USERS GUIDE TO THE PROGRAM DIMAD

Roger V. SERVRANCKX  
University of Saskatchewan  
Saskatoon Canada  
and  
Stanford Linear Accelerator Center  
Stanford California USA

Karl L. BROWN  
Stanford Linear Accelerator Center  
Stanford California USA

Lindsay SCHACHINGER  
SSC CDG, Lawrence Berkeley Lab  
Berkeley, California USA

David DOUGLAS  
CEBAF project. SURA  
Newport News Virginia USA

Peter G. TENENBAUM  
Stanford Linear Accelerator Center  
Stanford California USA

April 20, 2001

1Work sponsored by the Department Of Energy and by the National Science and Engineering Research Council of Canada
The present guide corresponds to the Next Linear Collider (NLC) program version dated April 20, 2001 (Version 2.8).

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. Please inform one of the following persons about any anomalies observed:

David Douglas at CEBAF 804/249 7512
Lindsay Schachinger at LBL 510 486 6590
Roger Servranckx at Saskatoon 306 966 6054.

For anomalies or other bugs in the NLC version, contact:

Peter Tenenbaum at SLAC 650 926-4429 (quarkpt@slac.stanford.edu).
Contents

1 INTRODUCTION ........................................... 7
  1.1 ELEMENT and MACHINE DATA INPUT .................. 7
  1.2 UNITS ............................................. 8
  1.3 GENERAL SYNTAX ................................... 8
  1.4 PARAMETERS ....................................... 9
  1.5 ELEMENT DEFINITIONS ................................ 9

2 ELEMENTS ............................................. 11
  2.1 drift ............................................... 11
  2.2 sbend .............................................. 11
  2.3 rbend ............................................... 11
  2.4 quadrupole ........................................ 12
  2.5 sextupole .......................................... 12
  2.6 quadsext .......................................... 12
  2.7 octupole .......................................... 12
  2.8 multipole .......................................... 12
  2.9 dimultipole ....................................... 13
  2.10 solenoid .......................................... 13
  2.11 rf cavity ......................................... 13
  2.12 lcavity ........................................... 14
  2.13 roll, srot ........................................ 14
  2.14 zrot, yrot ........................................ 14
  2.15 hkick, vkick ...................................... 14
  2.16 glkick ............................................ 15
  2.17 hmmon, vmon, monitor ............................ 15
  2.18 blmonitor, profile, wire, slmonitor, imonitor, instrument 15
  2.19 marker ............................................. 15
  2.20 ecollimator, rcollimator ........................ 16
  2.21 arbiter ........................................... 16
  2.22 mtwiss ........................................... 16
  2.23 matrix ............................................ 16
  2.24 BEAMLINE DEFINITIONS ......................... 16
  2.25 CONTROL FLOW ................................... 17
  2.26 OPERATION LIST DESCRIPTION .................. 18

3 IMPLEMENTED OPERATIONS ............................ 19
  3.1 ADIABATIC VARIATIONS ............................. 19
  3.2 BATMAN DISTRIBUTION ............................. 19
  3.3 BEAM MATRIX TRACKING ........................... 20
  3.4 BEAM GAS SCATTERING ............................. 21
  3.5 CHROMATIC PRECISION OPTION ................... 21
  3.6 CONSTANT DEFINITION ............................ 22
  3.7 DAMPING SET ..................................... 22
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.8 Detailed Chromatic Analysis</td>
<td>23</td>
</tr>
<tr>
<td>3.9 Effective R-Matrix Option</td>
<td>24</td>
</tr>
<tr>
<td>3.10 Generation of Particles</td>
<td>24</td>
</tr>
<tr>
<td>3.11 Geometric Aberrations in Multiple Turn Operation</td>
<td>25</td>
</tr>
<tr>
<td>3.12 Hardware Values Listing of Machine</td>
<td>26</td>
</tr>
<tr>
<td>3.13 Histogram Parameters</td>
<td>27</td>
</tr>
<tr>
<td>3.14 Interactive Control of Lattice</td>
<td>27</td>
</tr>
<tr>
<td>3.15 Least Square Fit</td>
<td>28</td>
</tr>
<tr>
<td>3.16 Line Geometric Aberrations (One Turn Computation)</td>
<td>30</td>
</tr>
<tr>
<td>3.17 Logbook Set</td>
<td>30</td>
</tr>
<tr>
<td>3.18 Machine and Beam Parameters Computations</td>
<td>31</td>
</tr>
<tr>
<td>3.19 MAD Particles</td>
<td>32</td>
</tr>
<tr>
<td>3.20 Matrix Computation</td>
<td>32</td>
</tr>
<tr>
<td>3.21 MCS Setup</td>
<td>33</td>
</tr>
<tr>
<td>3.22 Modification of Element Data</td>
<td>33</td>
</tr>
<tr>
<td>3.23 Movement Analysis</td>
<td>33</td>
</tr>
<tr>
<td>3.24 Observation Set</td>
<td>34</td>
</tr>
<tr>
<td>3.25 Output Control</td>
<td>35</td>
</tr>
<tr>
<td>3.26 Particle Analysis and Control</td>
<td>35</td>
</tr>
<tr>
<td>3.27 Print Selection</td>
<td>36</td>
</tr>
<tr>
<td>3.28 Radiation Lengths</td>
<td>37</td>
</tr>
<tr>
<td>3.29 R-Matrix Computation (6x6)</td>
<td>37</td>
</tr>
<tr>
<td>3.30 Seismic Perturbation Simulation</td>
<td>38</td>
</tr>
<tr>
<td>3.31 Set Fit Point</td>
<td>38</td>
</tr>
<tr>
<td>3.32 Set Limits to Variables</td>
<td>38</td>
</tr>
<tr>
<td>3.33 Set Symplectic Option On</td>
<td>39</td>
</tr>
<tr>
<td>3.34 SHO Values of Constants</td>
<td>39</td>
</tr>
<tr>
<td>3.35 Simple Fitting</td>
<td>40</td>
</tr>
<tr>
<td>3.36 Space Charge Computation</td>
<td>40</td>
</tr>
<tr>
<td>3.37 Tape File Output</td>
<td>41</td>
</tr>
<tr>
<td>3.38 Thermal Photon Scattering</td>
<td>41</td>
</tr>
<tr>
<td>3.39 Tracking of Particles</td>
<td>42</td>
</tr>
<tr>
<td>4 Operations Associated with Linear Acceleration</td>
<td>44</td>
</tr>
<tr>
<td>4.1 Charge Set</td>
<td>44</td>
</tr>
<tr>
<td>4.2 Energy Display</td>
<td>44</td>
</tr>
<tr>
<td>4.3 Phase Linac Cavities</td>
<td>45</td>
</tr>
<tr>
<td>4.4 Wakefield Options</td>
<td>45</td>
</tr>
<tr>
<td>5 Operations Used in Conjunction with Misalignments and Errors</td>
<td>47</td>
</tr>
<tr>
<td>5.1 Alignment Fitting</td>
<td>47</td>
</tr>
<tr>
<td>5.2 Baseline Definition</td>
<td>49</td>
</tr>
<tr>
<td>5.3 Block Misalignment</td>
<td>50</td>
</tr>
<tr>
<td>5.4 Corrector Data Definition</td>
<td>50</td>
</tr>
<tr>
<td>5.5 Errors Data Definition</td>
<td>51</td>
</tr>
<tr>
<td>5.6 Misalignment Data Definition</td>
<td>51</td>
</tr>
<tr>
<td>5.7 New Misalignments</td>
<td>52</td>
</tr>
<tr>
<td>5.8 Reference Orbit Display</td>
<td>52</td>
</tr>
<tr>
<td>5.9 Seed</td>
<td>53</td>
</tr>
<tr>
<td>5.10 Set Corrector Values</td>
<td>53</td>
</tr>
<tr>
<td>5.11 Set Errors of Elements</td>
<td>54</td>
</tr>
<tr>
<td>5.12 Set Misalignment of Elements</td>
<td>54</td>
</tr>
<tr>
<td>5.13 SHO Correctors</td>
<td>55</td>
</tr>
<tr>
<td>5.14 SHO Errors</td>
<td>55</td>
</tr>
</tbody>
</table>
Chapter 1

INTRODUCTION

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

In 1970 the first author had the good fortune of discovering the program OSECO (Optique du SECond Ordre) written by J.L. LA克莱 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].

Adapting the input code of MAD[3], Lindsay Schachinger transformed the program so it would enjoy a common input with MAD, thereby conforming to the input STANDARD. The present version of NLC-DIMAD uses the standalone parser library for Extended Standard Input Format (XSIF), which is also used by LIAR[4]. This simplifies the maintenance and expansion of the language and its capabilities and ensures that decks that run on one NLC simulation platform run unmodified on the others. A detailed description of XSIF syntax and technical description of the parser is available [5].

With ideas developed originally by E. Forest[6], David Douglas introduced the symplectic tracking options and the canonical variables.

1.1 ELEMENT and MACHINE DATA INPUT

The input format to dimad now conforms quite closely to the standard format, as laid out in [2]. This conversion of dimad to standard input was accomplished by taking the input subroutines from the program MAD[3] and making from these routines (with modifications) an input interface for dimad. One exception to the standard format is the units conventions.
Input to dimad can be in either transport units (indicated by the keyword \texttt{utransport}), or in standard units (indicated by the keyword \texttt{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 \texttt{quadsext}, \texttt{gkick}, and \texttt{mtwiss}. These elements are described more fully later. Also, in dimad, the solenoid can have a quadrupole field. Elements which are described in references 2, but which are not implemented in dimad are \texttt{separator} and \texttt{rbend}.

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

### 1.2 UNITS

The keyword \texttt{utransport} indicates that the input has the following units:

- angles in degrees except for $dx'$ and $dy'$ for the kicks
- lengths in meters
- energy in GeV
- electromotive force in kilovolts
- frequency in Hz
- field expansion is $B(x,0) = B\rho \sum_n K_n x^n$
- Solenoidal constant is $K_s \equiv 0.5 B_s / B\rho$
- positive $K_1$ is horizontally focussing.

