Design Of Newton Iterative Method For Matrix Inversion

Rashmi Pahal\textsuperscript{1}, Dinesh Kumar Verma\textsuperscript{2}

\textit{Electronics and communication Department\textsuperscript{1,2}}

P.D.M college of Engg\textsuperscript{1,2}

Rashmipahal04@gmail.com\textsuperscript{1}, erdineshverma@gmail.com\textsuperscript{2}

Abstract- Newton algorithm is efficient method to calculate the inverse of a matrix. It offers less complexity while calculating the inverse. This paper shows the initial choice of newton iteration. Initial choice matters a lot while defining the convergent behavior of the iterative process. This paper has shown the calculation of generalized inverse of matrix till 10\textsuperscript{th} iterations. It has been observed that as the iteration increases the accuracy of the algorithm increases. As at 4\textsuperscript{th} iteration residual matrix has higher values while at 7\textsuperscript{th} and 10\textsuperscript{th} iteration residual matrix converges to zero showing accurate work.

Index Term- SVD, Matrix, Algorithm

1. INTRODUCTION

Newton iterative method for matrix inverse is used, it compute the matrix inversion tin parallel time $O(\log^2 n)$, this computation is processor efficient. Algorithm essentially amounts to sequence of matrix-matrix multiplication, so it can be implemented on systolic arrays and on parallel computers with greater efficiency. It compute the result in parallel time proportional to $\log^2 n$ using number of processor that is within a factor of $\log n$ of the optimum. This method can calculate the inverse on nonsingular matrix which is not true for the earlier Polylog time matrix inversion method. This is simple to describe and analyze, and offers numerical stability\textsuperscript{[7]}.

Newton Iterative method for matrix inversion requires $n^2(k-1)$ less computation operation than the conventional approach, where $n$ is the order of matrix and $k$ is the total number of iterations. Iterative matrix inversion schemes involve successive improvement of the initial inverse calculated. Thus, iterative schemes for matrix inversion have three components namely 1). a scheme to compute the initial inverse, 2). a scheme to successively improve the inverse starting from initial inverse, 3). a stopping criterion to determine whether the desired solution has been achieved or not. For operation counts, use terms “flop” to mean a multiple-add pair \textsuperscript{[7]}. For unstructured matrix at each iteration, two matrix multiplications operation is essential, that counts to $1.5n^2$ flops. Various methods are provided \textsuperscript{[7]} to reduce this count to $n^2+1/2 n^2$.

Iterative method \textsuperscript{[10,13]} improves successive approximations until the solutions converges to desired result. Iterative methods for matrix computation are widely used, especially for large systems, because these methods tends to be simpler, more robust to numerical errors, and require less storage compared to direct methods. The speed of convergence of iterative methods, however, depends on the initial approximation. So the choice of initial solution is very important to make efficiently. The Gauss-Jordan method \textsuperscript{[9,12]} is a direct method for matrix inversion. The inverse of Matrix by this method involves applying series of transformations on the rows and columns of the matrix to convert it into an identity matrix. The corresponding operations on the unit matrix, transforms the identity matrix into inverse of the original matrix.

Cholesky method\textsuperscript{[12]} also known as the square-root method can be used to calculate the inverse of matrix. It decompose the matrix into lower triangular matrix first and then calculate the inverse by using the lower triangular matrix. So the inverse of lower triangular matrix is also lower triangular, the complexity of finding the inverse is reduced.

The iterative method\textsuperscript{[10,13]} for matrix inverse present a formulation that uses only the first two terms of an infinite series involved in representing the inverse. The formulation of the initial initial inverse using norms and their effects on the convergence of the iterative scheme determine the inverse has been studied. The effect of choice of the number of terms, on the error between the computed inverse and the actual inverse, for a fixed amount of computations, has been examined.
2. ITERATIVE MODEL AND ALGORITHM

Consider a \(n \times n\) nonsingular matrix, \(A\) whose inverse is to be computed. Newton Iterative provided by Schulz [6] for computing inverse of matrix \(A\).

Newton Iteration is given by\[7\]
\[X_{K+1} = X_K (2I - X_K) \quad \ldots (1)\]
Where \(X_{K+1}\) is the inverse of \(A\), obtained in the \(K\)th iteration.

