Description and Matching of Storage Rings in MAD

F. C. Iselin

Abstract

In MAD Version 8 there are problems with a correct description of the two beams in a storage ring. This note is an attempt to define changes to be implemented in MAD Version 9 to deal with these problems. The proposed solution is based on an object-oriented approach using the C++ language.

Geneva, Switzerland
January 25, 1995
1 The Problem

Version 8 of MAD poses some problems for a correct description of a storage ring:

1. The beams in the two rings rotate in opposite directions, and may contain particles of the same or of opposite charge. In each ring the sign of the main dipole field must be chosen such that the deflection occurs in the proper direction. For particles of the same charge the bend fields in the two rings must thus point in opposite directions.

2. Equivalent positions in the two rings should have the same azimuth to allow easy correlation between equivalent geometric positions in the two rings. To this effect the azimuth \( s \) should run in the same geometric direction (clockwise) in each ring. In one ring the beam then runs from \( s = 0 \) to \( s = C \), the machine circumference, while in the other ring it runs from \( s = C \) to \( s = 0 \). However, in MAD-8 conventions the beam elements would always be listed in beam order.

3. The two rings have all or part of the magnets in common. When the particles in the two beams have the same sign of charge, the fields in the common parts have opposite action on the two beams. Version 8 of MAD does not allow such a switching of sign, and the sign of magnetic fields is related to the sign and direction of the circulating particles.

4. In exceptional cases the two rings may be laid out for different reference momentum. Version 8 of MAD does not allow a renormalisation of multipole coefficients.

5. The phase lag in RF cavities is defined with respect to a reference particle running around the machine. In a storage ring there will be two such particles travelling in opposite directions, and it is not clear to which the phase lag should be related. If a cavity acts on both beams, the situation is further complicated.

6. For twin-bore magnets the field errors are normally measured with respect to a reference whose azimuth runs in the same direction in both bores. Entry of errors to MAD would be greatly simplified, if the magnets were listed in the same geometric order for both rings and the if field errors followed the same sign conventions as described above for the main fields.

2 An Object Oriented Solution

The future Version 9 of MAD, written in C++, makes a fully object-oriented solution possible. Data entry is the same as for MAD version 8, but with minor enhancements. Each ring is described as a C++ object, and so is every beam and every transfer map in the machine.

For matching, variable element parameters are defined as differential algebra objects, which automatically makes the transfer maps depend on those parameters. It is then possible to write very simple C++ routines which allow to match almost any global property of the machine in terms of these variables.

Data entry follows natural conventions:

1. Both rings have their elements listed in clockwise order.

2. The two rings are described in terms of beam lines and/or sequences like in MAD version 8. It is important that the common parts of the rings are defined as separate objects (a line or a sequence). In this way they can be used for the definition of both rings. It is however permissible to split those objects further, as shown in the example below.

3. The magnitudes of all multipole coefficients \( K_n \) follow MAD-8 conventions. Their signs corresponds to the magnetic fields seen by a positive particle, circulating in clockwise direction. A positive dipole component \( K_0 \) represents a magnetic field in the positive \( y \) direction. A positive multipole component \( K_n \) with \( n > 0 \) corresponds to a magnetic field in positive \( y \) direction for positive \( z \) values. The magnitude is scaled with a positive design momentum \( p_0 \). If both rings are designed for the same reference momentum, \( p_0 \) represents this reference momentum. Otherwise, any value can be chosen for \( p_0 \). The beam descriptions must then contain the design momentum as a multiple of \( p_0 \) for each beam.

4. Electrostatic fields are given with their actual magnitude in MV. A positive electric field points towards positive values of the corresponding coordinate.

5. If an RF cavity is common to both beams, and both beams see the same phase lag, it can be placed in a common section. Otherwise, it should be placed in a separated section. This allows the phase lag to be related to the single beam traversing it. A positive voltage in an RF-cavity is always accelerating, and the usual conventions for the phase lag apply. This convention needs further elaboration.
6. A solenoid field is positive, if it points towards increasing azimuth.

7. Field errors follow the same sign conventions.

To summarize the sign conventions, all field components are positive if they point towards increasing coordinate values. The conventions adopted make it possible to use the same data as before for single-ring machines.

The separation between the two rings is done in these lines using the following mechanisms:

1. Magnetic separation dipoles are specified as special elements SPLITTER, which have a straight reference orbit, but have opposite effects on the two beams. It is not permitted to have elements which change the direction of the reference (like SBEND and RBEND) in those regions.

2. All elements in the common regions are referred to the straight axis of the interaction region, but may be traversed off-centre by both beams.

