ending the current line with a "&".

A comment line begins with one of the characters "!", "*","(","@" in the first column.

A ";" is used to separate statements on the same line. Keywords are uniquely specified by the first five letters, and only those five must be entered. At most 8 letters in a keyword are checked.

The keyword NOECHO can be used to suppress transmission of the input data stream to the output files. The keyword ECHO reinstates the stream of the input data to the output files.

3.1 Parameters

Parameters are defined with a statement like

```
pname = value    or    pname := value
```

where pname is any parameter name. Parameters can then be used in element definitions. Value can also 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 π if needed must be defined as an input parameter (say as PI = π).

The value halfturn must be understood as either π radians or as 180 degrees depending on the units chosen.

4 Element Definitions

To define an element,

```
label: type [,pkeyw=value,......]
```

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). value can be a parameter name, or an expression involving parameters.

Examples:

b: sbend, l=lb, angle=lb/rho
d0: drift

4.1 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 $n = -\frac{\rho}{B_0}\frac{\partial B_0}{\partial x}$
e1	is the entrance edge angle.
e2	is the exit edge angle.
tilt	is the tilt angle. If tilt is entered with no value, halfturn/2 is assumed.

	k2	If the standard convention is being used, k2 is given by the field expansion in the UNITS section. If transport convention is being used, k2 is $\beta = \frac{\rho^2}{2B_0}\frac{\partial^2 B_0}{\partial x^2}$
	h1	is the entrance pole face curvature.
	h2	is the exit pole face curvature.
	hgap	is the entrance 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.
	fint	is the entrance fringe field integral, which defaults to 0.5. 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. hgapx is the exit half gap size. See hgap. fintx is the exit fringe field integral. See fint.

rbend The rbend is a parallel faced dipole magnet. Its parameters are the same as those of the sbend. Parameters e1 and e2 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.
k1	is the strength.
tilt	is the tilt angle. If tilt is entered with no value, halfturn/4 is assumed.
aperture	is the magnet aperture for the Hardware operation

sextupole

l	is the length.
k2	is the strength.
tilt	is the tilt angle. If tilt is entered with no value, halfturn/6 is assumed.
aperture	is the magnet aperture for the Hardware operation

quadsext

l	is the length.
k1	is the quadrupole strength.
k2	is the sextupole strength.
tilt	is the tilt angle. If tilt is entered with no value, halfturn/4 is assumed.
aperture	is the magnet aperture for the Hardware operation

octupole

l	is the strength.

	k3	is the strength.
	tilt	is the tilt angle. If tilt is entered with no value, halfturn/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.
	k0 - k20	are the strengths.
	t0 - t20	are the tilt angles. If tn is entered without a value, halfturn/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.

note1 : only the components with non zero 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.

note2 : 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*Bs/Brho.
	k1	is the quadrupole strength.
	tilt	is the tilt angle. If tilt is entered with no value, halfturn/4 is assumed.
	aperture	is the magnet aperture for the Hardware operation.

quadac (linac cavity containing a quadrupole element)

	l	is the length.
	k1	is the quadrupole strength
	deltae	is the energy increment per unit length in GeV/m given by the cavity to a particle riding on the crest of the wave.
	phi0	is the phase of the reference particle with respect to the RF wave.
	freq	is the frequency of the RF in MHz.
	e0	is the energy in GeV of the reference particle

	tilt	is the tilt angle. If tilt is entered with no value, halfturn/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) In program versions dated July 14 1989 or later the frequency is defined via the harmonic number. This has the advantage to match the frequency with full accuracy to the length of the machine. When the frequency must be not matched a non integer value for the harmonic number is entered. Note that the keyword has remained FREQ (until further notice). The particle type (electrons or protons) is selected via the constant definition operation.
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,

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.
	angle	is the angle through which the coordinates are rotated about the longitudinal axis.
	dz	is the longitudinal displacement.
	v	is the extrance-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.
hmon, vmon, monitor		These elements are horizontal, vertical, and horizontal and vertical monitors, respectively.
	l	is the monitor length.
	xserr,	not used presently
	yserr,	not used presently
	xrerr,	not used presently
	yrerr	not used presently
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	x collimator aperture
	ysize	y collimator aperture 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.
p1 - p20	are the parameters.

mtwiss

l	is the length.
mux,	x phase advance of twiss matrix
betax,	x beta value of twiss matrix
alphax,	x alpha value of twiss matrix
muy,	x phase advance of twiss matrix
betay,	y beta value of twiss matrix
alphay	y alpha value of twiss matrix betax and betay have default values of 1.

matrix This element is a general transfer matrix.

rij	first order ij element of matrix
tijk	second order ijk element of matrix, i,j and k range from 1 to 6, but j is always less than or equal to k.

5 Beamline definitions

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

```
label: line=(member1, member2, member3,.....)
```

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:

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.

6 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 without 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.

7 Translation to COSY input

The keyword COSY can be used after a beamline has been selected in the MAD input stream with the call

use, beamline name

and before the keyword dimat or newbeamline.

The keyword COSY will prompt the program to provide an input stream for the chosen beamline in a format compatible with the COSY program [7]

Not all elements are currently translated. When an element is not translated it is replaced by a drift.

The same keyword can also be used in the dimad operation input stream See the appropriate listing in the following section.

8 Operation list description

Each array specifying an operation starts with a title line of 80 characters or less. The first four non blank 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 "!","*","(","@" 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.

8.1 Implemented operations

9 Use and description of each operation

9.1 Adiabatic variation

This operation enables 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 in the same job.Three options are available : linear, sinusoidal and piecewize linear 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,
if nopt=3 then after nopt enter the following :
npts n_i,val_i for i=1 to npts

Parameter definitions :

name	name of element having a parameter to be varied
pkeyw	keyword of parameter to be varied (i.e. k1 for a quad strength)
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's 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 :

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

where turn is the current turn at which value is applied. Outside the turn interval p_1 and p_2 the original value is applied.
3 means the variation will be piecewise linear with npts nodes including the extremities. At turn n_1 value val_1 is set, at turn n_2 val_2 and so on up to turn n_{npts} when val_{npts} is set and maintained until the end of the tracking.

Examples

The following are two examples extracted from demo 13. The first varies the frequency of the cavities named cav and cav1 from turn 100 to turn 200 in a linear fashion between the harmonic number 336 and 336.01. The second varies the voltage of cavity cav. Refer to demo 13 to see what is done with this set up.

```
adiabatic variation of RF frequency
cav freq 1 100 200 336 336.01
cav2 freq 1 100 200 336 336.01
99,


adiabatic variation of RF voltage
cav volt 1 10 50 25 70
99,
```

9.2 Beam matrix tracing

Computes beam matrices at selected points of the machine from the initial beam matrix defined in the input of the operation. The equation of the six dimensional ellipsoid enclosing the beam is $X^t \Sigma^{-1} X = 1$

Input format

BEAM matrix tracing.... (up to 80 characters)

followed by :

$$\begin{array}{cccccc} \sigma_x & r_{xp_x} & r_{xy} & r_{xp_y} & r_{xl} & r_{x\delta} \\ & \sigma_{p_x} & r_{p_xy} & r_{p_xp_y} & r_{p_xl} & r_{p_x\delta} \\ & & \sigma_y & r_{yp_y} & r_{yl} & r_{y\delta} \\ & & & \sigma_{p_y} & r_{p_yl} & r_{p_y\delta} \\ & & & & \sigma_l & r_{l\delta} \\ & & & & & \sigma_\delta \end{array}$$

$$mprint \quad [mlist]$$

or :

$$\begin{array}{ccccc} 0 & & & & \\ \beta_x & \alpha_x & \eta_x & \eta_{p_x} & \epsilon_x \\ \beta_y & \alpha_y & \eta_y & \eta_{p_y} & \epsilon_y \\ \sigma_l & \sigma_\delta & & & \end{array}$$

$$mprint \quad [mlist]$$

or :

$$\begin{array}{ccccc} 0 & & & & \\ 0 & 0 & 0 & 0 & \epsilon_x \\ 0 & 0 & 0 & 0 & \epsilon_y \\ \sigma_l & \sigma_\delta & & & \end{array}$$

$$mprint \quad [mlist]$$

Parameter definitions

The relationship between the Σ beam matrix and the input coefficients is as follows :

