

NEW DEVELOPMENTS OF MAD-X USING PTC

P. K. Skowronski, F. Schmidt, R. de Maria, CERN, Geneva, Switzerland
E. Forest, KEK, Tsukuba, Japan.

Abstract

For the last few years the MAD-X program makes use of the Polymorphic Tracking Code (PTC) to perform calculations related to beam dynamics in the nonlinear regime. This solution has provided a powerful tool with a friendly and comfortable user interface. Its apparent success has generated a demand for further extensions. We present the newest features developed to fulfil in particular the needs of the Compact Linear Collider (CLIC) studies. A travelling wave cavity element has been implemented that enables simulations of accelerating lines. An important new feature is the extension of the matching module to allow fitting of non-linear parameters to any order. Moreover, calculations can be performed now with parameter dependence defined in the MAD-X input. In addition, the user can access the PTC routines for the placement of a magnet with arbitrary position and orientation. This facilitates the design of non-standard lattices. Lastly, the three dimensional visualization of lattices is now available.

INTRODUCTION

The MAD-X code [1] is ideally suited for very large accelerators but has also been frequently used for small machines and transfer lines. The strength of the code is the flexible MAD input language, a large collection of modules needed for accelerator design and most importantly the very elaborate matching techniques. However, for the smaller machines the users are more and more forced to use tricks to circumvent the many limitations of MAD-X. Moreover, given the rigid structure of the code one cannot easily upgrade it to fulfill the many requirements for lines or accelerators of all sizes.

It was therefore decided to complement MAD-X with the PTC library. This library has been developed and well debugged in recent years in view of providing proper physics models for all accelerator elements, using symplectic integrators to describe the accelerator models to arbitrary precision, allowing for full 3D freedom to place the elements on the accelerator floor and expanding the elements in a Taylor Map to arbitrary order, including parameter dependence, followed by a Normal Form analysis.

The combination of both codes, i.e. the flexible input, the many tools and the powerful matching mechanism of MAD-X together with the new modern features of PTC results in the enhanced "MAD-XP" code. To our knowledge MAD-XP is world-wide the most flexible and complete accelerator design code for any kind of accelerator.

The two independent codes are linked via a converter that transfers the accelerator structure and all el-

ements from MAD-X into PTC. The MAD-X language provides a new set of commands specific to PTC. The MAD-XP user must invoke an instance of PTC via the PTC.CREATE.UNIVERSE command. In a second step the accelerator description must be exported to PTC with the PTC.CREATE.LAYOUT command. There are presently three key commands in MAD-XP:

- PTC_TWISS - calculates all Twiss functions and also user defined parameters
- PTC_NORMAL - calculates global parameters of a ring using the Normal Form technique
- PTC_TRACK - performs symplectic particle tracking through thick elements. It should be noted that in MAD8 the thick-lens tracking was not symplectic. Therefore this module was excluded from MAD-X.

There are other commands to manipulate parameters of the lattice and to control data flow or processing. The MAD-XP code is in a period of full swing development: we are constantly extending the interface and the set of the available commands.

In this report we will high-light the newest developments of the MAD-XP code combination.

SUPPORT FOR TRANSFER AND ACCELERATING LINES

In the initial implementations of the MAD-XP transfer and accelerating lines were not properly treated. Therefore we have extended the interface between MADX and PTC, updating all the adequate data structures such that they are capable to handle linear layouts with longitudinally varying reference beam energy. We have added a switch that instructs the code to treat a given layout as a ring or a line. The code has been safeguarded against the misuse of the algorithms, e.g. the normal form cannot be performed for a linac. The appropriate user interface for the Traveling Wave Cavity, the TWCAVITY element, has been added. On request, the settings can be automatically adjusted to the user needs, e.g. the phase of the wave with respect to the reference particle.

KNOBS

PTC allows additional parameters in the calculations which we refer to as knobs. Then all lattice functions and map coefficients can be represented as functions (Taylor series) of the knobs.

We have developed the PTC.SETKNOB command that defines one or more of the following parameters as a knob:

