Approximate Neumann Series or Exact Matrix Inversion for Massive MIMO? (Invited Paper)

Oscar Gustafsson, Erik Bertilsson, Johannes Klasson and Carl Ingemarsson

Approximate Neumann Series or Exact Matrix Inversion for Massive MIMO? (Invited Paper)

Oscar Gustafsson, Erik Bertilsson, Johannes Klasson, and Carl Ingemarsson
Department of Electrical Engineering, Linköping University
SE-581 83 Linköping, Sweden, Email: oscar.gustafsson@liu.se

***Abstract*—Approximate matrix inversion based on Neumann series has seen a recent increased interest motivated by massive MIMO systems. There, the matrices are in many cases diagonally dominant, and, hence, a reasonable approximation can be obtained within a few iterations of a Neumann series. In this work, we clarify that the complexity of exact methods are about the same as when three terms are used for the Neumann series, so in this case, the complexity is not lower as often claimed. The second common argument for Neumann series approximation, higher parallelism, is indeed correct. However, in most current practical use cases, such a high degree of parallelism is not required to obtain a low latency realization. Hence, we conclude that a careful evaluation, based on accuracy and latency requirements must be performed and that exact matrix inversion is in fact viable in many more cases than the current literature claims.**

1. Introduction

In massive MIMO systems, i.e., where the number of base-station antennas, M, is significantly higher than the number of users, N, detection and pre-coding can be done using zero-forcing. Here, a matrix, $\mathbf{X} \in \mathbb{C}^{N \times N}$, is formed from the channel matrix, $\mathbf{H} \in \mathbb{C}^{M \times N}$, as $\mathbf{X} = \mathbf{H}^H\mathbf{H}$, where ^{H} denotes the Hermitian of the matrix, i.e., the conjugated transpose, and must be inverted. $\mathbf{X}$ is conjugate symmetric (Hermitian) and semi-definite, and, with uncorrelated channels and $M \gg N$, diagonally dominant.

Neumann series expansion has been suggested to obtain an efficient implementation of the matrix inverse [1], [2], [3], [4], [5], [6], [7], [8]. The motivation being a potentially lower complexity if few iterations are required. Also, the computation of a Neumann series has a potentially lower latency as it is more parallel.

The latency requirements is the time from that the pilot symbols are obtained to determine $\mathbf{H}$ to the data to be transmitted assuming the same channel conditions must be pre-coded [9]. This happens once for each coherence period, i.e., the time the channel is expected to be the same. Hence, the latency requirements are based on the frame format, which in turn is based on mobility parameters and used frequency range. Considering the commonly used frame format based on the current LTE standard, the frame time is 0.5 ms. The time to perform all computations from the last pilot symbol to the first transmit symbol is $\frac{3}{7}$ of that time [9]. Computing $\mathbf{X}$ requires $\frac{MN(N+1)}{2}$ MAD (multiply-and-add) operations. As a matrix inversion requires $\approx \frac{1}{2}N^3$ MAD operations, only a fraction of that time can be spent on the matrix inversion. Assuming 10%, the matrix inversion latency must be performed in 0.05 ms. The matrix inversion is in this scenario computed 2000 times per second, so commonly reported throughputs of Minv/s are of no use, since no other matrix inversions can be performed.

An algorithm with O operations, each with P pipeline stages implemented on Q processing elements (PEs), requires

$$C_{\text{alg}} \geq \max\left\{\left\lceil\frac{O}{Q}\right\rceil + P - 1, PC_{\text{latency}}\right\} \text{ cycles}, \qquad (1)$$

where C_{latency} is the minimum latency of the algorithm, i.e., the number of operations on the longest path. However, this is only a lower bound, so local parallelism limitations may introduce additional cycles. Also, this is without memory access aspects.

An overview of the most promising exact methods for inverting positive definite symmetric matrices, including complexity and finite word length simulation results, can be found in [10]. Based on [10], LDL decomposition combined with equation system solving is selected as it has low complexity and good numerical properties.

A K-term Neumann series matrix inverse approximation, $\mathbf{X}_K^{-1}$, using an initial guess $\mathbf{A} \approx \mathbf{X}^{-1}$, can be written as

$$\hat{\mathbf{X}}_K^{-1} = \left(\sum_{n=1}^{K} (\mathbf{I} - \mathbf{A}\mathbf{X})^{n-1}\right)\mathbf{A}, \qquad (2)$$

where $\mathbf{I}$ is the identity matrix. The main difference between most of the earlier works has been how to determine $\mathbf{A}$. In [1], [3], [4], [6], [7] the reciprocal diagonal elements was used. This leads to that $\mathbf{I} - \mathbf{A}\mathbf{X}$ has a zero diagonal, and, hence, that some computations can be removed. It was argued in [2] that using a tri-diagonal $\mathbf{A}$-matrix with an approximate inverse lead to better results. In [8], an alternative $\mathbf{A}$ with the diagonal and one column was used with the motivation as [2] has sequential reciprocal operations, and, hence, higher latency. The accuracy of all these methods depends on the structure of the matrix, where roughly better results are obtained with fewer iterations the larger the quotient $\frac{M}{N}$.