Also, the \texttt{utransport} keyword has implications for the field expansion coefficients in the \texttt{sbend} element.

The keyword \texttt{ustandard} indicates that the input has the following units:

- angles in radians
- lengths in meters
- energy in GeV
- electromotive force in megavolts
- frequency in Megahertz
- field expansion is $B(x,0) = B\rho \sum_n K_n x^n / \pi t$
- Solenoidal constant is $K_s \equiv B_s / B\rho$
- positive $K_1$ is horizontally focussing.

In addition to these keywords, an additional units keyword \texttt{umad} is supported. This allows use of the older MAD units, in which the signs of $K_n$ for $n > 0$ is reversed. This option is offered for backwards-compatibility with older decks and is not recommended for use in the development of new beamlines.

For all elements which take a TILT parameter, use of the TILT keyword with no argument or with a positive argument causes the element to be rotated clockwise when viewed from upstream.

### 1.3 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 a “!”. Any line containing one of the characters “*”, “%”, “@” in the first column is treated as a comment. A “;” is used to separate statements on the same line. Keywords are uniquely specified by the first four letters, and only those four must be entered. At most 8 letters in a keyword are checked. The keyword \texttt{NOECHO} can be used to suppress transmission of the input data stream to the output files. The keyword \texttt{ECHO} reinstates the stream of the input data to the output files.
1.4 PARAMETERS

Parameters are defined with a statement like:

\[ \text{name} [:]= \text{value} \]

where \text{name} 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:

\[
\begin{align*}
\text{lslot}=100 \\
\text{lb} &= \text{lslot}/8 \\
\text{lh} &= \text{sqrt(lslot)}
\end{align*}
\]

FORTRAN functions which may be used in parameter definitions are: SQRT, LOG, EXP, SIN, COS, ATAN, ASIN, ABS, in addition to the familiar +,-,*,/. On initialization DIMAD sets up the values of several useful constants as parameters. These are PI, TWOPI, DEGRAD (radians to degrees), RADDEG (degrees to radians), E (natural number), EMASS, PMASS, CLIGHT. New values for these parameters can be entered in the same fashion as all other DIMAD parameters. Note that when PI, TWOPI, RADDEG, EMASS, or CLIGHT are changed the equivalent DIMAT parameters are also changed (as if altered by a CONSTANT DEFINITION command).

While DIMAD recognizes the MAD CONSTANT statement:

\[
\text{ACON : CONSTANT}= 1.0
\]

the “constant” is defined in DIMAD as a parameter which can be redefined (unlike MAD, which does not permit CONSTANTS to be changed after the first definition). The recognition of the CONSTANT keyword is provided for compatibility only.

1.5 ELEMENT DEFINITIONS

To define an element,

\[ \text{elname: elmkw [,pkeyw=value,...]} \]

where \text{elname} is the name of the element, elmkw is an element keyword (see below), and pkeyw is a parameter keyword appropriate for the element (see below). value again can be a parameter name, or an expression involving parameters.

Examples:

\[
\begin{align*}
\text{B: SBEND, L=LB, ANGLE=LB/RHO} \\
\text{DO: DRIFT}
\end{align*}
\]

Conversely, the values of element parameters can be accessed by surrounding the name of the parameter by square brackets.

Examples

\[
\begin{align*}
\text{B01A: SBEND, L=LB, ANGLE=LB/RHO} \\
\text{DO: DRIFT, L=B01A[L]} \\
\text{BANGL=B01A[ANGLE]}
\end{align*}
\]

The statements above define a sector bend magnet, a drift whose length equals the length of the bend magnet, and a parameter whose value equals the bend angle of the bend magnet.

NLC-DIMAD can also make use of the MAD Element Class construct, in which a previously-defined element name can be used in place of a recognized keyword.

Examples
The statements above define a quad Q1, and a second quad Q1A which is identical in every parameter except K1 to Q1.

Element names can be from 1 to 8 characters in length; the first character must be a letter, and characters 2-8 can be letters, digits, underscores ('_'), dollar signs ('$'), or periods ('.'). All elements must have unique names.

In addition to its name, each element may have a 16-character TYPE designation and a 24-character LABEL designation. TYPE and LABEL are both alphanumeric, and need not be unique. The first character in a TYPE or LABEL must be a letter or digit, and the remainder may be letters, digits, underscores, dollar signs, periods, or colons (':').
Chapter 2

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.

2.1 drift

1 is the length.

2.2 sbend

1 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 conventions are being used, k2 is 

\[ \beta = \frac{\rho}{B_0} \frac{\partial^2 B_0}{\partial x^2}. \]

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.

2.3 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.
2.4 quadrupole

\begin{verbatim}
1 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.
\end{verbatim}

2.5 sextupole

\begin{verbatim}
1 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.
\end{verbatim}

2.6 quadsext

\begin{verbatim}
1 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.
\end{verbatim}

2.7 octupole

\begin{verbatim}
1 is the length.
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.
\end{verbatim}

2.8 multipole

\begin{verbatim}
l or lrad is the length.
k0l - k20l are the integrated strengths.
t0 - t20 are the tilt angles. If tn is entered without a value, halfturn/2(n+1) is assumed.
kzl is the integrated longitudinal strength, kzl = BzL/Bρ.
krl is the integrated radial strength, krl = BrL/Bρ.
theta is the beam crossing angle with respect to the symmetry axis of the solenoidal component.
z is the distance in z to the point at which the beam and solenoidal axes meet.
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.
\end{verbatim}

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.

NOTE3: Parameters kzl, krl, theta, and z are used to simulate a beam passing at an angle through a solenoidal field slice.

NOTE4: The k0l parameter is somewhat different from the MAD parameter, in that MAD assumes that the K0L represents a design bend field (which alters the central trajectory), while DIMAD treats it as a corrector magnet (which deflects the beam off of the central trajectory).

2.9 dimultipole

This element is the “traditional” DIMAD multipole. It is used to preserve compatibility with existing DIMAD decks, since the keywords of the MULTIPOLE have been changed to allow compatibility with MAD.

- **l** is the length. If the length is zero, the strengths are interpreted as integrated strengths.
- **k0 - k20** are the integrated strengths.
- **t0 - t20** are the tilt angles. If t0 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.

2.10 solenoid

- **l** is the length.
- **ks** is the solenoid strength.
- **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.

2.11 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.
- **harmon** is the harmonic number of the cavity. If both a frequency and a harmonic number are provided, computations will use the harmonic number and ignore the frequency.
- **energy** is the energy.(GeV) If no energy is entered, all calculations which require it will assume that the beam is ultrarelativistic (1/γ ≈ 0).
- **eloss** This is the energy loss factor of the cavity, in V/coulomb.
- **lfile** This is a filename of up to 78 characters enclosed in double quotes. The file contains longitudinal wakefield data for the cavity, in units of V/coulomb/meter. The filename may use the $PATH construct to refer to an alternate path (see Control Flow, below).
tfile  This is a filename of up to 78 characters enclosed in double quotes. The file contains transverse wakefield data for the cavity, in units of V/coulomb/m². The filename may use the $PATH construct to refer to an alternate path (see Control Flow, below).

nbin, binmax These are provided for compatibility with the NLC-MAD system for simulating wakefields, and are not used by DIMAD.

aperture is the aperture for the Hardware operation.

2.12 lcavity

This element is a cavity or structure for linear acceleration.

l is the length.
e0 is the injection energy, typically supplied only for the first cavity.
deltae is the energy gain on crest without beam loading (kV for TRANSPORT).
phi0 is the phase offset for a reference particle (degrees in TRANSPORT, radians/2π in STANDARD/MAD units). A positive phase indicates that the RF crest is ahead of the bunch.
freq is the frequency (Hz for TRANSPORT).
eloss is the cavity beam loading factor in V/Coulomb.
lfile is a 78-character filename, enclosed in double quotes. The file contains the cavity’s longitudinal wakefield Green’s Function in V/C/m. The filename may use the $PATH construct to refer to an alternate path (see Control Flow, below).
tfile is a 78-character filename, enclosed in double quotes. The file contains the cavity’s transverse wakefield Green’s Function in V/C/m². The filename may use the $PATH construct to refer to an alternate path (see Control Flow, below).
nbin, binmax are not used by DIMAD, and are present only for compatibility with NLC-MAD’s simulation of wakefields.
aperture is the aperture.

2.13 roll, srot

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.

2.14 zrot, yrot

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.

2.15 hkick, vkick

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

kick a horizontal (vertical) kick of size kick
tilt rotation angle about the longitudinal axis
2.16  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 \( dp/p \).
- \( \text{angle} \) is the angle through which the coordinates are rotated about the longitudinal axis.
- \( dz \) is the longitudinal displacement.
- \( v \) is the entrance-exit parameter of the kick. \( v \) is positive for an entrance kick, and negative for an exit kick. The absolute value of \( v \) is used to force the kick to be applied every \( \text{abs}(v) \) 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 \( p_0 \) with amplitude \( (p-p_0) \) 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.

2.17  hmon, vmon, monitor

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

- \( l \) is the monitor length.
- \( xserr \), \( yserr \), \( xrerr \), \( yrerr \) are the \( x \) and \( y \) systematic and random errors. NOTE: the errors are not used in this form presently. Errors are introduced via the misalignment operations.

2.18  blmonitor, profile, wire, slmonitor, imonitor, instrument

These are different types of instrumentation. They are treated as drifts by the program and are useful primarily in layout of beamlines.

- \( l \) is the length.

