

# The MAD Program

H. Grote, F. C. Iselin, and E. Keil, CERN, CH-1211 Geneva 23  
 J. Niederer, Brookhaven National Laboratory, Upton NY, 11973

**Abstract** The Methodical Accelerator Design program MAD for circular accelerators is in regular use for the design and modelling of LEP, LHC, PS and SPS at CERN, and in 10 to 20 other laboratories. It includes the following new accelerator physics features: (i) Machine imperfections, e.g. random and systematic errors in the alignment and excitation of the magnetic elements and read-out errors of the beam-position monitors, (ii) closed orbit correction using the MICADO algorithm, (iii) Lie-algebraic techniques for concatenating beam lines, for tracking trajectories and for higher-order orbit functions, (iv) systematic energy loss due to synchrotron radiation and its effects on the closed orbit and the orbit functions in all 3 degrees of freedom, (v) spin dynamics in SMILE style. The data describing the machine are held as data modules in a dynamically managed memory pool, including garbage collection and overflow onto external files. Graphics is interfaced to standard GKS and used for plotting orbit functions along the orbit and versus the momentum error, and phase space plots, Fourier spectra of particle orbits, etc. MAD is implemented on Cray, IBM, Nord, and VAX systems and on Apollo and Sun workstations.

## 1 User Interface

The accelerator structure and the computational commands are entered in a format-free input language which contains the standard language for describing accelerators and beam lines [1] as a subset. An accelerator or beam line is described as a *sequence of elements*, and by *parameters*. Parameters and elements are defined by statements such as

```
MU := PI/3
SM2 := SIN(MU/2)
LP := 100
PHI := 0.01
D: DRIFT,L=LP/4
B: MULTIPOLE,KOL=PHI/2
QFH: MULTIPOLE,K1L=2*SM2/LP, &
      K2L=4*SM2*SM2*QFH[K1L]/(LP*PHI*(1+SM2/2))
QDH: MULTIPOLE,K1L=-QFH[K1L], &
      K2L=-QFH[K2L]*(1+SM2/2)/(1-SM2/2)
```

Sequences are defined by statements such as

```
THINCELL: LINE=(QFH,D,B,D,2*QDH,D,B,D,QFH)
```

The above data may be recognized as the description of a FODO cell composed of thin elements, implemented with the MULTIPOLE command, with the quadrupoles and sextupoles superimposed, the quadrupole strengths chosen to obtain the phase advance  $\mu$ , and the sextupole strengths adjusted such that the chromaticity vanishes. The values of the element attributes may be arithmetic expressions. The language also contains powerful features to describe sequences which are not completely periodic. Sequence definitions can be nested to any level, thus providing

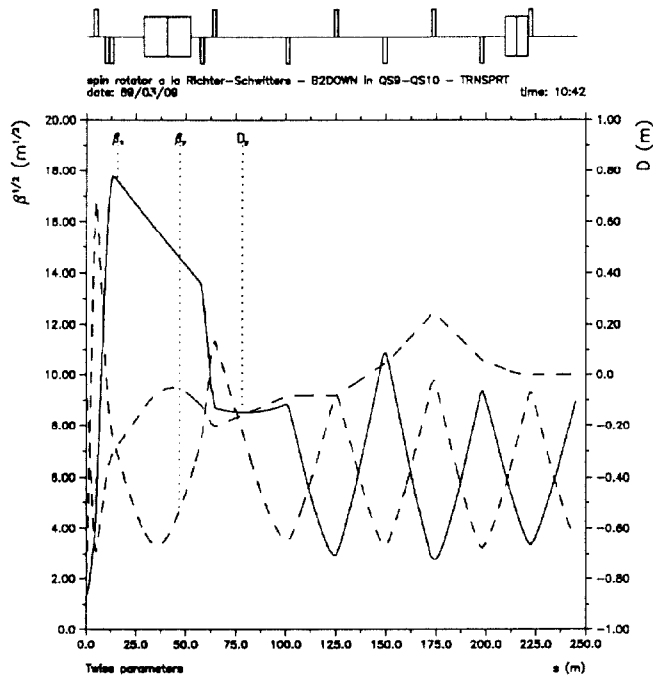


Figure 1: Orbit functions for spin rotator in LEP

very compact descriptions of large machines with much repetition. As an example let us consider the description of the complete CERN SPS machine. All items not defined below are simple elements. The total number of elements in the SPS is 1188.

```
SPS: LINE=(6*SUPER)
SUPER: LINE=(7*P44,INSERT,7*P44)
INSERT: LINE=(P24,2*P00,P42)
P00: LINE=(QF,DL,QD,DL)
P24: LINE=(QF,DM,2*B2,DS,PD)
P42: LINE=(PF,QD,2*B2,DM,DS)
P44: LINE=(PF,PD)
PD: LINE=(QD,2*B2,2*B1,DS)
PF: LINE=(QF,2*B1,2*B2,DS)
```

Commands for calculations are expressed in a similar way. Several examples are shown below.

