Available online at www.sciencedirect.com

SCIENCE DIRECT®

Nuclear Instruments and Methods in Physics Research A 558 (2006) 346–350

NUCLEAR INSTRUMENTS & METHODS IN PHYSICS RESEARCH Section A

www.elsevier.com/locate/nima

COSY INFINITY Version 9

Kyoko Makino*, Martin Berz

Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, USA

Available online 6 December 2005

Abstract

In this paper, we review the features in the newly released version of COSY INFINITY, which currently has a base of more than 1000 registered users, focusing on the topics which are new and some topics which became available after the first release of the previous versions 8 and 8.1. The recent main enhancements of the code are devoted to reliability and efficiency of the computation, to verified integration, and to rigorous global optimization. There are various data types available in COSY INFINITY to support these goals, and the paper also reviews the feature and usage of those data types.

Keywords: COSY INFINITY; Verified computation; Differential algebra; Taylor model; Interval arithmetic

1. Introduction

Since the previous versions 8 and 8.1, the main enhancements of the code COSY INFINITY were focused on computational reliability and speed. Interval arithmetic, which serves as a stepping stone to guaranteed computations, is performed with fully verified software rounding in COSY INFINITY. The method is connected to the Differential Algebraic (DA) technique to provide a further efficient method of reliable computation, the Taylor model method, by utilizing the capability of high order computation of the DA technique. Both of these methods benefit from enhanced sparsity support including the ability to treat different variables to different orders.

Besides such basics to support the quality of the computation, there are several particle optical elements, analyzing tools and beam physics concepts newly added in the code since the last version. In particular now there are various ways to treat the dynamics of particle beams traveling in matter.

Through the changes of versions, emphasis has been placed on backwards compatibility and portability of the code to different platforms as well as transparent portability to four separate language platforms, namely F77, F90, C and C++.

2. Verified computations

2.1. Data types

COSY INFINITY supports various data types, starting from RE (double precision REal number), DA (Differential Algebra vector), and the GR (GRaphics) data type. Table 1 lists all the data types supported in the standard version of COSY INFINITY Version 9. Since the code is object oriented, new data types and the associated operations can be easily added and removed [1]. As some of the readers may have noticed, the Ordered Interval (OI) and Ordered interval Vector (OV) data types are not supported anymore. Besides the STring (ST), LOgical (LO) and GRaphics (GR) data types, all others are numerical computation objects based on double precision real numbers. If any higher precision computation environment like quadruple precision computation mode is available, all these numerical computation objects can be straightforwardly ported to the higher precision mode.

A REal number (RE) data type object corresponds to a double precision number; a CoMplex number (CM) data type object occupies two double precision numbers for the real and imaginary parts. A real number VEctor (VE) data type object consists of several double precision numbers in

*Corresponding author.
E-mail addresses: makino@msu.edu (K. Makino), berz@msu.edu (M. Berz).

0168-9002/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.nima.2005.11.109

Table 1
Data types supported in COSY INFINITY

RE	Real Number
ST	String
LO	Logical
DA	Differential Algebra Vector
VE	Real Number Vector
CM	Complex Number
IN	Interval
IV	Interval Vector
GR	Graphics
CD	Complex Differential Algebra Vector
TM	Taylor model (Remainder-enhanced Differential Algebra Object)

a vector form, in a similar fashion as a one dimensional double precision array. The VE data type object is created by concatenating two RE data type objects using the operator "&", and more components can be added by further concatenating RE data type objects and/or VE data type objects; its advantage lies in enhanced performance in vectorizable operations.