2.19  marker

A marker is a drift element of zero length. It has no parameters.
2.20  **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.

\[
\begin{align*}
\text{l} & \quad \text{is the length.} \\
x\text{size} & \quad \text{x and y collimator apertures. The default apertures are 1 meter.} \\
y\text{size} & \\
\end{align*}
\]

2.21  **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.

\[
\begin{align*}
\text{l} & \quad \text{is the length.} \\
p1 - p20 & \quad \text{are the parameters.} \\
\end{align*}
\]

2.22  **mtwiss**

This is a general transfer matrix expressed in terms of its matched Twiss parameters and phase advance.

\[
\begin{align*}
\text{l} & \quad \text{is the length.} \\
mux & \\
\text{betax} & \\
\text{alphax} & \\
muy & \\
\text{betay} & \\
\text{alphay} & \quad \text{are the twiss parameters for this transfer matrix. betax and betay have default values of 1.} \\
\end{align*}
\]

2.23  **matrix**

This element is a general transfer matrix.

\[
\begin{align*}
\text{rij} & \quad \text{are the matrix elements.} \\
t\text{ijk} & \quad \text{i, j, and k range from 1 to 6, but } j \text{ is always less than or equal to } k. \\
\end{align*}
\]

2.24  **BEAMLNE DEFINITIONS**

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

\[
\text{label: line}=(\text{member1, member2, member3,.....})
\]

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

Examples:

\[
\begin{align*}
df\text{.line} & = (dq, oo, b, oo qf) \\
f\text{dstar}\text{.line} & = (qf, sf, b, sd, qd) \\
arc\text{.line} & = (df, 64*(f\text{dstar},df)) \\
\end{align*}
\]

Beamlines can also have formal arguments. An example is:
\begin{align*}
\text{fdstar}(sf, sd) & : \text{line} = (qf, sf, b, sd, qd) \\
\text{where} \text{sf} \text{ and} \text{sd} \text{ are not defined elements, but variables. So} \\
\text{super} & : \text{line} = (\text{fdstar}(sd1, sf1), df, \text{fdstar}(sd2, sf2))
\end{align*}

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

\section{2.25 CONTROL FLOW}

DIMAD can be called with a command file as a command-line argument:

\begin{verbatim}
  dimad commandfile.dimad
\end{verbatim}

In this case, DIMAD will open \texttt{commandfile.out} as its command output (logical unit 9), and \texttt{commandfile.echod} as its echo file (logical unit 8). The file \texttt{commandfile.dimad} will itself be opened as logical unit 7. If no command file is specified, DIMAD will expect to find its commands in the standard input (logical unit 5, or keyboard); output and echoes will be written to the same units, but without file names (ie, \texttt{fort.9} and \texttt{fort.8}). Echo file output is also directed to logical unit 6 (screen).

Files can be opened for read/write access by dimad via the \texttt{OPEN} statement:

\begin{verbatim}
  open, nn
  filename
\end{verbatim}

where \texttt{nn} is an integer and \texttt{filename} is the name of the file; this command causes file \texttt{filename} to be opened to logical unit number \texttt{nn}. The command \texttt{close, nn} closes the file. Once a file is opened via the \texttt{OPEN} statement, standard input can be switched to this file by the \texttt{CALL} statement:

\begin{verbatim}
  call, nn
\end{verbatim}

Redirection of standard input is terminated by a RETURN statement in the CALLe\textit{d} file or by end-of-file in the CALLe\textit{d} file. Up to 32 nested CALL statements can be made.

The default path which is searched for files by the \texttt{OPEN} command is the directory from which DIMAD was launched. An alternate path may be specified by the \texttt{PATH} command:

\begin{verbatim}
  path
  pathname
\end{verbatim}

When a filename is specified which contains the string \$\texttt{PATH}, DIMAD replaces \$\texttt{PATH} with the present value of the specified path. The \$\texttt{PATH} string can also be used in the pathname given in the \texttt{PATH} command; in this way, the selected absolute pathname can exceed the 80 character input line limit.

Beamline definitions are followed by a use statement in the form

\begin{verbatim}
  use, beam linename
\end{verbatim}

This causes the beamline \texttt{beam linename} to be the current machine for dimad. Next comes the statement

\begin{verbatim}
  dimat
\end{verbatim}

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

A `;` or the \texttt{DIMAT} command MADIN 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:
The last command “newbeamo” passes control back to dimad. Observe that without newbeamo 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.

### 2.26 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 132 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 XSIF input parser. DIMAD execution is terminated when the command STOP is encountered by either the XSIF parser or by the DIMAT command processor.

In all tracking operations the particle coordinates are checked at the entrance of some element. Particles are lost when the radial excursion is greater than the expulsion factor. It is set at the default value of 1 meter. Its value can be changed via the constant definition operation.
Chapter 3

IMPLEMENTED OPERATIONS

3.1 ADIABATIC VARIATIONS

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 same job. Two options are available: linear and sinusoidal variation.

Input format:

ADIAbatic variations of some parameters (up to 80 char)
name pkeyw nopt p1 p2 val1 val2 val3 val4
.......... name pkeyw nopt p1 p2 val1 val2 val3 val4
99,
if nopt = 3 then after nopt enter the following:
npts n(i) val(i) for i = 1 to npts

Parameters:

name name of element having a parameter to be varied
pkeyw keyword of parameter to be varied (i.e. k1 for a quad)
nopt option number

1 means variation will be linear according to the following rule: The parameter remains constant at value val1 until turn p1 then varies linearly to achieve the value val2 at turn p2. In this case only two parameter p’s and only two values vali are present in the input format.

2 means the variation will be sinusoidal between turn p1 and turn p2. The variation is done according to the formula:

\[ \text{value} = \text{val1} + \text{val2}\times\sin((2\pi\times\text{turn}/\text{val3})+\text{val4}) \]

where turn is the current turn at which value is applied. Outside turns p1 and p2 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.

3.2 BATMAN DISTRIBUTION

Sets up the beam energy according to a “Batman” (i.e., twin-horned) distribution. This is the distribution which is typical from a heavily-loaded linac with BNS damping phase profile. The distribution is simulated by generating a parabolic energy distribution.
Input format:
BATMan distribution...(up to 80 char)
\( \delta_0 \) width ratio,

Parameters:
\( \delta_0 \) relative momentum \((p-p_0)/p_0\), about which the distribution is to be centered
width half-width of the distribution
ratio ratio between the height of the distribution at the peaks and at the center. If a value less than 1.0 is entered, a flat distribution is used.

3.3 BEAM MATRIX TRACKING

Computes beam matrices at selected points of the machine from the initial beam matrix defined in the input of the operation. If \( \sigma_i \) and \( \sigma_o \) denote the beam sigma matrices at the entrance and exit of a beam line section then
\[
\sigma_o = R\sigma_i \ast R^t
\]
where \( R \) and \( R^t \) are the transformation matrix of the section and its transpose.

Input format:
BEAM matrix tracking computations..(up to 80 char)

\[
\begin{array}{cccccccc}
\sigma_x & r_{xx'} & r_{xy} & r_{xy'} & r_{xl} & r_{xp} \\
\sigma_{x'} & r_{x'y} & r_{xy'} & r_{x'y'} & r_{x'p} \\
\sigma_y & r_{yy'} & r_{yl} & r_{yp} & & \\
\sigma_{y'} & r_{y'lr} & r_{y'p} & & \\
\sigma_l & r_{lp} & & & \\
\sigma_p & & & & \\
\end{array}
\]
mprint [list]

or
\[
0
\]
\( \beta_x \ a_x \ \eta_x \ \eta_x' \ \epsilon_x \)
\( \beta_y \ a_y \ \eta_y \ \eta_y' \ \epsilon_y \)
(\sigma_l \ \sigma_p \)
mprint [list]

or
\[
0
\]
0 0 0 0 \( \epsilon_x \)
0 0 0 0 \( \epsilon_y \)
(\sigma_l \ \sigma_p \)
mprint [mlist]

Parameters:
\( \sigma_{...} \) sigma extension of the beam.(as defined [1] and [7])
\( r_{ij} \) correlation cosines as defined in [1].
\( \beta_{...} \eta_{...} \) initial values of twiss parameters used to define an uncoupled beam.
\( \epsilon_x, \epsilon_y \) emittances in x and y of the input beam. If \( \epsilon_x < 0 \) or \( \epsilon_y < 0 \), then the normalized emittances are assumed to be \( |\epsilon_x| \) and \( |\epsilon_y| \), and emittances calculated by the PART operation will also be normalized. This option is only valid if the energy at each point in the machine has been previously set by a PHASE command (see below).
NOTE: when $\beta_x$ etc are zero the values are obtained from a previous move-
ment analysis calculation made within a MATRix operation or a Fit
operation that generates a matrix calculation.

mprint -2 no computation is done. The operation serves only to define a beam
as needed in the operations beam TRACking 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.

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

3.4 BEAM GAS SCATTERING

defines the options related to the simulation of beam-gas scattering in the beamline. Either elastic (Mott)
scattering or inelastic (Bremsstrahlung) scattering may be selected. Further discussion of the simulation
technique can be found in [9].

Input format:
BGAS... (up to 80 characters)
bgopt bgturn energy prob cutoff,

Parameters
bgopt 1 Selects Mott scattering
2 Selects beam-gas Bremsstrahlung
bgturn Number of turns during which scattering is to be simulated
energy Beam energy in GeV. If beam energy is set by a PHASE command, the
PHASE energy overrides the energy specified here.
prob Scattering probability per particle per meter.
cutoff If Mott scattering is selected, the cutoff is the minimum scattering angle
which is to be simulated, in radians; if Bremsstrahlung is selected, the
cutoff is the minimum relative change in particle energy which is to be
simulated. Note that in either case the cross-section diverges for cutoff
= zero.

3.5 CHROMATIC PRECISION OPTION

selects the desired chromatic precision of tracking.

Input format:
CHROmatic precision option...(up to 80 characters)
nopt,

Parameters
Extended chromatic precision is used. At the entrance of bend magnets, the sixth coordinate is converted from $\delta$ to $\bar{\delta} = \delta/(1+\delta)$; when tracking through quadrupoles, sextupoles, or quad-sexts the transfer matrix is recalculated for each particle based on its actual energy. This allows tracking through a lattice which is correct to all orders in $\delta$.

