Improving the Performance of a Verified Linear System Solver Using Optimized Libraries and Parallel Computation

Mariana Kolberg, Gerd Bohlender and Dalcidio Claudio

Pontifícia Universidade Católica do Rio Grande do Sul, PPGCC
Av. Ipiranga, 6681 – 90619-900 – Porto Alegre, Brazil
Phone: +55 51 3320-3611 Fax: +55 51 3320-3621
Universität Karlsruhe, Institut für Angewandte und Numerische Mathematik
Kaiserstr. 12 – 76128 – Karlsruhe, Germany
Phone: +49 721 608 2049 Fax: +49 721 608 6679
{mkolberg,dalcidio}@inf.pucrs.br, bohlender@math.uka.de
topics of the conference list: Numerical algorithms for CS&E, Parallel or Distributed Computation and Cluster Computation.

Abstract. A parallel version of the self-verified method for solving linear systems was presented in [16, 15]. In this research we propose improvements aiming at a better performance. The idea is to implement an algorithm that uses technologies as MPI communication primitives associated to libraries as LAPACK, BLAS and C-XSC, aiming to provide both self-verification and speed-up at the same time. The algorithms should find an enclosure even for very ill-conditioned problems. In this scenario, a parallel version of a self-verified solver for dense linear systems appears to be essential in order to solve bigger problems. Moreover, the major goal of this research is to provide a free, fast, reliable and accurate solver for dense linear systems.

Many real problems are simulated and modeled using dense linear systems of equations like $Ax = b$ with an $n \times n$ matrix $A \in \mathbb{R}^{n \times n}$ and a right hand side $b \in \mathbb{R}^n$. This is true for functional linear equations that occur like partial differential equations and integral equations that appear in several problems of Physics and Engineering [4]. Many different numerical algorithms contain this task as a subproblem.

There are numerous methods and algorithms which compute approximations to the solution x in floating-point arithmetic. However, usually it is not clear how good these approximations are, or if there exists a unique solution at all. In general, it is not possible to answer these questions with mathematical rigor if only floating-point approximations are used. These problems become especially difficult if the matrix A is ill conditioned. The use of self-verified methods can lead to more reliable results [9]. Verified computing provides an interval result that surely contains the correct result [17, 14]. Like that the algorithm also proves the existence and uniqueness of the solution of the problem. The algorithm will, in general, succeed in finding an enclosure of the correct solution. If the solution is not found, the algorithm will let the user know. One possibility to implement verified computing is using interval arithmetic combined with suitable algorithms.

There is a multitude of tools and algorithms that provide verified computing. Among them, an option is C-XSC (C for eXtended Scientific Computing) [14]. C-XSC is a free and portable programming environment for C and C++ programming languages, offering high accuracy and automatic verified results. This programming tool allows the solution of many standard problems with reliable results. The Matlab toolbox for self-verified algorithms, INTLAB [12], is also an option. Like C-XSC, it also provides interval arithmetic for real and complex data including vectors and matrices, interval arithmetic for real and complex sparse matrices, rigorous real interval standard functions, rigorous complex interval standard functions, rigorous input/output, accurate summation, dot product and matrix-vector residuals, multiple precision interval arithmetic with error bounds, and more. However, this solver can be used just together with the commercial MATLAB environment, what can increase the costs to prohibitive values. To understand performance and accuracy issues, a comparison among C-XSC, INTLAB and LAPACK [1] was performed. LAPACK is a well known linear algebra package, considered very fast and optimized.

The results show that C-XSC has the most reliable results and the highest accuracy. LAPACK is the one that presents the best performance, but results are not verified, and in some cases less accurate. INTLAB is the best compromise between performance and accuracy. However, as said before, it requires Matlab which is not free. The tests show that the method used in C-XSC is a good choice, but it should be optimized to gain performance.

The method for solving linear systems with high accuracy can be found in [14, **?**], where the verified method for solving linear system using the C-XSC library is based on the enclosure theory described in [25].

These enclosure methods are based on the following interval Newton-like iteration:

$$x_{k+1} = Rb + (I - RA)x_k, k = 0, 1, ... \tag{1}$$

This equation is used to find a zero of $f(x) = Ax - b$ with an arbitrary starting value x_0 and an approximate inverse $R \approx A^{-1}$ of A. If there is an index k with $[x]_{k+1} \overset{\circ}{\subset} [x]_k$ (the $\overset{\circ}{\subset}$ operator denotes that $[x]_{k+1}$ is included in the interior of $[x]_k$), then the matrices R and A are regular, and there is a unique solution x of the system $Ax = b$ with $x \in [x]_{k+1}$. We assume that $Ax = b$ is a dense square linear system and we do not consider any special structure of the elements of A.

