C-XSC and Closely Related Software Packages

Werner Hofschuster[1], Walter Krämer[1], and Markus Neher[2]

[1] Scientific Computing/Software Engineering, Bergische Universität Wuppertal
D-42097 Wuppertal, Gaußstraße 20, Germany
{hofschuster, kraemer}@math.uni-wuppertal.de
[2] Institut für Angewandte und Numerische Mathematik, Universität Karlsruhe (TH)
D-76128 Karlsruhe, Germany
markus.neher@math.uni-karlsruhe.de

Keywords. Mathematical software, reliable computing, C-XSC.
MSC Subject Classifications: 68N30, 68N19, 65F99, 65G20, 65G30.

1 Introduction

C-XSC [11,12] is a C++ class library for verified computations, using interval arithmetic [2]. C-XSC has been developed and maintained for almost twenty years to enable and facilitate the implementation of reliable numerical methods on computers. Today, it is one of the most sophisticated software libraries available for this purpose (most wide-spread alternatives include INTLAB [19], filib++ [15] or the Boost interval arithmetic library [6]).

C-XSC is distributed under the terms of the GNU Library General Public License. The software and its source code are retrievable at the following site [1]:

`http://www.xsc.de`

Several platforms are supported: Linux, MS Windows, Mac OS and Sun Solaris [1]. Besides the cited literature, which describes the state of C-XSC at the time of publication, there is also a new online api documentation, generated by the documentation system Doxygen [1] and covering the latest evolution of the software.

2 Scope of C-XSC

Apart from the data types that are usually employed on computers (integers, reals, etc.), C-XSC provides advanced data types for verified computation: `interval`, `complex`, `cinterval` (complex interval data type), and the corresponding dotprecision data types `dotprecision` (for reals), `cdotprecision`, `idotprecision`, and `cidotprecision`. The dotprecision data types enable the evaluation of sums or scalar products of vectors with 1 ulp accuracy. This is achieved by a long accumulator, into which intermediate results of a scalar product are stored without roundoff errors.

Several kinds of multiple precision data types are also available: `l_real`, `l_interval`, `L_real`, `L_interval`, `DotK`, `IDotK`, etc. These types are implemented

Dagstuhl Seminar Proceedings 08021
Numerical Validation in Current Hardware Architectures
http://drops.dagstuhl.de/opus/volltexte/2008/1442

in the so-called staggered correction format [20]. Furthermore, corresponding vector and matrix types are provided for all scalar data types.

An extensive set of elementary mathematical functions for real and complex interval arguments of high accuracy is implemented in C-XSC, both for the basic interval types and for the multiple precision formats [1,11,12].

3 C-XSC Toolbox for Verified Computing and Related Software

The C-XSC toolbox contains procedures for the verified solution of various basic problems in scientific computing [9]:

- One-dimensional problems: Extended interval division, evaluation of polynomials, automatic differentiation, nonlinear equations of one variable, global optimization, accurate evaluation of arithmetic expressions, zeros of complex polynomials.
- Multi-dimensional problems: Linear systems of equations, linear optimization, automatic differentiation for gradients, Jacobians, and Hessians, nonlinear systems of equations, global optimization.

There is also a large collection of additional software for verified computation which relies on and which can be used in connection with the C-XSC library:

- One- and multidimensional interval Taylor arithmetic [3],
- one- and multidimensional (interval) slope arithmetic [5],
- CoStLy (complex interval standard functions library) [16],
- ACETAF (automatic computation of estimates for Taylor coefficients of analytic functions) [7],
- CLAVIS (classes for verified integration over singularities) [21],
- elementary functions of high accuracy [1],
- self-verifying solvers for dense systems of linear equations [10],
- FastILSS (fast verified solvers for dense linear (interval-)systems) [22],
- ParLinSys (solving parametric interval linear systems) [17],
- MPI extension for the use of C-XSC in parallel environments [8], and
- a modified staggered correction arithmetic with enhanced accuracy and very wide exponent range [4].

4 Future development of C-XSC

The development of C-XSC library is being continued. Possible future extensions include the implementation of containment sets [18], parallel solvers for sparse matrices, simplified output procedures, extended sets of numerical test cases, etc.

Finally, C-XSC could benefit from hardware support for interval arithmetic, which would make the computations much faster than in the current implementation. It has been pointed out frequently [13,14] that the increase of complexity in current processors would only be moderate. Hence, the authors hope that in the further development of floating-point standards, the time will be right for including hardware support for interval arithmetic.

