# Efficient, Accurate and Scalable Reflectarray Phase-Only Synthesis Based on the Levenberg-Marquardt Algorithm

**Daniel R. Prado, Jana Álvarez, Manuel Arrebola, Marcos R. Pino, Rafael G. Ayestarán, and Fernando Las-Heras**

Department of Electrical Engineering
University of Oviedo, Gijón, Asturias, 33203, Spain
{drprado, jalvarez, arrebola, mpino, rayestaran, flasheras}@tsc.uniovi.es

***Abstract*** **─** The Levenberg-Marquardt algorithm is used to perform an efficient phase-only synthesis for shaped-beam reflectarray antennas. A thorough analysis of the problem and algorithm allows to improve greatly its performance and its results with regard to those in the literature. Specifically, several optimizations in the Jacobian matrix computation are detailed, including its fully parallelization thanks to the fact that each column of the Jacobian can be independently computed. A Cholesky factorization based solver is used to obtain the updating vector at each iteration of the algorithm, which is the fastest exact solver for linear equation systems. Finally, some guidelines to choose proper values for the algorithm parameters are presented. Due to the high dimensionality of the problem, a good control in the algorithm evolution is important to guarantee good convergence and avoid non-desired local minima. The synthesis of a LMDS shaped beam for a dual-polarized reflectarray is proposed to test the developed algorithm and the results are compared with others in the literature, showing improvements in accuracy and time efficiency. The framework established for the phase-only synthesis can be used in a broader problem to directly optimize the geometry of the reflectarray through full wave analysis of the unit cell.

***Index Terms*** **─** Levenberg-Marquardt algorithm, local multipoint distribution system, parallelization, phase-only synthesis, reflectarray antennas.

## I. INTRODUCTION

Shaped-beam antennas are demanded in many applications that require non-canonical beams, such as direct broadcast satellite (DBS) missions, local multipoint distribution service (LMDS) technology or multibeam antennas [1]. Traditionally, parabolic reflectors and phased arrays have been used for these applications, although in recent years reflectarray antennas have proven to be a feasible solution while providing other benefits such as low cost, low weight and reduced physical dimensions [1]. As parabolic reflectors, they can be used in single configurations as well as dual-reflector setups [2]. However, reflectarrays present limitations mainly in the bandwidth [1,3], which can be overcome by choosing an appropriate element topology [1,4,5].

In order to achieve the required specifications, a number of algorithms has been used to synthesize the radiation patterns, for instance, analytical [6], steepest descent [7], conjugate gradient [8], intersection approach [9], Levenberg-Marquardt (LMA) [10], genetic algorithms [11] or particle swarm optimization [12], among others. However, the analytical approach has limitations when applied to complex shaped patterns, although they can be used to generate a starting point for a more powerful synthesis algorithm [13]. The steepest descent has a very slow convergence rate [14,15], which makes it impractical to synthesize arrays with a moderate number of elements. Conjugate gradient methods can be adapted to solve non-linear optimization problems and are faster than the steepest descent [14], but they tend to be both less efficient and less robust than quasi-Newton methods [15]. The intersection approach is very efficient when using only the fast Fourier transform (FFT), but suffers from the problem of traps, due to the non-convexity of the sets dealt with [16]. One manner of dealing with the trap problem is working with the far field squared amplitude instead of just the amplitude [16,17]. However, this approach causes that one of the projectors of the intersection approach cannot be implemented with FFT (the projector which recovers the reflected field on the reflectarray surface), and a minimization algorithm based on optimization techniques has to be used, reducing greatly the efficiency of the intersection approach algorithm (in [17], the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm is used, although other algorithms are also suitable, such as LMA). All these algorithms are local optimizers and depend

*Submitted On: June 30, 2015*
*Accepted On: November 8, 2015*

strongly on the starting point to converge with success.

Genetic algorithms (GA) and particle swarm optimization (PSO) are global search algorithms, which in contrast with the previous local search algorithms, do not depend on the starting point. These algorithms are potentially able to find the global maximum at the expense of taking many iterations. However, as the number of variables increase, the search space size grows exponentially, making it harder for these algorithms to find a suitable solution. Another aspect of evolutionary algorithms is that due to their non-deterministic approach, two instances using the same parameters will yield different results, in contrast with the deterministic approach of the local optimizers mentioned above. GA and PSO have been demonstrated capable of synthesizing phased arrays, although at the cost of several thousand iterations [18]. Each iteration involves several evaluations of the cost function, one for each member of the population, thus making their computing times very sensitive to the time cost of the cost function (also known as fitness in evolutionary algorithm terms). Both algorithms seem to have similar performance with small arrays, although PSO is easier to implement [18]. Recently, the PSO has been used to synthesize several reflectarray radiation patterns [12,19]. In [12], a single-fed reflectarray of 848 elements with asymmetric multiple beams was synthesized, taking more than 70000 iterations to converge and 44 hours. In [19], a reflectarray of 900 elements was synthesized, taking 5500 iterations to converge. Increasing the size of the antenna would dramatically increase computing times to achieve convergence with these algorithms because of their global search approach, unless a suitable starting point was used and the PSO was set up to prioritize local search.