Standard chromatic precision is used. No conversion of $\delta$ is performed in bends, and magnetic matrices are statically calculated. This is substantially faster than option 1, and usually more than adequate for beams with relatively small momentum spreads.

### 3.6 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 radius. The scale factors ETAFA$C$ and SIGFA$C$ are also accessible via this operation. Use the operation SHO Constant to examine the constants. Note that changing $\pi$ or the speed of light will also change the relevant parameters available in the MAD input section.

**Input format:**

```
CONStant definition .....(up to 80 Characters)
n(1),val(1),...,n(p),val(p)
```

**Parameters:**

- **n(i)** is the order number of the ith constant to be redefined according to the following order:
  1. $\pi$
  2. velocity of light
  3. particle mass
  4. particle radius
  5. particle charge
  6. reference energy used in taylor series expansions
  7. least square fit initial tolerance
  8. factor for maximum function calls in least square fit
  9. the expulsion factor
  10. scale factor for eta function
  11. scale factor for sigma values
  12. particle type (0 for electrons and 1 for protons, the default is 0). Presently when the particle type is changed the particle mass and radius is NOT changed. The particle type selection only affects the rf cavity functions: the particle velocity is taken into account to compute the phase of the particle relative to the RF.

- **val(i)** new value of the constant $n(i)$.

### 3.7 DAMPING SET

Selects which phase planes will be subject to synchrotron radiation damping. This option is only meaningful for storage rings (linear accelerators have adiabatic damping enabled at all times).
Input format:
DAMPing set...(up to 80 characters)
nopt,

Parameters

<table>
<thead>
<tr>
<th>nopt</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Neither longitudinal nor transverse damping is enabled.</td>
</tr>
<tr>
<td>1</td>
<td>Only longitudinal damping is enabled: synchrotron radiation emission is calculated for a particle’s actual energy including its offset from the design energy. This is only valid for SYNCH with options 10 through 13.</td>
</tr>
<tr>
<td>2</td>
<td>Only transverse damping is enabled: the angle of particles tracked through an RFCAV is reduced by the fractional energy gain.</td>
</tr>
<tr>
<td>3</td>
<td>Both longitudinal and transverse damping are enabled.</td>
</tr>
</tbody>
</table>

3.8 DETAILED CHROMATIC ANALYSIS

Traces particles (2 per plane, per momentum) to determine the linearized transfer matrix from the initial point to any related point in the lattice. A twiss function computation is then done at these points. The initial central particle position is assumed to be \( x_0 \) \( y_0 \). 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 energy \( e_i \), particles are generated around the point \( P_0 (x_0, y_0) \) to compute the elements \( R_{ij} \) of the matrix describing the linear motion around the trajectory defined by the point \( P_0 \).

Input format:
DETAiled chromatic analysis...(up to 80 characters)
NH NV NHV
\( x_0 x'_0 y_0 y'_0 \)
\( dx dx' dy dy' \)
\( \beta_x \alpha_x \eta_x \eta'_x \)
\( \beta_y \alpha_y \eta_y \eta'_y \)
Nener Ncoef
\( e_1 e_2 \ldots e_{Nener} \)
MLOCAT [LIST]

Parameters:

<table>
<thead>
<tr>
<th>NH</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Only ( xx' ) motion is traced and computed. ( \beta_x, \alpha_x, ) and ( \nu_x ) are computed.</td>
</tr>
<tr>
<td>0</td>
<td>( xx' ) motion alone is not analyzed.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NV</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Only ( yy' ) motion is traced and computed. ( \beta_y, \alpha_y, ) and ( \nu_y ) are computed.</td>
</tr>
<tr>
<td>0</td>
<td>( yy' ) motion alone is not analyzed.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NHV</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Coupled motion ( xx', yy' ) is traced. The full beam matrix is computed.</td>
</tr>
<tr>
<td>2</td>
<td>Coupled motion is computed. A short print is provided with ( x' x y y' ) ( \beta_x \alpha_x \beta_y \alpha_y \beta_x \alpha_x \beta_y \alpha_y \nu_x \nu_y ) for all energies requested</td>
</tr>
<tr>
<td>3</td>
<td>In this case three energies have to be defined: ( \delta_0, \delta_0 - \epsilon, \delta_0 + \epsilon ). This enables computing the basic machine parameters associated with energy ( \delta_0 ). Choose ( \epsilon ) comfortably close to zero for accurate computation of the ( \eta ) functions. A short print of the machine parameters is provided. The values of ( \eta ) and ( \eta' ) are affected by a scale factor ( \text{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.</td>
</tr>
</tbody>
</table>
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 definition operation.

Same as three with the code added in the first column to facilitate some plotting work NOTE: the input beam must have been defined previously in a BEAM MATRIX TRACKING operation.

coupled \( xx', yy' \) motion is not analyzed. When NH, NV, and NHV are all zero, the program prints the centroid positions only.

\[ dx \ dx' \ dy \ dy' \]

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

\[ \beta_x, \ldots \eta_y' \]

Initial values used in the twiss function computations.

\begin{itemize}
  \item \textbf{Nener} number of energies for which the analysis is done (maximum 15).
  \item \textbf{Ncoef} number of coefficients used in the Taylor series expansion as a function of momentum (max 6).
  \item \textbf{e1} \ldots \textbf{eNener} momentum values in the form \( (p - p_0)/p_0 \)
  \item \textbf{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.
  \item \textbf{LIST} a set of pairs of numbers each of which indicate the beginning and end position (in the order list of machine elements) of each of mlocat intervals in which printing takes place. List may contain up to mmlist numbers (set at 40 initially)
\end{itemize}

3.9 EFFECTIVE RMATRIX OPTION

This option allows the matrix-transport operations (BEAM, MACHINE, and MATRIX) to use the matrices computed by a previous RMATRIX operation in their calculation, rather than the standard element matrices. This allows the user to preserve the state of the machine with misalignments and errors, and study its linear-level optics in detail. This option is only valid after an RMATRIX calculation.

Input format:

\begin{verbatim}
EFFEctive rmatrix option...(up to 80 characters)
nopt,
\end{verbatim}

Parameters

\begin{itemize}
  \item \textbf{nopt 0} Standard matrices are used, misalignments and errors are ignored.
  \item \textbf{1} Effective R-matrices are used in optics operations.
\end{itemize}

3.10 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:

\begin{verbatim}
GENEration of particles
nopt \( \sigma_1 \ldots \sigma_6 \) scale npart
\( x_0, x'_0, y_0, y'_0, \delta_0 \)
\end{verbatim}

Parameters

\begin{itemize}
  \item \textbf{nopt 0} Standard matrices are used, misalignments and errors are ignored.
  \item \textbf{1} Effective R-matrices are used in optics operations.
\end{itemize}
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 \( \sigma \) 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.

4 the particles are generated according to a grid in up to six dimensions. This option is still somewhat experimental.

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

\( \sigma_i \)

For nopt=1 or nopt=3, the number of sigmas above which the gaussian distribution is truncated for each of the six variables \( x, x', y, y', l, \delta \). The beam is defined by a previous BEAM operation which is assumed to define the one sigma distribution. For nopt=4, grid is generated with particles evenly spaced from \(-\sigma_i\) to \(\sigma_i\). Setting \( \sigma_i \) to zero indicates that no grid is to be generated in the \( i \) dimension (ie, \( x_i \) is set to the centroid coordinate specified for \( i \) for all particles).

scale scales the beam size by the given factor.

npart number of particles to be generated. If NPART < 0, the existing particles will be kept and abs(NPART) new particles generated and added. As of version 2.8, particle tables are dynamically allocated and there is no limit in DIMAD to the number of particles which may be generated and tracked.

\( x_0...\delta_0 \)

centroid coordinates around which the beam is generated.

3.11 GEOMETRIC ABERRATIONS IN MULTIPLE TURN OPERATION

This operation traces particles that are placed on ellipses with nominal emittances \( \epsilon_x(i), \epsilon_y(i) \) for many turns. It then fits an ellipse to the output points obtained. From this fitted ellipse it determines the average values for \( \beta_x, \alpha_x, \beta_y, \alpha_y, \nu_x, \nu_y, \epsilon_x, \) 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
xc0,xc0',yc0,yc0',ener
ncase,nturn,njob
nplot,nprint
\epsilon_x(1), \epsilon_y(1)