Let Σ_{ij}, with i and j taking the values x, p_x, y, p_y, l and δ, be the elements of the symmetric positive definite matrix Σ. Then the following formulae describe the relationship between Σ and the input parameters :

$$\begin{array}{rcll} \sigma_i & = & \sqrt{\Sigma_{ii}} & = \text{ value of maximum extent of beam} \\ r_{ij} & = & \frac{\Sigma_{ij}}{\sigma_i \sigma_j} & = \text{ correlation factor} \end{array}$$

$\beta\alpha$ Twiss parameter values defining a beam ellipsoid

η η -function values to define momentum dependent terms in the beam ellipsoid

ϵ_x and ϵ_y are the horizontal and vertical emittances. In this case, the beam is assumed uncoupled .

NOTE :when using the third input option (β_x is 0) then the twiss parameter values are obtained from a previously run movement analysis calculation done with a matrix operation

mprint

-2 no computation is done. The operation serves only to define a beam as needed in the operations BEAM tracing and DETAiled analysis with parameter nvh=1.

-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.

mlist 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 pairs of numbers.

Units

$\sigma_x,\ \sigma_y,\ \sigma_l,\ \beta_x,\ \beta_y,\ \eta_x,\ \eta_y$ are measured in meters

$\sigma_{p_x},\ \sigma_{p_y},\ \eta_{p_x},\ \eta_{p_y}$ are measured in radians.

$\epsilon_x,\ \epsilon_y$ are measured in m-radians.

σ_δ unit is one.

The correlation coefficients r_{ij} are dimensionless and vary from -1 to $+1$

Examples

Three examples are given. The first is extracted from demo 7 , the second and third are extracted from demo 2.

The first shows the definition of the beam using the extension σ 's.

The second defines the transverse part of the beam by using the twiss parameters and the emittances.

The third shows a definition used in conjunction with a previous matrix analysis. It uses the twiss parameters obtained in that analysis and the value

of the tranverse emittances.

```
BEAM MATRIX TRACING
0.00012157290 0 0 0 0 0
  0.00000246750 0 0 0 0
    0.0000826254 0 0 0
       0.0000036309 0 0
                    0.002 0
                      0.005
-1,

   beam definition
   0
   1.0 0 0 0 1.0e-06
   2.0 0 0 0 1.0e-06
   0.001 0.001
   0,

MATRIX : FIRST ORDER CELL MATRIX.
1 -1,
BEAM MATRIX
0
0 0 0 0 1.0E-06
0 0 0 0 1.0E-06
0.02 0.001
-1;
```

9.3 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 : Pi, the velocity of light (in m/sec), the electron mass (or particle mass) (in GeV), the electron (or particle) radius, the electron (or particle) charge. The reference relative momentum (dp/p) that is used in some Taylor expansion with delta 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. The particle type : 0 for electrons 1 for protons . Use the operation SHO Constant to examine the constants.

Input format:

CONStant definition .....(up to 80 Characters)

followed by :

n_1 , val_1 $\cdots$ n_p , val_p

Parameter definitions

n_i is the order number of the ith constant to be redefined according to the following order :

1	pi
2	velicity of light
3	particle mass
4	particle radius
5	particle charge
6	reference relative momentum
7	tolerance parameter in LSQ fit
8	maximum factor in LSQ
9	expulsion factor for particles
10	scale factor for eta function
11	scale factor for sigma values
12	particle type(0:electron, 1:proton)

val_i new value of the constant n_i.

Examples

The example given is extracted from demo9 and shows the change of the current particle type to a proton and sets the rest mass to the value of the rest mass of a proton.

```
CONSTANT DEF
12 1 3 0.938259,
```

9.4 COSY translation

This operation will translate a beamline, with its field and alignment errors and possible corrections into a COSY [7] input compatible format.

It is also possible to concatenate parts of the beamline lying between individual elements defined in a list. This allows for faster fitting operations in COSY.

Since the COSY program operates in canonical variables it is ESSENTIAL that the symplectic option with output in canonical variables be turned on.

Input format

COSY translation ..... (up to 80 characters)
nopt
$name_1 ... name_n$ end

Parameter definitions

nopt

0 no concatenated sections of the beam line will be created
1 concatenated sections of the beam line will be created

namei names of the elements which separate the different concatenated sections of the beamline. Present only if the parameter nopt is equal to 1

Examples

```
cosy
0;
stop

cosy
1
mag1 mag2 end;
stop
```

9.5 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 $v_0 = (x_0\ x'_0\ y_0\ y'_0)$.

NOTE: no kicks simulating synchrotron oscillation (parameter $T < 0$) may exist in the lattice. Results are meaningless in the presence of such kicks. For each chosen momentum, particles are generated around the point v_0 to compute the elements R_{ij} of the matrix describing the linear motion around the trajectory defined by the point v_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
δ_1 δ_2 ... δ_{nener}
mlocat [list]

Parameter definitions

nh

1 only x x' motion is traced and computed. β_x α_x and ν_x are computed.
0 x x' motion alone is not analyzed.

nv

1 only y y' motion is traced and computed. β_y α_y and ν_y are computed.
0 y y' motion alone is not analyzed.

nhv

1 coupled motion x x', y y' is traced. The full beam matrix is computed.

2 coupled motion is computed. A short print for x x' y y' β_x α_x β_y α_y ν_x ν_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 1.0e03(say)) 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 sigmas (not the correlation coefficients) are affected by the scale factor SIGFAC which can be set via the constant defintion operation.

5 same as three with the code added in the first column to facilitate some plotting work

0 coupled x x', y y' motion is not analyzed. When NH, NV, and NHV are all zero, the program prints the centroid positions only.

NOTE: the input beam must have been defined previously in a BEAM MATRIX TRACKING operation.

dx dx' dy dy'

increments at which the off orbit particles are placed to compute the sx, cx, sy, cy functions (see reference [1]).

β_x α_x η_x ${\eta'}_x$ β_y α_y η_y ${\eta'}_y$

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 (max 6).

$\delta_1 \ldots \delta_{nener}$

momentum values

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

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 pairs of numbers.

Examples

The first example, taken from demo7, produces a multiple momentum analysis of the beam centroid and beam matrix at the end of the lattice.

The second example, taken from demo9, gives a "short" printout of the beam centroid and the twiss parameters at every point of the lattice, for the on momentum motion.

```
DETAILED CHROMATIC ANALYSIS
1 1 1
0 0 0 0
0.000001 0.000001 0.000001 0.000001
49.26950355 0 0 0 22.75617615 0 0 0
7 5
0.0 -.001 0.001 -.003 0.003 -.005 0.005
-1,

detailed
1 1 2
0 0 0 0
1.0e-06 1.0e-06 1.0e-06 1.0e-06
1.76806 -0.0032 0 0
15.0736 -0.0034 0 0
1 0
0.00
0;
stop
```

9.6 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
$x_0, x'_0, y_0, y'_0, al_0, del_0$

Parameter definitions

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 sigmai 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.

σ_i

the number of sigmas above which the gaussian distribution is truncated for each of the six variables x,x',y,y',al,delta. The beam is defined by a previous BEAM operation which is assumed to define the one sigma distribution.

scale

scales the beam size by the given factor.

npart

number of particles to be generated. Maximum number mxpart (initially set at 1000 or 10000 depending on versions)

$x_0, x_0', y_0, y_0', al_0, del_0$
centroid coordinates around which the beam is generated.