Acknowledgment

We wish to thank all personal friends and colleagues (see http://www.math.uni-wuppertal.de/˜xsc/xsc/history.html) who have been contributing to the development of C-XSC.

References

1. Download: http://www.xsc.de/
   http://www.math.uni-wuppertal.de/˜xsc/
2. Alefeld, G.; Herzberger, J.: Introduction to Interval Computations. Academic Press, New York, 1983.
3. Blomquist, F.; Hofschuster, W.; Krämer, W.: Real and complex Taylor arithmetic in C-XSC. Preprint 2005/4, Universität Wuppertal, 2005.
4. Blomquist, F.; Hofschuster, W.; Krämer, W.: Real and complex staggered (interval) arithmetic with wide exponent range (in German). Preprint 2008/1, Universität Wuppertal, 2008.
5. Bräuer, M.; Hofschuster, W.; Krämer, W.: Steigungsarithmetiken in C-XSC. Preprint 2001/3, Universität Wuppertal, 2001.
6. Brönnimann, H.; Melquiond, G.; Pion, S.: The design of the Boost interval arithmetic library. Theor. Comput. Sci., **351** (2006), 111–118.
7. Eble, I., Neher, M.: ACETAF: A software package for computing validated bounds for Taylor coefficients of analytic functions, ACM TOMS **29** (2003), 263–286.
8. Grimmer, M., Krämer, W.: An MPI extension for the use of C-XSC in parallel environments. Int. Conf. on Scientific Computing (CSC'07, Worldcomp'07) Las Vegas, June 25-28, 2007, Proceedings pages 111 to 117, Arabnia et al. (editors), 2007.
9. Hammer, R.; Hocks, M.; Kulisch, U.; Ratz, D.: C++ Toolbox for verified computing – basic numerical problems. Springer-Verlag Berlin Heidelberg, 1995.
10. Hölbig, C. A., Krämer, W., Diverio, T. A.: An accurate and efficient self-verifying solver for systems with banded coefficient matrix. In: Parallel Computing: Software Technology, Algorithms, Architectures & Applications.1 ed. Amsterdam : Elsevier Science B.V., pp. 283–290, 2004.
11. Hofschuster, W.; Krämer, W.: C-XSC 2.0: A C++ library for extended scientific computing. In: Numerical Software with Result Verification, Lecture Notes in Computer Science, **2991**, Springer-Verlag, Heidelberg, pp. 15–35, 2004.
12. Klatte, Kulisch, Wiethoff, Lawo, Rauch: C-XSC – A C++ Class Library for Extended Scientific Computing, Springer-Verlag, Heidelberg, 1993.
13. Kirchner, R., Kulisch, U.: Hardware support for interval arithmetic. Reliable Computing, **12** (2006), 225–237.
14. Kulisch, U.: Computer Arithmetic and Validity – Theory, Implementation. To appear, 2008.

15. Lerch, M.; Tischler, G.; Wolff von Gudenberg, J.; Hofschuster, W; Krämer, W.: filib++, a fast interval library supporting containment computations. ACM TOMS **32** (2006), 299–324.
16. Neher, M.: Complex standard functions and their implementation in the CoStLy library. ACM TOMS **33** (2007), 20–46.
17. Popova, E.; Krämer, W.: Inner and outer bounds for the solution set of parametric linear systems. J. Comp. Appl. Math. **199** (2007), 310–316.
18. Pryce, J.: Interval arithmetic with containment sets. Computing **78** (2006), 251–276.
19. Rump, S.: INTLAB – INTerval LABoratory. In Csendes, T,. editor, *Developments in Reliable Computing*, pp. 77–104. Kluwer, Dordrecht, 1999.
20. Stetter, J.: Sequential defect correction for high-accuracy floating-point arithmetic. In: Numerical Analysis (Proceedings, Dundee 1983), Lecture Notes in Mathematics **1066** (1984), 186–202.
21. Wedner, S.: Verifizierte Bestimmung singulärer Integrale – Quadratur und Kubatur. Dissertation, Universität Karlsruhe, 2000.
22. Zimmer, M.: Laufzeiteffiziente, parallele Löser für lineare Intervallgleichungssysteme in C-XSC. Master thesis, University of Wuppertal, 2007.