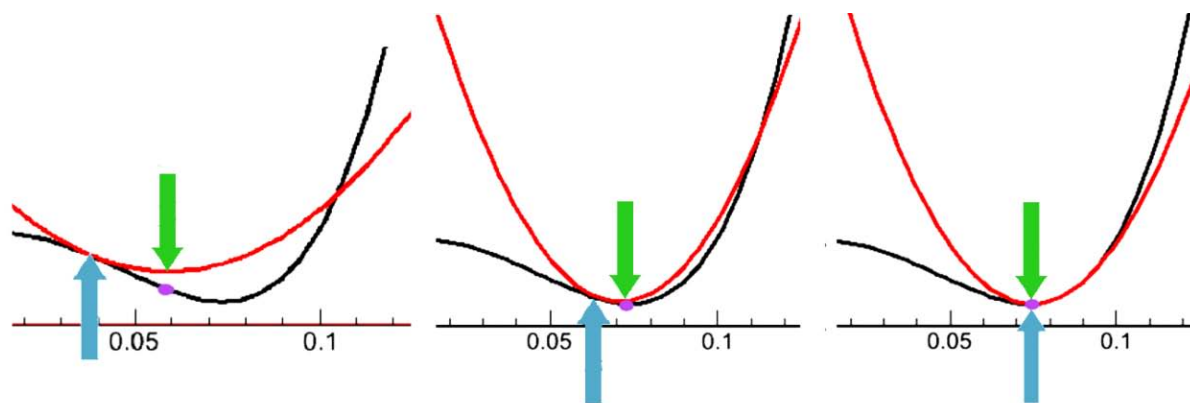


Figure 1: Schema of matching employing knobs on a simple example with one variable. Horizontal axis represents the variable value, vertical - penalty function value. The black line represents a penalty function. The red one shows its Taylor approximation. The blue up-pointing arrow indicates the currently guessed value. The green arrow shows minimum of the approximation and the violet dot corresponding value of the penalty function. The minimum of approximation becomes a new guess point. The algorithm stops if the found minimum coincides within required precision with both, the current guessed point and value of the penalty function at this point.

- any field component up to order 100
- voltage, frequency, phase of a cavity
- field strength of a solenoid
- Twiss parameters (only uncoupled) at the beginning of a line

All standard lattice functions as well as map coefficients and other nonlinear parameters selected by the user are buffered in the form of polynomials. With the help of the appropriate commands these functions can be:

- displayed on the screen
- written to a file in several formats
- exported to a MAD-X table, from where they are accessible for further processing by other MAD-X commands
- plotted with the help of the RPLOT tool

A knob value is the deviation from the nominal value around which the Taylor series is expanded. The command PTC.SETKNOBVALUE sets a given knob to a certain value. As a consequence, the scalar values of the lattice functions stored in the tables are re-evaluated using the polynomial representation.

MATCHING OF PARAMETERS TO ARBITRARY ORDER

The matching module of MAD-X has been updated in order to extend its functionality to more sophisticated fitting procedures, some of which are based on PTC calculations [2]. The user can define a macro that is executed at each matching step. It may contain an arbitrary set of MAD-X commands including calls to PTC. Additionally, the matching constraints may refer to an arbitrary cell in a table or a user-defined expression. The two new commands, PTC.SELECT and SELECT.PTC.NORMAL, instruct the

PTC.TWISS and PTC.NORMAL procedures, respectively, to store the selected parameters of arbitrary order in the specified tables.

The example below shows how to match the parameter QDK1 to obtain a zero component H3000 at position PSR\$END while keeping the second and third order detuning terms small.

```
create,table=H3,withname,column="h0000";
ptc_select,table=H3,column="h0000",
    polynomial=3, monomial="400000";
select_ptc_normal, dq1=1;
select_ptc_normal, dq1=2;
dQxdX:= table(normal_results,value,1);
dQxdX2:=table(normal_results,value,2);
match, use_macro;
vary, name=QDK1, step=0.01;
m1: macro={
  ptc_create_universe;
  ptc_create_layout,method=6,
    exact=true, closed_layout=true;
  ptc_twiss,closed_orbit, no=4;
  ptc_normal,closed_orbit,normal,no=4;
  ptc_end;
};
constraint,expr=
    table(H3,PSR$END,h0000) = 0.0;
constraint, expr= dQxdX <1e-3;
constraint, expr= dQxdX2 <1e-3;
lmdif, tolerance=1e-30 ,calls=200;
endmatch;
```

FAST MATCHING WITH THE HELP OF KNOBS

The high complexity of the calculations performed by PTC is associated with large computational times. It is es-

pecially noticeable in the case of the matching procedures, where the calculation must be performed several hundred or even thousands times. The performance of the procedure is significantly increased if faster but approximate results are exploited.

The algorithm that we have developed is illustrated in Fig.1. The variables that the user aims to match are set as knobs. All the lattice functions and other parameters used in the constraints become Taylor series expanded around the first (guessed) starting point. Then the minimum is found on the basis of the Taylor series, which is a very fast procedure. Of course, the search must be limited to some user defined validity range where the approximations are sufficiently accurate. The minimum as found in this procedure becomes the new guessed value, the lattice parameters are updated accordingly and the PTC calculation is repeated. The algorithm stops if both of the following conditions are fulfilled:

- The minimum is found for sufficiently small values of the knobs (Taylor series is exact at the point where it was expanded)
- the value of the penalty function evaluated for these knobs is within the required precision close to the one obtained with Taylor series

Please note, that the second point needs an additional PTC run.

