

## COSY INFINITY Version 6

**Martin Berz**

Department of Physics and Astronomy and  
National Superconducting Cyclotron Laboratory,  
Michigan State University, East Lansing, Mi 48824

### **Abstract.**

COSY INFINITY is an arbitrary order map code for the design and analysis of particle optical systems. It is based on differential algebraic methods, which provide an efficient and elegant framework for typical tasks faced in practice. For example, the computation of maps of arbitrary order and their dependence on system parameters, normal form techniques for the analysis and resonance correction of the behavior of repetitive systems, techniques for aberration correction and variety of other features can all be phrased very directly in terms of these methods.

As the user interface, COSY employs a structured object oriented language, which is simultaneously used for the coding of the physics. This approach allows a seamless connection of the work of the user with the tools provided by the code and directly allows the adaptation of the code to specific tasks. Optimization and fitting is supported at the language level, allowing rather direct and general optimization of essentially any quantity. Lattices can be described in terms of COSY's language environment or alternatively in the standard MAD input.

For the sake of portability, the language interface of COSY is standard FORTRAN 77; the compiler and executer of COSY as well as the libraries to which the language links are all written in this language. The language environment supports a variety of interactive and printer or plotter-based graphics drivers, and new drivers are added easily. Currently COSY INFINITY is used by approximately 100 registered users.

### **1. Introduction**

The properties of particle optical systems [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] can be conveniently described by a map relating final phase space coordinates  $\vec{z}_f$  to initial coordinates  $\vec{z}_i$  and system parameters  $\vec{\delta}$  in the following way:

$$\vec{z}_f = \mathcal{M}(\vec{z}_i, \vec{\delta}). \quad (1)$$

Since except for trivial cases it is impossible to express the transfer map  $\mathcal{M}$  in closed form, it is customary to represent it in terms of its Taylor expansion, a method used in many codes[11, 12, 13, 14, 15]. With the advent of the differential algebraic methods[16, 17, 18, 19], these Taylor methods were generalized in a rather straightforward way from the orders of two or three used by other programs to any

order, and including the dependence on system parameters. COSY INFINITY (see also [20, 21, 22, 23, 15]) is a program that uses these techniques for the computation as well as for the analysis of these maps in a variety of ways.

## 2. The Physics in COSY

The actual computation of maps of particle optical systems is done in several different ways suitable for different situations. The main fields of elements are computed using the Lie derivative [17]  $L_f = \vec{f} \cdot \vec{\nabla} + \partial/\partial t$  associated with the differential equation

$$\frac{d}{dt} \vec{x} = \vec{f}(\vec{x}, t), \tag{2}$$

which allows the computation of the higher order time derivatives of any function  $g(\vec{x}(t))$  defined on the solution of the differential equation:

$$\begin{aligned} d/dt g &= L_f g \\ d^2/dt^2 g &= L_f^2 g \\ d^3/dt^3 g &= L_f^3 g. \end{aligned} \tag{3}$$

Hence in a formal sense we have  $g(t) = \exp(L_f) g(0)$ ; by passing to DA operations, the convergence of the sequence becomes guaranteed. By choosing  $g$  to be the individual phase space coordinates, one can compute their final values from the initial values and hence the transfer map.

The loss of the highest derivative in the derivation operation  $\partial$  in DA entails that the mechanism is only applicable to cases in which  $\vec{f}$  does not depend on the independent variable, which in the case of particle optics means that the fields be constant. The treatment of non-constant fields is done by numerical integration with DA operations [24, 16, 17], which is more time consuming. However, for the most important case of such fields, the fringe fields, there is a very efficient perturbative method with high accuracy based on symplectic scaling, which is described in detail in [25].

Besides these general methods, some particle optical elements are treated in a separate way. For example, for multipoles of up to order two, analytical formulas are used in the flavor or TRANSPORT [11], which for these cases gains about a factor of two in speed. The drift and the dipole with circular and tilted edges are treated geometrically, which allows the computation of the aberrations to arbitrary order in an analytical way and sometimes a tenfold speed gain.