3. At the junction of the common region to the separated arcs the reference orbit must be displaced by a special element SHIFT to make it coincide with the reference orbit of the separated arcs. The sign of the horizontal or vertical shift is the displacement of the reference system with respect to a global system when travelling in clockwise direction.

Each beam is described by a “beam object”. A beam object contains at least the following data:

- Direction of travel, clockwise or anti-clockwise,
- Particle charge, expressed as a signed multiple of elementary charges,
- Particle name,
- Particle mass in GeV,
- Reference momentum as a multiple of $p_0$, to allow proper scaling of forces,
- Bunch flag, true, if beam is bunched,
- Current per bunch in A, if beam is bunched,
- Radiation mode. This can express the conditions radiation on/off, damping on/off, and quantum excitation on/off.

A beam object may also contain some derived quantities:

- Particle energy in GeV,
- Relativistic $\gamma$ and $\beta$,
- Particle momentum in GeV/c,
- Number of particles per bunch,
- Emittances for the three modes,
- Tunes,
- Chromaticities,
- Momentum compaction,
- Transition energy,
- Machine circumference,
- Revolution frequency,
- Energy loss per particle and turn,
- Classical particle radius,
- Damping partition numbers.
3 Input Example

This section gives an extremely simplified example. To make the discussion easy it assumes full symmetry between quadrants and has been further simplified to the extreme. It is meant to show the principles and does of course not model any real design.

! ========== arc definitions

! coordinate shifts, assuming a separation of 2*180 millimetres
SHP:  SHIFT, DX=+0.18
SHM:  SHIFT, DX=-0.18

! arcs and coordinate shifts
ARCI: LINE = (SHM,...,SHP) ! inner arc
ARCO: LINE = (SHP,...,SHM) ! outer arc

! ========= interaction region definitions
! interaction point
IP: MARKER

! splitter magnets, assuming an angle of 18 mrad
SPP: SPLITTER, ANGLE=+0.018
SPM: SPLITTER, ANGLE=-0.018
DSP: DRIFT, L=10

! separation of the two beams
! position  interaction region
SPLIT1L: LINE = (SPM,DSP,SPP) ! downstream odd
SPLIT1R: LINE = (SPP,DSP,SPM) ! upstream odd
SPLIT2L: LINE = (SPP,DSP,SPM) ! downstream even
SPLIT2R: LINE = (SPM,DSP,SPP) ! upstream even

! interaction regions
! position  interaction region
IR1L: LINE = (SPLIT1L,...) ! downstream odd
IR1R: LINE = (...),SPLIT1R) ! upstream odd
IR2L: LINE = (SPLIT2R,...) ! downstream even
IR2R: LINE = (...),SPLIT2R) ! upstream even

! ========= definition of the two main rings
RING1: LINE = (4 * (IP, IR1R, ARCI, IR2L, IP, IR2R, ARCO, IR1L))
RING2: LINE = (4 * (IP, IR1R, ARCO, IR2L, IP, IR2R, ARCI, IR1L))

The two lines IRxx contain all elements common, and ARCI and ARCO contain all elements which are not separated for the two rings. To show that the common parts can be split into several parts, the elements separating the two beams have been put into their own beam lines SPLITxx.

4 Sign Conventions for Particle Tracking and Optics

Here comes an excerpt of the code to be used:

// declare the two beam objects
// the parameters specify direction of travel, charge, etc.
Beam beam1(...);
Beam beam2(...);

// declare the two transfer maps
both are set to identity to start with
Map map1;
Map map2;

declare the two rings
the structure has been read in
Lattice ring1;
Lattice ring2;

now compute the transfer maps
map1 = ring1.Propagate(beam1, map1);
map2 = ring2.Propagate(beam2, map2);

continue with evaluation of the transfer maps
map1.NormalForm();
map2.NormalForm();

track a bunch of particles through one revolution in ring 1
Bunch bunch;
ring1.Propagate(beam1, bunch);
// etc.

The azimuth serves only to identify positions in the ring. It should not be confused with the arc length which always runs in beam direction. The lattice functions are therefore always computed with respect to the beam direction, and they are saved together with the azimuth. Among other things, this allows to plot them on the same plot.

4.1 Tracking Positive Particles in Clockwise Direction

The beam line to be tracked is defined by the selected line object, and the mode of tracking is given by the selected beam object. When tracking positive particles in clockwise direction the sign conventions of MAD 8 are valid.

