

# GPU-based Electromagnetic Optimization of MIMO Channels

Alfonso Breglia, Amedeo Capozzoli, Claudio Curcio, Salvatore Di Donna, and  
Angelo Liseno

Università di Napoli Federico II, Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione  
via Claudio 21, I 80125 Napoli, Italy  
a.capozzoli@unina.it

**Abstract** — Strategies to accelerate MIMO channel capacity optimization on GPUs are outlined. The optimization scheme is dealt with by properly facing the main computational issues. In particular, the propagation environment is described by ultrafast Geometrical Optics (GO), singular values are computed by a very fast scheme and the optimizer is a parallel version of the differential evolutionary algorithm. The unknowns are given proper representations to reduce the number of optimization parameters.

**Index Terms** — CUDA, differential evolutionary, Geometrical Optics, GPU, MIMO channel, optimization, singular values.

## I. INTRODUCTION

Multiple Input - Multiple Output (MIMO) [1, 2] is a wireless communication technology using multiple antennas in both transmission and reception modalities to increase the channel capacity (CC) over that of a conventional SISO (Single Input - Single Output). CC is a crucial parameter to assess the performance of a communication system, and the problem arises of how defining the most convenient configuration of transmitting (TX) and receiving (RX) antennas to maximize it for a given SNR (Signal to Noise Ratio) and propagation environment.

Different approaches have been proposed to optimize the MIMO CC, see [3, 4] for two representative examples. As it appears from [3, 4], besides signal processing factors (e.g., modulation), two critical aspects emerge when optimizing the performance of a MIMO channel: one is the electromagnetic environment, since the electromagnetic propagation influences the properties of the channel matrix, and the other is the optimization scheme itself. Since both aspects pose a significant computational question, the issue thus arises of how computationally addressing the problem of optimizing a MIMO channel, by firstly determining the most convenient computational resources and algorithms to be exploited and that could make the challenge feasible. Then, how much a MIMO channel can be improved, how

the CC depends on the accuracy of the employed electromagnetic model and what can be obtained by a modeling grasping only the essential aspects of the problem should be pointed out, giving general guidelines at the design stage. These points have been up to now overlooked throughout the literature. Our purpose is facing the first point, namely, how much accelerated analysis and optimization can make the goal viable. This entails understanding how to push the performance of both, the employed algorithms and computational resources. Several computational key points should be then considered, since each performance can degrade the problem to unfeasibility:

1. Properly choosing and accelerating a global optimizer since a local optimizer is typically not enough to find the best solutions;
2. Properly choosing and accelerating the approach to compute the MIMO channel matrix;
3. Being the CC related to the singular values (SVs) of the channel matrix, accelerating their computation depending on the problem size (conventional - 4x4, 6x6 - MIMO vs. massive MIMO [5]);
4. Properly representing the unknowns, to manage only the essential optimization parameters;
5. Properly exploiting massively parallel computing platforms as Graphics Processing Units (GPUs).

The paper is organized as follows. The MIMO CC is briefly recalled in Section II, just to provide a formal introduction. Section III addresses points 1) and 4), Section IV point 2) and Section V points 3) and 5). Finally, Sections VI and VII present numerical results and conclusions, respectively.

## II. CHANNEL MODEL

Let us consider a narrowband, flat-fading channel, whose CC depends on the distribution of the SVs of the channel scattering matrix [1, 2]. Indeed, given  $N_{TX}$  transmitting and  $N_{RX}$  receiving antennas embedded in a complex 3D deterministic electromagnetic scenario, the MIMO channel can be described by its complex,  $N_{TX} \times N_{RX}$  matrix  $\underline{H}$  [1]. The generic element  $h_{ij}$  of  $\underline{H}$  can be expressed as:

$$h_{ij} = h(\underline{r}_i^{TX}, \underline{r}_j^{RX}) = \sum_{m=1}^M G_m(\underline{r}_i^{TX}, \underline{r}_j^{RX}), \quad (1)$$

where  $\underline{r}_i^{TX}$  represents the position of the  $i$ -th transmitting antenna,  $\underline{r}_j^{RX}$  represents the position of the  $j$ -th receiving antenna,  $M(i, j)$  is the number of relevant multi-paths between  $\underline{r}_i^{TX}$  and  $\underline{r}_j^{RX}$  and  $G_m(\underline{r}_i^{TX}, \underline{r}_j^{RX})$  is proportional to the voltage induced on the  $j$ -th receiving antenna by the  $m$ -th multipath originated at the  $i$ -th transmitting antenna.