## 2 Graphics

Recently, the emphasis for the interface between MAD output and the user has changed from printed tables to graphics. CERN has accepted the ISO standard GKS (Graphics Kernel System) for the type of plots required by MAD. Since GKS contains only the most basic plotting functions such as drawing lines or symbols, one of the authors has written, in FORTRAN 77, a stand-alone interface package called GXPLOT that provides higher level functions such as curve smoothing, automatic scaling, drawing axes with labels and text, and the like. This package is part

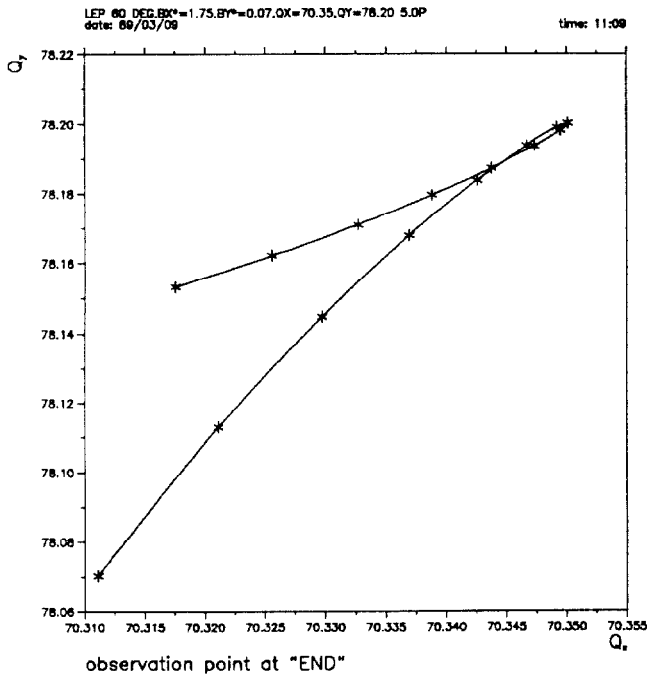


Figure 2: Working line for LEP

of MAD and should be rather easy to implement at sites where a GKS version exists. Several examples of graphics output are shown below.

Fig. 1 shows a plot of  $\beta_x, \beta_y$  and  $D_y$  for half a spin-matched Richter-Schwitters spin rotator [2] for an experimental interaction point in LEP, obtained with the commands:

```
USE, PERIOD=SEC4M
TWISS, BETX=1.75, BETY=0.07, DPY=-0.155301, SAVE
PLTDAT: SETBPLOT, SQRT, DMIN=-1, DMAX=1
BPLOT, BETX, BETY, DY, DATA=PLTDAT, RANGE=#1/117
```

Fig. 2 shows the working line for LEP, i.e. the vertical tune  $Q_z$  vs. the horizontal tune  $Q_x$ , as an example for plotting any one against any other of the parameters  $\alpha_x, \alpha_y, \beta_x, \beta_y, D_x, D_y, D'_x, D'_y, \mu_x, \mu_y$  at a given point along the beam line and against  $\Delta E/p_0c$ . This plot is obtained with the commands:

```
USE, PERIOD=LEP
TWISS, SAVEDP, DELTAP=-0.014:+0.014:0.002
DPLLOT, HAXIS=MUX, VAXIS=MUY
```

Fig. 3 shows the result of tracking several particles through a lattice consisting of a simple FODO cell with phase advance  $\mu$  and a single sextupole. It is well known [3], [4] that this arrangement corresponds to the famous Hénon map [5] if the motion is restricted to the horizontal plane.

### 3 Data Structure

MAD is written in FORTRAN 77, a language that offers only common blocks to set up data structures. Applications such as MAD, but as well high energy physics analysis programs, require a dynamic management of the memory resources since it is normally out of question to allocate the maximum possible sizes to all common block arrays that the program may be using, even on computers with "virtual" memory.

Dynamic memory management uses one or several big "pools" of memory where the executing program module can reserve areas for temporary or permanent usage. The temporary space can be returned on leaving the module, and re-used by other modules. If the available memory fills up nevertheless, an overflow mechanism stores certain areas on random access disk files, or some similar storage medium, such as the SSD (=Solid-state Storage Device) on the Cray computer.