Regarding memory usage, the LMA presents a disadvantage with regard to other synthesis algorithms. The LMA needs to store a Jacobian matrix, which is generally bigger than the Hessian matrix (or its approximation) used by Newton and quasi-Newton methods. Additionally, the LMA also needs to store the approximation of the Hessian. In contrast, the conjugate gradient only needs to store a smaller matrix (same size as the Hessian) and a few vectors, while the steepest descent only stores vectors. The intersection approach uses one to four small matrices with dimension the number of samples of the radiation pattern. (When an optimization algorithm is introduced in the intersection approach, the storage needs would be the same as the ones of the optimization algorithm plus the needs of the intersection approach.) Evolutionary algorithms, such as GA or PSO, store one solution per member of the population. In the phase-only synthesis case, the solution is a vector or matrix with the number of elements equal to the number of unknowns. Exact storage needs for the mentioned algorithms will vary according to their implementations, although they have been roughly laid out for the main data structures. The LMA trades more memory usage for a more robust algorithm for non-linear optimization when compared with other gradient methods, and a more flexible framework when compared with the intersection approach. It is also simpler and easier to implement than other quasi-Newton methods, and is faster than evolutionary algorithms due to its local search nature.

As an example, a reflectarray of $M \times N$ elements (which correspond to the unknowns of the problem) is considered, computing the radiation pattern only for one polarization in a grid with $U \times V$ points. The evolutionary algorithms are considered to have $L$ members in their population. Note that, in general, $M \cdot N \leq U \cdot V$. In this case, the size of the Jacobian matrix is $M \cdot N \cdot U \cdot V$, the size of the Hessian or its approximation is $M^2 \cdot N^2$ and the size of a solution is $M \cdot N$. Then, the memory usage is $O(M \cdot N \cdot U \cdot V + M^2 \cdot N^2)$ for the LMA, corresponding to the Jacobian and approximation of the Hessian; $O(M^2 \cdot N^2)$ for quasi-Newton methods and the conjugate gradient, corresponding to the Hessian; $O(U \cdot V + M \cdot N)$ for the intersection approach, corresponding to a solution and the computed far field; and $O(L \cdot U \cdot V + L \cdot M \cdot N)$ for the GA and PSO, corresponding to one solution and one computed far field per member of the population. Auxiliary vectors and matrices might be used depending on the implementation, but the main data structures shown above take up most of the memory used by the algorithm.

In this work, an optimization framework based on the LMA described in [10], is presented with a number of improvements derived from a deeper insight into the problem and the algorithm itself that allows to improve both the performance and results, yielding a faster and more robust method for the synthesis of shaped radiation patterns. Specifically, the proposed improvements focus on accelerating the computing times by parallelizing different operations, which makes the algorithm scale well with the number of available processors, taking advantage of farm servers and new desktop computers to perform faster synthesis for a given problem size, or to increase the size for the same computing times. Also, a faster method for solving the equation system is used, taking advantage of the nature of the problem. The algorithm accuracy and convergence is improved yielding better results in less iterations, which also leads to further reduction in computing times. Finally, a general phase-only framework based on the LMA is established, which can be further extended to include direct optimization of the geometry through full wave analysis of the unit cell. This last feature can dramatically increase the computing times of the cost function, which makes it impractical to implement in evolutionary algorithms such as GA or PSO. Some of the improvements in the implementation of the LMA for reflectarray synthesis can be used in other optimization algorithms, such as BFGS. The resulting framework based on the LMA is able to further optimize previous synthesis performed by other algorithms, or to carry out new ones in a more efficient way, provided a suitable starting point.

This paper is organized as follows. Section II describes how the reflectarray is analyzed to obtain the radiation pattern from the optimization variables, and also presents the approximations considered in the phase-only synthesis case along with their justification. Section III details the algorithm and all the optimizations implemented as well as a description of its parameters and implementation. Section IV shows some results for different synthesis. Finally, Section V has the conclusions.

## II. ANALYSIS OF THE REFLECTARRAY

### A. Computing co-polar far fields for phase-only synthesis

The scheme of an offset printed reflectarray is shown in Fig. 1. The reflectarray is illuminated by a feed which generates an incident electric field, $\vec{E}_{\text{inc}}$, on the surface of the reflectarray. This field can be expressed for both polarizations as:

$$\vec{E}_{\text{inc}}^{X/Y}(x,y) = E_{\text{inc},x}^{X/Y}(x,y)\hat{x} + E_{\text{inc},y}^{X/Y}(x,y)\hat{y}, \qquad (1)$$

where the superscripts indicate the polarization of the feed and the subscripts the component of the field with regard to the reflectarray coordinate system ($E_x^X$ would be the $\hat{x}$ component of the projected field over the reflectarray surface when the feed radiates in X-polarization). Note that each polarization has, in general, two components of incident field, the desired and the cross-polar, due to the feed not being ideal and the projection of the field over the reflectarray surface. In a similar way, the field reflected on the reflectarray surface can be written as:

$$\vec{E}_{\text{ref}}^{X/Y}(x,y) = E_{\text{ref},x}^{X/Y}(x,y)\hat{x} + E_{\text{ref},y}^{X/Y}(x,y)\hat{y}. \qquad (2)$$

The relation between the incident and reflected fields at each element $(m,n)$ of the reflectarray is given by a matrix of reflection coefficients that characterize that element:

$$\vec{E}_{\text{ref}}^{X/Y}(x_m,y_n) = R^{mn} \cdot \vec{E}_{\text{inc}}^{X/Y}(x_m,y_n), \qquad (3)$$

where

$$R^{mn} = \begin{pmatrix} \rho_{xx}^{mn} & \rho_{xy}^{mn} \\ \rho_{yx}^{mn} & \rho_{yy}^{mn} \end{pmatrix}, \qquad (4)$$

and $(x_m,y_n)$ are the coordinates of the $(m,n)$th element.

