General Activity Report From EST3 Fellowship

Project : Marie Curie Host Fellowship: Research Training at an EaSTern European Site with Tradition in Computational Science and Engineering-EST3.

Author : Kandula Hanumantha Rao

Affiliation(s): Laboratorul de Metode Numerice-LMN, CIEAC Universitatea Politechnica Bucuresti.

Author's Signature

General Activity Report from the EST3 Fellowship

Author: Kandula Hanumantha Rao

Between 27-09-2007 and 27-01-2008 I had a chance to take advantage of the project called Research Training at an EaSTern European Site with Traidition in Computational Science and Engineering-EST3 within the Marie Curie Fellowship in Politechnical University of Bucharest, Romania.

I have spent the period under scientific supervision of Prof. Daniel Ioan at the computational Electromagnetics Group, LMN laboratory (Numerical Method Lab). During this time I was involved in chameleon RF and other LMN activities. I had an open access to all the local resources, including library, electronic archives and an account on the LMN server.

From the scientific point of view, I had an opportunity to learn about modern methods developed there to build electromagnetic models of passive nano-electronic devices. I gathered also knowledge considering all aspects of such procedures. From my part , I had started researches on Evaluate existing solvers (direct / iterative) for the solving of complex linear systems of equations from the modeling of passive on chip components which is mentioned in scientific report. To make it possible I was given a possibility to use LMN computers and dedicated software.

The time of my research training let me experience reality of working, doing researches and communicating in the international scientific environment, which helped me in improving my abilities to prepare presentations and write scientific papersin a foreign languages. Every opportunity to talk and exchange thoughts and ideas with other scientists was also very valuable. I have received also sufficient scientific support in formulating my research field of my interest. During this time I have learned a lot about the country, its history and culture, quite different from my own.

Last but not least it was the time when I met nice people not onlyfrom Romania but from all over the Europe. I had a chance to meet researchers as well as foreign students.

I think it was a great experience for me , not only in a scientific but also social way. I hope that everything I have learned in Bucharest will help me in developing my future career.

Project : **EST3**

Title : Evaluation of solvers for sparse complex linear systems of equations, arising from the modeling of passive on-chip components.

Author(s) : Kandula Hanumantha Rao

Affiliation(s): Laboratorul de Metode Numerice- LMN, CIEAC, Universitatea Politechinca Bucureti, www.lmn.pub.ro

Acknowledgements

I would like to thank for the scientific support I have received from **Prof**. **Daniel Ioan,** www.lmn.pub.ro/~daniel , my research supervisor at LMN.

The work was financed by **Marie Curie EST3 Project** http://est3.lmn.pub.ro/ .

I am thankful to my wife Anuradha Devi and my parents for their faith in me and moral support they had given me.

Finally, I thank one and all of LMN team for their co-operation.

Evaluation of solvers for sparse complex linear systems of equations, arising from the modeling of passive on-chip components

Kandula Hanumantha Rao, Bucharest

Abstract

In this report study and evaluation of solvers for sparse complex linear systems and the modelling of passive on-chip components are presented. And also explained types of solvers for the sparse complex linear systems are discussed.

1. Introduction

In various application fields such as fluid dynamics and structural analysis, industry is increasingly relying on computer simulations. The physical structures to be analyzed on the computer are discretized by means of complex grids. The finer the resolution of these grids, the higher the accuracy of the corresponding numerical simulation. Unfortunately, however, increasing the grid resolution also increases the size of the corresponding (sparse) systems of equations which have to be solved numerically. Problems with many millions of degrees of freedom (unknowns) are being tackled nowadays and grid sizes will substantially grow further in the near future. The resulting large systems of equations can no longer be solved efficiently with standard numerical approaches such as conjugate gradient combined with classical preconditioners. Instead, hierarchically operating solvers are required.

The classical multigrid or multilevel approach, representing one of the major work areas of SCAI, was the first hierarchical approach to reach maturity. Rather than operating merely on the given (very fine) grid, corresponding methods combine the numerical information resulting from a (pre-defined) hierarchy of increasingly coarse grids. Compared to classical one-level solvers, the main advantage of properly designed multilevel approaches is their numerical scalability. That is, the computational work to solve the underlying systems of equations grows only proportionally with the number of unknowns. Depending on the concrete application, the computational gain may be enormous, and it increases further with increasing problem size.

2. Theory/Background Material

complex unsymmetric sparse systems of linear equations, say

1) Ax=b, within multithreaded applications.
2) Direct solvers work directly with the matrix A, iterative algorithms solve a related problem PAx= Pbwhere P is a pre-conditioned matrix.
3) The solution of a linear system is a four-step process.
4) Define the linear system Ax=b and how it will be solved.
5) Pre-order the matrix A to improve the effectiveness of static and diagonal pivoting. (optional)
6) Factor either the coefficient matrix A or the pre-conditioned matrix P.
7) Solve for Ax=b or PAx=Pb

We have three objectives when writing linear system solvers , they are Time, memory, and accuracy.This is a multi-objective optimization problem with no one best solution but several solutions with different trade-offs. Extremely large linear systems bring real challenges and the direct solvers run out of memory where as Iterative solvers converge slowly if at all.

3. Types of Solvers:

There are two types of solvers: direct and iterative, each with its own advantages and limitations.

i) **<u>Direct Solvers</u>**:

BERKELEY: A slightly improved version of the original SPARSE code original sparse solver written by Kenneth S. Kundertat Berkeley University. Real linear systems with LU factorization and modified Markowitzpivoting.