Examples

The example shown is taken from dem07 . It will generate 500 particles inside the ellipsoid defined by a previous beam matrix definition using a gaussian distribution truncated at 6 sigmas for the transverse coordinates and 1 sigma for the longitudinal coordinates.

```
GENERATION OF PARTICLES
3 6 6 6 6 1 1 1.0 500
0 0 0 0 0 0,
```

9.7 Geometric aberrations in multiturn operation

This operation traces particles that are placed on ellipses with nominal emittances ϵ_{x_i}, ϵ_{y_i} 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, $\epsilon_x, and\ \epsilon_y$. It also computes the maxima and minima

emittances which informs about the diffusion pattern of the motion. Variances of the tunes are also computed.

Input format

GEOMetric aberrations .....(up to 80 characters)
$\beta_x, \alpha_x, \beta_y, \alpha_y$
$x_{co}, x'_{co}, y_{co}, y'_{co}$,ener
ncase,nturn,njob
nplot,nprint
$\epsilon_{x1}, \epsilon_{y1} \ldots \epsilon_{xncase}, \epsilon_{y_{ncase}}$
anplprt

Parameter definitions

$\beta_x, \alpha_x, \beta_y, \alpha_y$

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 0. The values corresponding to the first energy are used. This includes the parameters x_{co} to ener.

$x_{co}, x'_{co}, y_{co}, y'_{co}$, ener

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

ncase number of cases analyzed (maximum mxgaca)
nturn number of turns for tracing (maximum 1000)

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 = nturn.

ϵ_x, ϵ_y

ncase values for the chosen nominal emittances in x and y using the unit mm-mrad (E-06 m-rad)

anplprt

parameter selecting the fast fourier transform options. When 0 no fourier transform is performed. 1 the fourier transform components are printed. 10 the amplitude of the fourier transform is printer-plotted. 100 an analysis of the peaks is provided. A combination of those values is allowed eg: 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 analysed.

Examples

The first example comes from demo5 and provides an FFT analysis with the plotting of the FFT spectrum but no printout of the spectrum.

The second example, taken from demo12, illustrates how the geometric aberration operation uses the twiss parameters computed in a previous movement analysis operation.

```
GEOMETRIC ABERRATIONS
2 0 .11 0
0 0 0 0
0
2
1000
1
1 -1
4.5 2.25
9 4.5
110;
stop
* The following is to illustrate the use of rmat and geometric
```

```
* aberrations in conjunction with movement analysis
movement analysis
1 1 1 -3 1 0 0.00001
0 0 0 0 0 0.002
0,
geometric aberration
0 0 0 0
0 0 0 0 0
1 100 1
1 -2
10 10,
```

9.8 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

Parameter definition

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

Examples

The example is taken from demo1. It produces a listing of the hardware information about the elements that are defined in the input section of the data for the energy 1GeV.

```
hardware
1.0,
```

9.9 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 developped and debugged!

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

Various versions of this operations exist. Interested users should contact David Douglas or Matt Bickley at CEBAF.

Input format

INTEractive control of ..(up to 80 char)
niopt nivar
name keyword (repeated nivar times)

Parameter definitions

niopt	option parameter not used now .
nivar	number of parameters to be varied (maximum 8)

9.10 Layout printing of machine

Computes the geometry of the lattice

Input format

LAYOut printing of machine..(up to 80 char)
s x y z θ ϕ ψ
conv mprint [list]

Parameter definitions

s x y z θ ϕ ψ

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 $\theta, \phi,$ andψ, the axis z 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(with a positive bend angle and positive radius). 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)

Examples

The example taken from demo1 will give the coordinates of the endpoint of the lattice and the angles defined by the tangent to the reference trajectory within an absolute reference system. The units used are meters.

```
layout
0 0 0 0 0 0 0 1.0 -1,
```

9.11 Least square fit

This operation handles most fitting problems. 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 overflow condition is developed in the subroutine LMDIF, the solution sought is probably not a practical one. The 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 del(i) have no influence. We have kept them to avoid changes in the input format.

Input format

LEASt square fit of .....(up to 80 char)
nstep nit nvar ncond
$\beta_x, \alpha_x, \eta_x, \eta'_x, \beta_y, \alpha_y, \eta_y, \eta'_y$
name_i pkeyw_i del_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

Parameter definitions

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).

ncond number of conditions to be met(max:mxlcnd). Note that nvar and ncond must satisfy the condition

$$nvar <= ncond$$

$\beta_x \ldots \eta'_y$

initial values needed for the function computation.When betax value is entered as zero, then the program uses the betax...etapy 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

del_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 values obtained from the stable motion analysis of the total matrix. They are used in the following order : compf (compaction factor), nux, etax, etapx, alphax, betax, dmux/ddelta, chromx (chrmoaticity in x), dalphax/ddelta, dbetax/ddelta, muy, nuy, etay, etapy, alphay, betay, dmuy/ddelta, chromy (chromaticity in y), dalphay/ddelta, dbetay/ddelta.

Numbers 65 to 68 refer to detax/ddelta detapx/ddelta detay/ddelta detapy/ddelta as computed in the stable motion analysis of the total matrix.

Numbers 21 to 30 refer to betax alphax etax etapx nux betay alphay etay etapy nuy 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 betax nuy 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 betax nuy computed at the first and second fit point defined by the preceding SET Fit point operation.(ie.:v2-v1)

Numbers 41 to 61 refer to the beam values sigx ... sigp and the rij 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.(ie.:v2-v1)

Numbers 93 to 98 fit the average chromatic errors for betax, alphax, betay, alphay nux, nuy 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(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 parameter to which the associated parameters are connected. It must be present in the list of basic parameters.

npas
: number of parameters to be associated to name1 (max:6).

name$_k$

name of one element having a parameter associated to name1.

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$_k$.

Examples

The first two examples come from demo1. The second example is given to illustrate the use of associated parameters. Refer to demo1 for the meaning of the element names.

The third example comes from demo4. It illustrates the use of associated parameters and shows how beam matrix elements can be fitted.

```
least square FIT OF QUADRUPOLE STRENGTH FOR A 90 DEGREE CELL.
1 1 2 2
* since this a cell fit the twiss values are irrelevant but
* values must be put in (beta =/ 0)
1.0 0 0 0
1.0 0 0 0
QFONE K1 .001
QD1 K1 .001
2 .25 1.0 12 .25 1.0
0;

least square FIT OF QUADRUPOLE STRENGTH FOR A 90 DEGREE CELL.
1 1 2 2
1.0 0 0 0
1.0 0 0 0
QFONE K1 .001
QD1 K1 .001
2 .25 1.0 12 .25 1.0
1
QFONE  1
QFTWO K1 1.0,
```

```
LEAST SQUARE FIT FOR COMPENSATION OF SOLENOIDAL FIELD
1 2 8 12
0.53 0 0 0 0.04 0 0 0
Q1 K1 .001 Q2D K1 .001 Q3D K1 .001
QSKEW1 K1 .001 QSKEW2 K1 .001
QSKEW3 K1 .001 KQ5E ANGLE .001 EFSA L .0001
21 8.975 1 22 1.3341 1 26 20.709 1 27 1.2077 1
130 0 1 140 0 1 230 0 1 240 0 1
310 0 1 320 0 1 410 0 1 420 0 1
2
EFSA 3
E41X L -1
E42E L 1
FSB L -1
KQ5E 1
KQ4X ANGLE -1,
```

9.12 Line geometric aberrations :one turn computation

This operation traces npart particles, placed on ellipses with nominal emittances $\epsilon_{xi}, \epsilon_{xi}$ for one turn. It then fits an ellipse to the output points obtained. From this fitted ellipse it determines the average values for $\beta_x,\ \alpha_x,\ \beta_y,\ \alpha_y,\ \nu_x,\ \nu_y,\ \epsilon_x$, and ϵ_y.