The components of $R^{mn}$ are complex numbers and fully characterize the behaviour of the element. This method of analysis takes into account three sources of cross-polarization: the feed, the projection of the incident field over the reflectarray surface and the reflectarray elements through the use of $R^{mn}$. However, considering lossless ideal phase shifters instead of real elements, the cross-polarization generated by them is not taken into account and the amplitude of $\rho_{xx}^{mn}$ and $\rho_{yy}^{mn}$ is set to one. Therefore, the synthesis is defined as a phase-only array synthesis where only the co-polar patterns are considered [20]. In this case, the reflected field in the aperture can be expressed as [13]:

$$\begin{aligned} E_{\text{ref},x}^{X}(x_m,y_n) &\approx \rho_{xx}^{mn} \cdot E_{\text{inc},x}^{X}(x_m,y_n), \\ E_{\text{ref},y}^{Y}(x_m,y_n) &\approx \rho_{yy}^{mn} \cdot E_{\text{inc},y}^{Y}(x_m,y_n). \end{aligned} \qquad (5)$$

Once the tangential electric field has been obtained using (5), the co-polar far field can be efficiently computed using the FFT [1], obtaining $\vec{E}_{cp}^{X/Y}(u,v)$, with:

$$u = \sin\theta\cos\varphi \quad ; \quad v = \sin\theta\sin\varphi. \qquad (6)$$

The far field pattern specifications can be given in form of masks with upper and lower bounds [1,9,10], $M_U(u,v)$ and $M_L(u,v)$, which can be defined in the whole UV grid or in the area of interest.

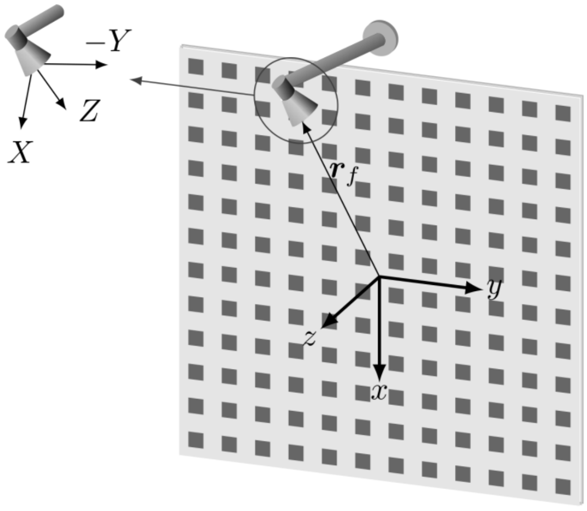


Fig. 1. Geometry of an offset printed reflectarray.

### B. Phase-only synthesis considerations

When performing phase-only synthesis, the only variables to be optimized are the phases of the reflection coefficients $\rho_{xx}^{mn}$ and $\rho_{yy}^{mn}$. The approximations made to obtain the co-polar far field patterns using the tangential field of (5) have proven to be valid in the case of reflectarray antennas [20], providing a co-polar pattern which is very close to the exact one.

Usually, the reflectarray is placed at far field distance of the feed, in which case can be modelled as a $cos^q\,\theta$ function [21], with very accurate results [22,23]. This feed model can also be used in the preliminary steps of the synthesis process and later use full wave simulations of the feed in order to accurately predict the far field [24-26]. However, the co-polar far fields obtained using both methods are very similar between each other. In any case, the algorithm described in the present work allows to use the real incident field from the feed, either by full wave simulations or near field measurements. The incident field is fixed throughout the synthesis process and only the phase shift that the reflectarray elements have to introduce is optimized in order to obtain the required co-polar pattern.

Due to the simplifications made on the $R^{mn}$ matrix, the cross-polar pattern that could be computed using (5) would not take into account the cross-polarization introduced by the reflectarray elements, which in practise is an important contribution to the cross-polar pattern. As it is explained in detail in [13,20], the co-polar pattern obtained with the simplifications made on the $R^{mn}$ matrix is still accurate. However, the cross-polar pattern is not [13], hence a phase-only synthesis only deals with co-polar requirements.

Once the synthesis is finished and the phases of $\rho_{xx}^{mn}$ and $\rho_{yy}^{mn}$ are obtained, a design can be carried out that takes into account the real element behaviour. By using full wave simulations based on local periodicity [27-29], the full $R^{mn}$ is computed, taking into account mutual coupling between elements. During this step, a zero finding routine is used that calls iteratively a full wave simulator to adjust the required phase shift for each element. Because the synthesized phases are accurately adjusted using this procedure, the resulting phase distribution will have a small error with regard to those obtained after the phase-only synthesis, and hence the co-polar pattern will be very close to the one synthesized.

## III. IMPROVED SYNTHESIS BASED ON LMA

### A. Cost function definition

In order to use the LMA, a proper cost function to be minimized has to be defined. The same cost function as in [10] will be used, where the residuals are defined as:

$$\begin{aligned}F_t^{X/Y} = C(\vec{r}_t)\Big[&\Big(M_U^2(\vec{r}_t) - \left|E_{\rm cp}^{X/Y}(\vec{r}_t)\right|^2\Big)\\ &\cdot\Big(M_L^2(\vec{r}_t) - \left|E_{\rm cp}^{X/Y}(\vec{r}_t)\right|^2\Big)\\ &+\left|M_U^2(\vec{r}_t) - \left|E_{\rm cp}^{X/Y}(\vec{r}_t)\right|^2\right|\\ &\cdot\left|M_L^2(\vec{r}_t) - \left|E_{\rm cp}^{X/Y}(\vec{r}_t)\right|^2\right|\Big],\end{aligned} \tag{7}$$

and thus, the cost function is:

$$F^{X/Y} = \sum_{t=1}^{T}\left(F_t^{X/Y}\right)^2. \tag{8}$$

In (7) and (8), each $t = 1, \dots, T$ describes a $\vec{r}_t = (u, v)_t$ point in which the UV grid is discretized; $C(\vec{r}_t)$ is a weight function; $E_{\rm cp}^{X/Y}(\vec{r}_t)$ can be either of both co-polar far fields, which are synthesized independently; and $F^{X/Y}$ is the total error, contributed by all the far field samples that lay outside the specified masks. This cost function penalizes the samples that lie outside the specified bounds (upper and lower) while it sets the error to zero when the samples are within bounds. It represents a non-convex search space [30] due to the non-convexity of the lower bound [31] and multiple solutions are possible. There are potentially a large number of solutions with minimum error. If the lower and upper bounds are too confined, the specifications might be too stringent and the algorithm might not find a solution, either because it does not exist or because the starting point is too far off from the solution. In that case, the specifications should be relaxed and/or the antenna optics redefined.