SPARSE: A completely re-engineered sparse solver for real and complex linear systems. Uses different data structures for different steps of the computation. Five pivoting algorithms that can be mixed into 16 pivoting strategies. Two variants: one where every element created can always be accessed directly by the user (akapersistent data) and a more memory efficient one. The pivotlessvariant (refactor) is parallel.

FAST: based on a pivotlessLU factorization of the matrix preceded by a symbolic factorization to estimate the amount of fill and schedule the diagonal pivots. Two re-ordering algorithms to control the amount of fill and the pivoting order: Minimum Degree (MD) and Reverse Cuthill-McKee (RCM). The numerical factorization part is parallel.

ii) **Iterative Solvers**:

BCGST: An improved variant of the Enhanced BiCGstab(L) algorithm by Fokkema, Sleijpenand van derVorst.

GMRES: an implementation of the algorithm by Y. Saad. It can solve real and complex linear systems.

Preconditioning: Two preconditionersare available: one is ILU(k): an incomplete LU factorization a specified number k of fill-ins, and the other isILU(p): an incomplete LU factorization with a threshold on the size of fill-ins retained

The solution of a linear system involves three steps: pre-order, factor of A or P, and solve

1) The pre-ordering step arranges the matrix to improve the effectiveness of diagonal pivoting. This step can also remove structural zeros if required.
2) The factor step Once the pivoting order and schedule are known from the first factorization, fast threaded specialized routines are available for subsequent factorization of similarly structured matrices.
3) The solve step is fast for direct solvers and may accounts for the bulk of the work for iterative solvers.

SPARSE: User requests a pointer to each element and these pointers remain valid until the matrix is destroyed .Elements of real and complex matrices can also be entered one by one directly into a data structure built by the solverliblibrary.Elements must be indexed for their addresses to remain valid when fill-in elements appear .

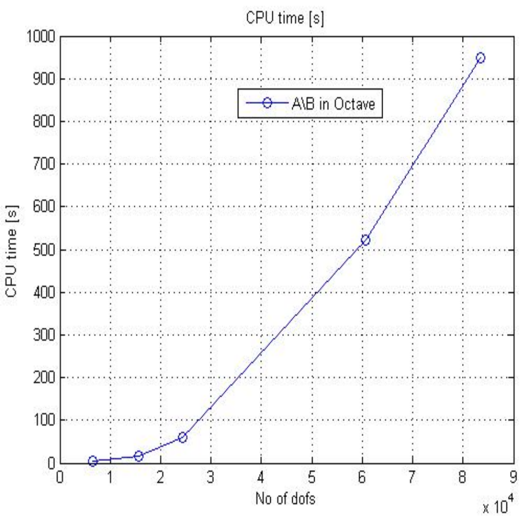
The following points are to close the gap between the direct and iterative solvers

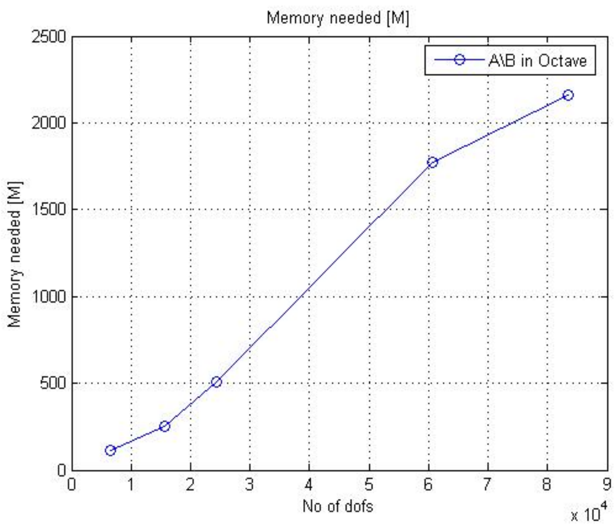
i) Better pre-ordering algorithms can make FAST more reliable and speeds up the convergence of iterative solvers GMRES and BCGST.
ii) Tidied up numerics, use uniform tests for null pivots.
iii) Faster pivoting for SPARSE on large matrices by using a priorityqueue for pivot candidates along the diagonal.
iv) Tighter memory management with different adaptive data structures for different parts of the SPARSE solver.

4. Results Expected :

The following results are expected by extending any one of the software tool for that The resources needed to solve problems are as follows. This is the reference result you can use. You should enlarge the table (and graphs) with as many tests (for other solvers, implementations) .

Test no	1	2	3	4	5	6
No of dofs	**6608**	**15620**	**24465**	**60658**	**83410**	101126
No of in/outs	2	2	2	2	2	2
CPU time [s]	**3**	**17**	**60**	**523**	**950**	Failure of solver
Memory used [M] during computation	**110**	**254**	**505**	**1770**	**2162**	Failure of solver





5. Conclusion and Future work:

This report is study and evaluation of the solvers for sparse complex linear systems of equations, arising from the modeling of passive on-chip components. I tried to get the expected results using C++ ,but they were not reach the correct results and we have to develope the software tool using the given conditions for the expected results.

6. References:

[1]. http://www.silvaco.co.jp/pdf/pdf_dwnld/wrkshp2007

[2]. http://www.scai.fhg.de

[3]. http://www.ercim.org/publication/Ercim_News/enw50/stueben.html

[4]. DUFF, I.S. Practical comparisions of codes for the solution of sparse linear systems. In Sparse Matrix Proceedings 1978.

[5]. D. Ioan, M.Radulescu, G.Ciuprina: Scientific Computing in Electrical Engineering,Vol. 4, 248–256, Springer (2004)