2. Results

The results in terms of complexity and latency in number of MADs and reciprocals are shown in Table 1. All the considered algorithms can be written as MAD operations, so for consistency this was used as the fundamental operation. As reciprocals are often implemented using Newton-Raphson iterations, later the reciprocals are replaced with MADs. All reciprocals are real-valued, while it is assumed that all MADs are complex valued. However, since some of the MADs has either one real-valued input or results in a real-valued output and it is sometimes of interest to count the number of real-valued MADs as in [3], these numbers are also given.In a few cases both the input and the output is real-valued. These are counted as real-valued outputs here. For the Neumann series methods, care has been taken to only compute the required partial results. For example, since the inverse is known to be Hermitian, only one half of the highest order matrix product must be computed. Typically, the numbers given are clearly lower compared to the original papers. In [8], the resulting matrix is not Hermitian. However, here, we only compute one half, assuming that the result should be Hermitian. Furthermore, we assume that an element in a matrix product is computed by sequential MAD operations. This is only a restriction from a latency perspective for the Neumann method when there are more than about $\frac{1}{2}N^2$ PEs.

Clearly, if $K = 2$, the arithmetic complexity is significantly lower compared to the exact methods. However, in many situations more iterations are required and for $K = 3$, the arithmetic complexity is slightly higher compared to the exact method, but the latency is lower. Hence, the potential benefit is that the resulting

TABLE 1. ARITHMETIC COMPLEXITY AND LATENCY FOR INVERSION OF AN $N \times N$ SYMMETRIC POSITIVE DEFINITE COMPLEX MATRIX.

Method	Operations				Latency	
	Total MADs	Real input MADs	Real output MADs	Reciprocals	MADs	Reciprocals
Exact method [10]						
LDL$^\intercal$+EQU	$\frac{1}{2}N^3+\frac{1}{2}N^2-N$	N^2-N	N^2-N	N	$4N-4$	N
Neumann series						
Diagonal, $K=2$	N^2-N	N^2-N	-	N	2	1
[1], [3], [4], [6], [7], $K=3$	$\frac{1}{2}N^3+N^2-\frac{1}{2}N$	$\frac{3}{2}N^2-\frac{1}{2}N$	N^2-N	N	$N+1$	1
Tri-diagonals [2], $K=2$	$3N^2+7N-10$	N^2+5N-6	$7N-6$	$2N-1$	$2N+5$	N
$K=3$	$\frac{1}{2}N^3+6N^2+\frac{1}{2}N-2$	$\frac{5}{2}N^2+\frac{9}{2}N-7$	N^2+7N-6	$2N-1$	$3N+5$	N
Diag. + column [8], $K=2$	$\frac{3}{2}N^2+\frac{5}{2}N-4$	N^2+N-2	$3(N-1)$	N	$N+2$	1
$K=3$	$\frac{1}{2}N^3+\frac{5}{2}N^2-2N-1$	$\frac{5}{2}N^2-\frac{5}{2}N$	N^2+N-2	N	$2N+1$	1

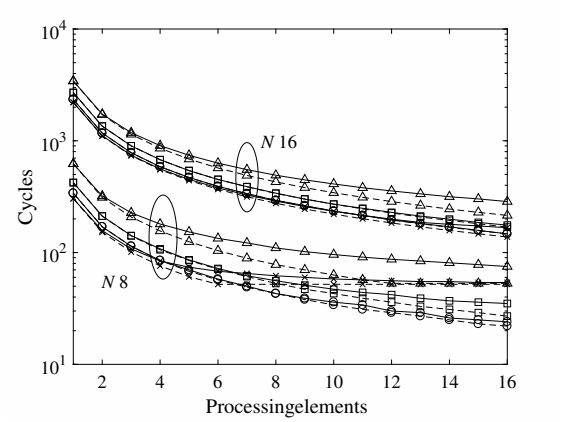

Figure 1. Number of cycles with single cycle latency MAD. Solid line is actual number of cycles, dashed is lower bound (1). × LDL+EQU, ◯ Neumann – diagonal, △ Neumann – tri-diag, □ Neumann – diag + column.

number of cycles for a parallel implementation. We have not considered $K \geq 4$ since the arithmetic complexity is significantly higher compared to exact matrix inversion.

