
Haifeng Qian
Stochastic Preconditioning for Iterative Linear Equation Solvers

1000 8th Street SE
Apt 107
Minneapolis
MN 55414
qianhf@ece.umn.edu
Sachin Sapatnekar

Stochastic techniques, namely random walks, have been used to form linear equation solvers since 1940's, but have not been used for preconditioning, at least not effectively, to the best of our knowledge. In this talk, We present a new stochastic preconditioning approach: we prove that for symmetric diagonally-dominant M-matrices, an incomplete LDL factorization can be obtained from random walks, and used as a preconditioner for an iterative solver, e.g., conjugate gradient. The theory can be extended to general matrices with nonzero diagonal entries.

The stochastic preconditioning is performed by a random walk “game” defined as follows. Given a finite undirected connected graph representing a street map, a walker starts from one of the nodes, and goes to one of the adjacent nodes every day with a certain probability. The walker pays an amount of money, m_i at node i , to a motel for lodging everyday, until he/she reaches one of the homes, which are a subset of the nodes. Then the journey is complete and he/she will be rewarded a certain amount of money, m_0 . The problem is to determine the gain function:

$$f(i) = E[\text{money earned} \mid \text{walk starts at node } i] \quad (1)$$

These gain values satisfy the following linear equations:

$$\begin{aligned} f(i) &= \sum_{j \in \text{neighbors of } i} p_{i,j} f(j) - m_i \quad , \quad \forall i \\ f(\text{a home node}) &= m_0 \end{aligned} \quad (2)$$

where $p_{i,j}$ is the transition probability of going from node i to node j , and note that j can be a home node. Thus a random walk game is mapped onto a system of linear equations. Conversely, it can be verified that given $A\mathbf{x} = \mathbf{b}$, where A is a symmetric diagonally-dominant M-matrix, we can always construct a random walk game that is mathematically equivalent, in which the set of f values is equal to the solution vector \mathbf{x} . To find the i^{th} entry of \mathbf{x} , one may run a number of walks from node i and take the average of the results; to get

the complete solution, one may repeat the process for every entry of \mathbf{x} . We alter this normal procedure by adding the following rule: every calculated node becomes a new home in the game with an award amount equal to its calculated f value. Without loss of generality, suppose the nodes are in the natural ordering $1, 2, \dots, N$, then for walks starting from node i , the node set $\{1, 2, \dots, i-1\}$ are homes where walks terminate (in addition to homes generated from the strictly-diagonally-dominant rows of A), while the node set $\{i, i+1, \dots, N\}$ are motels where walks pass by.

Define the operator $\text{rev}(\cdot)$ on square matrices such that it reverses the ordering of the rows and reverses the ordering of the columns: $\text{rev}(A)_{i,j} = A_{N+1-i, N+1-j}$, $\forall i, j \in \{1, 2, \dots, N\}$. Let the exact LDL factorization of $\text{rev}(A)$ be $\text{rev}(A) = L_{\text{rev}(A)} D_{\text{rev}(A)} (L_{\text{rev}(A)})^T$. Again, in the random walk game, assume that the nodes are in the natural ordering $1, 2, \dots, N$, and that node i corresponds to the i^{th} row of A . We prove the following relations:

$$(L_{\text{rev}(A)})_{i,j} \approx -\frac{M_{N+1-j, N+1-i}}{W_{N+1-j}}, \quad \forall i > j \quad (3)$$

$$(D_{\text{rev}(A)})_{i,i} \approx \frac{W_{N+1-i} A_{N+1-i, N+1-i}}{J_{N+1-i}} \quad (4)$$

where W_k is the total number of walks that are carried out from node k , M_{k_1, k_2} is the number of walks that start from node k_1 and end at k_2 , and J_k is the number of times that the W_k walks from node k pass node k itself. These equations show that we can approximate an LDL factorization by collecting information from random walks. We further prove that if $(L_{\text{rev}(A)})_{i,j} = 0$ then $M_{N+1-j, N+1-i} = 0$; in other words, the nonzero pattern of the L factor produced by random walks is a subset of nonzero pattern of the exact $L_{\text{rev}(A)}$. Therefore, we conclude that an incomplete LDL factorization can be obtained from random walks.

We argue that the obtained incomplete LDL factors have better quality, i.e., better accuracy-size tradeoffs, than the incomplete Cholesky factor obtained by a traditional method based on Gaussian elimination. Our argument is based on the fact that each row in the L factor is independently calculated and has no correlation with the computation of other rows. Therefore we avoid the error accumulation in traditional incomplete factorization procedure.

We also discuss, by defining a new set of game rules, how this theory can be extended to general matrices, given that the diagonal entries are nonzero.

To evaluate the proposed approach, a set of benchmark matrices are generated by Y. Saad's SPARSKIT by finite-difference discretization of the 3D Laplace's equation $\nabla^2 u = 0$ with Dirichlet boundary condition. The matrices correspond to 3D grids with sizes 50-by-50-by-50, 60-by-60-by-60, up to 100-by-100-by-100, and a right-hand-side vector with all entries being 1 is used with each of them. We compare the proposed solver, i.e., random walk preconditioned conjugate

gradient, against ICCG with ILU(0) and ICCG with ILUT. The complexity metric is the number of double-precision multiplications needed at the iterative solving stage, in order to converge with an error tolerance of 10^{-6} . The results show up to 2.1 times speedup over ICCG, and a trend that the larger and denser a matrix is, the more the proposed solver outperforms ICCG.

This talk is partially based on [1], and the implementation is available to the public [2].

Bibliography

- [1] H. Qian and S. S. Sapatnekar, “A hybrid linear equation solver and its application in quadratic placement,” *IEEE/ACM International Conference on Computer Aided Design Digest of Technical Papers*, pp. 905-909, 2005.
- [2] H. Qian and S. S. Sapatnekar, The Hybrid Linear Equation Solver Binary Release, available at
<http://www.ece.umn.edu/users/qianhf/hybridsolver>