Input format

LINEgeometric aberrations....(up to 80 characters)
$\beta_x,\ \alpha_x,\ \beta_y,\ \alpha_y$
$x_{co},\ x'_{co},\ y_{co},\ y'_{co}$, ener
ncase,npart,ncoup
nplot,nprint,mlocat,[list]
$\epsilon_{xi},\ \epsilon_{y_i}$ for $i = 1$ to ncase

Parameter definitions

$\beta_x,\ \alpha_x,\ \beta_y,\ \alpha_y$

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

$x_{co},\ x'_{co},\ y_{co},\ y'_{co}$, ener

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 300).

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_{xi}, ϵ_{y_i}

ncase values for the chosen nominal emittances in x and y using the unit mm-mrad (E-06 m-rad)

Examples
This example is fictitious.

```
line geometric aberrations
0.19518E+01 -0.5862785371E-01 0.1816010291E+02  0.3831956152E-01
0 0 0 0 0
3 100 0
1 -1 0 40
38 38
75 75
150 150;
```

9.13 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)
e1 e2 de nlum dnu nint nbunch
β_x α_x η_x η_{p_x}
β_y α_y η_y η_{p_y} mprint [list]

Note: if e1 is zero then nlum is assumed 0 and the input twiss parameters values are those obtained in a previous matrix analysis. If e1 is non zero 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. **Parameter definitions**

e1 start momentum for beam data and luminosity computations.

e2 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

$\beta_x ... \eta_{p_y}$
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)

Examples

The first example, from demo2, shows a general use of this operation.

The second example from demo9 shows how to use this operation when one needs only the list of the twiss parameters around the lattice. Notice that in this case the operation must be preceded by a matrix analysis and that both motions (horizontal and vertical) must be stable.

```
MACHINE AND BEAM PARAMETERS. NO LUMINOSITY COMPUTATIONS.
1 1.2 .2 0 0.025 1 1
21.357376 0 0 0 3.940971 0 0 0 1 1 53,

MACH
0,
```

9.14 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]

Parameter definitions

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 prin
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 th
Gxxxxxxx element.
When norder is -11 or -12 th computation is to order 1 and 2 and the format of the outpu
the standard program output for matrices
When norder is -110 the program prints all the matrices of the beamline in a standar
output format (added Dec 1998)

mprint

-2 no printing of matrix.

-1 print matrix at end of machine only.

0 print matrices at all intermediary points and at final point

n where n> 0, used with list and indicates the number of intervals in which printing is to occur.

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

Examples
All three examples come from demo1.

```
maTRIX COMPUTATION AT END OF CELL: FIRST ORDER COMPUTATION
2 1 4 4,

matrix in input format
-2 1 1 4,

MATRIX COMPUTATION AFTER FIT : SECOND ORDER COMPUTATION.
2 -1;
```

9.15 Modification of element data

This operation 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_i pkeyw_i value_i for $i = 1$ to n.

Parameter definitions

n number of modifications to be made (i.e. number of 'name pkeyw value' entries which follow).

name_i name of the machine element which is to be varied.

pkeyw_i keyword of the parameter in the given element which is to be changed.

value_i value that the parameter will assume.

Examples
The example comes from demo3 .

```
MODIFY SEXTUPOLE STRENGTH TO CHANGE CLOSED ORBIT POSITION.
1
SEXTUPOL K2 5.0,
```

9.16 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 (gkick parameter T < 0) may exist in the lattice. Results are meaningless in the presence of such kicks.

Input Format
MOVEment analysis ..... (up to 80 char)
nprint nturn nanal nit nener ncoef dist
x x' y y' l $\delta_1 \ldots \delta_{\text{nener}}$
naplt delmin delmax dnumin dnumax dbmin dbmax ncol nline

Parameter definitions

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 fixed at 0.01(1momentum 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',l estimate of the coordinates of the closed orbit.

δ_i Momenta(dp/p) for which the analysis is performed. i = 1 to nener

naplt

0 no plot of the Taylor expansion is required

1 a plot is required

delmin delmax min max of dp/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.

Examples

The first example comes from demo2 where it is used to analyse the momentum dependence of the parameters of a stable motion.

The second, taken from demo3, illustrates the study of a third order resonance, and the use of a subsequent tracking operation to obtain the separatrices associated to the third order resonance.

```
MOVEMENT ANALYSIS (CHROMATIC EXPANSION) NO GRAPH
1 1 1 -4 15 6 0.00001
0 0 0 0 0
0.0 -.00001 0.00001 -.00003 0.00003
-.0001 0.0001 -.0003 0.0003 -.001 0.001
-.003 0.003 -.006 0.006
0,

MOVEMENT ANALYSIS TO FIND UNSTABLE CLOSED ORBIT NEAR .03 .002 0 0
1 3 3 -4 1 0 .00001 0.03 0.002 0.0 0.0 0.0 0.00 0,
tracking the 10 particles generated by the movemenet analysis just above
1 -2 0 1600
0
11
-.010 0.010 -.002 0.002
-.0006 0.0006 .0003 0.0003
51 51,
```

9.17 Output control

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

Input format

OUTPut control .....(up to 80 characters)
nopt

Parameter definitions

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 . The fifth coordinate will then be the phase relative to an RF cavity instead of the path length.

Note : not all output is affected by the option 14 in the present version of the program. The print operation cannot appear in the Standard format input section. In this section the command NOECHO can be used to suppress printout and the NOECHO can be reversed by use of the command ECHO.

Examples

All three examples come from demo6 and demo7 where they are use extensively to reduce the amount of output produced by the program.

```
OUTPUT CONTROL
3,

OUTPUT CONTROL
1,

OUTPUT CONTROL
0,
```

9.18 Particle distribution analysis

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

Input format

PARTicle distribution analysis ...(up to 80 characters)
nopt

Parameter definitions

nopt must be a non zero number for the analysis to be printed. Its non zero value is immaterial to the running of this operation.

Examples

The example, taken from demo7 illustrates its use in conjunction with the generation of particles and subsequent tracking.

```
GENERATION OF PARTICLES
3 6 6 6 6 1 1 1.0 500
0 0 0 0 0 0,
TRACKING OF THE ABOVE five hundred PARTICLES
-1 -2 0 1,
PARTICLE ANALYSIS
10;
STOP
```

9.19 Print selection

This operation allows 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 supercede 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(i) as many as needed
99
end,

Parameter definitions

The different keywords are : interval, name, type

Interval allows 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 shoulduse 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 ab will produce printing.

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

Examples
The example comes from demo12.

```
* The following is to illustrate the use of the PRINT
* operation
print
interval
m1 m2
99,
name
qf*
99,
type
bend
99,
end
machine : inthis case the print request of -1 supersedes the
* request found in the previous print command (the same would
* be true if if the request were 0 : print everywhere)
1 1.1 0.1 0 0.025 1 1
0 0 0 0
0 0 0 0
-1,
machine: in this case the printing occurs as requested by the
* previous print command
1 1.1 0.1 0 0.025 1 1
0 0 0 0
```

```
0 0 0 0
1,
```

9.20 Rmatrix computation

Computes in chosen intervals the 6X6 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 behaviour of beamlines affected by errors and misalignements. The program also provides the entrance and exit orbit displacements. They are needed to correctly use the matrix. Using the given emittances ϵ_1 ϵ_2 for the two eigenmodes (when they are stable) the program computes the projected emittances on the x-plane and the y-plane. Given an injected beam by its beta alpha and emittance values, the program computes the coupled envelope and then prints the projected emittances of that envelope.