Under the hypothesis of narrowband, flat-fading channel, AWGN noise at the receivers and equally distributed power among the transmitters [1], the Shannon CC, say  $C$ , expressed in bit/s/Hz can be calculated by first normalizing  $\underline{H}$  to its Frobenius norm as [6]:

$$\tilde{\underline{H}} = \underline{H} / \sqrt{\frac{\sum_{i=1}^{N_{TX}} \sum_{j=1}^{N_{RX}} |h_{ij}|^2}{N_{TX} N_{RX}}}, \quad (2)$$

and then computing,

$$C = \sum_{k=1}^r \log_2 \left( 1 + \frac{SNR \sigma_k^2}{N_{TX}} \right), \quad (3)$$

where  $r$  is the rank of  $\tilde{\underline{H}}$  and  $\sigma_k^2$  is its  $k$ -th SV.

The approach is illustrated in Fig. 1, where the flow-chart boxes correspond to the titles of the Sections III-V.

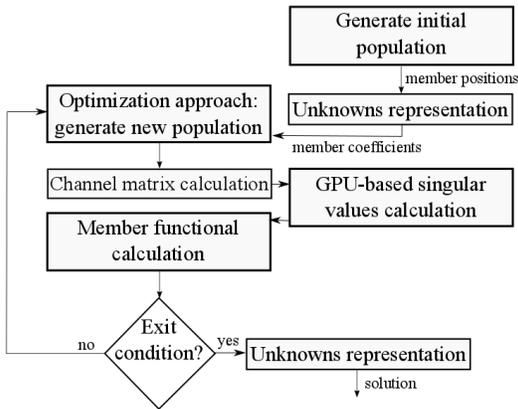


Fig. 1. Flow chart of the approach.

### III. THE OPTIMIZATION APPROACH

To enable a satisfactory exploration of the objective functional landscape, the global optimization approach should be chosen to exhibit good convergence properties and to profit of the massive parallelization. In this sense, a less “complex”, but massively parallelizable algorithm should be preferred to a more “involved”, but “more sequential” scheme. The “genetic-like” differential evolutionary approach has been then chosen as the global optimization scheme due to its main features matching

both the above mentioned requirements [7].

The approach exploits a population of  $N_p$  members, each member being represented by an array of  $D$  values, where  $D$  coincides with the number of optimization unknowns. At the  $g$ -th iteration (generation), the  $k$ -th member of the population is denoted by the  $D$ -dimensional array  $\underline{p}_{k g} = (p_{k g}^{(1)}, p_{k g}^{(2)}, \dots, p_{k g}^{(D)})$ . The initial population is randomly generated, accounting for some physical constraints enforced by the problem. Starting from the initial population, the algorithm generates a new one by first defining new arrays as (mutation):

$$\underline{m}_{k g+1} = p_{k_1 g} + F(p_{k_2 g} - p_{k_3 g}), \quad (4)$$

where  $k_1, k_2$  and  $k_3$  belong to  $\{1, 2, \dots, N_p\}$  and three indices mutually different and different also from  $k$ , and  $F \in [0, 2]$  is a user defined real and constant factor representing a scale factor of the differential variation

$$\underline{p}_{k_2 g} - \underline{p}_{k_3 g}.$$

Following mutation, new trial arrays  $\underline{t}_{k g+1} = (t_{k g+1}^{(1)}, t_{k g+1}^{(2)}, \dots, t_{k g+1}^{(D)})$  are generated as follows (crossover):