..............
\epsilon_x(ncase), \epsilon_y(ncase)
anplprt
```

Parameters:

\( \beta_x, \alpha_x, \beta_y, \alpha_y \)

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 \( xc0 \) to \( ener \) coordinates and momentum of the closed orbit around which the aberrations are to be computed. When \( \beta_x = 0 \) these parameters are obtained from a previous movement analysis operation. Values corresponding to the first energy are used.
ncase    number of cases analyzed (maximum 10)
nturn    number of turns for tracing (maximum 100)
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.
\( \epsilon_x(i), \epsilon_y(i) \)    ncase values for the chosen nominal emittances in x and y using the
                                unit mm-mrad \( (E-06 \ m\text{-rad}) \)
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.

3.12 HARDWARE VALUES LISTING OF MACHINE

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

Input format:
HARDware layout and element parameters..(up to 80 char)
E s x y z \( \theta \ \phi \ \psi \) conv mprint [list]
or
E

Parameters:
E      momentum \((\text{GeV/c})\) used for computation of field values. When only
one data is entered only the hardware values are printed, the layout
printing is left out.
s x y z coordinates of starting point in some absolute reference coordinate sys-
tem. The coordinate s is the length of arc along the reference traject-
cy. To justify the choice of the angles \( \theta, \phi, \) and \( \psi \), the axis z should
coincide more or less with the longitudinal axis of the beam. The angles
\( \theta, \phi, \) and \( \psi \) describe the motion needed to bring the absolute system of
reference in coincidence with the local system of coordinates. The local
system of coordinates is the system used by the program. Its z axis
is tangent to the reference trajectory. The x axis (uniquely defined
by the bends) is in the midplane of symmetry and points outwards
of the bend. The y axis completes the local right handed system of
reference. To bring the absolute system in coincidence with the local
reference system, one executes the following rotations (strictly in the
order indicated):
A rotation \( \theta \) around the y axis (positive when the z axis turns towards the x axis)
A rotation \( \phi \) 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 \( \psi \) (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 \( m\times list\) numbers (set at 40 initially)

### 3.13 HISTOGRAM PARAMETERS

This operation allows control over the parameters which are used in binning the beam for wakefield or luminosity-equivalent sigma calculations (see PART operation).

**Input format:**

HISTogram parameters...(up to 80 characters)

nbin centr nsigma,

**Parameters:**

nbin number of bins to use
centr center position for the bins
nsigma number of standard deviations to be included. If \( nsigma < 0 \), the binning occurs about the mean of the distribution, rather than about the position specified in centr.

**NOTE:** If the HIST command is not given, the default is to use a number of bins equal to \( \sqrt{\text{npart}}/2 \), centered at the average value of the co-ordinate which is being histogrammed, and binning out to \( \pm 6 \) standard deviations.

### 3.14 INTERACTIVE CONTROL OF LATTICE

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

**Input format:**

INTEractive control of ..(up to 80 char)

niopt nivar
name keyword (repeated nivar times)

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

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

3.15 LEAST SQUARE FIT

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

Input format:

LEASt square fit of ..... (up to 80 char)
nstep nit nvar ncond
βx αx ηx η'x
βy αy ηy η'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(k) npas
name(k) pkeyw(k) coef(k) for k = 1 to npas

Parameters:

nstep number of steps taken to approach final fit.
nit number of iterations used in final step of fit.
nvar number of independent variables(max:20).
ncond number of conditions to be met(max:20).
βx ... η'y initial values needed for the function computation. When βx value is entered as zero, then the program uses the βx ... η'y values computed in the last matrix operation preceding the present operation.
name(i) name of element with an independent parameter to be varied.
pkeyw(i) variable element parameter keyword
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 the values of the stable motion analysis of the total matrix in the same order as mentioned in paragraph 2.8. of the SIMP operation. Numbers 21 to 30 refer to $\beta_x \alpha x \eta_x \eta_x' \beta_y \alpha y \eta_y$ at the end of the machine. These values are computed from the initial values present in the second line of the input format. Numbers 31 to 40 refer to the values $\beta_x \nu y$ computed at the first fit point defined by the preceding SET Fit point operation. Numbers 1031 to 1040 refer to the difference between the values $\beta_x \nu y$ computed at the first and second fit point defined by the preceding SET Fit point operation. (i.e.: v2-v1) Numbers 41 to 61 refer to the beam values $\sigma_x \ldots \sigma_p$ and the $r_{ij}$ at the end of the machine. See operation BEAM for the meaning of these parameters and the order in which they appear. These values can only be fitted if a BEAM operation defining the beam values at the beginning of the machine has preceded the fitting operation. Numbers 71 to 91 refer to the same beam values computed at the first fit point defined by the operation SET Fit point. Numbers 1071 to 1091 refer to the differences of the same beam values computed at the first and second fit point defined by the operation SET Fit point. (i.e.: v2-v1) Numbers 93 to 98 fit the average chromatic errors for $\beta_x \alpha x \nu x \beta_y \alpha y \nu y$ as computed in the detailed chromatic analysis operation. A fit on these elements can only be done after a previous detailed chromatic analysis operation is done which serves to define the parameters needed for the computation. Selected numbers 110 to 666 specify matrix elements in the following fashion:

ij0 represents the first order matrix element $R(i,j)$.

ijk represents the second order matrix element $T(i,j,k)$ (as in the TRANSPORT notation).

Numbers 2001 through 2007 refer to $\sigma_x^*, \epsilon_x, \sigma_y^*, \epsilon_y, \sqrt{\sigma_{LEx}^2 \sigma_{LEY}^2}, \sigma_{LEx}, \sigma_{LEY}$ at the end of the machine as determined after tracking. In order to use these a set of particles must be set up either through the GENE command or by reading initial coordinates into the TRAC command. The "*" superscript indicates values at the waist closest to the end of the line, while the subscript $LE$ indicates luminosity-equivalent sigma (see PART operation). Note that these operations are computationally intensive.

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.

name1 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 name1.

coef(k) coefficient with which the BASE parameter (that of name1) is to be multiplied to obtain the value of the parameter of name(k).
3.16  LINE GEOMETRIC ABERRATIONS (ONE TURN COMPUTATION)

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

Input format:

LINEgeometric aberrations....(up to 80 characters)
\[ \beta_x, \alpha_x, \beta_y, \alpha_y, \]
\[ xc0, x c'0, y c0, y c'0, \text{ener} \]
\[ ncase, npart, ncoup \]
\[ npplot, nprint, mlocat, [\text{list}] \]
\[ \epsilon_x(1), \epsilon_y(1) \]
\[ \ldots \ldots \]
\[ \epsilon_x(ncase), \epsilon_y(ncase) \]

Parameters:

$\beta_x, \alpha_x, \beta_y, \alpha_y$  input values of the twiss parameters at the entrance of the line.
$xc_0$ ... 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)
$\epsilon_x(i), \epsilon_y(i)$  ncase values for the chosen nominal emittances in x and y using the unit mm-mrad (E-06 m-rad)

3.17  LOGBOOK SET

Selects values which are computed during a PARTicle analysis option for output to a flat, uncommented file for ease of plotting. At present, the output of logbook values goes to unit 24.

Input Format:

LOGBook set... (up to 80 char)
nitem
repeat the following nitem times
\[ \text{item} \]

Parameters:

nitem  number of values per PART operation which are to be written to the logbook. If nitem = 0, logging is turned off.
Index number for the desired value. Values will be written to the logging file in the order in which they appear in the LOGBOOK command.

Numbers 1 to 6 correspond to the beam average position in $x$, $x'$, $y$, $y'$, $l$, $\delta$, respectively.

Numbers -1 through -6 correspond to the RMS size of the beam in $x$, $x'$, $y$, $y'$, $l$, $\delta$, respectively.

Numbers 11 through 66 correspond to the beam sizes and normalized correlation coefficients, $\sigma_i$ and $r_{ij}$ from the beam matrix.

Numbers -11 through -66 correspond to the true sigma matrix elements, $\sigma_{ij}$.

Numbers 101, 102, 201, 202 correspond to the horizontal beam size at the nearest waist, the horizontal emittance, the luminosity-equivalent horizontal beam size at the end of the beamline, and the luminosity-equivalent horizontal beamsize at the nearest waist, respectively.

Numbers 103, 104, 203, 204 correspond to the vertical beam size at the nearest waist, the vertical emittance, the luminosity-equivalent vertical beam size at the end of the beamline, and the luminosity-equivalent vertical beamsize at the nearest waist, respectively.

Numbers 301, 302, 303 correspond to the RMS size along the semimajor axis of the beam, the RMS size along the semiminor axis of the beam, and the $xy$ tilt of the beam, respectively.

3.18 MACHINE AND BEAM PARAMETERS COMPUTATIONS

Computes $\beta$, $\alpha$, $\eta$, $\eta'$, $\nu$ 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

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

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

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 $\beta_x$ is zero, the first line of parameters must be given and the initial twiss parameters values will be those of the preceding matrix analysis.

Parameters:

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.

3 Transport-line chromaticities and one-turn growth in normalized emittances due to synchrotron radiation are calculated. For this option, if beamline energy has been set with a PHASE command the PHASE energy overrides the use of E1, E2, and dE.

DNU $dv$ value used for optimum luminosity computation.
NINT number of interaction regions
NBUNCH number of bunches
\[ \beta_x \eta_y \]

function values at starting point of lattice.

mprint

-2 no printing of results
-1 print final result only.
0 print all intermediary and final results.

n

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

list

beginning and end positions of each interval in which printing is done.

List is a set of pairs of numbers. List may contain up to \( m \times \text{list} \) numbers (set at 40 initially)

3.19 MAD PARTICLES

Converts the coordinates of all presently valid particles (ie, particles which have not been lost) to canonical coordinates \((x, px, y, py, -t, deltap)\), and writes the coordinate values to standard output in a format suitable for input into the MAD particle tracking module. This allows use of the DIMAD GENERATION operation to generate a beam with desired parameters, which can then be tracked through MAD.

Input Format:

MAD Particles... (up to 80 characters) noption

Parameters:

noption When noption=1, particle output is generated; when noption=0, no action is taken.

3.20 MATRIX COMPUTATION

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

Input Format:

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

NORDER MPRINT [LIST]

Parameters:

NORDER

1 first order matrix only is printed.
2 second order terms are also printed.
< 0 computation is done to order \(|\text{NORDER}|\). MPRINT must be > 0 and the program will print matrices of the beam line situated between and including the element pairs defined in LIST. When norder is -1 or -2 the format of the output is identical to the input format of the Gxxxxxxx element. When norder is -11 or -12 the computation is to order 1 or 2 and the format of the output the standard program output for matrices

MPRINT

-2 no printing of matrix.
-1 print matrix at end of machine only.
0 print all intermediary matrices plus final matrix.

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

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 \( n \times \text{list} \) numbers (set at 40 initially)
3.21 MCS SETUP

Initializes variables which permit simulation of multiple coulomb scattering in collimators. When this option is enabled, collimators are not treated as perfect absorbers but rather as elements which can cause transverse scattering and energy loss when encountered by beam particles. See also the command RADIation length, below.

**Input Format:**

```
MCS Setup ... (up to 80 char)
nopt δmin seed,
```

**Parameters:**

- `nopt` if nopt=1, MCS is simulated in collimators; otherwise, if nopt=0, any particle which is outside the defined aperture of a collimator is stopped.
- `δmin` Momentum cutoff for tracking: 0 ≥ δmin ≥ −1. A particle which, after MCS simulation, has a relative momentum less than this value is considered a stopped particle and is no longer transported.
- `seed` random-number generation seed for MCS.

3.22 MODIFICATION OF ELEMENT DATA

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

**Input Format:**

```
MODIfication of input parameters...(up to 80 char)
 n
 name pkeyw value
 name pkeyw value . . .
 name pkeyw value
```

**Parameters:**

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

3.23 MOVEMENT ANALYSIS

This operation finds closed orbits and analyses both stable and unstable motions for up to 15 different momenta. NOTE: no kicks simulating synchrotron oscillation (parameter T < 0 ) 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 e_{1,...,ener}
naplt δmin δmax dνmin dνmax dβmin dβmax ncol nlinc
```
Parameters:

- **nprint**: print action for the closed orbit information
  - 0: action after every element
  - n: action after every n turn

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

- **nanal**: no stability analysis is done, only the closed orbit is computed.
  - 0: no analysis
  - 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 \(\eta_p\) versus momentum. The reference momentum in the expansion is fixed at 0.01 (1 momentum can be changed by the CONStant definition operation.