In order to alleviate the notation, from here on a generic $F_t$ will be used, avoiding the superscripts and knowing that it can represent the residual of either of both polarizations ($F_t^X$ or $F_t^Y$).

### B. Jacobian matrix calculation

The LMA requires the calculation of a Jacobian matrix, which is a $T \times P$ matrix, where $P$ is the number of variables to be optimized. Any element $(t, p)$ of the Jacobian is calculated as:

$$J(t, p) = \frac{\partial F_t(\alpha)}{\partial \alpha_t}, \tag{9}$$

where $\alpha = [\alpha_1, \dots, \alpha_P]$ is an array with the optimization variables. Now, Equation (9) can be evaluated analytically, as in [10], as long as the analytical expression of the far field as a function of the optimization variables is provided. In [10], the partial derivatives are obtained deriving the cost function with respect to the tangent of the phases. However, in this work derivation is done with respect to the phases themselves, improving the performance of the algorithm greatly by making it converge faster, as it will be shown later.

In the case where the analytical expressions cannot be used, the derivative can be calculated using finite differences of the form [14]:

$$\frac{\partial F_t(\alpha)}{\partial \alpha_p} \approx \frac{F_t(\alpha + he_p) - F_t(\alpha - he_p)}{2h}, \tag{10}$$

where $h$ is a small positive scalar and $e_p$ is the $p$th unit vector. Because the evaluation of the cost function (8) can be computationally expensive, a one-sided-difference can be used instead of the central difference of (10) in order to reduce by half the number of evaluations required. Also, a proper choice of $h$ can minimize the error of the evaluation of the derivative in (10). For the central difference the optimum choice of $h$ [14] is:

$$h = \sqrt[3]{u_r}, \tag{11}$$

with $u_r$ being the unit roundoff, whose value will depend on the precision of the real numbers used in the implementation of the algorithm. The optimum choice for a lateral difference would be the value $h = \sqrt{u_r}$. Note that, for the central difference there is an error of $O(h^2)$, while for the lateral one, the error is $O(h)$. Since $h \in (0, 1)$, the error will be lower for the central difference, although the evaluation of the derivative will be twice as expensive, in time consuming terms.

An important point to consider when computing the Jacobian matrix is the fact that the columns of $J$ are independent from each other because the derivatives are calculated with respect to one variable. Hence, the evaluation of the Jacobian can be fully parallelized by means of OpenMP [32], computing one column per available thread. Furthermore, each column can be obtained by just two calls to the cost function when the central difference is used (one call for the lateral difference). Also, the far field is computed efficiently by means of the FFT, and only one FFT is needed per cost function call. This way, one of the most time-consuming operations of the algorithm is performed efficiently and will scale well with the number of available processors, allowing the optimization of large problems.

### C. Solving the matrix equation

Once the Jacobian matrix is calculated, the LMA can be applied iteratively as:

$$[J_i^T \cdot J_i + \mu_i \cdot \mathrm{diag}(J_i^T \cdot J_i)] \cdot \delta_i = -J_i^T \cdot F_{t,i}, \qquad (12)$$

which can be compactly written as:

$$A_i \cdot \delta_i = b_i. \qquad (13)$$

In (12), the subindex $i$ represents the current iteration, $\mathrm{diag}(\cdot)$ is the diagonal matrix, $\delta_i$ is the updating vector which satisfies the equality and $\mu_i$ is a real positive number. The choice of $\mathrm{diag}(J_i^T \cdot J_i)$ instead of any other positive diagonal matrix, such as the identity, is to reduce the effects of poor scaling in the optimization variables by using an ellipsoidal trust region. This way, the algorithm is invariant under diagonal scaling of the components of α [14].

The matrix multiplication $J_i^T \cdot J_i$ and other matrix-vector operations can be computationally very expensive if the dimension is large. Nevertheless, these operations can be performed by routines from libraries such as OpenBLAS [33] or MKL [34], which take advantage of highly optimized and fully parallelized algorithms and low-level hardware operations in order to improve their performance and computing times. Also, since the resulting matrix is symmetric, only the upper or lower triangular part of it can be computed, further reducing computing times.

In [10], (13) is solved by forming its normal equation applying the Conjugate Gradient Squared (CGS) method. This is unnecessary because (13) is already a square matrix system, and the CGS additionally solves another system of normal equations, thus squaring the condition number of matrix $A_i$, which can lead to poor convergence of the CGS depending on the initial Jacobian matrix.

Nevertheless, because $A_i$ is at least positive semi-definite, a Cholesky factorization based solver can be used [35], which is the fastest exact solver for this type of problems [36] since it takes advantage of the symmetric nature of the matrix. Compared with other methods, the Cholesky factorization involves $P^3/3$ floating-point operations, while LU takes $2P^3/3$ and SVD $12P^3$ [36]. Although SVD is more robust, in this case the Cholesky factorization is enough, being 26 times faster than SVD and twice as fast as LU.