A DA data type object carries all the non-negligible coefficients of the differential algebra ${}_nD_v$ with order n and dimensionality v [2], where vanishing coefficients and coefficients below the cutoff threshold ε_c in magnitude are not retained. The value of the cutoff threshold ε_c can be specified via the COSY intrinsic procedure DAEPS, its default is set to 10^{-20}. The sparsity handling is particularly important for beam physics applications where usually half of coefficients vanish due to symmetry, and we have been striving to realize and keep superb performance of DA data type objects by the efficient sparsity handling algorithms [3]. A Complex Differential algebra vector (CD) data type object turns DA data type objects to be complex by carrying a set of DA coefficients for the real part and a set for the imaginary part. DA and CD data type objects truncate small coefficients below cutoff threshold, on the other hand, the TM (Taylor model, Remainder-enhanced Differential algebra object) data type objects keep errors associated to order truncation, cutoff and round-off in the data structure. It is worth noting that among various numerical algorithms in the code, those based on the DA-fixed point theorem [2] are particularly powerful, by achieving an nth order DA solution in at most $(n+1)$ iterations of the DA-fixed point operation.

The newly implemented feature of weighted order computation enables to carry different variables x_i to different orders w_i, which can account for the fact that certain variables are more important than others. This fact is particularly helpful for the problem of integration of transfer maps for time-dependent systems with or without verification, where the length of the time step typically significantly exceeds the range of initial conditions, i.e. beam coordinates. This is achieved by simply "seeding" original variables as $x_i^{w_i}$ instead of x_i. In all subsequent operations, only multiples of w_i appear as powers of x_i. Using the feature, optimal reduction of speed can be achieved by sparsity.

2.2. Data types for verified computations

The other data types INterval (IN), Interval Vector (IV) and TM are objects for verified computation. An IN data type object consists of two double precision numbers; one for the lower bound and one for the upper bound of an interval, and an IV data type object describes a vector with interval components. The concept of the IV data type is similar to that of the VE data type, except for that each component is an interval consisting of two double precision numbers. Interval methods for numerical computations express a set of numbers by an interval, and through various computational operations, interval arithmetic rigorously keeps all the possible outcomes in the resulting interval. (Refer to, for example [4] as well as references therein, and many more.) Floating point rounding errors are unavoidably associated with numerical computations, thus a correct implementation of interval methods requires proper handling of these rounding errors. The interval library in COSY INFINITY, which is the base for the verified computations for the data types IN, IV and TM, supports the directed rounding to assure the verification [5]. The COSY ASCII output of intervals is further rounded outward to avoid confusion caused by system dependent output truncation. To facilitate interval-related applications, some utility procedures are newly available. Particularly, INTSEC, INTUNI and INTINC are used for intersection, union and inclusion check, respectively.

The data type TM represents Taylor model objects. The DA data type objects truncate Taylor power series at order n, but the TM data type objects keep the contribution from the Taylor remainder term in an interval, the so-called remainder bound interval. The bulk amount of the functional dependency is kept in the polynomial part that has the same data structure as the DA data type objects. Thus, this data type is also called the (R)emainder-enhanced (D)ifferential algebra type. The remainder bound interval part is used conveniently to absorb other errors like floating point rounding errors and coefficient cutoff errors. This allows us to use floating point numbers for polynomial coefficients, while assuring verified computations.

The utilization of floating point number coefficients has numerous benefits. First, a Taylor model implementation can utilize a big part of the DA library code by mere DA subroutine calls. This helps to reduce the code implementation and maintenance effort as well as the size of the code, but it requires special care to properly handle floating point rounding errors and cutoff errors. Secondly, as summarized in an exhaustive paper on the method of Taylor models [6], this idea is one of key points to enable the suppression of the dependency problem. This is one source of overestimation often observed in interval based computations, and it limits the applicability of many verified computations. Thirdly, the concept of DA-fixed point theorem can be extended straightforwardly to various Taylor model algorithms, allowing them to achieve the same level of computational efficiency to that of DA fixed point based algorithms. One superb example of this is the Taylor model algorithm for verified ODE (ordinary differential equation) integrations [7].

The arithmetic on Taylor models is introduced for binary operations and intrinsic functions. The implementation of Taylor models in COSY INFINITY utilizes the COSY DA and the COSY interval libraries optimally so that the efficiencies achieved for the DA and IN data types can be carried over to the TM data type. Refer to [6] for the theoretical background and implementation details. Furthermore, a binary output/input capability is newly added to COSY INFINITY to avoid any error growth associated to Taylor model file input and output. The binary I/O capability is supported for the RE and TM data types.