Input format

RMATrix.....(up to 80 characters)
x_0 x'_0 y_0 y'_0 l_0 δ_0
dx dx' dy dy' dl dδ
norder mprint [nlist]
nmopt
ϵ_1 ϵ_2
β_x $alpha_x$ ϵ_x $beta_y$ $alph_y$ ϵ_y

P**Parameter definitions**

x_0 x'_0 y_0 y'_0 l_0 δ_0

initial coordinates of reference orbit When δ_0 = 1 then the dx dx' coordinates are the coordinates of the closed orbit computed in a previous movement analysis. The values corresponding to the first energy are used.

dy dy' dl dδ

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

-2 no printing of matrix.

-1 print matrix at end of machine only.

0 print matrices at all intermediary points and at final point

n where n> 0, used with list and indicates the number of intervals for which the matrix is computed.

nlist mprint pairs of numbers defining the intervals for which the matrix will be computed.

nmopt option number. when 0 the following data is not needed as the coupled beam emttances are not computed

ϵ_1 ϵ_2 Emittances present in each eigenmode used to compute the projected emittances on both the horizontal and the vertical planes

β_x α_x ϵ_x Parameters defining the horizontal projection of an injected beam

β_y α_y ϵ_y Parameters for the vertical projection of the injected beam

Examples

The first example comes from demo9.

The second, which illustrates its use in conjunction with a movement analysis operation, comes from demo12.

```
rmatrix
0 0 0 0 0 0
1.0e-06 1.0e-06 1.0e-06 1.0e-06 1.0e-06 1.0e-06
1 -1
0,

* The following is to illustrate the use of rmat and geometric
* aberrations in conjunction with movement analysis
movement analysis
1 1 1 -3 1 0 0.00001
0 0 0 0 0 0.002
0,
geometric aberration
0 0 0 0
0 0 0 0 0
1 100 1
1 -2
10 10,
```

```
print
interval
m1 m2
99,
end,
rmat
0 0 0 0 0 0
1.0e-6 1.0e-6 1.0e-6 1.0e-6 1.0e-6 1.0e-6
1 1,
```

9.21 Show values of constants

TS 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.

9.22 Seismic perturbation simulation

This operation sets transverse misalignements 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)
xlambs axseis phixs
ylambs ayseis phiys
beginname endname

Parameter definitions

xlambs,ylambs	Wave length(in m) for x and y waves
axseis,ayseis	Amplitudes of the x and y waves
phixs,phiys	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, markers preferably.

9.23 Set fit points

This operation sets intermediate fit points to be used with the least square fit operation.

Input format

SET Fit point.....(up to 80 characters)
n Position1 [Position2]

Parameter definitions

n Number of fit points defined (maximum 2)

Positioni 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.

Examples

The example comes from demo10. It illustrates the use of this operation in conjunction with the least square fit operation.

```
set fitpoints
2
fit1
fit2
least square fit
1 2 11 11
2.8 0 0 0
0.11 0 0 0
q1h k1 0.0001
q2h k1 0.0001
q3  k1 0.0001
qf1 k1 0.0001 qd1 k1 0.0001
qf2 k1 0.0001 qf3 k1 0.0001
qf  k1 0.0001 qd  k1 0.0001
qs1 k1 0.0001 qs2 k1 0.0001
22 0.0 1.0 24 0.0 1.0 27 0.0 1.0
25 1.773333333 1.0  30 1.5166666666 1.0
1031 0.0 1.0 1032 0.0 1.0 1033 0.0 1.0 1034 0.0 1.0
1036 0.0 1.0 1037 0.0 1.0
0,
```

9.24 Set limits to variables

This operation sets boundaries on the variables used in the least square fitting operation. This is done via a supplementary constraint that uses an internally defined penalty function. When boundaries are in use the achieved value for the

fit function will depend strongly on the distance between the "ideal" minimum and the boundary values when that minimum is outside the boundaries. Choice of weights will change greatly the final values achieved.

Input format

SET Limits on variables .....(up to 80 characters)
name keyword upper lower weight distance
.....
name keyword upper lower weight distance
99,

Parameter definitions

Name Name of the element for which some parameter will be affected by boundaries.
Keyword Keyword of the parameter to be affected by the boundaries.
Upper Lower The upper and lower boundaries affecting the parameter.
Weight The weight that is applied to the boundary constraint.
Distance A distance parameter that serves to indicate some flexibility in the boundary constraint. That distance is progressively reduced as the iterations step increases.

9.25 Set symplectic option on

This operation 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,px,y,py,-tau=-t*c=-al,DE/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 and 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 (eg: in alignment fitting when steering only is involved)

Input format

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

Parameter definitions

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 non symplectic ray trace is done using the canonical matrices.

1 Fast version of ray trace is done with the variables x, x', y, y', l, δ and using the canonical matrices.
2 Fast version of ray trace is done with the variables $x, p_x, y, p_y, -\tau, \Delta E/E$ and using the canonical matrices.
3 Slow version of ray trace is done with the variables x, x', y, y', l, δ and using the canonical matrices.
4 Slow version of ray trace is done with the variables $x, p_x, y, p_y, -\tau, \Delta E/E$ and using the canonical matrices.

Energy Energy of nominal particle (in GeV)

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

Examples

The example comes from demo11. Other examples can be found in demos 2 and 9.

```
set symplectic option: lagrangian variables, fast ray trace
1 6.,
```

9.26 Space charge set up

This operation sets up the parameters to compute the effect of space on particle tracking. Presently only a gaussian distribution is used, though the provision of other distribution is possible. The space charge remains in effect for all the operations that use tracking. (eg: rmat, movement analysis tracking etc.).

The program uses one reference particle that is situated at a user chosen sigma in the space charge distribution. At regular intervals the program checks the approximate linear motion traced by this particle.

This operation MUST ALWAYS be preceded by a MACHine operation providing the list of the twiss parameters at every point of the lattice, and by a REFErence orbit run, and a Constant definition changing the mass of the particle to that of the particle for which the space charge computation is done.

Input format

SPACe charge set up.....(maximum 80 characters)
Option chkturn sigma nturn energy epsilonx epsilony delta lambda

Parameter definitions

option option number presently not used

chkturn number of turns after which the space charge distribution is reevaluated, based on the results of the tracking of the reference particle. This reference particle is used in evey tracking operation. It is placed last in the sequence of particles

sigma the number of sigmas within the gaussian space charge distribution at which the reference particle is located. Care has to be taken that the number chosen does NOT place the reference particle on an unstable depressed tune. If that is the case another value for sigma should be chosen. A value of two is a good choice to define the distribution if the the motion of the reference particle is relatively smooth

nturn the number of turns the setup routine will track the reference particle. This number must be a multiple of chkturn. It should be chosen so that an equilibrium has been reached for the emittance of the distribution.

energy The reference energy in GeV for which the space charge distribution kicks are computed

epsilonx (y) the one sigma emittance of the gaussian beam distribution in x-x' and y-y' planes. This is used to define the distribution at every point in the ring and to locate the reference particle

delta the relative energy deviation for which the space computation are done

lambda The linear charge density in Coulombs per meter

Examples

The example is taken from demo9. It is given together with the operations that must precede it. This operation requires caution in the interpretation of the results.

```
CONSTANT DEF
12 1 3 0.938259,
output
0,
REFERENCE
12 0.002 0.002
0 0 0 0 0 0
0,
output
3,
set symplectic on
1 0.5,
space charge set up
```

```
* new input structure
1 80 2 80 0.45 15 15 0 2.981e-8,
* slightly reduced linear density to avoid error on movement analysis
*1 80 2 80 0.45 15 15 0 1.000e-8,
* old input structure
*1 80 0.45 60 60 0 2.981e-8,
*stop
*1 80 0.45 60 60 0 1.0e-8,
GEOM
1.661 0 14.792 0.0002
0 0 0 0 0
2 320 1
1 -1
1 1
60 60
;
stop
```

9.27 Tracking of particles

This operation tracks up to maxpar particles around the machine. Depending on the version maxpar may be set initially to 1000 or 10000. 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, δ for all particles)
MLOCAT LIST NGRAPH XMIN XMAX XPMIN XPMAX YMIN YMAX
YPMIN YPMAX NCOL NLINE [ALMIN ALMAX DELMIN DELMAX]

Parameter definitions

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 [t]number of turns to be traced.

x, x', y, y', l, δparticle coordinates for NPART particles.