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

- **l**: estimate of the coordinates of the closed orbit.

- **e\(_1\)...e\(_nener\)**: Momenta(dp/p) for which the analysis is performed.

- **naplt**: no plot of the Taylor expansion is required
  - 0: no plot is required
  - 1: a plot is required

- **\(\delta_{\min}\) \(\delta_{\max}\)**: min max of dp/p for the plot.

- **\(d\nu_{\min}\) \(d\nu_{\max}\)**: min max for \(d\nu\) (the first momentum serves as the reference to compute the tune difference \(d\nu\)).

- **\(d\beta_{\min}\) \(d\beta_{\max}\)**: min max for relative difference in betas.

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

### 3.24 OBSERVATION SET

Enables or disables computation of observables (centroid positions, second moments, and number of particles remaining) during tracking.

**Input format:**

```
OBSErvation set...(up to 80 characters)
nopt,
```

**Parameters**

- **nopt**: 0: no physical observables are computed.
  - 1: Beam centroid position and number of particles are calculated at every monitor and printed to the output file on every turn.
Beam centroid position, second moments, and charge are calculated at each INST, BLMO, SLMO, WIRE, IMON, PROF and printed to the output file on every turn.

Both monitor and instrument observables are computed and printed on every turn.

### 3.25 OUTPUT CONTROL

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

**Input format:**

```
OUTPut control
nopt
```

**Parameters:**

- **nopt**
  - 0: all output is suppressed (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.
  - 5: Printing of stopped-particle positions is suppressed entirely; a summary of the number of particles stopped at each element is printed instead.
  - 14: Same as 4. The fifth coordinate will then be the phase relative to an RF cavity instead of the path length.

**Notes:**

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

### 3.26 PARTICLE ANALYSIS AND CONTROL

This operation is destined to provide some analysis of particle distribution. In addition to the beam positions and the beam matrix at the end of a TRACking run, the output now includes the estimated beam sizes at the nearest waist (useful when the waist position does not exactly correspond to the end of the beamline), the horizontal and vertical emittances, and the luminosity equivalent beam sizes at the end of the line and the nearest waist. The “luminosity equivalent sigmas” are computed according to an algorithm which de-emphasizes the importance of particles which are far from the center according to an algorithm by Pantaleo Raimondi. Finally, the beam RMS sizes along its semimajor and semiminor axes are computed, as well as the tilt angle of the $xy$ spot. As of version 2.8, this operation has new options related to particle input, output, reinitialization and deletion.

**Input format:**

```
PARTICLE ANALYSIS AND CONTROL
nopt,
```

or

```
nopt npart $x$ $x'$ $y$ $y'$ $l$ $\delta$ (npart repetitions)
```

or

```
nopt nunit,
```
Parameters:

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

2. Restore all particle coordinates to their values at the time they were loaded or generated.

3. Add new particles; in this case, `nopt` must be followed by `npart`, thence by the 6 coordinates of the `npart` particles which are to be added to the ray table. If `PART` has previously been called with option 7, the new particle coordinates will be read from the specified particle-input logical unit rather than the command stream.

4. Delete all particles in the ray table.

5. Change the logical unit number used for reading particles into the ray table. By default, DIMAD attempts to read the particle initial coordinates from the command input stream; this option allows the user to specify a separate unit for all particle-read operations. If this option is used, `nopt` must be immediately followed by `nunit`, the desired logical unit number. If `nunit==0`, the particle input unit is switched to the command input stream.

6. Change the logical unit number used for writing particle coordinates. By default, DIMAD attempts to write particle coordinates to the echo file; this option allows the user to specify a separate unit for all particle-write operations. If this option is used, `nopt` must be immediately followed by `nunit`, the desired logical unit number. If `nunit==0`, the particle output unit is switched to the echo file.

3.27 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 supersede 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,
```

Parameters:

- `keyword`: 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 should use markers. 99 is the flag that terminates the sequence of names. 99 is followed by another keyword or by end. 
- `name`: followed by names of elements at which printing is to occur. The maximum number is 10. The names may contain the wild character * . `AB*` means all names starting with ab will produce printing.
type: the types are: drift, bend, quadrupole, multipole, gkick, collimator, rf cavity, sextupole, solenoid, monitor, quad sext, matrix, mtwiss, arbitrary. Two types may be defined.

3.28 RADIATION LENGTHS

Allows the user to set the thickness of collimators in radiation lengths, for use with the multiple coulomb scattering simulation (see MCS Setup, above).

Input format:

- RADIation lengths... (up to 80 characters)
- name radlen

Parameters:

- name Element identifier: either its name or its element type string (specified with the “TYPE =” keyword during standard input); if the latter, it must be enclosed in double quotes.
- radlen Thickness of the element in radiation lengths.

3.29 RMATRIX COMPUTATION (6X6)

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 misalignments. The program also provides the entrance and exit orbit displacements. They are needed to correctly use the matrix. Using the given emittances $\epsilon_1$, $\epsilon_2$ 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.

In addition, the RMATRIX operation computes the first- and second-order transfer matrices of each element at the closed orbit using a total of 33 particles, and stores these for future use if EFFECTIVE RMATs are chosen.

Input format:

- RMATrix.......................(up to 80 characters)
- $x_0$ $x'_0$ $y_0$ $y'_0$ $l_0$ $\delta_0$
- $dx$ $dx'$ $dy$ $dy'$ $dl$ $d\delta$
- norder mprint (nlist)
- nmopt
- $\epsilon_1$ $\epsilon_2$
- $\beta_x$ $\alpha_x$ $\epsilon_x$ $\beta_y$ $\alpha_y$ $\epsilon_y$

Parameters:

- $x_0$ ... $\delta_0$ initial coordinates of reference orbit. When $\delta_0 = 1$ then the coordinates are the coordinates of the closed orbit computed in a previous movement analysis. The values corresponding to the first energy are used.
- $dx$ ... $d\delta$ increments used to generate the six particles surrounding the reference orbit.
- norder order of the computation: 1 or 11. The order of the computation is 1 when norder = 11 the output is in the standard input format.
- mprint number of intervals wanted; when mprint is -1, the calculation is done for the full machine from the first element to the last.
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 emittances are not computed
\( \epsilon_1 \epsilon_2 \) Emittances present in each eigenmode used to compute the projected emittances on both the horizontal and the vertical planes
\( \beta_x \alpha_x \epsilon_x \) Parameters defining the horizontal projection of an injected beam
\( \beta_y \alpha_y \epsilon_y \) Parameters for the vertical projection of the injected beam

3.30 SEISMIC PERTURBATION SIMULATION
Sets transverse misalignments according to sinwaves 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
```

Parameters:

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

3.31 SET FIT POINT
Sets an intermediate fit point to be used with the least square fit operation.

Input format:

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

Parameters:

- 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.

3.32 SET LIMITS TO VARIABLES
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 is from 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,

Parameters:
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.

3.33 SET SYMPLECTIC OPTION ON

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

Parameters:
Option Determines the mode of tracking. This affects only the operations based on tracking and not those based on matrix analysis (MATRICES, 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 $x, x', y, y', al, \delta$
and using the canonical matrices.
2 Fast version of ray trace is done with the variables $x, px, y, py, -\tau, DE/E$ and using the canonical matrices.
3 Slow version of ray trace is done with the variables $x, x', y, y', al, \delta$
and using the canonical matrices.
4 Slow version of ray trace is done with the variables $x, px, y, py, -\tau, DE/E$ and using the canonical matrices.

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

Energy Energy of nominal particle (in GeV)

3.34 SHO VALUES OF CONSTANTS

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

SHO Values of the basic constants
no parameters are used for this operation.

3.35 SIMPLE FITTING

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

Input format:

SIMPle fitting ........(up to 80 char)

nstep nit nvar
name pkeyw(i) del(i) (i = 1 to nvar)
valf valf (i = 1 to nvar)
nasp
repeat the following nasp times
name pkeyw npas
name(k) pkeyw(k) mult(k) add(k) k = 1 to npas

Parameters:

nstep number of steps to reach the final stage.
nit number of iterations performed in the last step in order to refine the variable values.
nvar number of variables and conditions (max:10).
name name of element containing a variable.
pkeyw keyword of the parameter to be varied in the element.
del increment by which the variable is to be varied. This number is divided by 5 in every iteration of the last step.
valf the values to be achieved in the final step.
nasp total number of parameters to be associated to the parameter pkeyw of name1.
name(k) name of the element which has a parameter to be associated with name 1.
pkeyw(k) parameter keyword to be associated.
mult(k),add(k) multiplicative and additive constants which define the value of the associated parameter according to the following formula

\[ \text{parvalue}(k) = \text{mult}(k) \times \text{parval} + \text{add}(k) \]

where parval is the value of the parameter used in the element name1 and to which pkeyw(k) is associated.