Above equation yield the inverse of matrix called as generalized inverse, \(A^+\). So (1) can be given as:
\[A^+ = X_K (2I - A X_K) \quad \ldots (2)\]
Ben-Israel and Cohen provide \[1],[2] that the iteration converges to \(A^+\) provided that \(X_0 = \alpha_0 A^T\) \[3\]
With \(\alpha_0\) positive and sufficiently small. Initial choice of \(\alpha_0\) is crucial in the convergence process for calculating the inverse of matrix.

So the optimum choice of \(\alpha_0\) in (3) which minimize the \(\|I - X_0 A\|\). Where \(X_0\) is the generalized inverse at 0th iteration.

\(\alpha_0 = \frac{1}{\sigma_1^2 + \sigma_r^2} \quad \ldots (4)\)
It is very difficult to calculate the \(\sigma_0\) so we use the suboptimal method to calculate the \(\alpha_0\) which is given as:
\(\alpha_0 = \frac{1}{\|A\|_1 \|A\|_\infty} \quad \ldots (5)\)
Other choices of \(\alpha_0\) which do not require an estimate of \(\sigma_r\) are available \[1],[2],[4],[5\].

The residual matrix, \(E_K\) is a measure of deviation of computed inverse from the actual inverse of given matrix \(A\). The Residual matrix \(E_K\) after \(k\) iteration is given as
\[E_K = I - A^+ A \quad \ldots (6)\]
It is easy to show that in this method we have \(X_{K+1} = X_k^2\) revealing quadratical convergence. By increasing the number of iteration, accuracy of generalized inverse matrix also increases.

The error matrix by after the \(k\)th iteration is, \(\tilde{A}_k\) we have
\[A^+ = A^{-1} + \tilde{A}_k \quad \ldots (7)\]
All of our analysis is based on Singular value decomposition (here referred as SVD) of \(A\), matrix whose inverse is to be calculated, \(A \in \mathbb{R}^{m \times n}\), \(U\) and \(V\) are square orthogonal matrices.
\[A = U \Sigma V^T \quad \ldots (8)\]
\(U\) and \(V\) are square orthogonal matrices. Where \(U = [u_1, \ldots, u_m], V = [v_1, \ldots, v_n]\) and \(\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)\), \(r = \text{rank}(A)\). Here \(r = \text{rank}(A)\), the generalized inverse of \(A\) is
\[A^+ = U \Sigma^+ V^T \quad \ldots (9)\]
Where \(\Sigma^+ = \text{diag}(\sigma_1^{-1}, \sigma_2^{-1}, \ldots, \sigma_r^{-1}, 0, \ldots, 0)\).
So if \(A\) has SVD then \(\|A\|_2 = \sigma_1\) and \(\|A\|_F = \|\Sigma\|_F\).
This is provided by the Eckart-Young theorem \[3\].

3. PERFORMANCE RESULT

In this work, as discussed above initial choice decide the convergent behavior of the algorithm. In our work we calculate initial iteration as in form of norms. The initial iteration is given by (3),(5)
\(\alpha_0 = \frac{1}{\|A\|_1 \|A\|_\infty}\).
It include the \(A\) norm One and \(A\) norm Infinity. Both of them are computed using various mathematical formulations.

We analyze, while doing the work that to get the accurate result we need to take the algorithm to the higher iteration level, as at lower iteration the result obtained are not upto the mark. So the result analyzed at 4th iteration has high value in residue matrix while at 7th iteration residue matrix converges to zero. And at the same time it is also evaluated that the result at 10th iteration are similar to result at 7th iteration.

The algorithm is applied on \(4 \times 4\) nonsingular matrix and on \(8 \times 8\) nonsingular matrix. It has been evaluated that as the size of the matrix increases or as the matrix became complex than the number of iteration decreases to obtain the desired result.

We analyse the Residue matrix at each and every iteration, and calculate the difference between the Reside matrix at successive iterations.
4. SIMULATION

The implementation design in this work has been stimulated using Verilog-HDL. We have done our work on Xilinx ISE tool, and analyze all of the work by Stimulation provided by the same tool.

5. CONCLUSION

This paper considers the Newton Iterative method for calculating the inverse, to reduce the complexity of the inverse calculation. This method involves less computation complexity, robust and very much prone to errors. Further improvement can be done in algorithm at initial iteration.

REFERENCES