MLOCAT indicates the number of intervals in which printing is to occur. If MLOCAT is equal to 0, then printing occurs at end of lattice only. When MLOCAT is 0, 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)

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 δ, l or the δ, ϕ plots where l is the pathlength coordinate of the particles, 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).del 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 as regards the xx',yy' and xy plots and at the same time will produce an δ, ϕ plot identical to that of produced by the ngraph value of 16.

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 or 16 The are not present for the other values of NGRAPH For NGRAPH=16 AL is to be interpreted as PHI.

NCOL,NLINE number of columns and lines to be used in plot matrix.

Examples

The first two examples are taken from demo3. The second one shows how to use the operation to track already existing particles.

The third example comes from demo13. It illustrates how to plot the longitudinal phase space using the RF phase advance and the relative momentum.

```
TRACKING OF PARTICLE TO MAKE TRIANGLE APPARENT
1 -1 3 500
.001 0.0 0.0003 0.0 0.0 0.00
.0015 0.0 0.0004 0.0 0.0 0.00
.0020 0.0 0.0005 0.0 0.0 0.00
0
11
-.005 0.005 -.001 0.001
-0.0006 0.0006 -.0003 0.0003
51 51,
```

```
MOVEMENT ANALYSIS TO FIND UNSTABLE CLOSED ORBIT NEAR .03 .002 0 0
1 3 3 -4 1 0 .00001 0.03 0.002 0.0 0.0 0.0 0.00 0,
tracking the 10 particles generated by the movemenet analysis just above
1 -2 0 1600
0
11
-.010 0.010 -.002 0.002
-.0006 0.0006 .0003 0.0003
51 51,

tracking
1 -2 3 1000
0 0 0 0 0.001 0
0 0 0 0 0.001 0.0005
0 0 0 0 0.2 0
0
16
-0.01 0.01 -0.01 0.01
-0.01 0.01 -0.01 0.01
51 31
0 12.56 -0.003 0.003,
```

10 Operations used in conjunction with misalignments and errors

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 the detail but the general trends should be similar.

10.1 Alignment fitting

This operations allows the user to fit the 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 .

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$_i$ for i from 1 to nener
Origin
name$_i$, keyword$_i$, del$_i$ for i from 1 to npar
When nfit equals 1 or 2 the following group applies
CORR
name$_i$, pos$_i$, opt$_i$, param$_i$, del$_i$ for i from 1 to ncor
NOTE: ncor+npar=nval
mon$_i$, pos$_i$, val#$_i$, value$_i$, weight$_i$, error$_i$ for i from 1 to ncond
End of the group for nfit 1 or 2
If nfit equals 3 the following group applies :
CORR
mcorr
name$_i$, opt$_i$, param$_i$ for i from 1 to mcorr
nmon nskip
name$_i$, val#$_i$, value$_i$, weight$_i$, error$_i$) for i from 1 to nmon
end of group for nfit 3
nasp
repeat the following nasp times
name keywd npas
name$_k$, keyword$_k$, mult$_k$, add$_k$ for k from 1 to npas

Parameter definitions

nstep number of steps taken to approach final fit.

nit number of iterations used in final step. During these iterations the stepsizes del of the parameters are reduced by a constant factor(5).

nvar number of variables used (maximum 12).

ncond number of conditions fitted.

nfit selects the fitting procedure.

1 Newton's method is used. In this case ncond = nvar. We recommend that you do not use this fit option

2 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.

$\beta_x, \alpha_x, \eta_x, \eta'_x, \beta_y, \alpha_y, \eta_y, \eta'_y$

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 dx' dy dy' increments used in the computation of the cx sx cy sy functions needed to generate the transfer matrices around the nominal orbit

nener number of momenta traced (maximum 3).

ener values of the momenta (p-p0)/p0.

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

$name_i, keyword_i$ names of the elements having parameters to be varied. npar¡=nvar such elements can be used

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

CORR flag to signal 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 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

del_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 sigx value as read by monitor
=(iener-1)*4+4 sigy 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 1 the monitor mon(i) is affected by the error $error_i$.

nasp: number of associated parameters

name1 Name of element to which some parameters are to be associated.

keywd parameter keyword of element name1 to which some parameters are to be associated.

npas total number of parameters to be associated to the parameter (keywd) of name1.

$name_k$ name of the element which has a parameter to be associated with name 1.

par_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 name1 and to which par_k is associated.

Examples

The first two examples are taken from demo6.

The third set comes from demo7. Please refer to these demos for their relation with other operations.

```
* the following is an example of an Alignment fitting with nfit = 3
ALIGNMENT FITTING
0 1 8 8 3 0
3 0 0 0 1 0 0 0
0 0 0 0
0.00001 0.00001 0.00001 0.00001
1 0
MRKC2
CORR
2
HC 0 4
kcv 3 2
2 0
PMC 1 0.0 1.0 20E-06
PMC 2 0.0 1.0 20E-06
0,
```

```
* the following is an example of an Alignment fitting with nfit = 2
ALIGNMENT FITTING
0 1 6 6 2 0
3 0 0 0 1 0 0 0
0 0 0 0
0.00001 0.00001 0.00001 0.00001
1 0
```

```
MRKS1
corr
ksv 2 3 2 0.0001 ksv 16 3 2 0.0001 ksv 35 3 2 0.0001
ksh 7 3 1 0.0001 ksh 19 3 1 0.0001 ksh 32 3 1 0.0001
PMs 13 1 0.0 1.0 20E-06 PMs 13 2 0.0 1.0 20E-06
PMs 28 1 0.0 1.0 20E-06 PMs 28 2 0.0 1.0 20E-06
PMs 40 1 0.0 1.0 20E-06 PMs 40 2 0.0 1.0 20E-06
0,



* the following is an example of an Alignment fitting with nfit = 2
* and shows how to correct  off momentum orbits and beam sigma
* values
ALIGNMENT CORRECTION FINAL FOCUS ORBIT
0 1 6 6 2  0
49.26950355 0 0 0 22.75617615 0 0 0
0 0 0 0
0.000001 0.000001 0.000001 0.000001
2 0 -0.001
MFIN
KORS DXP 0.000001 KORS DYP 0.000001
KETA DXP 0.000001 KETA DYP 0.000001
KORT DXP 0.000001 KORT DYP 0.000001
PMINT 28 1 0.0 1.0 10E-06 PMINT 28 2 0.0 1.0 10E-06
PMINT 31 5 0.0 1.0 10E-06 PMINT 31 6 0.0 1.0 10E-06
PMINT 33 1 0.0 1.0 10E-06 PMINT 33 2 0.0 1.0 10E-06
0,
ALIGNMENT CORRECTION FINAL FOCUS SIGMAS
0 2 4 6 2  0
49.26950355 0 0 0 22.75617615 0 0 0
0 0 0 0
0.000001 0.000001 0.000001 0.000001
3 0 -0.003 0.003
MFIN
QC1 K1 0.0001  QS1 K1 0.0001
Q3XC K1 0.0001  Q2BA K1 0.0001
PMINT 31  3 1.2E-06 1.0 0.000 PMINT 31  4 1.2E-06 1.0 0.000
PMINT 31  7 1.2E-06 1.0 0.000 PMINT 31  8 1.2E-06 1.0 0.000
PMINT 31 11 1.2E-06 1.0 0.000 PMINT 31 12 1.2E-06 1.0 0.000
0,
```

10.2 Baseline definition

This operation defines a baseline resulting from surveying errors. The baseline must be considered 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 to the instruments used but also due 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 developped.Use at OWN RISK.

