Title:
New Multilevel Parallel Preconditioning Strategies of Sparse Matrix for Speeding up CAD Systems

Author:
Kai Wang, Kaiwang@lbl.gov, UC Berkeley Lawrence Berkeley National Lab of USA

Keywords:
Parallel preconditioning strategies, sparse matrix, MSP strategy, multilevel pre-conditioner, 3D spatial data processing, performance of CAD system

DOI: 10.14733/cadconfP.2014.152-154

Introduction:
With the more progress of CG software and hardware technology and growing demand of CAD application, the 3D scene objects and the complexity of the model increases rapidly, the realistic demand of rendering was extremely increased, the display resolution appears exponential increase. According to above characteristics of huge amount volume in spatial data processing, researching the problem that applying high performance parallel computing especially non-traditional model to practical solving process of large sparse matrices equations, which will have important significance for performance calculation of speeding up CAD system.

As we all know, the linear system may be solved by a direct solver based on a factorization of the sparse matrix (Gauss eliminations), which is known to be robust. However, the Gauss eliminations lack of inherent parallelism, and their O(n2) complexity of memory cost and O(n3) complexity of computational cost make them very expensive for solving large problems. So people turn to other methods and try to deal with sparse linear systems by taking the advantage of the sparse structure in the coefficient matrices, including LU factorizations (ILU(k), ILUT), other variants of multilevel ILU pre-conditioners, Sparse Approximate Inverse (SAI) and its developed versions, Multistep Successive Preconditioning strategy (MSP) and so on.

In this paper, we studied the use of the Multistep Successive Preconditioning strategies (MSP) based on the sparse approximate techniques that computes a sequence of low cost sparse matrices to achieve the effect of a high accuracy pre-conditioner, because of the inherent parallelism provided by the MSP algorithm, we do not need to use an independent set search related strategy to form the multilevel structure, which may be difficult to implement on parallel CAD computing systems. So a new parallel multilevel MSP pre-conditioner is presented, at each level, we use a diagonal value based strategy to permute the matrix into a 2 by 2 block from. During the preconditioning phase, we do forward and backward preconditioning to improve the performance of the pre-conditioner. Compared with the linear CAD computing systems, forward and backward preconditioning strategy is used in the pre-conditioner application (solution) phase to improve the performance of the resulted multilevel pre-conditioner.

Pre-conditioner Construction:
The convergence rate of a Krylov subspace solver applied directly to the linear matrix may be slow due to the potential ill-conditioning of the matrix A. In order to speed up the convergence rate of the iterative methods, we may transform $Ax = b$ into an equivalent system:

$$MAx = Mb \tag{1}$$

where M is a nonsingular matrix of order n. If M is a good approximation to A^{-1} in some sense, M is called a sparse approximate inverse of A. Then MA can be a good approximation to the identity matrix I. It follows that the equivalent system (1) will be easier to solve by a Krylov subspace solver.

The multistep successive preconditioning strategy can be applied to almost any of the existing sparse approximate inverse preconditioning techniques, including multistep successive preconditioning, multilevel preconditioning and multilevel pre-conditioner based on the MSP strategy. Then the MSP algorithm will create a series of matrices:

$$M_{\alpha l} M_{\alpha l-1} \cdots M_1 \approx D_\alpha^{-1} \tag{2}$$

where l is the number of steps.

Implementation details:

The success of general sparse linear system solvers depends on sophisticated implementations as heavily as on the innovative underlying ideas. In this section, we discuss several implementation issues that need to be addressed to build efficient software for distributed memory parallel computers.

To solve a sparse linear system on a parallel computer, the coefficient matrix is first partitioned by a graph partitioner and is distributed to different processors uniformly. Supposing initially the matrix is distributed row by row in each processor, where the coefficient matrix A is distributed in 4 processors. The shaded parts A_m, m = 1,…,4, in the figure form the block diagonal of A. From the view of graph partition, A_m can be called the local matrix of the processor P_m. And the entries of A_m are the local elements of the processor P_m.

Experimental discussion:

Convection-diffusion problem: A 3D convection-diffusion problem (defined on a unit cube).

$$u_{xx} + u_{yy} + u_{zz} + 1000(p(x,y,z)u_x + q(x,y,z)u_y + r(x,y,z)u_z) = 0 \tag{3}$$

is used to generate some large sparse matrices to test the scalability of multilevel MSP. Here the convection coefficients are chosen as:

$$\begin{aligned} p(x,y,z) &= x(x-1)(1-2y)(1-2z) \\ q(x,y,z) &= y(y-1)(1-2z)(1-2x) \\ r(x,y,z) &= z(z-1)(1-2x)(1-2y) \end{aligned} \tag{4}$$

The Reynolds number for this problem is 1000. Eq. (3) is discretized by using the standard 7-point central difference scheme and the 19-point fourth order compact difference scheme. The resulting matrices are referred to as the 7-point and 19-point matrices respectively.

level	*size*	*density*	*iter*	*setup*	*solve*	*total*
2		5.13	1843	22.5	417.1	4692
						439.6
4	524	6.88	238	18.0	71.8	89.8
6	60	6.86	140	17.3	41.8	59.1
8	8	6.86	137	17.3	40.1	57.4

Tab. 1: Solve the matrix (n = 14075) with different levels; ratio = 0.67, step = 3, ε = 0.02.