After the matrix system is solved, the solution is updated as:

$$\alpha_{i+1} = \alpha_i + \delta_i. \qquad (14)$$

### D. Choice of $\mu_0$

The parameter $\mu$ in (12) is used to control the convergence of the algorithm. It controls the behaviour of the algorithm ranging from the steepest descent when $\mu \to \infty$ and the Gauss-Newton method when $\mu = 0$ [15]. A small value of $\mu$ when the current solution is not near the minimum may cause the algorithm to diverge. Hence, it is recommended to start with a high value of $\mu$ and decrease it as the error diminishes. Conversely, if the error increases, it could be necessary to increase the value of $\mu$ to control the algorithm and prevent it from diverging to non-desired solutions. A new real parameter $\beta > 1$ is defined to control $\mu$. If the last $k_d$ iterations have decreased the error, $\mu$ is updated as $\mu_{i+1} = \mu_i/\beta$. On the other hand, if the last $k_i$ iterations have increased the error, $\mu$ is updated as $\mu_{i+1} = \mu_i\beta$. If neither of both conditions are fulfilled, $\mu$ remains the same.

Parameters $\beta$, $k_d$ and $k_i$ are artificially introduced in the algorithm in order to control $\mu$. On the one hand, $\beta$ controls how fast $\mu$ is decreased when the error diminishes and vice versa. A high value of $\beta$ causes $\mu$ to decrease initially very fast, and could lead the algorithm to diverge, hence rapidly increasing the value of $\mu$. These swings in $\mu$ can cause the algorithm to either diverge or converge to non-desired local minima. A low value of $\beta$ (close to 1) causes the algorithm to converge slowly, specially when $\mu$ is high. It has been found that values of $\beta$ between 1.05 and 1.5 provide a good trade-off for a good rate of convergence while keeping the algorithm from diverging. On the other hand, $k_d$ and $k_i$ control how tolerant the algorithm is to changes in $\mu$ when the error oscillates. When the algorithm reaches a flat valley in the search space, the error might behave irregularly due to the low gradient of the hypersurface around the current position. In order to control $\mu$ in this situation, $k_d$ and $k_i$ come into play, having complementary roles. If the error starts decreasing, $k_d$ prevents $\mu$ from decreasing too much if previously the error had increased. This tries to prevent the algorithm from diverging because it ensures that $\mu$ remains constant until the error decreases $k_d$ consecutive iterations. Conversely, the error has to increase during $k_i$ consecutive iterations in order to increase $\mu$. This prevents $\mu$ from increasing too much making the converge slower once the error begins to decrease again. Some reasonable values for these two parameters are within the range from 2 to 5.

However, the problem of choosing $\mu_0$ remains. Nevertheless, there is a suitable strategy to choose it. The optimization variables are the reflection coefficient phases. Hence, in order to have a reliable design, the phase distribution should be as smooth as possible, because that way, the physical dimensions of the elements of the reflectarray would vary smoothly from one element to the next (which is necessary because the reflectarray analysis is based on a full-wave analysis assuming local periodicity [1,26-29]). Following this, $\mu_0$ should be high enough to allow the phase at the initial iterations to vary smoothly and not to make jumps to valleys of the search space with noisy phase distributions. Due to the dimensionality of the problem being $P$ (usually of the order of hundreds or thousands of optimization variables, at least one per element of the reflectarray), it is very easy to make false steps into non-desired solutions during the first iterations of the algorithm from which it will be virtually impossible to escape. For that reason a high value of $\mu_0$, about the same order of magnitude of $P$ (for instance, between $0.5P$ and $5P$), is a good choice.

### E. Starting point

Another important issue is the starting point for the optimization process, which has been widely discussed in the literature [9,20]. As the LMA is a local optimizer, the starting point is of the utmost importance, since it will determine if the achieved solution is good enough. It has been determined [20] that a good initial point is that of a pencil beam pattern properly focused. Also, a pencil beam pattern can provide a smooth enough initial phase distribution in the center of the reflectarray, where the field amplitude is higher [1], depending on the placing of the feed antenna.

### F. Seeking the correct solution

In the previous sections, different aspects of the algorithm have been discussed that can prevent it from converging to a solution. First, the specifications should be reasonable; i.e., not too stringent, to allow the algorithm to converge from the first iteration. Also, the starting point should be good enough and $\mu_0$ high to ensure a soft descend towards the solution. In practise, the final solution should not only have a low error, but also a smooth phase distribution. With regard to having a low error, it means that the co-polar pattern is close to meet the specifications, which is the main goal of the synthesis. However, a smooth phase distribution is also needed for a real reliable design. Due to the local periodicity assumption when analysing the reflectarray unit cell [1], the design will perform better when the physical size of the reflectarray elements will vary smoothly across the surface of the antenna. The smoothness of the phase distribution is more critical in the centre of the reflectarray were the illumination level is higher.

### G. Numerical implementation

A homemade version of the algorithm including all the described optimizations has been implemented in Fortran using the Intel Fortran Compiler and MKL library [34]. Double precision is used for real numbers, which implies a unit roundoff of:

$$u_r = 2.22 \ldots \cdot 10^{-16}. \tag{15}$$

This means that the error for the lateral difference would be of the order of $O(10^{-8})$, while for the central difference would be $O(10^{-11})$. The difference between both errors is not as impressive due to the optimum choice of $h$ to minimize the error in the derivative taking into account errors produced in floating-point arithmetic [14]. Since the rest of the operations are performed in double precision, the error in the evaluation of the Jacobian will propagate and will be the limiting factor in the minimum achievable error. However, since a phase-only synthesis is done, the evaluation of the cost function is fast, and the central difference is used in the optimization.