Input format

BASEline definition
npen Kxxx Kxxx .... Kxxx
nsub σ
Kyyy Kyyy ..... Kyyy
σ_1 σ_2

Parameter definitions

npen Number of penetration points

Kxxx a set of npen Elements of the KICK type. There MUST be one such kick at the beginning and at the end of the lattice.

nsub Number of subdivision points.

σ the displacement sigma to be used in the generation of the coordinate displacements of the npen penetration points.

Kyyy nsub KICK elements defining the intermediate points. They may not coincide with any of the basepoints.

σ_1 angular sigma (in radians) of the systematic aim error

σ_2 angular sigma (in radians) of the random aim error

10.3 Block misalignment

This operation sets up misalignment condition for subsets (or blocks) of a beamline. Every such block is treated as if it was one element. However, any misalignment of individual elements as defined in the misalignement data definition will be superimposed.

A block is defined by indicating the names of its first and last element. Any subset of the lattice beginning and ending with these two names will be considered a a block of the same family and subjected to the same rms errors (not necessarily the same error values).

The line may contain up to mxbloc block families (initially set at 100). A family may contain as many blocks as needed.(there is no built in limit)

Blocks may be nested. An inside block must be closed before the outside block can be closed. Up to ten levels of nesting are allowed.

The present implementation is correct only for straight blocks or blocks with small bend angles : less than .1 radian)

Input format

BLOCk misalignement.....(maximum 80 characters)
nopt ranopt
Name1 name2 dx dx' dy dy' dz dzr ddel
.....
Name1 name2 dx dx' dy dy' dz dzr ddel
99,

Parameter definitions

nopt option number. It is not used at present, but must be present in the input stream

ranopt the option nmumber selecting the random generation that will affect the misalignment and field errors of the blocks. The allowed values for ranopt are 0 1 2 3 11 12 13 .See the set misalignment operation for their definitions.

name1 name2 name of two gkick elements whose names uniquely define the beamline interval to be misaligned as a block.

dx dx' dy dy' dz dzr ddel

one sigma rms values of the random generation of the x x' y y' z zr(rotation around the longitudinal axis) del (dp/p or the relative field error) offsets

10.4 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 and whose setting may be changed to achieve corrections of closed orbits or beam size at monitor locations.

Input format

CORRector data definition ....(maximum 80 characters)
Name i1 f1.....in fn,
.....
Name i1 f1 ....in fn,
99,

Note the comma ending each line defining a corrector. i1 f1 and in fn 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.

Examples
The example comes from demo6.

```
CORRECTOR DATA DEFINITION
HC 1 200,
ksh 1 200,
ksv 1 200,
kcv 1 200,
99,
```

10.5 Error 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 note the line containing 99, to end the input. Parameter and value are the parameter keyword of the element called name that is affected by the error. Value is the value of the error.

Examples
The example comes from demo7.

```
**** THE ERROR COMPUTATION IS TURNED OFF:IT IS TIME CONSUMING
*ERRORS DATA DEFINITION
*HFOC 3  40,
*HDOC 3  40,
*HFDC 3  40,
*HDEFOC 3  40,
*99,
```

10.6 Misalignment data definition

This operation defines the misalignments of different elements of the lattice. Up to mxmis(100) distinct elements can be misaligned.

Input format

MISAlignment data definition....(maximum 80 characters)
Name dm1 dm2 dm3 dm4 dz dzr ddel option
.....
Name dm1 dm2 dm3 dm4 dz dzr ddel option
99,

Parameter definitions

For all values of the option parameter, the parameters dz and dzr are the values of the longitudinal displacement and the rotation angle (in radians!!) around the longitudinal axis. The values dm1,dm2,dm3 and dm4 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 parameter to determine the nature of the misalignment

1 The element is misaligned around the tangent to the central trajectory at the entrance of the elements. In this case the parameters dm1,dm2,dm3 and dm4 are respectively dx,dxr,dy,dyr where dx and dy are the displacements along the axes x and y and dxr ,dyr are rotation angles (in radians) around the axes x and y respectively.

2 The element is misaligned around the chord defined by the two extreme points of the central trajectory. In this case the parameters dm1,dm2,dm3 and dm4 are dx1,dx2,dy1,dy2 where dx1,dy1 are the displacements at the entrance of the element along the axes x and y. The parameters dx2,dy2 are the displacements at the exit of the element along the axes x and y

3 The element is misaligned around the tangent to the central trajectory at the midpoint of the element. The parameters dm1,dm2,dm3 and dm4 have the same meaning as in the case of the option value 1.

4 This does not apply to dipole elements (Bends). In this case the parameters dm1 and dm3 represent displacements in x and y respectively. The particle is also subjected at the entrance and the exit to an angle kick of dm2/2 and dm/4 in x and y respectively (same sign at exit as at entrance).This enables to simulate baseline excursion in a similar way as that defined under baseline operation. Parameters dz dzr are in effect but not ddel.This should be mainly used on monitors.

Examples

The example comes from demo6. Please note that when an element has zero length, the option of the misalignment of that element must be 1.

```
MISALIGNMENT DEFINITION
DDCH  300E-06 300E-06 300E-06 300E-06 0.0010 0.001 0.0 2
DFCH  300E-06 300E-06 300E-06 300E-06 0.0010 0.001 0.0 2
HC    300E-06 300E-06 300E-06 300E-06 0.0010 0.000 0.0 2
DSH  300E-06 300E-06 300E-06 300E-06 0.0010 0.001 0.0 2
QS1D 300E-06 300E-06 300E-06 300E-06 0.0010 0.001 0.0 2
QS2F 300E-06 300E-06 300E-06 300E-06 0.0010 0.001 0.0 2
QS2D 300E-06 300E-06 300E-06 300E-06 0.0010 0.001 0.0 2
QS3F 300E-06 300E-06 300E-06 300E-06 0.0010 0.001 0.0 2
QS3D 300E-06 300E-06 300E-06 300E-06 0.0010 0.001 0.0 2
99,
```

10.7 Reference orbit display

This operation computes the orbit defined by the initial coordinates x x' y y' l δ, and provides either a printout or a printer-plot display.

Input format

REFErence .....(maximum 80 characters)

nprint sizex sizey
$x_0, x'_0, y_0, y'_0, l_0, \delta 0$
npos pos_i for i from 1 to npos.

Parameter definitions

nprint	controls display
1	a printout is provided
2	a printer-plot is provided
11,12	same as for 1 and 2 , the orbit is a 4-dimensional closed orbit in the space defined by the variables x,x',y,y'
21,22	same as for 1 and 2 , the orbit is a six-dimensional closed orbit in the space defined by the variables x,x',y,y',l,δ.

sizex,sizey They define the boundaries between which the coodinates x and y are plotted. They are always needed in the input stream

. $x_0, \ldots \delta$ 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 the end of lattice

Examples
The example comes from dem06.

```
REFERENCE ORBIT
12 0.0100 0.0100
0 0 0 0 0 0.00
0,
```

10.8 Seed selection

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)
ns,

Parameter definitions

ns seed value

0 a seed is generated by the program.

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

Examples
The first example shows the case where the seed 37 is chosen.
The second example is a case where the computer clock will choose a seed.

```
SEED
37,

seed
0,
```

10.9 Set corrector values

This operation is used to manually set corrector to some predetermined values

Input format

SET Corrector values
Name pos opt p1...p4,
.....
Name pos opt p1...p4,
99,

Parameter definitions

Name Name of corrector element whose value is to be set

pos position of corrector

p1...p4 the four parameters used to define the corrector.

opt option number defining the kind of corrector

0 p1 is dx,p2 is dy,p3 is dy' and p4 is ddel 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 Bx induction) The energy of the particle is changed by ddel (This simulates the effect of backleg windings providing a change in By)

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 ddel remain the same as above

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.