The Taylor model method provides sharp estimate while guaranteeing the result, and even only low-order Taylor models often perform better than sophisticated methods like the centered form and the mean value form [6,4]. Details can be found in Ref. [6], which also summarizes various Taylor model based algorithms and discusses some practical problems. The various Taylor model operations and intrinsics have been independently analyzed for rigor based on IEEE floating point standards [8] and subjected to extensive and challenging execution-based testing [9,10].

2.3. COSY-VI and COSY-GO

Verified integration of ODEs and global optimization require efficient computational methods with verification, and the method of Taylor models can be applied effectively. Based on the Taylor model implementation in COSY INFINITY, packages for those important applications are now available for release.

COSY-VI is the COSY Taylor model package for verified integration of ODEs, and the package offers various state of the art Taylor model algorithms for the task. The package also can be used for other types of problems like differential algebraic equations by reducing them to the form of ODEs [11]. Besides the general concern of controlling the dependency problem in verification problems, verified integrations of multidimensional ODEs exhibit a severe asymptotic overestimation problem of geometric nature, called the wrapping effect [4]. The striving for suppressing the wrapping effect has as long a history as the computer implemented interval method itself. The naive concept of Taylor models with multivariate polynomials allows not only the high-order Taylor expansion in time t, but also the high-order expansion in space variables $\vec{x}$. This feature enables a solution set at each integration time step to be enclosed by an nth order Taylor model, i.e. the set is approximated by an nth order polynomial while the approximation error is kept in a small remainder bound interval. Combined with preconditioning techniques, this approach much reduces the devastating wrapping effect. The COSY-VI package is further equipped with higher level algorithms like the method of shrink wrapping and various types of blunting for tighter control of error growth, details about which can be found in an exhaustive paper on the Taylor model ODE integrations [7]. In summary, the key features and algorithms of COSY-VI are

- High-order expansion not only in time but also in transversal variables.
- Capability of weighted order computation, allowing to suppress the expansion order in transversal variables.
- Shrink wrapping algorithm including blunting to control ill-conditioned cases.
- Pre-conditioning algorithms based on the Curvilinear, QR decomposition, and blunting pre-conditioners.
- Resulting data is available in various levels including graphics output.

COSY-GO is the COSY Taylor model package for global optimization. Different from intervals, Taylor models carry the information on local slope and convexity in the data structure. Utilizing the readily available information, various range bounding algorithms have been developed. The linear dominated bounder LDB provides fast multidimensional Taylor model range bounding, utilizing the linear part as a guideline on range enclosing and reducing the corresponding domain area. The quadratic dominated bounder QDB provides a thorough quadratic bounding of a multidimensional Taylor model by carrying out the convexity tests of the quadratic part. A v dimensional box has 3^v surfaces, consisting of 2^v one dimensional corner points, various higher dimensional surfaces, and finally the v dimensional box interior. Thus, a complete examination of stationary points for a v dimensional quadratic polynomial requires $3^v - 2^v$ tests if conducted naively, which becomes impractical quickly as v increases. The QDB bounder reduces the required efforts by utilizing the LDB and the efficient surface list handling, making high dimensional problems practically solvable. To facilitate the task of quadratic bounding for global optimization, a limited purpose quadratic bounder, the fast quadratic bounder QFB, is more practical. QFB is designed for a multidimensional Taylor model whose quadratic part is convex, which is characteristic of the most crucial bounding task, namely that of a Taylor model in the proximity of a local minimizer. This enables to eliminate the pure quadratic terms from the bounding task. The COSY-GO package is equipped with those state of the art Taylor model range bounding algorithms [12]. Since those quadratic bounders solve an infamous problem in global optimizations with verified methods, the so called cluster effect, COSY-GO makes various global optimization problems practically solvable, among them the long-term stability estimate of storage rings using Normal Form analysis, which is the original motivation of the development of Taylor models. The core features and algorithms of COSY-GO can be summarized as follows:

- List management of boxes not yet determined to not contain the global minimizer. Loading a new box. Discarding a box with range above the current threshold value. Splitting a box with range not above the threshold value for further analysis. Storing a box smaller than the specified size.
- Application of a series of bounding schemes, starting from mere interval arithmetic to naive Taylor model bounding, LDB, then QFB. A higher bounding scheme is executed only if all the lower schemes fail.
- Update of the threshold cutoff value via various schemes. It includes upper bound estimates of the local minimum by corresponding bounding schemes, the mid point estimate, global estimates based on local behavior of function using gradient line search and convex quadratic form.
- Box size reduction using LDB.
- Resulting data is available in various levels including graphics output.

3. Particle optical elements and analyzing tools

COSY INFINITY offers various methods for particle trackings via transfer maps that relate the initial condition $\vec{z}_{\rm i}$ to the final condition $\vec{z}_{\rm f}$ via $\vec{z}_{\rm f} = \mathscr{M}(\vec{z}_{\rm i})$. In the code, transfer map $\mathscr{M}$ is represented by the DA data type. For the purpose of gaining speed in computations, the VE data type is employed to represent the particle coordinates $\vec{z}$ in the tracking algorithm. Tracking can be performed not only in regular particle coordinates that provide intuitive understanding of the dynamics, but also in normal form coordinates that provide a means of quantitative analysis of the dynamics. When studying the dynamic aperture, it is important to be able to utilize appropriate symplectification. In the current version of COSY INFINITY, mere particle tracking without symplectification is performed by specifying the tracking mode $\text{TY} = -21$ for the command TR [13]. The symplectic tracking modes with generating functions of types F_1 to F_4 require the user to find the optimal generating function by trial and error for each problem. A new approach, the EXPO (The EXtended POincare generating function type), employs the optimal generating function for symplectification [14,15]. COSY INFINITY offers this method to promote the easy usage of symplectic trackings, and the feature is performed by specifying the tracking mode $\text{TY} = 0$. Another minor but useful tool is the command TRT, which allows the user to incorporate bookkeeping information in a tracking picture produced by the command TR. It is also possible to mark a specific particle in tracking pictures by coloring the particle via the command SR. Due to the additional memory consumption, this feature is turned off by default. To activate the feature, a few lines in the file cosy.fox have to be altered, which can be easily identified by searching a string "color."

Some of modern particle optical devices have large acceptance, for example those to be used for the various muon accelerator scenarios, where the difficulty and expense of cooling require the ability to manipulate a beam of unusually large emittance. This naturally has led to the usage of COSY INFINITY for such systems, because of the necessity of high-order nonlinear computations. As a result, several new algorithms and tools have been developed. This includes an extensive collection of solenoidal elements, efficient propagation of beams with tremendously large emittance, and treatment of dynamics of particle beams traveling through matter while experiencing scatterings. Refer to [16,17], for details.

4. Standard features and supported languages

With the rapid expansion of computer techniques in recent years, it is not a simple task to maintain a scientific computation code like COSY INFINITY with numerous users and a variety of computer environments, to adjust to newly emerging techniques and the disappearance of others. To efficiently confront this situation, we have strived to keep backward compatibility and portability of the code COSY INFINITY as much as possible in order to protect users from additional effort due to sudden code changes based on syntax modification.

There are some items worthwhile to mention in this paper about the current official distributions at the COSY web site cosy.pa.msu.edu. For the interactive graphics output purpose, the PGPLOT graphics library has been stable in the last years, and thus we keep the PGPLOT graphics drivers in COSY as the standard interactive graphics package [1]. On the other hand, the GKS graphics library is quickly becoming obsolete, so we demoted the GKS graphics drivers. The GKS drivers are merely commented in the code, so they are still easily available for the user. We keep the VGA graphics drivers for Lahey Fortran and the graPHIGS graphics drivers in the same commented form. The long swing between MicroSoft Windows PCs and Linux/UNIX for the COSY Fortran77 sources seems to be settled into "UNIX" version. It is because the COSY "PC" version was meant mostly for Lahey Fortran, and popular Fortran compilers lately available for MicroSoft Windows PCs are compatible with the COSY "UNIX" version.