## IV. VALIDATION

### A. Antenna specifications

An outline of the reflectarray is shown in Fig. 1. It consists of a planar rectangular reflectarray with dual linear polarization formed by 900 elements $(30 \times 30)$ and a feed horn modelled as a $\cos^{q}\theta$ function with a q-factor of 37, which produces an illumination taper of $-19.5$ dB on the surface of the reflectarray. The feed horn points to the centre of the reflectarray and its phase centre is placed at $\vec{r}_f = (-94, 0, 214)$ mm with regard to the centre of the reflectarray. The working frequency is 25.5 GHz and the periodicity of the elements is $5.84\ \text{mm} \times 5.84\ \text{mm}$, which is approximately half a wavelength [37]. Also, the far fields are discretized in a $128 \times 128$ UV grid, being $T = 16384$. Note that, according to Fig. 1, the X-polarization corresponds to the vertical polarization (V) because the electric field in $\hat{x}$-direction is vertical, while Y-polarization corresponds to the horizontal polarization (H).

The chosen pattern is a LMDS, which presents a 30º-sector beam in azimuth and a square cosecant beam in elevation [13]. Templates in the main cuts will be presented along with the results of the optimization in the next section.

### B. Optimization of previous synthesis

In order to test the described procedure, a synthesis for a LMDS pattern was carried out. The first example uses as a starting point the final result of [13] for both polarizations. This constitutes an excellent initial radiation pattern since it is very close to the final specification. The geometry of the antenna is the same as in [37], with the LMA parameters set to $\mu_0 = 1800, \beta = 1.05, k_d = 3$, $k_i = 2$ and $C(\vec{r}_t) = 1$. The initial error is 2.99 and 2.95 for vertical and horizontal polarizations, according to (8). The convergence is very similar for both polarizations. In the case of vertical polarization, after 500 iterations of the LMA, the error was $7.8 \times 10^{-10}$. The algorithm was left to complete 999 iterations. However, after iteration 500, approximately, it stagnates. The lowest error was $5.57 \times 10^{-10}$ at iteration 990. For the horizontal polarization, the lowest error was $3.78 \times 10^{-10}$ at iteration 995, stagnating around iteration 650.

The simulated radiation pattern for vertical polarization is shown in Fig. 2. The main cuts for horizontal polarization are shown in Fig. 3 along with the results of [13]. With a global error of the order of $10^{-10}$, the radiated fields fit very well to the masks, improving back lobes and the coverage zone with regard to [13] while maintaining a similar shape of the synthesized phases, which are shown in Fig. 4.

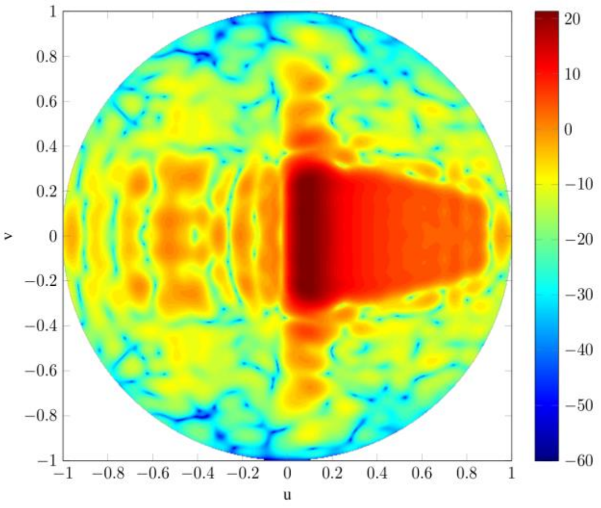
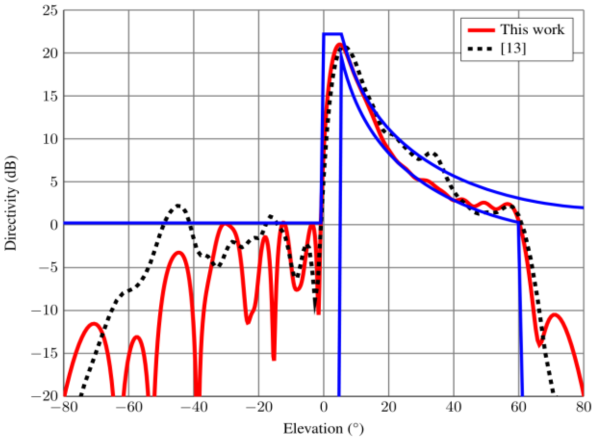
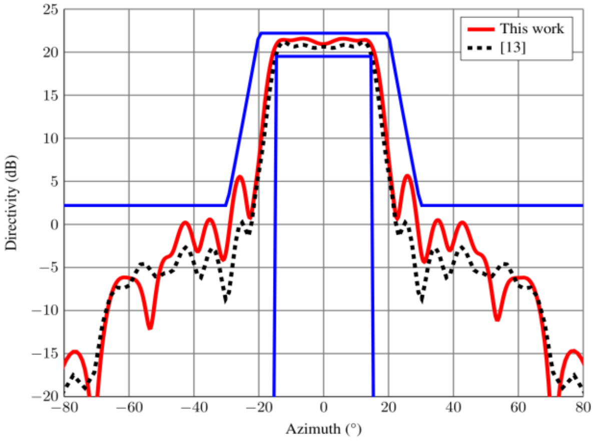


Fig. 2. Three dimensional synthesized radiation pattern in directivity (dB) for vertical polarization.



