A Parallel Algebraic Multigrid through Lifting Technique, with Application in Circuit Simulation Matrices

Fabio H. Pereira[a,b] and Silvio I. Nabeta[b]
[a]Industrial Engineering Post Graduation Program, Nove de Julho University – PMEP/UNINOVE
[b]Electrical Machine and Drivers Lab – São Paulo University – GMAcq/USP
Francisco Matarazzo, Av. 612, 05001100, São Paulo - Brazil
fabiohp@uninove.br; nabeta@pea.usp.br

Abstract — Especially in the context of large scale problems and massively parallel computing, the most desirable property of the multigrid approaches is its potential for algorithmic scalability. So, the need to solve linear systems arising from problems posed on extremely large, unstructured grids has been generating great interest in parallelizing algebraic multigrid. However, the current approaches have some limitation, mainly regarding the coarsening process in setup phase. This paper presents a new parallel algebraic multigrid through lifting technique that eliminates the grid coarsening process simplifying its implementation on distributed memory machines. The method is used as a parallel black box solver in some numerical problems concerning to circuit simulation matrices.

I. Introduction

Although its efficiency for solving large sparse linear systems of algebraic equations arising from irregular domains and unstructured meshes, the use of algebraic multigrid method (AMG) presents some drawbacks, especially regarding to the coarsening process. The traditional coarsening scheme can lead to computational complexity growth as the problem size increases, resulting in an elevated memory use and execution time, and in a reduced scalability [1]. Moreover, the coarsening process is inherently sequential in nature that makes difficult the implementation on distributed memory machines [2].

Over the last few years, several works have exploited the similarities between the multigrid methods and wavelets trying to overcome these difficulties. The use of the discrete wavelet transform in the construction of the matrices hierarchy and the transfer operators in the AMG method was proposed by these authors in [3], producing a new method called WAMG that eliminates the standard coarsening process. A parallel version was also proposed [4], but despite the ease of implementation, it still presented some difficulties related to the size of the column vectors in each processor, which should be equal to 2^k, k in Z.

The method has revealed to be very efficient and promising for several problems related to the computation of electromagnetic fields and, consequently, further researches have been carried out for its improvement [5]-[6]. One of those works was presented recently in [6] which proposed the use of the lifting scheme [7] to accomplish the wavelet transform in the multigrid. The lifting technique has some numerical advantages in relation to the standard wavelet transform, as a reduced number of floating point operations and, especially, the capacity of transforming signals (vectors) with an arbitrary length (need not be 2^k), which is very appropriate in the WAMG context and allows us to solve the difficult of the parallel algorithm. This next natural step is presented in this paper.

II. Wavelet Transform and the Lifting Technique

The standard discrete wavelet transform (dwt) corresponds to the application of low-pass and high-pass filters, followed by the elimination of one out of two samples (decimation or sub sampling). The discrete signal, which in one dimension is represented by a vector of values, is filtered by a set of digital filters that are associated to the wavelet adopted in the analysis. Then, starting from a vector $y(N)$ at level 0, two sets of coefficients are generated in each level l of the process: a set d_l of wavelets coefficients (detail coefficients) and a set c_l of approximation coefficients. This procedure can be applied again, now using c_l as an input vector to create new coefficients c_{l+1} and d_{l+1}, and so on.

In 2-D case, the dwt is obtained through the application of successive steps of this 1-D transform into the rows and columns of the matrix. In both cases, one and two dimensions, the approximation coefficients keep the most important information of the discrete signal, whereas the detail coefficients possess very small values, next to zero.

In [7] Sweldens has shown that every wavelet filter can be decomposed into lifting steps. Some advantages of this technique over the classical wavelet transform are:

a) Smaller memory;

b) Efficiency: reduced number of floating point operations);

c) Parallelism: inherently parallel feature;

d) Transforms signals with an arbitrary length: need not be 2^k.

The representations of the Daubechies 2 wavelets in the lifting form are presented in equation (1) [8]

$$\begin{aligned} d_{l+1}^{(1)}[n] &= c_l[2n+1] - c_l[2n], \\ c_{l+1}^{(1)}[n] &= c_l[2n] + \frac{1}{2}\cdot d_{l+1}^{(1)}[n], \\ c_{l+1}[n] &= \sqrt{2}\cdot c_{l+1}^{(1)}[n], \\ d_{l+1}[n] &= \frac{1}{\sqrt{2}}\cdot d_{l+1}^{(1)}[n]. \end{aligned} \tag{1}$$

The coefficients $c_{l+1}[n]$ and $d_{l+1}[n]$ in (1) are, respectively, the approximation and the detail coefficients, at level l+1, of the input signal.

III. The Algebraic Multigrid through Lifting Technique

The key point of the proposed method is the application of an incomplete lifting transform to generate the hierarchy of matrices in the algebraic multigrid. The approximation coefficients keep the most important information of the discrete signal, which is essentially a low resolution version of the signal and represent a coarse version of the original data. So, the procedure defined by first tree equations in (1) can be used as an algebraic restriction operator in the multigrid context. The prolongation operator is defined using the inverse lifting transform, which is easy to find and it has exactly the same complexity as the forward transform, and the coarse matrix is calculated applying the procedure in (1) on the rows and columns of the original matrix [8].