Besides the computation of the maps, COSY allows the analysis of maps in a variety of ways. For repetitive systems, normal form methods play an important role in the analytical computation and correction of tune shifts, pseudo invariants and resonance strengths [26]. They are based on an iterative order by order simplification of the map by nonlinear changes of variables

$$\mathcal{M}_n = \mathcal{A}_n^{-1} \circ \mathcal{M}_{n-1} \circ \mathcal{A}_n. \tag{4}$$

An example of the correction of resonances with COSY can be found in a companion paper [27].

For the long term analysis of systems, it is important to be able to perform symplectic tracking, which is achieved by means of one of a variety of generating function representations of the map

$$\begin{aligned}
 F_1(q_i, q_f) \text{ satisfying } (\vec{p}_i, \vec{p}_f) &= (+\vec{\nabla}_{q_i} F_1, -\vec{\nabla}_{q_f} F_1) \\
 F_2(q_i, p_f) \text{ satisfying } (\vec{p}_i, \vec{q}_f) &= (+\vec{\nabla}_{q_i} F_2, +\vec{\nabla}_{p_f} F_2) \\
 F_3(p_i, q_f) \text{ satisfying } (\vec{q}_i, \vec{p}_f) &= (-\vec{\nabla}_{p_i} F_3, -\vec{\nabla}_{q_f} F_3) \\
 F_4(p_i, p_f) \text{ satisfying } (\vec{q}_i, \vec{q}_f) &= (-\vec{\nabla}_{p_i} F_4, +\vec{\nabla}_{p_f} F_4),
 \end{aligned} \tag{5}$$

which in the DA framework can be computed in a very direct way [17, 18].

For the case of single pass systems, there are tools to output the aberrations of systems as well as to compute resolutions of spectrographs in a variety of operating modes. Particularly useful is the method of reconstructive correction, which computationally eliminates all aberrations via measurements in two planes [28, 29]

### 3. The COSY Language

The COSY language is a structured language with the flavor of PASCAL. Different from PASCAL, it allows the use of arbitrary data types and operations thereon that can be specified. In particular, this allows a direct use of the DA (differential algebra) data type, which is essential for an efficient use of DA in practice. It is also helpful for the manipulation of pictures, which are stored as picture objects. Type checking is performed at execution time, which allows the use of the same procedures with DA or real arguments, a feature which makes the computation of maps depending on parameters all but trivial. The commands of the COSY language are as follows:

```

BEGIN ;                END ;                { Begins and ends program          }
INCLUDE ;             SAVE ;                { Includes and saves compiled code }
VARIABLE ;           ;                       { Declares a local variable        }
PROCEDURE ;         ENDPROCEDURE ;         { Declares a local procedure       }
FUNCTION ;          ENDFUNCTION ;         { Declares a local function        }
< assignments > ;   ;                       { Sets value of variable           }
< procedure calls > ; ;                     { Calls previously defined procedure }
IF ;                ENDIF ;               { Executed once if argument is true }
WHILE ;             ENDWHILE ;            { Executed while argument is true   }
LOOP ;              ENDLOOP ;             { Stepping argument                }
FIT ;               ENDFIT ;              { varying arguments to fit conditions }

```

Except for the last one, the flow control statements are rather standard. The FIT block is executed over and over again as long as the sum of squares of the objective variables listed in the ENDFIT statement can be reduced by modifying the values of the free variables listed in the FIT statement. The ENDFIT statement also contains the number of the optimizer to be used as well as the tolerance and the maximum number of iterations allowed.

Module Name	Contents	Language	Lines
FOXY.FOR	Compiler/Executor	FORTRAN 77	5,000
DAFOX.FOR	Low-level libraries	FORTRAN 77	12,000
FOXFIT.FOR	Various optimizers	FORTRAN 77	5,000
FOXGRAF.FOR	Graphics interfaces	FORTRAN 77	2,000
MADCOSY.FOR	MAD processor	FORTRAN 77	5,000
COSY.FOX	Physics library	COSY	2,000
DEMO.FOX	Demos of key tools	COSY	1,000

**Table 1.** The modules comprising COSY INFINITY

The nesting of procedures and functions as well as the visibility of modules and variables follows standard practice of modular languages. The flow control structures IF, WHILE, LOOP and FIT can be nested. In practice, the direct availability of various optimization methods at the language level via the FIT command proves very helpful. Another feature of the COSY language is that it is rather directly possible to interface to FORTRAN subroutines, which allows easy connection to other codes.