Up to now, MAD used a memory manager developed specifically for this application and nowhere else in use. This had the advantage that it could be tailored to the needs of MAD, in particular to the use of double precision variables on 32-bit computers. The disadvantage lies in its uniqueness: since such a system is rather complicated, it unavoidably contains errors which are not found quickly by other applications since there are none, but tend to show up over the years. Furthermore, the scope of MAD kept expanding, and a more powerful memory manager became desirable.

In view of this, MAD is currently being re-written at CERN and now uses a rather recent memory manager ZEBRA [6] developed there. ZEBRA is very powerful in that it offers structures such as trees and linear chains which is of great help to the programmer, and is used by many experiments for the analysis of their data. However, the casual user of MAD, wanting to add a few routines to the existing program hooks would have to learn this system before he could do so. In order to avoid this inconvenience, the new version of MAD contains standard interface routines that present the variables in the familiar form of subroutine arguments or common blocks, and allow both to retrieve and to store information in MAD without knowing anything about ZEBRA.

The data describing the accelerator, and the commands to be executed are stored in the form of ZEBRA data banks, and linked by pointers. Data can be deleted at any time and the memory freed can be reclaimed. Nevertheless the complete program is written in standard FORTRAN 77.

### 4 Definitions in MAD

MAD uses full  $6 \times 6$  matrices to allow coupling effects to be treated, and the canonical variable set  $x, p_x/p_0, y, p_y/p_0, -c\Delta t, \Delta E/p_0c$ , as opposed to other programs most of which use the set  $x, x', y, y', -\Delta s, \delta$ . Like the program MARYLIE [7], MAD uses the relative energy error  $\Delta E/p_0c$  which is related to the relative momentum error  $\delta$  by  $\Delta E/p_0c = \beta \Delta p/p_0 = \beta \delta$ .

The choice of canonical variables in MAD leads to slightly different definitions of the orbit functions. In MAD the Courant-Snyder invariants [8] take the form  $W_x = \gamma_x x^2 - 2\alpha_x x p_x + \beta_x p_x^2$ . Comparison to the original form  $W_x = \gamma_x x^2 - 2\alpha_x x x' + \beta_x x'^2$  shows that the orbit functions cannot be the same. A more detailed analysis, using  $x' = p_x/(1 + \delta)$ , shows that all formulas can be made consistent by defining the MAD orbit functions as

$$\beta_{xM} = \beta_x(1 + \delta), \quad \alpha_{xM} = \alpha_x, \quad \gamma_{xM} = \gamma_x/(1 + \delta). \quad (1)$$

For constant  $\delta$  along the beam line the original orbit functions can therefore easily be recovered. In a machine where  $\delta$  varies along the circumference, e.g. in a linear accelerator or in an  $e^+e^-$  storage ring, the definition of the Courant-Snyder invariants must be generalized. The MAD invariants have the advantage that they remain invariants along the beam line as they should.

BEAM, PARTICLE=POSITRON,ENERGY=50,RADIATE  
 NORMAL, ORDER=6,RFCAVITY=RFC,DELTAP=-.01:+.01:.005

The phase of RFC is adjusted to obtain the correct revolution time, and the starting value of  $\Delta s$  is adjusted to make the initial and final values of  $\Delta E/p_0c$  the same. The variable DELTAP is now interpreted as the *average* value of  $\Delta E/p_0c$  along the orbit.