$$t_{k g+1}^{(i)} = \begin{cases} m_{k g+1}^{(i)}, & \text{if } \text{rand}(j) \leq CR \text{ or } j = \text{randD}(k) \\ p_{k g}^{(i)}, & \text{if } \text{rand}(j) > CR \text{ or } j \neq \text{randD}(k) \end{cases}, \quad (5)$$

where  $\text{rand}(j) \in [0, 1]$ ,  $j=1, 2, \dots, D$ , is the  $j$ -th evaluation of a random uniform generator,  $CR \in [0, 1]$  is a user defined crossover constant and  $\text{randD}(k) \in \{1, 2, \dots, D\}$  is a randomly chosen integer to ensure that the trial array contains at least a mutated element. Following the crossover, the cost values of  $\underline{p}_{k g}$  and  $\underline{t}_{k g+1}$  are compared and the one with the largest cost becomes the new population member at generation  $g+1$  (selection).

The operations involved in mutation, crossover and selection and the random number generations are inherently parallel. An issue of the crossover stage is the “random” global memory access, so that particular care has been given to improve memory coalescence.

### Unknowns representation

The problem concerns the optimization of the TX and RX antenna locations to maximize the MIMO CC. To profit from a reduction of the number of unknowns, both the TX and RX antennas are assumed to be located on lines and their positions are indirectly searched for by representing them by Legendre polynomials [8] as:

$$x_n = \sum_{k=0}^{K-1} c_k \psi_k(\xi_n). \quad (6)$$

In Eq. (6),  $x_n$  is the generic antenna coordinate on the optimization line,  $K$  is the number of polynomials,  $\psi_k$  is the  $k$ -th Legendre polynomial, the  $\xi_n$ 's are uniformly spaced points in  $[-1, 1]$  and the  $c_k$ 's become the actual unknowns to be sought for. If  $K$  is less than the involved antennas, the number of problem parameters is reduced.

Notice that the representation in Eq. (6) is also amenable to enforcement of constraints on minimum and maximum antenna spacings [9].

#### IV. CHANNEL MATRIX CALCULATION

The method exploited to calculate  $\tilde{\underline{H}}$  should trade off computational accuracy and speed to execute in an iterative optimization. Calculating at each generation the matrix  $\tilde{\underline{H}}$  for a large number of antenna configurations by a full wave method would be unfeasible, especially for large scenarios. Opposite to that, Geometrical Optics (GO) is very appealing to quickly provide an approximate solution to Maxwell's equations.

Nevertheless, for electrically large scenarios GO must be properly accelerated by adequate algorithmic structures capable to properly handle the intersections of the rays with the scene objects. Indeed, ray tracing involves two main steps: the search for the intersection between a ray and the geometric primitives (e.g., triangles), and the electromagnetic field transport. The first step can be definitely the most time consuming one, if not properly managed. A brute force approach would be indeed unfeasible due to the large number of intersection tests to be performed. Fortunately, the problem can be faced by tree-like structures which, if properly setup, managed and explored, can significantly reduce the computational complexity. Data structures like KD-tree and BVH (Bounding Volume Hierarchy) [10] can be effectively applied to this purpose and may profit from a high degree of parallelization. Here, the Split BVH (SBVH) scheme set up in [10] has been exploited.

#### V. GPU-BASED SVs CALCULATION

Computing the SVs of small or large matrices should be dealt with different approaches. Accordingly, the computational scheme to be employed differs if considering conventional or massive MIMOs. In this paper, we address the former case. Furthermore, the number of involved matrices is related to the number of population members of the differential optimizer. Then, at each generation, the SVs of a large number of small sized matrices have to be computed. This task can be efficiently and effectively performed on a GPU as in [11].