The physics in COSY is written almost entirely in COSY's own input language and contained in the macro file COSY.FOX. Using the COSY commands SAVE and INCLUDE, this macro file can be compiled ahead of time and included as needed in user code. The use of the features in COSY is via the call of procedures and functions.

For example, particle optical elements are just procedures, which when invoked update the current transfer map of the system, the coordinates of rays to be tracked, and a variety of other quantities. In this framework it is rather directly possible to build lattices in a convenient way. For example, if we want to build a simple FODO consisting of a dipole (DI), a quadrupole (MQ) as well as some drifts (DL), which is then used six times with slightly different values of quad strengths due to random errors, the input would look like this:

```

PROCEDURE CELL PHI Q ;
  DL 5 ; DI 10 PHI .1 ; DL 5 ; MQ .5 Q .1 ; DL 5 ;
  ENDPROCEDURE ;
.
.
.
UM ;                               { Sets the transfer map to unity }
LOOP I 1 6 ; CELL 60*(1+.1*RANF()) .2*(1+.01*RANF()) ; ENDLLOOP ;
TS MUX MUY MUZ ;   { Computes the tune dependence on emittance }

```

Altogether, the COSY INFINITY package consists of six different modules plus a demo file showing the use of the most important features, with a total of about 30,000 lines of code, a breakdown of which is shown in Table 1.

#### 4. Elements in COSY

The library of macros in COSY.FOX contains a large list of pre-defined beamline elements, each of which can be calculated to arbitrary order including the dependence on parameters. Besides the ones listed here, it is possible to develop new ones in a variety of ways, for example using the procedure GE, which allows the computation of the transfer map from a table of data for multipole strengths. The following elements can be used directly in COSY:

- Bending magnets including inhomogeneities and edge angles
- Electrostatic deflectors including inhomogeneities
- Magnetic and electric multipoles, any order and superimposed
- Wien filters, with or without net deflection, including inhomogeneities
- Wigglers with various field models
- Thin cavities including nonlinearities
- General element GE, described by a table of multipole strengths
- Magnetic and electric round lenses including various solenoids
- Fringe fields for all magnetic and electric elements
- Glass lenses, spherical and aspherical, mirrors, prisms
- Misalignments (shifts, tilts of reference orbit, rotation around reference orbit)

#### 5. Analysis with COSY

The macro package of analysis tools contained in COSY.FOX contains a large variety of tools that are useful for various problems. Using the open approach of COSY, it is also rather easily possible to develop new analysis tools. The currently available features include

- Transfer maps and aberrations of arbitrary order including system parameters
- Various generating function and Lie factorization representations
- Closed orbit and linear lattice parameters including parameter dependence
- Tracking of rays through the system and layout of the system
- Analytical calculation of amplitude and parameter dependent tune shifts
- Analytical calculation and correction of resonance strengths of repetitive systems
- Repetitive Tracking, if desired with one of several symplectification methods
- Computation of resolutions of spectrographs
- Reconstructive correction of aberrations