The use of verified computing makes it possible to find the correct result. However, finding the verified result often increases the execution times dramatically [20]. The research already developed show that the execution time of verified algorithms are much larger than the execution time of algorithms that do not use this concept [11, 10].

To compensate the lack of performance of such verified algorithms, some works suggest the use of midpoint-radius arithmetic to achieve a better performance, since its implementation can be done using only floating-point operations [23, 24]. This would be a way to increase the performance of verified methods.

The advent of parallel computing and its impact in the overall performance of many algorithms on numerical analysis can be seen in the past years [6]. The use of clus-

ters plays an important role in such a scenario as one of the most effective manner to improve the computational power without increasing costs to prohibitive values. The parallel solutions for linear solvers found in the literature explore many aspects and constraints related to the adaptation of the numerical methods to high performance environments [5, 22, 8, 28, 18, 21]. However, those implementations do not deal with verified methods. In the field of verified computing many important contributions have been done in the last years. Some works related to verified solvers for dense linear systems [7, 14, 9, 26, 19] can be found in the literature. However, only a few papers on verified solvers for dense systems together with parallel implementations were found [13, 27, 24], but these authors implement other numerical problems or use a parallel approach for other architectures than clusters.

The self-verified method presented above is divided in several steps. By tests, the computation of R, the approximate inverse of matrix A, takes more than 50% of the total processing time. Similarly, the computation of the interval matrix $[C]$ that contains the exact value of $I - RA$ (iterative refinement) takes more than 40% of the total time, since matrix multiplication requires O(n^3) execution time, and the other operations are mostly vector or matrix-vector operations which require at most O(n^2). Due to this, both parts of the algorithm were optimized using mathematical concepts and parallelization. The evaluation of $[C]$ is composed of operations that are suitable for parallelization, since all components can be computed independently.

A parallel version of the self-verified method for solving linear systems was presented in [16, 15]. In this research we propose the following improvements aiming at a better performance:

- Calculation of R using just floating-point operations;
- Avoid the use of C-XSC elements that could slow down the execution;
- Use of the fast and highly optimized libraries: BLAS and LAPACK in the first sequential version (for the parallel version PBLAS and SCALAPACK respectively);
- Use of both interval arithmetics: infimum-supremum and midpoint-radius (as proposed by Rump [24]);
- Use of techniques to avoid the switching of rounding mode (proposed by Bohlender [2, 3]).

The new algorithms should find an enclosure even for very ill-conditioned problems. Moreover, the major goal of this research is to provide a free, fast, reliable and accurate solver for dense linear systems.

New algorithms based on the C-XSC methods were implemented, but using just libraries like BLAS and LAPACK to achieve better performance. The idea of reducing the switching of rounding mode presented by Bohlender was implemented as well as an optimization of the residuum based on the INTLAB method. In other words, the new implementations try to join the best aspects of each library.

To ensure that an enclosure will be found, interval arithmetic was used. To find the best arithmetic for this method, the algorithms for point and interval input data were written using both infimum-supremum and midpoint-radius arithmetic. Until now, the implemented algorithms show a good performance.

Aiming at a better performance, the algorithms are being parallelized using the libraries SCALAPACK and PBLAS. The idea of using popular and highly optimized libraries to gain performance will also be maintained in the parallel version.

One important advantage of the presented algorithm is the ability to find a solution even for very ill-conditioned problems (in tests on personal computers an enclosure could be found for condition number up to 10^{15}) while most algorithms may lead to an incorrect result when it is too ill-conditioned (above condition number 10^8). Our main contribution is to increase the use of verified computing through its optimization and parallelization, once without parallel techniques it becomes the bottleneck of an application.