3.36 SPACE CHARGE COMPUTATION

This operation computes the linear effect of space charge on the motion of individual particles. It only affects operations that use tracking. This operation MUST be used in conjunction with a gkick simulating synchrotron oscillation or with an rfcaivity. Particles at the centre of the bunch are affected by the maximum
tune shift and particles at the edge see a zero tune shift. The simulation is done by changing the effective
strength of each quadrupole according to the relative particle momentum position with respect to the mo-
mentum spread assumed to be present in the beam. This operation is introduced on an experimental basis
only. We recommend caution in the interpretation of the results.

Input format:
SPACe charge computation...(up to 80 char)
option dpmax dkxmax dkymax

Parameters:
option 0 sets the computation off, 1 sets it on.
dpmax The maximum dp/p present in the beam
dkxmax dkymax The maximum relative change of the x focusing and the y focusing
quadrupoles that will produce the required maximum space charged
tune shifts as determined by the theory of linear tuneshift produced by
space charge. The link between dkxmax and dkymax and the maximum
space charge tune shift is generally given by the natural chromaticities
of the machine.

3.37 TAPE FILE OUTPUT

Provides an imitation of a MAD Tape file output, with extensions described in [8]. This option operates by
changing the output formats of the BEAM, HARDWARE, and MACHINE commands.

Input format:
TAPE file output...(up to 80 characters)
nopt
nopt 0 standard DIMAD format output is generated by BEAM, HARDWARE,
and MACHINE
1 BEAM, HARDWARE, and MACHINE operations generate output in
MAD tape format.

3.38 THERMAL PHOTON SCATTERING

Sets options for the simulation of thermal photon scattering. The simulation is described in detail in [9].

Input format:
THERmal photon... (up to 80 char)
TEMP ENERGY PROB TURN,

Parameters:
TEMP Vacuum chamber temperature in degrees CENTIGRADE
ENERGY Beam energy in GeV. If the beam energy has been set by PHASE
command, the PHASE energy overrides the energy specified here.
PROB Scattering probability per particle per meter
TURN Number of turns for which scattering should be simulated.
3.39 TRACKING OF PARTICLES

Tracks unlimited numbers of particles around the machine. Tracking will terminate on the last turn requested if an element whose name is STOPTRAC is encountered.

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)

Parameters:

NPLOT 0 plot action after every element.
-1 no plot action.
n action occurs after n turns (used in conjunction with MLOCAT and LIST) at MLOCAT locations specified by LIST elements.

NPRINT 0 print action after every element.
-1 printing at beginning and end only.
-2 no printing occurs.
-3 printing at end only.
n action occurs after n turns (used in conjunction with MLOCAT and LIST) at MLOCAT locations specified by LIST elements.
NOTE: when nplot=-1 and nprint=-2 then mlocat and list do not appear. MLOCAT and LIST are the same for plot and print.

NPART number of particles traced.
< 0 abs(npart) particles are added to particles already present from previous operation.
0 existing particles kept - none added.
> 0 previously used particles deleted. NPART new particles introduced.

NTURN number of turns to be traced.

x ... δ particle data for NPART particles. If PART operation with option 7 has been used, the particle data may be omitted but will be read from the specified particle-input unit.

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 beginning and end position (in the order list of machine elements) of each of MLOCAT intervals in which printing takes place. List may contain up to mxlist numbers (set at 40 initially)

NGRAPH 1 plot x,x' plane
2 plot y,y' plane
3 plot x,y plane
4 plot all planes
11 - as above but the graphs are accumulated and the plot is printed at the end.
accumulates the $l, \delta$ or the $\phi, \delta$ plots where $l$ is the pathlength coordinate of the particles, $\phi$ 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), and $\delta$ 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 guarantees 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 $E \phi$ plot identical to that of produced by the ngraph value of 16.

XMIN ... YMAX limits for the plotting windows.

ALMIN ... DELMAX limits for the plotting windows for the cases NGRAPH=15 or 16. They 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 number of lines to be used in plot matrix.
When linear acceleration structures (LCAVs) are present in a beamline, DIMAD computes the beam energy at each element in two ways. The design, or ideal, energy profile is computed using the energy gain of each cavity, its phase, the bunch charge, and the ELOSS parameter. This is done at DIMAT initialization, or when the cavity phases are set (either by the PHASE operation or by least-squares fitting with phases as independent variables). The real energy at each location is computed during tracking or matrix operations (TRAC, MATR, MACH, BEAM, etc.), and includes errors in phases and voltages, or changes in the bunch charge since the last PHASE operation. If the ideal and real energies are not equal, the $K_1$ and $K_2$ factors of quads, sextupoles, and quadsexts are scaled by a factor of $E_{\text{ideal}}/E_{\text{real}}$ during tracking (though this is not done during MATRIX, MACH, or BEAM operations).

During tracking the energy gain of each particle is calculated using the cavity phase and voltage, the longitudinal position of the particle, the sinusoidal variation of energy gain with $z$, and the longitudinal wakefield (if enabled). The difference between the particle’s actual energy and the real energy at that point is stored as the sixth particle coordinate.

### 4.1 CHARGE SET

This operation sets the bunch charge (in units of the fundamental electron charge) for use in calculations regarding wakefields or the ELOSS cavity parameter.

**Input format:**

```
CHARge set...(up to 80 characters)
npart,
```

**Parameters:**

- `npart` is the number of particles per bunch.

### 4.2 ENERGY DISPLAY

This operation displays the current values of $E_{\text{ideal}}$ and $E_{\text{real}}$ at selected locations down the beamline, along with energy gains and phases of LCAVs.

**Input format:**

```
ENERgy display...(up to 80 characters)
nopt,
```
Parameters:

- nopt is the display option
  - 0: no display is performed
  - 1: initial and final energies are displayed
  - 2: displays initial and final energies, as well as the phase and DELTAE of each cavity and the ideal and real energies at the entrance of each cavity
  - 3: as in option 2, except that the display is for all elements in the beamline

4.3 PHASE LINAC CAVITIES

This operation adjusts phases and voltages of cavities according to a desired injection and extraction energy, and a desired phase profile along the beamline. This option allows the user to set phases for each cavity independent of the phases entered in the MAD input section; each cavity may have a unique phase, even if all the cavities are represented by a small number of distinct MAD elements (i.e., two or three LCAVs defined in MAD can be used to stand for up to MX_LCAV=5000 cavities, each with a different phase). These phases are used unless a LEAST operation is performed with cavity phases as a parameter to be fitted; in this case DIMAT reverts to using the phases defined in the MAD input section.

Note that the PHASE command can be used to set the beam energy of a line which does not accelerate (transport line, storage ring, or bunch compressor). In this case the Nregion parameter should be set to zero. Once the energy at each element is set, beam emittances in a BEAM command can be specified as normalized emittances. The energy set by the PHASE command will take be used, if set, to calculate the beam energy at RFCAVs, and to set the nominal energy in synchrotron radiation operations. The energy set by the PHASE command also overrides energies entered in the following operations: BGAS, MACHINE (NLUM option 3 only), THERMAL, SYNCHROTRON RADIATION.

Input format:

PHASE linac cavities...(up to 80 characters)
Einit Efinal
Nregion
φ(i) E(i) for i=1 to Nregion-1
φ(Nregion)
ndisp,

Parameters:

- Einit is the injection energy (GeV)
- Efinal is the final energy (GeV)
- Nregion is the number of phase regions.
- φ(i), E(i) cavity phase (always in degrees!) and endpoint energy for phase region i. When phasing is performed, all cavities from the point where E=E(i-1) to the point where E=E(i) are set to phase φ(i). Since E(i=Nregion) must be equal to Efinal, the last region requires only a phase specification.
- Ndisp is the display option (identical to the display option for ENERGY operation).

4.4 WAKEFIELD OPTIONS

This operation specifies the options desired for use of longitudinal and transverse wakefields during tracking. If either type of wakefield is selected for use, all the wakefield files of that type (longitudinal or transverse, or both) will be read at the time this operation is carried out.
Input format:
  WAKEfield options...(up to 80 characters)
  lopt topt,

Parameters:
  lopt  is the longitudinal wakefield option
         0  no longitudinal wakefields are to be used
         1  longitudinal wakefields are to be used in tracking. This is accomplished
             by longitudinally “binning” particles in a histogram, and computing
             the wakefield energy loss at each particle due to each bin. The parti-
             cles are re-binned if a bend magnet is encountered due to the nonzero
             momentum compaction.
  topt  is the transverse wakefield option
         0  no longitudinal wakefields are to be used. This is the only option at
             present, since transverse wakefields are not as yet simulated.
Chapter 5

OPERATIONS USED IN CONJUNCTION WITH MISALIGNMENTS AND ERRORS

As of Version 2.6 of NLC-DIMAD, element misalignments generated with the MISALIGNMENT and SET MISALIGNMENT command sequence are generated statically (in previous versions they were generated dynamically during tracking). Consequently, while the limit on the number of misalignments entered at one time is still set by MXMIS (300 in this version), the total number of misalignments implemented is now unlimited: users can use several subsequent MISA...SET MIS... commands. In addition, misalignments generated using these commands are additive. A new command, NEW MISALIGNMENTS, is provided to delete all existing misalignments.

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 though the trends will be similar.