4.2 Tracking Positive Particles in Anti-Clockwise Direction

For tracking positive particles in anti-clockwise direction the following changes are required:

1. Run backward through the beam line.
2. Change $s$ to $C - s$.
3. The positive $z$-axis now points towards the centre of the ring.
4. The forces due to electrostatic fields are unchanged.
5. The direction of flight is inverted. This changes the sign of all magnetic forces. For dipoles this changes the sense of deflection. For quadrupoles the inversion of strength causes interchange of focusing and defocusing.
6. As for all magnets the sign of deflection in SPLITTER magnets is inverted.
7. The sign of displacement in SHIFT elements is inverted.
8. Energy losses due to synchrotron radiation depends on the absolute values of the deflections only and are conserved.
9. With the conventions adopted, energy gains in cavities can always be computed without sign change.
10. The initial beam axis direction for survey must be inverted. All changes of reference change their sign due to the inverted direction of flight.
4.3 Tracking Negative Particles in Clockwise Direction

For tracking negative particles in clockwise direction the following changes are needed:

1. No change of direction is needed.
2. The forces due to electrostatic fields change their signs.
3. The charge is inverted. This changes the sign of all magnetic forces. For dipoles this changes the sense of deflection. For quadrupoles the inversion of strength causes interchange of focusing and defocusing.
4. As for all magnets the sign of deflection in SPLITTER magnets is inverted.
5. The sign of displacement in SHIFT elements is kept.
6. Energy losses due to synchrotron radiation depends on the absolute values of the deflections only and are conserved.
7. With the conventions adopted, energy gains in cavities can always be computed without sign change.
8. The initial beam axis direction for survey must be inverted. All changes of reference due to magnetic elements change their sign due to the inverted charge.

4.4 Tracking Negative Particles in Anti-Clockwise Direction

For tracking negative particles in anti-clockwise direction the following changes are required:

1. Run backward through the beam line.
2. Change $s$ to $C - s$.
3. The positive $x$-axis now points towards the centre of the ring.
4. The forces due to electrostatic fields change their signs.
5. The direction of flight and the sign of the charge is inverted. This keeps the sign of all magnetic forces unchanged.
6. As for all magnets the sign of deflection in SPLITTER magnets is kept.
7. Due to the inversion of the direction of flight the sign of displacement in SHIFT elements is inverted.
8. Energy losses due to synchrotron radiation depends on the absolute values of the deflections only, they are conserved.
9. With the conventions adopted, energy gains in cavities can always be computed without sign change.
10. Survey coordinates start at $\theta = \pi$. All changes of reference must be adapted to the inverted direction of flight and to the inverted charge. Since electrostatic separators do not change the reference, these adaptations cancel.

5 Status of MAD-9

5.1 Decoder for Elements, Lines and Sequences

The decoder for beam lines and sequences is complete. It has been enhanced slightly:

- Beam lines now have a read-only parameter which is the total length. This can be used in expressions.
- A beam sequence now requires its length as a single parameter on the first line. This avoids the need to introduce an artificial marker at its end which only serves to define the length. Also, the length is available in other computations.
- A beam line can be embedded in a sequence like a single element.

The decoder for beam elements is able to recognize most element types. The missing elements are being added shortly.
5.2 Differential Algebra

A differential algebra package in C++ has been written and tested. It is being used to compute the transfer maps for the elements already implemented. For most of these elements the transfer maps to an arbitrary order can be given with coefficients up to machine precision.

Application of differential algebra for determining the dependence of transfer maps on system parameters is not yet implemented.

5.3 Normal Form Analysis

A 2N-dimensional normal form analysis has been implemented along the ideas by M. Berz, E. Forest, and J. Irwin [1]. As opposed to the method in [turchetti] it is only limited in the number of dimensions by the available computer resources.

5.4 More Than One Beam

The use of more than one beam in the machine has not yet been implemented. However, it is straightforward to extract the relevant data from a beam object instead of using global data. A slightly more complex problem consists of adapting the signs and scaling of the element parameters for inverted charge and/or beams.

5.5 Time Schedule for MAD-9

It is expected that the following time is required to implement the required operations:

- Implementation of missing elements – 1/2 month,
- Application of differential algebra for variable system parameters – 1 month,
- Application of more than one beam objects – 1 month,
- Simple matching – 1 month,
- Complex matching – another 2–3 months.

Towards summer a usable version could probably be available.

6 Conclusions

The object-oriented approach allowed in C++ makes it possible to get a much more flexible program. A full use of the flexibility is possible if users are willing to contribute their own matching methods by writing simple C++ functions. I am perfectly willing to teach interested people how to do this, once the program is in a state permitting its use.

References