Finally, for the increasing population and demand of non-Fortran77 languages, COSY INFINITY provides interface packages for Fortran90 and C++ to enhance portability. All the data types and the associated operations, functions and intrinsic procedures in COSY INFINITY are accessible via the interface packages as C++ classes for the C++ user, and as Fortran90 modules for the F90 user. These COSY interface C++ classes and Fortran90 modules outperform independent attempts of creating DA packages in C++ and Fortran90. Refer to [1,5] for details on the interface packages.

Acknowledgements

This work was supported by the US Department of Energy, the National Science Foundation, the German National Merit Foundation, an Alfred P. Sloan Fellowship, and the Illinois Consortium for Accelerator Research.

References

[1] M. Berz, J. Hoefkens, K. Makino, COSY INFINITY Version 8.1—programming manual, Technical Report MSUHEP-20703, Department of Physics and Astronomy, Michigan State University, East Lansing, MI, 2002, see also http://cosy.pa.msu.edu.

[2] M. Berz, Modern Map Methods in Particle Beam Physics, Academic Press, San Diego, 1999, also available at http://bt.pa.msu.edu/pub.

[3] M. Berz, Forward algorithms for high orders and many variables, in: A. Griewank, G.F. Corliss (Eds.), Automatic Differentiation of Algorithms: Theory, Implementation and Application, SIAM, Philadelphia, 1991.

[4] R.E. Moore, Methods and Applications of Interval Analysis, SIAM, Philadelphia, 1979.

[5] J. Hoefkens, Verified methods for differential algebraic equations, Ph.D. Thesis, Michigan State University, East Lansing, MI, USA, 2001.

[6] K. Makino, M. Berz, Int. J. Pure Appl. Math. 6 (3) (2003) 239 available at http://bt.pa.msu.edu/pub.

[7] K. Makino, M. Berz, Suppression of the wrapping effect by Taylor model based validated integrators, submitted. Also MSUHEP-40910, available at http://bt.pa.msu.edu/pub.

[8] N. Revol, K. Makino, M. Berz, J. Logic Algebraic Program. 64 (1) (2004) 135–154.

[9] G.F. Corliss, J. Yu, Interval testing strategies applied to COSY's interval and taylor model arithmetic, in: R. Alt, et al. (Ed.), Numerical Software with Result Verification, Lecture Notes in Computer Science, vol. 2991, Springer, Berlin, 2004, pp. 91–106.

[10] G.F. Corliss, Testing COSY's INSRF, Technical Report MSUHEP 31228, Michigan State University, 2003.

[11] J. Hoefkens, M. Berz, K. Makino, Verified high-order integration of DAEs and higher-order ODEs, in: W. Kraemer, J.W.v. Gudenburg (Eds.), Scientific Computing, Validated Numerics, Interval Methods, Kluwer, Boston, 2001, pp. 281–292.

[12] M. Berz, K. Makino, Y.K. Kim, Long-term stability of the Tevatron by verified global optimization, Nucl. Instr. and Meth. A, in press, doi:10.1016/j.nima.2005.11.035.

[13] M. Berz, K. Makino, COSY INFINITY Version 8.1—user's guide and reference manual, Technical Report MSUHEP-20704, Department of Physics and Astronomy, Michigan State University, East Lansing, MI, 2002, see also http://cosy.pa.msu.edu.

[14] B. Erdélyi, M. Berz, Phys. Rev. Lett. 87 (11) (2001) 114302.

[15] B. Erdélyi, M. Berz, Int. J. Pure Appl. Math. 11 (3) (2004) 241 available at http://bt.pa.msu.edu/pub.

[16] K. Makino, M. Berz, C.J. Johnstone, D. Errede, Nucl. Instr. and Meth. A 519 (2004) 162.

[17] M. Berz, K. Makino, C.J. Johnstone, Propagation of a large-emittance muon beam through a straight, quadrupole-based precooling channel, in: NuFact03, AIP Conference Proceedings, 2004.