(a)



(b)

Fig. 3. Radiation pattern of the synthesized reflectarray considering an ideal model of the feed horn in dual polarization. Main cuts for horizontal polarization in (a) elevation and (b) azimuth.

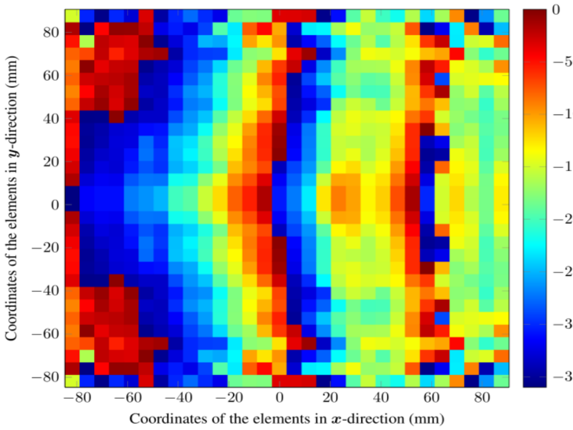
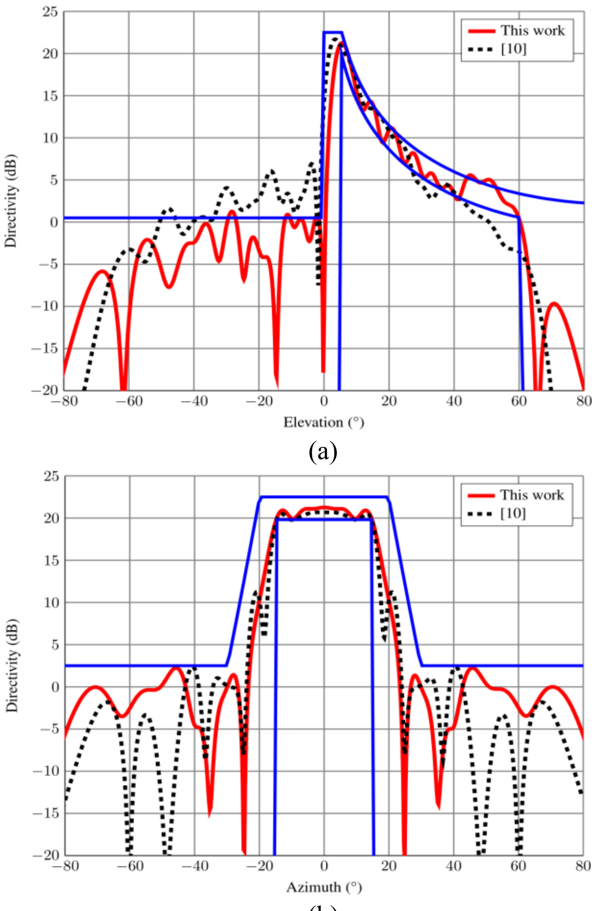


Fig. 4. Synthesized phase distribution of the reflection coefficient for the vertical polarization (degrees).

### C. Synthesis with a pencil beam as starting point

In order to compare more faithfully the results with [10], a new synthesis was carried out employing the same initial point; i.e., the phases of a pencil beam pattern pointing to $(\theta, \varphi) = (5.4°, 0.0°)$, which corresponds to the area of maximum directivity in the masks. The LMA parameters for this case were $\mu_0 = 500, \beta = 1.1, k_d = 3$, $k_i = 2$ and $C(\vec{r}_t) = 1$. First, the H-polarization was synthesized from the phase distribution of the pencil beam. The initial error was 53.00 and after the iteration 450 (where the error was $5.05 \times 10^{-7}$) it stagnates. The lowest error achieved was $3.87 \times 10^{-7}$ at iteration 998, out of 999. After the synthesis of the H-polarization, the V-polarization was synthesized starting with the synthesized phases of the H-polarization. This resulted in an initial error of $7.03 \times 10^{-3}$ because the pattern is closer to the masks than the pencil beam, although higher than the final error for the H-polarization because the incident field is different for both polarizations. The lowest error achieved was $2.97 \times 10^{-9}$ at iteration 471. As comparison, the final error in [10] is $5.60 \times 10^{-3}$, which is several orders of magnitude higher than the error obtained in this work.

Figure 5 shows the main cuts for the horizontal polarization of the new synthesized radiation pattern. Because the starting point is not as good as in the previous case, the final pattern obtained now is slightly worse, although it greatly improves the results of [10]. In particular, the back lobes are reduced by about 6 dB and the coverage zone improves for large angles. Also, the results were obtained in less iterations (less than 500 vs. 3900, for each polarization) and with a final error several orders of magnitude lower, which accounts for the better results in the radiation patterns. Finally, Fig. 6 shows the synthesized phase distribution when using a pencil beam pattern as starting point.



(a)

(b)

Fig. 5. Radiation pattern of the synthesized reflectarray considering an ideal model of the feed horn in dual polarization with starting point a pencil beam pattern. Main cuts for horizontal polarization in (a) elevation and (b) azimuth.

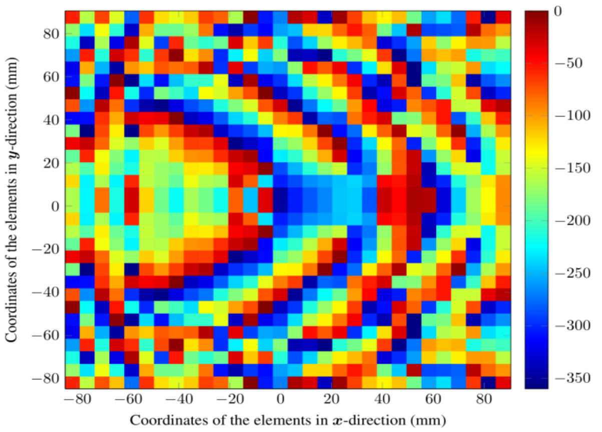


Fig. 6. Synthesized phase distribution of the reflection coefficient for the vertical polarization using a pencil beam as starting point (degrees).