The problem of computing the SVs can be recast to the computation of the SVs of a real-valued matrix  $\underline{A}$ . To this end, the approach in [7] consists of three steps. The first step amounts at reducing  $\underline{A}$  to a bidiagonal matrix, say  $\underline{B}$ , as:

$$\underline{A} = \underline{P} \underline{B} \underline{Q}^T, \quad (7)$$

where  $\underline{B}$  is a  $N_{TX} \times N_{RX}$  upper bidiagonal matrix, and  $\underline{P}$  and  $\underline{Q}$  are  $N_{TX} \times N_{TX}$  and  $N_{RX} \times N_{RX}$  orthogonal Householder matrices, respectively. The bidiagonalization

step consist of applying a sequence of Householder transformations [12] to the matrix  $\underline{A}$ , which zero the elements below the diagonal and to the right of the first superdiagonal. In the second step,  $\underline{B}$  is transformed to a tridiagonal matrix  $\underline{T} = \underline{B}^T \underline{B}$ . Finally, in the third step, the symmetric tridiagonal eigenvalue problem is solved by a bisection method based on the use of Sturm sequences by restricting the search range using the Gershgoring circle theorem [11].

The motivation for computing the tridiagonal matrix  $\underline{T}$  as above is due to the fact that the explicit formation of  $\underline{T}$  should be avoided for numerical reasons since it may introduce non-negligible relative errors, especially in the computation of the smallest SVs [12]. However, the exploited approach is meant for those applications, as the one at hand, in which the smallest SVs have very low relative weight and may be considered irrelevant.

In summary, the problem of computing the SVs is recast as a "guided" bisection, which is amenable to parallelization.

#### VI. NUMERICAL RESULTS

The optimizer, the ray tracer and the SVs calculation have been implemented in parallel GPU (CUDA) and multi-core CPU (C++ with OpenMP directives) languages. For the CPU case, the SVs have been achieved using the third party Eigen library.

We consider a circular cylinder with radius  $10\lambda$  centered at the origin of the  $Oxyz$  reference system and a plate of width  $50\lambda$ , parallel to the  $yz$  plane and located at  $x=30\lambda$ . The cylinder and the plate are perfectly conducting and have a height of  $15\lambda$ . The scene has been discretized with 95458 triangles. This example points out how much computation time can be saved by the approach and provides an answer to the points raised in Section I and a perspective to design tools.

The optimizer can position an arbitrary number of transmitting and receiving antennas on lines with arbitrary spatial orientations. Here,  $N_{TX} = 4$  and  $N_{RX} = 4$ . The TX and RX antennas have been located on lines lying on the  $xy$  plane, parallel to the  $x$ -axis and passing by  $(15\lambda, -40\lambda, 0\lambda)$  for the TX and by  $(15\lambda, 40\lambda, 0\lambda)$  for the RX antenna. The antenna positions have been represented using  $K=3$  and minimum and maximum spacing of  $\lambda/4$  and  $2\lambda$ , respectively, have been enforced to control the maximum array size and mutual coupling. The SNR has been fixed to 20 dB.

For computational convenience, the optimization has been run with a population of 1000 elements, grouped in 10 subgroups including those configurations sharing the same positions of the TX antennas and different positions of the RX ones. The optimization has been run for a number of 50 generations, with  $CR=0.4$  and  $F=0.7$ .

The code has been run on a workstation equipped with two Intel Xeon E5-2650 2.00GHz, Eight core processors each and four NVIDIA Kepler K20c cards, but with multi-GPU disabled. Figure 2 displays an OpenGL rendering of the MIMO channel with the optimized antenna locations. The figure also depicts the GO rays connecting the TX antennas (the pink dots) with the RX ones (not appearing in the image). As it can be seen, multiple interactions have been accounted for as well as diffraction from the plate border. Diffraction from the plate corners and the cylinder ends have been neglected for simplicity. The optimized antenna positions are reported in Table 1. As it can be seen, the TX and RX antennas occupy almost symmetric locations due to the problem symmetry. The GPU code has run in about 4.5 hours, gaining a speedup of about 5 as compared to the CPU execution obtained by running 32 CPU threads. The optimized channel capacity has been 22.3 bps/Hz, a value which well agrees with the statistical distribution of channel capacities for random channels with 4 transmitting and 4 receiving antennas reported in [2, Fig. 7].