In Fig. 1 the number of cycles required for the different algorithms and four different matrix sizes are shown, assuming $P = 1$ or $P = 3$ MADs and three MADs per reciprocal. It can be seen that when many PEs are used, the exact algorithm and the tri-diagonal approach of [2] are limited by the latency for $N = 8$ (for $N = 16$ this happens for more PEs). However, for $N = 8$ and one PE, 304 cycles are required for the exact algorithm. For this to finish in 0.05 ms a clock frequency of only 6.08 MHz is required. Assuming a clock frequency of just below 280 MHz, a matrix with $N = 30$ can still be inverted with a single PE in 0.05 ms.

Assuming that the P levels of pipelining can be introduced in a fully balanced way, decreasing the longest logic path the same factor and ignoring the setup and hold time of the registers, the time to perform the matrix inversion is shown in Fig. 2. Again, for $N = 8$, the exact method has limited parallelism as seen for $P = 2,\ 3$ and more processing elements. However, as it works for any channel condition, it should still be an attractive option in many cases.

3. Conclusion

The complexity of three-term Neumann series based matrix inversion is slightly higher than the best exact algorithms, both about $\frac{1}{2}N^3$. Hence, for the complexity to be lower, either two terms should be used or the complete third term should not be computed, as in [6]. The latency is higher for exact methods, and, hence, for very low latency implementations this may be a

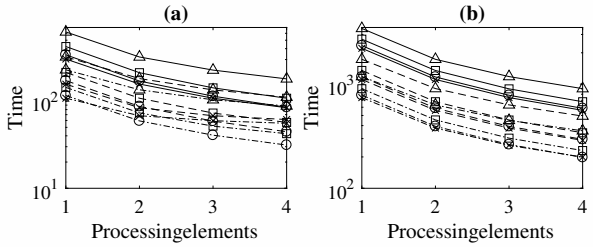

Figure 2. Time in single cycle latency operations for matrix inversion with one (solid), two (dashed), and three (dash-dotted) pipeline stages: (a) $N = 8$ and (b) $N = 16$. Markers as in Fig. 1.

limitation. However, in typical contemporary cases, it is generally not a limiting factor. Hence, exact methods are viable options, both from complexity and latency perspectives. In addition, the exact methods do not rely on the channel leading to a diagonally dominant matrix, and, hence, can be used in any scenario.

References

[1] H. Prabhu *et al.*, "Approximative matrix inverse computations for very-large MIMO and applications to linear pre-coding systems," in *Proc. IEEE Wireless Commun. Networking Conf.*, Apr. 2013.

[2] ——, "Hardware efficient approximative matrix inversion for linear pre-coding in massive MIMO," in *Proc. IEEE Int. Symp. Circuits Syst.*, Jun. 2014, pp. 1700–1703.

[3] M. Wu *et al.*, "Large-scale MIMO detection for 3GPP LTE: Algorithms and FPGA implementations," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 916–929, Oct. 2014.

[4] F. Wang *et al.*, "Efficient matrix inversion architecture for linear detection in massive MIMO systems," in *Proc. IEEE Int. Conf. Digital Signal Process.*, Jul. 2015, pp. 248–252.

[5] D. Zhu, B. Li, and P. Liang, "On the matrix inversion approximation based on Neumann series in massive MIMO systems," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2015, pp. 1763–1769.

[6] B. Kang, J. H. Yoon, and J. Park, "Low complexity massive MIMO detection architecture based on Neumann method," in *Proc. Int. SoC Design Conf.*, Nov. 2015, pp. 293–294.

[7] F. Rosário, F. A. Monteiro, and A. Rodrigues, "Fast matrix inversion updates for massive MIMO detection and precoding," *IEEE Signal Process. Lett.*, vol. 23, no. 1, pp. 75–79, Jan. 2016.

[8] S. M. Abbas and C.-Y. Tsui, "Low-latency approximate matrix inversion for high-throughput linear pre-coders in massive MIMO," in *Proc. IFIP/IEEE Int. Conf. Very Large Scale Integration*, Sep. 2016.

[9] E. Bertilsson, O. Gustafsson, and E. G. Larsson, "A scalable architecture for massive MIMO base stations using distributed processing," in *Proc. Asilomar Conf. Signals Syst. Comput.*, 2016.

[10] C. Ingemarsson and O. Gustafsson, "On fixed-point implementation of symmetric matrix inversion," in *Proc. Europ. Conf. Circuit Theory Design*, 2015, pp. 440–443.