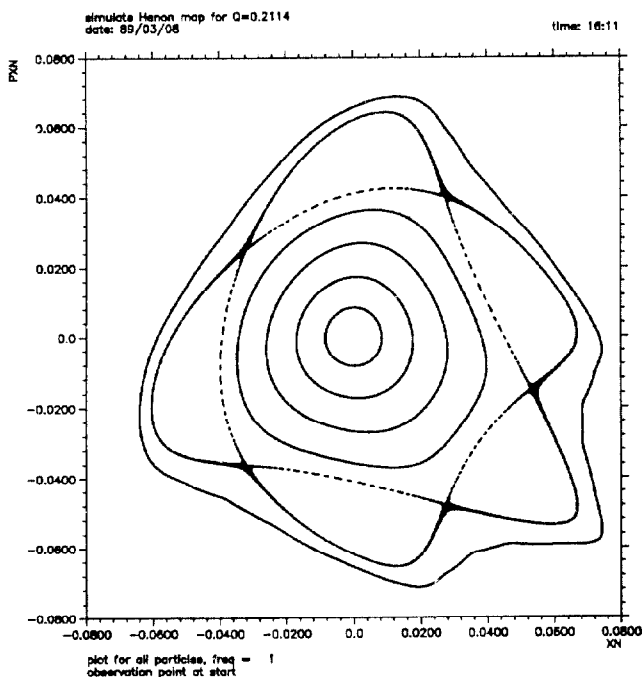


Figure 3: Tracking of the Hénon map

## 5 Alignment and Excitation Errors

It is very easy in MAD to specify alignment and excitation errors for the elements [9] which are subsequently taken into account in the calculation of the closed orbit, the tunes, orbit functions, etc. The closed orbit distortions caused by these errors may be corrected by the MICADO algorithm [10]. The commands needed for tilting the dipoles named B6, and the quadrupoles of type MQ, with Gaussian distributions truncated at 2.5 standard deviations, to misalign the same quadrupoles in the  $x$  and  $y$  direction, to define readout errors on the monitors MHV, and to define relative field errors on the dipoles named B6, and to correct the closed orbit are shown below:

```
USE, LEP
EALIGN, B6,DPSI=24E-5*TGAUSS(2.5)
EALIGN, TYPE=MQ,DX=.14E-3*TGAUSS(2.5), &
DY=.14E-3*TGAUSS(2.5),DPSI=24E-5*TGAUSS(2.5)
EALIGN, MHV,MREX=.3E-3*TGAUSS(2.5), &
MREY=.3E-3*TGAUSS(2.5)
EFIELD, B6,DBLR=12E-4*TGAUSS(2.5)
MICADO, ITERATE=2,ERROR=2E-3,NCORR=96
```

## 6 Synchrotron Radiation Losses

The systematic synchrotron radiation losses in dipoles are calculated and used when the RADIATE flag in the BEAM command is true. By convention, MAD assumes that positively charged particle run forwards and negatively charged particles run backwards through the beam line. Positively (negatively) charged particles lose (gain) energy in dipoles and gain (lose) energy in RF cavities. The NORMAL command finds the closed orbit in six dimensions, tracks and prints the eigenvectors along it, and prints the fractional tunes for all three modes. Internally, it uses Lie-algebraic maps [11], concatenated up to  $2 \leq \text{ORDER} \leq 6$ . The NORMAL command is used in statements such as:

## 7 Polarization

The program SMILE [12] allows to calculate the equilibrium polarization for electron or positron storage rings according to the Derbenev-Kondratenko formula. The calculation is based on linear orbit dynamics. However, the spin motion—as described by the Thomas-BMT equation—contains non-linear terms in the orbital variable  $z$ . In this way the influence of higher order synchrotron sideband resonances on the polarization is taken into account.

A first version of SMILE was implemented in the previous MAD version. The coupling of the two programs was, however, loose in the sense that the SMILE part used only the element description from MAD, but performed otherwise its own orbit calculations before calculating the spin motion, exactly as in the stand-alone version. This implementation was never distributed.

The current plans are to integrate the new and improved SMILE version with the new MAD version under preparation. This time, MAD will perform the orbit calculations for SMILE, of which only the spin part will be left.

## 8 Final Remarks

Since earlier presentations of the Methodical Accelerator Design program (MAD) [13], the physics modules listed in the abstract have been added. MAD offers a user interface and a data structure which should make it easy to add program modules to perform new physical calculations and to interface them to the data structure. The following documentation exists: A User's Reference Manual intended for the user of the standard version of MAD as distributed by the CERN Program Library, and a Programmer's Reference Manual for programmers who wish to write their own modules in the MAD framework. A MAD Newsletter is published about every six months.

## References

- [1] D.C. Carey and F.C. Iselin, Proc. 1984 Summer Study on the Design and Utilization of the Superconducting Super Collider, Snowmass, Colorado (1984) 389.
- [2] A. Blondel and E. Keil, CERN 88-06 (1986) 250.
- [3] L. Michelotti, FERMILAB Pub-86/104 (1986).
- [4] S.G. Peggs and R.G. Talman, Ann. Rev. Nuclear and Particle Sci. 36 (1986) 287.
- [5] M. Hénon, Q. Appl. Math. 27 (1969) 291.
- [6] R. Brun and J. Zoll, CERN Program Library code Q100.
- [7] A. Dragt, F. Neri et al., These Proceedings.
- [8] E.D. Courant and H.S. Snyder, Ann. Phys. 3 (1953) 1.
- [9] E. Nordmark, CERN/LEP-TH/85-15 add.
- [10] B. Autin and Y. Marti, CERN ISR-MA/73-17 (1973).
- [11] L. Healy, Doctoral Thesis, Maryland Univ. (1986), and private communication.
- [12] S.R. Mane, Phys. Rev. A36 (1987) 120.
- [13] F. C. Iselin, E. Keil and J. Niederer, Conf. Computing in High-Energy Physics, Amsterdam (1985) 316.