Reduction ratio and number of levels: The sizes of the current level matrix and the next level (reduced) matrix are decided by the reduction ratio. Reduction ratio is an important parameter for deciding the number of levels and influencing the property of the multilevel.

Comparison of MSP and multilevel MSP: This experiment gives a few experiment results for using MSP and MMSP to solve a few different matrices. For MSP, we adjust the parameter ε and number of steps and try to give the best performance results for solving these matrices. For all results of multilevel MSP we fix the reduce ratio to be 0.67, ε to be 0.05, the number of steps at each level to be 2 and the

number of forward and backward iterations to be 5. The number in the parentheses of MSP is the number of steps, and the number in the parentheses of multilevel MSP is the number of constructed levels.

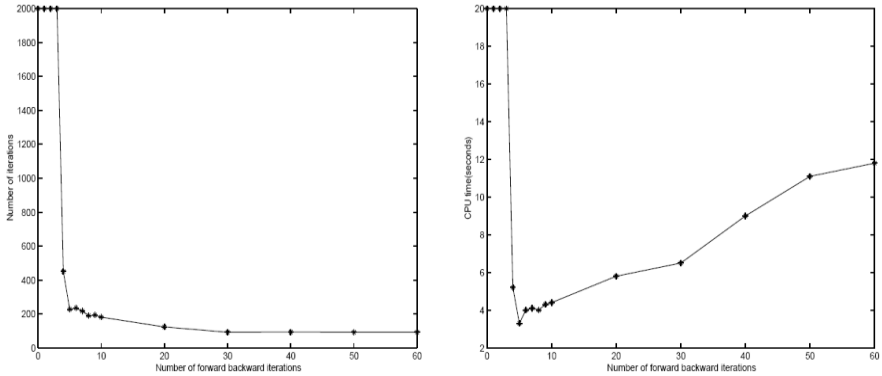

Fig. 1: Convergence behavior of multilevel MSP using different number of forward and backward iterations for solving the UTM1700B matrix. (ratio = 0.67, step = 2, ε = 0.05, density = 3.48, level = 7.) Left: the number of forward and backward iterations versus the number of outer iterations to make the preconditioned system converge. Right: the number of forward and backward iterations versus the total CPU time for solving the preconditioned system.

Scalability: We use the 3D convection-diffusion problem (3) to test the implementation scalability of our multilevel MSP pre-conditioner. The results are from solving a 7-point matrix with n = 1003 and nnz = 6940000 using different number of processors. Because the memory limitation of our parallel computers, we can only test the problems starting from 4 processors. To be convenient, we set the speedup in 4 processors case to be 4. We can see that the multilevel MSP scales well. In particular, we point out that it is not the same as the MSP algorithm, in which the number of the MSP iterations is not influenced by the number of processors when the whole problem size fixed.

Conclusion:

We have developed a multilevel sparse approximate inverse pre-conditioner based on the MSP strategies for solving general sparse matrices in 3D computing and displaying problems. A prototype implementation is tested to show the robustness and computational efficiency of this class of multilevel pre-conditioners.

From the numerical results presented, we can see that forward and backward preconditioning is an important strategy for the convergence performance of the multilevel MSP pre-conditioner. A number of forward and backward preconditioning iterations helps the convergence of the multilevel MSP pre-conditioner and a carefully chosen number of iterations will make the multilevel MSP pre-conditioner converge fast.

In addition, the number of levels also influences the convergence and memory cost of the multilevel MSP pre-conditioner. A large number of levels will produce a good pre-conditioner with a high memory cost. A small number of levels will create a cheap pre-conditioner with low memory cost. The same thing happens when refer to the number of MSP steps used at each level of the multilevel MSP pre-conditioner. Fortunately, because the MSP algorithm creates a series of preconditioning matrices, we can use a two Schur complement strategy to decrease the memory cost, even sometimes the computational cost for the preconditioning phase may increase.

When compared with MSP, multilevel MSP is much cheaper and more robust. The scalability of multilevel MSP is also tested by a 3D scene computing and displaying problem. The multilevel MSP pre-conditioner seems to scale well.