5.1 ALIGNMENT FITTING

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

Input format:

ALIGNment fitting .....(maximum 80 characters)
   nstep nit nvar ncond nfit nopter
   βx αx ηx η′x
   βy αy ηy η′y
   x0 x′0 y0 y′0
   dx dx′ dy dy′
   nener ener(1) ... ener(nener)
   Origin
   name(i) keywd(i) del(i) for i = 1 to npar

When nfit equals 1 or 2 the following group applies
CORR
   name(i) pos(i) opt(i) param(i) del(i) for i = 1 to ncor

NOTE: ncor+npar=nvar
mon(i) pos(i) val#(i) value(i) weight(i) error(i) for i = 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=1 to mcorr
  nmon nskip
  name(i) val#(i) value(i) weight(i) error(i) for i=1 to nmon
end of group for nfit 3
nasp
repeat the following nasp times
name keywd npas
name(k) keywd(k) mult(k) add(k) k = 1 to npas

Parameters:
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.
  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 randonly 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 - 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...\eta_y' x_0 x'_0 y_0 y'_0 initial values of nominal orbit.
dx dy dx' 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).
enener values of the momenta (p-p_0)/p_0.
origin position used as current origin to position the correctors used later. This position is be specified by the name of an element.
name(i) names of the elements having parameters to be varied. npar=i=nvar
keywd(i) such elements can be used
del(i) 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 conjunction 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 par# of name1.

name(k) name of the element which has a parameter to be associated with name1.

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.

5.2 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

upen Kxxxxxxx Kxxxxxxx .... Kxxxxxxx

nsub sigma
Parameters

- npen: Number of penetration points
- Kxxxxxxx: 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.
- sigma: 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.
- sigma1: Angular sigma (in radians) of the systematic aim error
- sigma2: Angular sigma (in radians) of the random aim error

5.3 BLOCK MISALIGNMENT

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

Input format:

- BLOck misalignement ...........(maximum 80 characters)
- Name1 name2 dx dy dz dzr ddel
- Name1 name2 dx dy dz dzr ddel
- 99,

Parameters:

- name1 name2: Name of two gkick elements whose names uniquely define the beamline interval to be misaligned as a block.
- dx ... ddel: One sigma values of the random generation of the x, y, z offsets, the roll around the longitudinal axis, and the relative field offset. NOTE: only 10 distinct intervals are allowed

5.4 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 may be changed and/or whose setting may be changed to achieve corrections of closed orbits and/or beam size at the monitor locations. At the moment only dipoles with small bend angles can be used as correctors.

Input format:

- CORRector data definition .......(maximum 80 characters)
- Name 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.
5.5 ERRORS DATA DEFINITION

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

Input format:

ERRORs data definition ....(maximum 80 Characters)
Name Parameter value .... parameter value;
............
Name Parameter value .... parameter value;
99,

Note the semicolon ending each line defining errors for one element and 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 , and the value of the error , respectively.

5.6 MISALIGNMENT DATA DEFINITION

This operation defines the misalignments of different elements of the lattice. Up to 300 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,

Note the comma ending each line defining the misalignments The list is terminated with 99, as in the first element list.

Parameters:

Name Identifier of the element to be misaligned. The identifier can be either the element name, or it can be the element type (entered with the TYPE keyword); in the latter case the identifier must be enclosed in double quotes. The identifier can include a wildcard ('*') character.

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 displacements or angles). The program uses approximate formulae to set up the misaligned element. The parameter option can take the values 1,2,3 or 4 which determines 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.
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.

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 dm4/4 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.

5.7 NEW MISALIGNMENTS

This operation deletes all misalignments previously set with the SET MISALIGNMENT command.

Input format:

NEW Misalignments........(maximum 80 characters)
nopt,

Parameters:

ns  0  No action is taken
    1  existing misalignments are deleted.

5.8 REFERENCE ORBIT DISPLAY

This operation computes the orbit defined by the initial coordinates x x' y y' al delta (delta=(p-p0/p0)), and provides either a printout or a printer-plot display.

Input format:

REFErence ........(maximum 80 characters)
nprint sizex sizey
x0 x0' y0 y0' al δ
npos pos(1) .... pos(npos)

Parameters:

nprint controls display
    1  a printout is provided
    2  a printer-plot is provided
    11, same as above, the orbit is 4-dimensional closed orbit defined by the
    12  variables x,x',y,y'.
    21, same as above, the orbit is a six-dimensional closed orbit on the vari-
    22  ables x, x', y, y', al, δ.

sizex, sizey always needed. They define the boundaries between which the coordi-

nates x and y are plotted.

x0...δ 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),end of lattice
5.9 SEED

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

Input format:

SEED........(maximum 80 characters)
ns,

Parameters:

ns 0 a seed is generated by the program.
not 0 ns must be positive. The program insures ns is an odd number and prints the number used as the seed.

5.10 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,

Parameters:

Name Name of corrector element whose value is to be set
pos position of corrector
opt option number defining the kind of corrector
p1...p4 the four parameters used to define the corrector.

opt = 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)

opt = 1 the parameters have the same definition as above. The displacement dx and dy are imposed at the entrance of the magnet. The exit point of the magnet is assumed fixed. The operation of dy’ and ddel remain the same as above

opt = 2 As for option 1 the parameters keep their definition. This time the entrance is fixed and the displacements dx and dy are imposed at the exit of the magnet. In both cases 1 and 2 a momentum independent slope is computed and imposed on the magnet.

opt = 3 In this option the corrector acts as a pure dipole steering magnet. Parameter 1 is dxp and parameter 2 is dyp. Parameter 3 and 4 are not used. The angle kicks dxp and dyp are inversely proportional to the momentum of the particle (ie: dxp and dyp are divided by 1+delta where delta is the relative momentum of the particle traced.)
5.11 SET ERRORS OF ELEMENTS

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

Input format:

```
SET Errors ....(maximum 80 characters)
nopt
nerr
  name nint nb(1) nf(1) ....nb(nint) nf(nint) ,
  ........
  name nint nb(1) nf(1) ....nb(nint) nf(nint) ,
```

Parameters:

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

5.12 SET MISALIGNMENT OF ELEMENTS

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

Input format:

```
SET Misalignment .....(maximum 80 characters)
nopt
nmis
  name nint nb(1) nf(1) ....nb(nint) nf(nint) ,
  ........
  name nint nb(1) nf(1) ....nb(nint) nf(nint) ,
```

Parameters:

```
nopt  choice option for the random generators
  0   the elements are misaligned by the fixed values given in the MISA... operation. No randomness is introduced,
  1   The misalignment values are obtained by multiplying the values given in the MISA... operation by +1 or -1 randomly generated.
  2   A uniform distribution is generated having the rms values defined by the MISA... operation.
  3   A Gaussian distribution truncated above two standard deviations is generated with the rms value defined by the MISA... operation.
  4   A Gaussian distribution truncated above six standard deviations is generated with the rms value defined by the MISA... operation.
  11 - 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). If nmis < 0, the number of misaligned element types is given by |nmis|, and the misalignment algorithm will attempt to identify blocks of elements which should be misaligned together (split elements, elements with BPMs or correctors integrated into them, etc.) and give them a common misalignment.
```
name
name of the family of misaligned element. These names must match
the names given in the MISA... command, and if element type strings
are used they must be enclosed in double quotes.
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.

5.13 SHO CORRECTORS

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

Input format:
SHO Correctors
option ....

Parameters:
option determines the information to be printed out.
1 The rms, maxima and minima of the values are displayed
2 name This option prints the values 1 and 2 of the correctors with the label
name . These values are multiplied by the scale factor SIGFAC as
defined in Constant definition.
3 Under this option the values of the correctors are printed out in the
format accepted as input by the SET Correctors operation. This can
be used to set up an input file for a misaligned and corrected machine
which can be then studied without executing the alignment correction
procedures.

5.14 SHOERRORS

(not implemented yet)

5.15 SHO MISALIGNMENT

This operation displays the actual values of the displacement generated in previous operations.

Input format:
SHO Misalignment
nrange ............

Parameters:
nrange 0 then all misalignements are printed
> 0 then nrange intervals ni mi are used. The misalignements are printed in
these intervals

5.16 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. Note that in order to simulate longitudinal damping, one of options
10 through 13 must be used.
**Input format:**
SYNChrotron radiation.....(maximum 80 characters)
Energy option randomoption,

**Parameters:**

<table>
<thead>
<tr>
<th>Energy</th>
<th>Initial nominal energy in GeV. This is used only if the energy has not been set by a PHASE command.</th>
</tr>
</thead>
<tbody>
<tr>
<td>option</td>
<td>no synchrotron radiation effect is simulated.</td>
</tr>
<tr>
<td>0</td>
<td>synchrotron radiation loss is computed in every dipole. Particles lose that energy at the entrance and exit of the magnet.</td>
</tr>
<tr>
<td>1</td>
<td>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.</td>
</tr>
<tr>
<td>2</td>
<td>both the radiation loss and the emittance growth are simulate in the dipoles only.</td>
</tr>
<tr>
<td>3</td>
<td>synchrotron radiation is simulated in the dipoles as in option 2 to which is added a synchrotron quantum fluctuation in the quadrupoles.</td>
</tr>
<tr>
<td>4</td>
<td>synchrotron radiation is simulated in the dipoles as in option 3 to which is added a synchrotron quantum fluctuation in the quadrupoles.</td>
</tr>
<tr>
<td>5</td>
<td>synchrotron radiation losses and emittance growths are computed for each particle in dipole bends according to a more realistic simulation developed by Ghislain Roy [10].</td>
</tr>
<tr>
<td>10</td>
<td>as in 10 both radiation losses and growths are computed in bends, quadrupoles, sextupoles, quadsexts, and multipoles.</td>
</tr>
<tr>
<td>11</td>
<td>Only the emittance growths are computed in bends with the Ghislain Roy's simulation.</td>
</tr>
<tr>
<td>12</td>
<td>Only the emittance growths are computed in bends, quadrupoles, sextupoles, and multipoles using Ghislain Roy's simulation.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>randomoption</th>
<th>choice of the random generator, applies to above options 2 through 5.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>the random generator is binary + and - randomly affecting the energy spread creating the emittance growth.</td>
</tr>
<tr>
<td>2</td>
<td>the random generator is uniform</td>
</tr>
<tr>
<td>3</td>
<td>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.</td>
</tr>
<tr>
<td>11</td>
<td>- 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.</td>
</tr>
</tbody>
</table>
Bibliography