References

1. E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.
2. G. Bohlender. Faster Interval Computations Though Minimized Switching of Rounding Modes. 2002. presented at PUCRS, Porto Alegre, Brazil.
3. G. Bohlender, M. Kolberg, and D. Claudio. Modifications to Expression Evaluation in C-XSC. Technical report, preprint BUW-WRSWT 2005/5, Universität Wuppertal, DE, 2005. presented at SCAN04, Fukuoka, Japan.
4. D. M. Claudio and J. M. Marins. *Cálculo Numérico Computacional: Teoria e Prática*. Editora Atlas S. A., São Paulo, 2000.
5. J.J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A van der Vorst. *Numerical Linear Algebra for High-performance Computers*. SIAM press, Philadelphia, USA, 1998.
6. I. S. Duff and H. A. van der Vorst. Developments and Trends in the Parallel Solution of Linear Systems. Technical Report RAL TR-1999-027, CERFACS, Toulouse, France, 1999.
7. A. Facius. *Iterative solution of linear systems with improved arithmetic and result verification*. PhD thesis, University of Karlsruhe, Germany, 2000.
8. A. Frommer. *Lösung linearer Gleichungssysteme auf Parallelrechnern*. Vieweg-Verlag, Universität karlsruhe, Germany, 1990.
9. R. Hammer, D. Ratz, U. Kulisch, and M. Hocks. *C++ Toolbox for Verified Scientific Computing I: Basic Numerical Problems*. Springer-Verlag, New York, Inc., Secaucus, NJ, USA, 1997.
10. C. A. Hölbig, P. S. Morandi Júnior, B. F. K. Alcalde, and T. A. Diverio. Selfverifying Solvers for Linear Systems of Equations in C-XSC. In *Proceedings of Parallel and Distributed Programming (PPAM)*, volume 3019, pages 292–297, 2004.
11. C. A. Hölbig, W. Krämer, and T. A. Diverio. An Accurate and Efficient Selfverifying Solver for Systems with Banded Coefficient Matrix. In *Proceedings of Parallel Computing (PARCO)*, pages 283–290, Germany, September 2003.
12. INTLAB. INTerval LABoratory. http://www.ti3.tu-harburg.de/ rump/intlab/.
13. T. Kersten. *Verifizierende rechnerinvariante Numerikmodule*. PhD thesis, University of Karlsruhe, Germany, 1998.
14. R. Klatte, U. Kulisch, C. Lawo, R. Rauch, and A. Wiethoff. *C-XSC- A C++ Class Library for Extended Scientific Computing*. Springer-Verlag, Berlin, 1993.
15. M. Kolberg, L. Baldo, P. Velho, L. F. Fernandes, and D. Claudio. Optimizing a Parallel Self-verified Method for Solving Linear Systems. In *PARA - Workshop on State-of-the-art in Scientific and Parallel Computing*, 2006.

16. M. Kolberg, L. Baldo, P. Velho, T. Webber, L. F. Fernandes, P. Fernandes, and D. Claudio. Parallel Selfverified Method for Solving Linear Systems. In 7^{th} *VECPAR - International Meeting on High Performance Computing for Computational Science*, pages 179–190, Rio de Janeiro, Brazil, 2006.
17. U. Kulisch. and W. L. Miranker. *Computer Arithmetic in Theory and Practice*. Academic Press, New York, 1981.
18. G. C. Lo and Y. Saad. Iterative Solution of General Sparse Linear Systems on Clusters of Workstations. Technical Report umsi-96-117, msi, uofmad, 1996.
19. T. Ogita, S. M. Rump, and S. Oishi. Accurate Sum and Dot Product with Applications. In *2004 IEEE International Symposium on Computer Aided Control Systems Design*, LNCS, pages 152–155, Taipei, Taiwan, September 2004. IEEE Press.
20. T. Ogita, S. M. Rump, and S. Oishi. Accurate Sum and Dot Product. *SIAM Journal on Scientific Computing*, 26(6):1955–1988, 2005.
21. J. C. Cabaleiro P. Gonzalez and T. F. Pena. Solving Sparse Triangular Systems on Distributed Memory Multicomputers. In *Proceedings of the Sixth Euromicro Workshop on Parallel and Distributed Processing*, pages 470–478. IEEE Press, January 1998.
22. V. Pan and J. Reif. Fast and Efficient Parallel Solution of Dense Linear Systems. *Computers & Mathematics with Applications*, 17(11):1481–1491, 1989.
23. S. Rump. INTLAB - INterval LABoratory. *Developments in Reliable Computing*, pages 77–104, 1998. http://www.ti3.tu-harburg.de/ rump/intlab.
24. S. Rump. Fast and Parallel Interval Arithmetic. *Bit Numerical Mathematics*, 39(3):534–554, 1999.
25. S. M. Rump. *Kleine Fehlerschranken bei Matrixproblemen*. PhD thesis, University of Karlsruhe, Germany, 1980.
26. S. M. Rump. Solving Algebraic Problems with High Accuracy. In *IMACS World Congress*, pages 299–300, 1982.
27. A. Wiethoff. *Verifizierte globale Optimierung auf Parallelrechnern*. PhD thesis, University of Karlsruhe, Germany, 1997.
28. J. Zhang and C. Maple. Parallel Solutions of Large Dense Linear Systems Using MPI. In *International Conference on Parallel Computing in Electrical Engineering, PARELEC '02*, pages 312–317. IEEE Computer Society Press, 2002.