## 6. Distribution of COSY

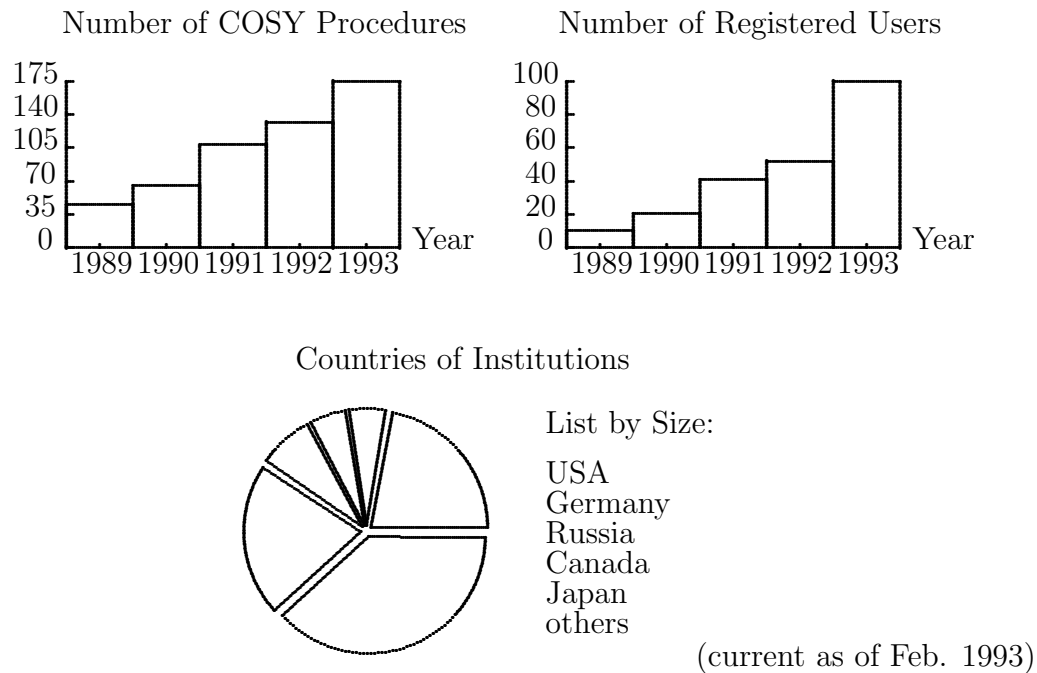
COSY INFINITY has been made available to users since 1989. Currently, version 6 of the code is being distributed. The code can be obtained from the author by electronic mail after signing a simple registration form including a nonproliferation agreement. Currently, the code is being used by about 100 registered users in about 50 laboratories. Figure 1 (which incidentally was created using COSY's graphics features and output directly using the  $\text{\LaTeX}$  graphics driver) shows the history of the number of users as well as the number of procedures since its first release.

Due to the rigorous use of plain FORTRAN 77, COSY can easily be adjusted to any FORTRAN environment, and in most cases the necessary changes concern only file handling as well as very few low-level library routines like the ones returning CPU time. For most popular environments, a dedicated version can be generated from the master version by running a small program supplied with COSY which selectively comments and un-comments certain commands. The list of environments that are currently contained in the master version in this way are

- VAX (VMS and UNIX)
- SUN (Unix)
- HP (Unix)
- CRAY
- IBM 6000 (Unix)
- IBM PC (Lahey)
- IBM Mainframe

COSY graphics at its lowest level is based on seven elementary routines which can be interfaced rather easily with any environment. The routines are BEGIN, END, MOVE, DRAW, CHARACTER, LINEWIDTH, and COLOR, and so it is rather easy to write drivers for a variety of environments. With the normal shipment of COSY, the following drivers are supplied automatically:

- GKS-based X-Windows
- GKS-based VMS-Windows
- IBM PC VGA (Lahey)
- Direct Tektronix
- Direct Postscript
- $\text{\LaTeX}$
- GKS-based HP7475
- GKS-based HP Paintjet
- Low Resolution ASCII



**Figure 1.** Statistics about Users of COSY (Picture generated in L<sup>A</sup>T<sub>E</sub>X with COSY's graphics environment)

## 7. Acknowledgements

For increasing help as well as stimulating and valuable discussions, I would like to thank my colleagues Ralf Degenhardt, Kyoko Fuchi, Georg Hoffstätter, Weishi Wan, and Meng Zhao. For financial support, I would like to thank the National Science Foundation as well as the Alfred P. Sloan Foundation.