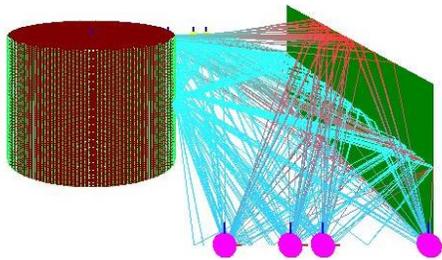


Fig. 2. OpenGL rendering of the MIMO channel with optimized antenna locations.

Table 1: Optimized TX and RX antenna positions

Antenna	$x$ -coord.	Antenna	$x$ -coord.
TX 1	$7.0\lambda$	RX 1	$7.08\lambda$
TX 2	$12.6\lambda$	RX 2	$12.9\lambda$
TX 3	$18.0\lambda$	RX 3	$17.1\lambda$
TX 4	$20.9\lambda$	RX 4	$22.0\lambda$

## VII. CONCLUSIONS

A GPU-based approach to accelerate MIMO CC optimization has been presented using ultrafast Geometrical Optics (GO), a very fast SV calculation scheme and a parallel version of the differential evolutionary algorithm. A speedup of 5 has been achieved against a multi-core CPU implementation.

## ACKNOWLEDGMENT

Work partially funded by the Italian Ministry of Education, University and Research (MIUR), project PON01\_02425 "Services for wireless network Infrastructure beyond 3G" (SIRIO).

## REFERENCES

- [1] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Commun.*, vol. 6, no. 3, pp. 311-335, Mar. 1998.
- [2] J. Bach Andersen, "Array gain and capacity for known random channels with multiple element arrays at both ends," *IEEE J. Selected Areas Commun.*, vol. 18, no. 11, pp. 2172-2178, Nov. 2000.
- [3] U. Olgun, et al., "Optimization of linear wire antenna arrays to increase MIMO channel using swarm intelligence," *Proc. of the 2<sup>nd</sup> Europ. Conf. on Antennas Prop.*, Edinburgh, UK, pp. 1-6, Nov. 11-16, 2007.
- [4] M. A. Mangoud, "Optimization of channel capacity for indoor MIMO systems using genetic algorithm," *Progr. Electromagn. Res. C*, vol. 7, pp. 137-150, 2009.
- [5] E. G. Larsson, et al., "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186-195, Feb. 2014.
- [6] N. Noori and H. Oraizi, "Evaluation of MIMO channel capacity in indoor environments using vector parabolic equation method," *Progr. Electromagn. Res. B*, vol. 4, pp. 13-25, 2008.
- [7] R. Storn and K. Price, "Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Opt.*, vol. 11, no. 4, pp. 341-359, Dec. 1997.
- [8] A. Capozzoli, et al., "Field sampling and field reconstruction: a new perspective," *Radio Sci.*, vol. 45, RS6004, pp. 31, 2010, doi: 10.1029/2009RS004298.
- [9] A. Capozzoli, et al., "FFT & aperiodic arrays with phase-only control and constraints due to super-directivity, mutual coupling and overall size," *Proc. of the 30<sup>th</sup> ESA Antenna Workshop on Antennas for Earth Observ., Science, Telecomm. and Navig. Space Missions*, Noordwijk, The Netherlands, May 27-30, 2008, CD ROM.
- [10] A. Breglia, et al., "Comparison of acceleration data structures for electromagnetic ray tracing purposes on GPUs," *IEEE Antennas Prop. Mag.*, vol. 57, no. 5, pp. 159-176, Oct. 2015.
- [11] A. Capozzoli, et al., "Massive computation of singular values of small matrices on GPUs," *Proc. of the Int. Workshop on Comput. Electromagn.*, Izmir, Turkey, pp. 36-37, July 1-4, 2015.
- [12] G. H. Golub and C. Reinsch, "Singular values decomposition and least squares solutions," *Numer. Math.*, vol. 14, pp. 403-420, 1970.