The user can implement this kind of algorithm within the MAD-X script. In this case she/he is free to choose the penalty functions and convergence conditions suitable for her/his particular problem. It is possible to implement rather sophisticated algorithms. As usual the examples are available on the MAD-X web pages.

For convenience we have incorporated as a basic version the algorithm within the MATCH routine of MAD-X that is able to cope with the most common problems. In order to use it, the user must add the USE_PTCKNOB switch. In these cases variables are defined using PTC_SETKNOB commands. Within the scope of the MATCH command the following calls:

- CONSTRAINT
- PTC_CREATE_UNIVERSE
- PTC_CREATE_LAYOUT
- PTC_TWISS

are stored in memory. They are used after calling the matching routine (JACOBIAN, MIGRAD, LMDIF, etc.).

ELEMENT PLACEMENT AT ARBITRARY POSITION AND ORIENTATION

PTC routines can be used that permit to displace elements to arbitrary positions and orientations (see 2). This feature facilitates the design of non-standard lattices. The user can define new element position and orientation with respect to:

- current position
- end face of the previous element
- origin of the global coordinate system

If requested, the program adjusts automatically the positions of all elements downstream of the displaced element.

3D VISUALIZATION OF LATTICES

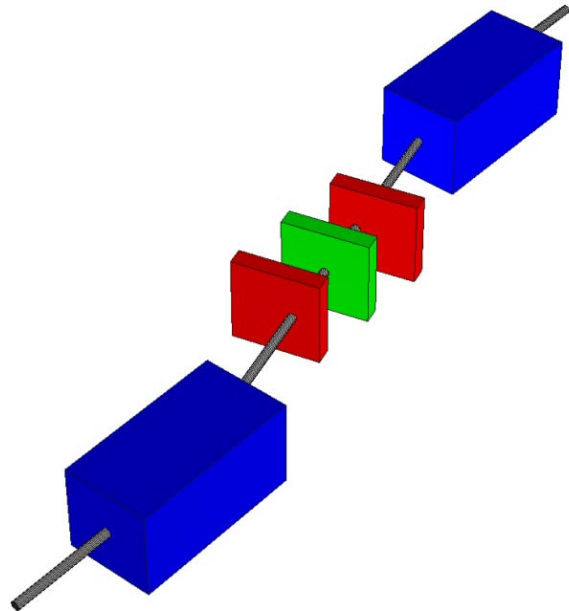


Figure 2: Example of 3D lattice visualization: a dogleg chicane with quadrupoles displaced off beam axis. Its position is matched to obtain the required momentum compaction factor.

The details of the geometry described in PTC can be written to a file using the PTC_PRINTFRAMES command. One of the supported formats is a ROOT macro [3]. When executed with ROOT, 3D image of the of the layout are displayed, see Fig.2. Root supports intuitive mouse based navigation as well as keyboard based one. The user can zoom and rotate the image in an arbitrary plane. The hope is that this feature is not just a gimmick but will be used to facilitate lattice design that may require non-trivial geometric transformations. It might even be useful to verify that a particular ring closes as expected.

PLUG-INS

We have defined a C interface to MAD-XP that opens the possibility for the development of plug-ins that extend the standard functionality of the code. The user may define new commands that are implemented in the specified shared library. On request of such a command the library is dynamically loaded and the specified function is called. This interface allows for the interaction and data exchange between user defined modules and MAD-XP.

RPLOT

The first available plug-in is RPLOT. It is a ROOT [3] based graphical tool that plots the parametric lattice functions. These functions can be depicted

- at a given location as a function of one or two parameters
- along the layout

This interface allows to control the knobs and format the plots which can be saved in the formats: Postscript, PDF, JPG, GIF and others.

CONCLUSIONS

We have described the newest features of MAD-X that take advantage of the PTC capabilities leading to increased functionality in the combined code MAD-XP. The new commands extend the existing MAD language and are easy to use.

Linear machines are now better supported in MAD-XP. Thanks to the implementation of the parametric knobs the user is able to study more effectively properties of a lattice as a function of its parameters. Consequently, the matching module could now be extended to deal with parameters of any order. Moreover, the parametric matching allows for a significantly speed-up of the matching procedure.

The user can now displace the elements to arbitrary positions. New ways to visualize a layout in 3D are also provided. The concept of plug-ins have now been implemented into MAD-X, first of which is RPLOT that uses the ROOT capabilities for plotting the parametric lattice functions.

By interfacing MAD-X with PTC (MAD-XP) we have provided a more complete and user friendly program for particle accelerator design and modelling. Driven by the needs of future machines like CLIC, MAD-XP is constantly improved to tackle ever more complex tasks.

REFERENCES

- [1] <http://mad.home.cern.ch>
- [2] R. de Maria, F. Schmidt, P.K. Skowronski, "Advances in matching in MAD-X", ICAP'06, October 2006, Chamonix, <http://www.JACoW.org>.
- [3] <http://root.cern.ch>