IV. The Parallel algorithm using Lifting

The parallelization strategy starts dividing equally the rows of the matrix among the processors, in such a way that the 1-D transform defined in (1) may be applied efficiently in the rows. As the lifting allows transforming signals with an arbitrary length, the transform also can be applied in the part of the columns in each processor without any communication. It means the resulting coarse matrix will be calculated entirely locally.

In the solver phase the communications among processors are necessary only for operations involving matrices. This task is accomplished by using the MPI collective communication function MPI_Allgather [9].

V. The Numerical Results

The parallel algorithm uses the version one of the Message Passing Interface (MPI) that provides a standard for message passing for parallel computers and workstation clusters. The method has been implemented using in C++ and tested in a homogeneous Beowulf cluster with 5 machine nodes (Core 2 Duo, 2Gb RAM) connected to the switch with fast Ethernet network.

A hybrid Jacobi-Gauss method was used as smoother and the V(1,0)-cycle was applied as a resolution scheme to solve two systems of linear equations with circuit simulations matrices from [10] (Table I).

TABLE I
CHARACTERISTICS OF TEST MATRICES. THE DIMENSION OF THE PROBLEM IS *NN* AND THE NUMBER OF NONZERO IS *NNZ*

Matrices	*NN*	*NNZ*
trans4	116835	749800
trans5	116835	749800

The results are presented in Table II. In all cases, the convergence is defined by $\|r^n\|/\|b\| < 10^{-4}$, where r^n is the residual vector at the nth iteration and the right hand side vector b is chosen so that the solution is a unitary vector. The results report only the CPU time, which is fair once processes that are waiting for synchronization are still consuming full CPU-time. Moreover, as there is no interprocessor communications in the setup phase, the CPU-time and the wall clock time are the same.

TABLE II
NUMERICAL RESULTS FOR SETUP TIME (*tm*), SOLVER TIME (*ts*) AND NUMBER OF ITERATIONS (*n*)

Problem	Sequential		Parallel Method				
	ILU and BiCGStab	Lifting AMG	2	3	4	5	
trans4	901.3 235.1 11	30.3 356.6 26	15.6 311.2 2	12.9 314.2 2	11.9 310.7 2	11.5 311.7 2	*tm* *ts* *n*
trans5	784.2 1941.7 56	30.3 171.4 134	15.6 303.6 2	12.9 375.4 2	12.0 482.6 2	11.5 460.1 2	*tm* *ts* *n*

VI. Conclusions and Comments

The parallel algorithm uses a hybrid Jacobi gauss-seidel smoother that has not been reproduced in the sequential version of the method. This can help us explain the difference in the number of iterations. A possible influence of the coarse grid systems dimensions in each processor also should be investigated and it will be discussed in the full paper.

On the other hand, the fast Ethernet network shows to be an important bottleneck, which was aggravated by the load unbalance (test matrices has a large number of nonzero elements in the first rows). A new way to divide the matrix among the processors is under investigation.

VII. References

[1] G. Haase, M. Kuhn, and S. Reitzinger, "Parallel Algebraic Multigrid Methods on Distributed Memory Computers", *SIAM J. Sci. Comput.*, vol.24, pp. 410-427, 2002.

[2] H.D. Sterck, U.M. Yang, and J.J. Heys, "Reducing Complexity in Parallel Algebraic Multigrid Preconditioners", *SIAM J. Matrix Anal. Appl.*, vol.27, no. 4, pp. 1019-1039, 2005.

[3] F.H. Pereira, *et al.*, "A Wavelet-based Algebraic Multigrid Preconditioning for Iterative Solvers in Finite Element Analysis". *IEEE Trans. on Magn.*, vol. 43, no. 4, pp. 1553-1556, 2007.

[4] F.H. Pereira, M.F. Palin, S.L.L. Verardi, S.I. Nabeta, "A Parallel Wavelet-based Algebraic Multigrid black-box Solver and Preconditioner", *16th Compumag - Conference on the Computation of Electromagnetic Fields*, 2007, Aachen.

[5] V.M. Garcıa, L. Acevedo, and A.M. Vidal, "Variants of algebraic wavelet-based multigrid methods: Application to shifted linear systems," *Appl. Math. Comput.*, vol. 202, no. 1, pp. 287–299, 2008.

[6] F.H. Pereira, S.I. Nabeta, "Wavelet-Based Algebraic Multigrid Method Using the Lifting Technique", *J. of Microwaves, Optoelectronics and Electromagnetic Applications*, vol. 9, no. 1, pp. 1-9, 2010.

[7] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets", *Appl. Comput. Harmon. Analysis*, vol. 3, no. 2, pp. 186-200, 1996.

[8] A. Jensen, and A. la Cour-Harbo, *The Discrete Wavelet Transform*, Ripples in Mathematics, Springer, Berlin, 2001.

[9] H.J. Sips, and H.X. Lin, *High Performance Computing Course MPI Tutorial*, Delft University of Technology Information Technology and Systems, September 16, 2002.

[10] Davis, T., *University of Florida Sparse Matrix Collection*, NA Digest, 1997.