3 In this option the corrector acts as a pure dipole steering magnet. Parameter p1 is dxp and parameter p2 is dyp. Parameters p3 and p4 are not used. The angle kicks dxp and dyp are inversely proportional to the momentum of the particles(ie: dxp and dyp are divided by 1+delta where delta is the relative momentum of the particle traced.

10.10 Set errors of elements

This operation specifies the random generation mode and the name of the elements that should actually be affected by the errors, and the intervals locations in the beam line where the errors are applied.

Input format

SET Errors ....(maximum 80 characters)
nopt
nerr
name nint nb_i nf_i for i from 1 to nint
.....
name nint nb_i nf_i for i from 1 to nint

Parameter definitions

The meaning of the parameters are the same as those of the SET MIS.... operation which follows. A special value of nopt : 5 , is used to read the errors sequentially from the fortran input file f007. In this case however one needs to know the order of the element parameters internal to Dimad. Please contact Lindsay Schachinger to obtain the information needed for correct use of this option.

Examples

The example comes from demo7. Just as for error data definition this operation is deactivated in demo7.

```
*SETERROR RANGES
*3
*4
*HFOC 0 HDOC 0 HFDC 0 HDEFOC 0,
```

10.11 Set misalignment of elements

This operation specifies the random generation mode and the name of the elements that should actually be affected by the misalignment errors, and the intervals locations in the beam line where the errors are applied.

Input format

SET Misalignment .....(maximum 80 characters)
nopt
nmis
name nint nb_i nf_i for i from 1 to nint
.....
name nint nb_i nf_i for i from 1 to nint

Parameter definitions

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 element with that name are misaligned. In this case no interval range is given.
- -1 no element 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.

Examples

The first example is taken from demo6.

In the second example a gaussian disbtribution truncated at 2 sigmas will be used for the alignment errors of the elements.

```
SETMISALIGNMENTS
12 9
DFCH 0 DDCH 0 HC 0
DSH  0 QS1D 0 QS2F 0 QS2D 0 QS3F 0 QS3D 0,

SETMISALIGNMENTS
13 9
DFCH 0 DDCH 0 HC 0
DSH  0 QS1D 0 QS2F 0 QS2D 0 QS3F 0 QS3D 0,
```

10.12 Show corrector values

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

Input format

SHO Correctors
option [name]

Parameter definitions

option determines the information to be printed out.

1 The rms, maxima and minima of the values are displayed For this option the parameter name may be omitted

2 This option requires the parameter name. It is the name of the correctors whose values will be printed. 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.

Examples
The example comes from demo7.

```
SHO CORRECTOR VALUES
1,
```

10.13 Show or save misalignment values

This operation displays the actual values of the displacement generated in previous operations and provides manipulation of the misalignement features.

Input format

SHO Misalignment
nopt [ni m_i for i from 1 to nopt]
[name [name$_{b1}$name$_{b5}$...name$_{b5}$name$_{e5}$,]
....
[name [name$_{b1}$name$_{b5}$...name$_{b5}$name$_{e5}$,]
99,
MISFAC

Parameter definitions

nopt option number deciding on the action to be taken according to the description following

0 All misalignement values are printed

>0 nopt intervals are defined with the nopt pairs of numbers n_i m_i . The misalignment values are printed in these intervals

-1 Arrays containing all the misalignement data are set up. Tracking execution proceeds faster. Once this operation (sho mis -1) has been executed, a new set of misalignment data may be defined. This new set is superimposed on the stored misalignment data. A following operation (sho mis -1) will save the accumulated misalignment data and one could if needed define some other misalignment data to superimpose on the previous.

-2 This choice of the parameter nopt allows to add some misalignment to chosen elements in chosen intervals to study effects of added andom errors. Basically this possibility is covered more easily by the use of the operation sho mis -1. Up to five element names "name" and for each name up to five intervals name_{bi} name_{ei} may be chosen. The names defining the intervals must be unique (use markers). The operation adds to the elements namei 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 misalignement procedure. This operation must be preceded by a SHO MIS operation with option -1 to be successful.

-10 This operation provides information of the average lateral displacement of the element labeled name using the scale factor SIGFAC.

Examples

The different examples given all come from demo6.

```
SHOMIS
0,

****** following operations sets up array of misalignement to increase
****** the computation speed
Sho mis
-1,

****** the following shows a listing of the average lateral displacement
****** of a given misaligned element
sho mis
-10 hc,

**** the following adds a random fractional misalignment to some
**** element. For this operation to be operational a 'sho mis' with
**** option -1 must have preceded. The additional misalignement is
**** added to the previously defined misalignement.
```

```
sho mis
-2
hc mrks1 mrks2,
99
0.5,
```

10.14 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,

Parameter definitions

Energy Initial nominal energy in GeV

option option deciding on type of simulation.

0 no synchrotron radiation effect is simulated.

1 synchrotron radiation loss is computed in every dipole. 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. This effect is simulated only in the dipoles.

3 both the radiation loss and the emitttance growth are simulate in the dipoles only.

4 synchrotron radiation is simulated in the dipoles as in option 2 to which is added a synchrotron quantum fluctuation in the quadrupoles.

5 synchrotron radiation is simulated in the dipoles as in option 3 to which is added a synchrotron quantum fluctuation in the quadrupoles.

10 synchrotron radiation losses and emittance growths are computed for each particle in dipole bends according to a more realistic simulation developed by Ghislain Roy.

11 as in 10 both radiation losses and growths are computed in bends and quadrupoles.

12 Only the emittance growths are computed in bends with the Ghislain Roy's simulation.

13 Only the emittance growths are computed in bends and quadrupoles using Ghislain Roy's simulation.

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.

Examples

The examples given come from demo7.

```
SYNCHROTRON RADIATION DEFINITION
50 1 3,

*SYNCHROTRON RADIATION DEFINITION
*50 3 3,
```

References

[1] K.L.Brown,D.C.Carey,Ch.Iselin,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 Particl 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.

[3] 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.

[4] F. Christoph Iselin "The Mad Program Reference Manual," CERN, LEP Division November 1, 1984.

[5] David Douglas, Etienne Forest, Roger 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.

[6] Karl L. Brown, Roger V. Servranckx First- and Second-Order Charged Particle Optics. SLAC-PUB-3381. July 1984 .Stanford Linear Accelerator Center.

[7] M.Berz COSY INFINITY version 5, User's Guide and Reference Manual. Technical report MSUCL-811, December 1991

11 Appendix: in Dimad, but not yet in the Manual.

```
Date: Sat, 16 Sep 2000 07:39:04 -0700
From: Roger Servranckx <servranc@home.com>
To: kaltchev@triumf.ca
Cc: Craddock@triumf.ca
Subject: fitting using movement

Dear Dobrin
The introduction of fitting using movement analysis is now done on dimad
in the directory
~roger/dirnewdim.
It seems to work ok though more tests need to be done.
The example file is in the same directory : testlsq
the corresponding output file is : lsqout

The numeric codes are 701 to 712 for the following variables :
  nux chromx betax alphax etax etapx nuy chromy betay alphay etay etapy

The result is not quite as accurate as with the regular fitting because
we use movement analysis.
We may need to check on the choice of ddelta.
I also use three energies to be able to fit more variables than just the
chromaticities.

Changes in the program are as follows :
Common CTUNE in dimad 09,12(twice),18,24,28
Common FOUT in dimad 09 12(twice) 17 24
Common FITL in dimad 06,07(twice) 09,11,12(three times),13,15,17,27


Subroutine SETOUT in dimad 24
Subroutine Fitmov in dimad 12
Subroutine fitlsq in dimad 12 (An error was corrected : 780 becomes 760
)
Function FFFCT in dimad09

I think that is all. All changes have been marked DK & RVS Sept 2000 (I
hope!!)

Bye for now
Roger
```