## References

- [1] H. Wollnik. *Charged Particle Optics*. Academic Press, Orlando, Florida, 1987.
- [2] D. C. Carey. *The Optics of Charged Particle Beams*. Harwood, 1987.
- [3] P. W. Hawkes and E. Kasper. *Principles of Electron Optics*. Academic Press, London, 1989.
- [4] Ed. A. Septier. *Focusing of Charged Particles*. Academic Press, New York, 1967.
- [5] A. P. Banford. *The Transport of Charged Particle Beams*. Spon, London, 1966.
- [6] H. Buchdahl. *An Introduction to Hamiltonian Optics*. Cambridge University Press, Cambridge, MA, 1970.
- [7] R. K. Luneburg. *Mathematical Theory of Optics*. University of California, Berkeley Press, Berkeley, CA, 1964.
- [8] A. B. El-Kareh and J. C. J. El-Kareh. *Electron Beams, Lenses and Optics, 2 Vols.* Academic Press, New York, 1970.
- [9] J. Grosser. *Einführung in die Teilchenoptik*. Teubner, Stuttgart, 1983.
- [10] P. Dahl. *Introduction to Electron and Ion Optics*. Academic Press, New York, 1973.
- [11] K. L. Brown. The ion optical program TRANSPORT. Technical Report 91, SLAC, 1979.
- [12] H. Wollnik, B. Hartmann, and M. Berz. Principles behind GIOS and COSY. *AIP Conference Proceedings*, 177:74, 1988.

- [13] T. Matsuo and H. Matsuda. Computer program TRIO for third order calculations of ion trajectories. *Mass Spectrometry*, 24, 1976.
- [14] A. J. Dragt, L. M. Healy, F. Neri, and R. Ryne. MARYLIE 3.0 - a program for nonlinear analysis of accelerators and beamlines. *IEEE Transactions on Nuclear Science*, NS-3,5:2311, 1985.
- [15] M. Berz. Computational aspects of design and simulation: COSY INFINITY. *Nuclear Instruments and Methods*, A298:473, 1990.
- [16] M. Berz. Differential algebraic description of beam dynamics to very high orders. *Particle Accelerators*, 24:109, 1989.
- [17] M. Berz. Arbitrary order description of arbitrary particle optical systems. *Nuclear Instruments and Methods*, A298:426, 1990.
- [18] M. Berz. Differential algebraic treatment of beam dynamics to very high orders including applications to spacecharge. *AIP Conference Proceedings*, 177:275, 1988.
- [19] M. Berz. Differential algebraic description and analysis of trajectories in vacuum electronic devices including spacecharge effects. *IEEE Transactions on Electron Devices*, 35-11:2002, 1988.
- [20] M. Berz. COSY INFINITY Version 6 reference manual. Technical Report MSUCL-869, National Superconducting Cyclotron Laboratory, Michigan State University, East Lansing, MI 48824, 1993.
- [21] M. Berz. COSY INFINITY Version 5 reference manual. Technical Report MSUCL-811, National Superconducting Cyclotron Laboratory, Michigan State University, East Lansing, MI 48824, 1991.
- [22] M. Berz. New features in COSY INFINITY. In *Third Computational Accelerator Physics Conference*. AIP Conference Proceedings, 1993.
- [23] M. Berz. COSY INFINITY. In *Proceedings 1991 Particle Accelerator Conference*, San Francisco, CA, 1991.
- [24] M. Berz. The method of power series tracking for the mathematical description of beam dynamics. *Nuclear Instruments and Methods*, A258:431, 1987.
- [25] G. Hoffstätter and M. Berz. Efficient computation of fringe fields using symplectic scaling. In *Third Computational Accelerator Physics Conference*. AIP Conference Proceedings, 1993.
- [26] M. Berz. Differential algebraic formulation of normal form theory. in: *M. Berz, S. Martin and K. Ziegler (Eds.), Proc. Nonlinear Effects in Accelerators*, 1993.
- [27] R. Servranckx. Optics programs at Triumf. in: *M. Berz, S. Martin and K. Ziegler (Eds.), Proc. Nonlinear Effects in Accelerators*, 1993.
- [28] M. Berz, K. Joh, J. A. Nolen, B. M. Sherrill, and A. F. Zeller. On-line correction of residual aberrations in spectrographs. In *Proceedings 1991 Particle Accelerator Conference*, San Francisco, CA, 1991.
- [29] M. Berz, K. Joh, J. A. Nolen, B. M. Sherrill, and A. F. Zeller. Reconstructive correction of aberrations in nuclear particle spectrographs. *Physical Review C*, 47,2:537, 1993.