### D. Improvements in computing times

With the optimizations detailed in previous sections, the computing times were greatly reduced. In [10], it is reported that each iteration takes less than a minute. Here, each iteration takes about 5.7 seconds using the same computer (Intel Core 2 Duo with a 2.4 GHz processor), which along with the improved convergence of the LMA, reduces significantly the computing times of the synthesis process. The time for each iteration is reduced approximately by a factor of 10. Moreover, taking into account that the synthesis process took less than 500 iterations for each polarization (about eight times faster), the overall improvement in computing times is by a factor of 80.

## V. CONCLUSION

An improved phase-only synthesis for reflectarrays based on the Levenberg-Marquardt algorithm with an ellipsoidal trust region has been developed, improving the accuracy and efficiency with regard to other works in the literature. By optimizing each building block of the algorithm, a great computational efficiency is achieved that will allow for more powerful synthesis techniques implemented with the same algorithm. For instance, it will be possible to implement a direct optimization of the geometry of the reflectarray through full wave analysis of the reflectarray unit cell [27-29] within reasonable computing times.

In particular, the Jacobian matrix is obtained through finite differences which allows to avoid using the analytical expressions for complex problems. By choosing the appropriate value of the increment in the finite difference equation, the error evaluating the derivative is minimized. Also, the columns of the Jacobian can be computed independently from each other, which allows to fully parallelize its evaluation. By deriving with respect to the reflection coefficient phases instead of the tangent of the phases, the convergence of the algorithm is improved.

Further improvements were made regarding the choice of the solver for the equation system, where a Cholesky factorization based solver was selected to take advantage of the symmetry of the resulting matrix. Also, computationally expensive operations such as matrix and matrix-vector multiplications were performed using highly optimized and parallelized routines. Since the result of the matrix multiplication is symmetric, only the lower or upper triangular part needs to be computed.

In addition, due to the intrinsic high dimensionality of the problem, a few guidelines have been laid out in order to control the evolution of the synthesis, which allows for a better control of the obtained solution. In particular, it has been shown how the initial point of the synthesis is a key factor in a local search optimizer such as the LMA. Also, a suitable choice of the parameters of the LMA is important in order to control the speed of convergence as well as the initial evolution of the algorithm, which can determine the path to a good or bad solution.

Finally, two test cases of a LMDS pattern were shown to validate the proposed solution. The performance of the algorithm has proven to be better than others in the literature. The results are more accurate, reducing back lobes and better controlling the coverage zone, while reducing the computing times by a factor of 80.

## ACKNOWLEDGMENT

This work was supported in part by the European Space Agency (ESA) under contract ESTEC/AO/1-7064/12/NL/MH; by the Ministerio de Economía y Competitividad, under project TEC2014-54005-P (MIRIIEM); by the Gobierno del Principado de Asturias/FEDER under project GRUPIN14-114 and under grant BP12-063; and contract FUO-EM-151-09.

## REFERENCES

[1] J. Huang and J. A. Encinar, *Reflectarray Antennas*, John Wiley & Sons, 2008.

[2] M. Arrebola, E. Carrasco, and J. A. Encinar, "Beam scanning antenna using a reflectarray as sub-reflector," *Appl. Comp. Electro. Society (ACES) Journal*, vol. 26, no. 6, pp. 473-483, Jun. 2011.

[3] B. Devireddy, A. Yu, F. Yang, and A. Z. Elsherbeni, "Gain and bandwidth limitations of reflectarrays," *Appl. Comp. Electro. Society (ACES) Journal*, vol. 26, no. 2, pp. 170-178, Feb. 2011.

[4] J. A. Encinar and J. A. Zornoza, "Broadband design of three-layer printed reflectarrays," *IEEE Trans. Antennas Propag.*, vol. 51, no. 7, pp. 1662-1664, Jul. 2003.

[5] S. H. Yusop, N. Misran, M. T. Islam, and M. Y. Ismail, "Design of high performance dual frequency concentric split ring square element for broadband reflectarray antenna," *Appl. Comp. Electro. Society (ACES) Journal*, vol. 27, no. 4, pp. 334-339, Apr. 2012.

[6] P. M. Woodward, "A method of calculating the field over a plane aperture required to produce a given polar diagram," *J. Inst. Elec. Eng.*, vol. 93, pt. III, no. 10, pp. 1554-1558, 1946.

[7] J. Perini and M. Idselis, "Note on antenna patterns synthesis using numerical iterative methods," *IEEE Trans. Antennas Propag.*, vol. 19, no. 2, pp. 284-286, Mar. 1971.

[8] T. S. Fong and R. A. Birgenheier, "Method of conjugate gradients for antenna pattern synthesis," *Radio Sci.*, vol. 6, no. 12, pp. 1123-1130, Dec. 1971.

[9] O. M. Bucci, G. Franceschetti, G. Mazzarella, and G. Panariello, "Intersection approach to array pattern synthesis," *IEE Proc. Microw. Antennas Propag.*, vol. 137, no. 6, pp. 349-357, Dec. 1990.

[10] J. Álvarez, M. Arrebola, R. G. Ayestarán, and F. Las-Heras, "Systematic framework for reflectarray synthesis based on phase optimization," *Int. J. Antennas Propag.*, vol. 2012, pp. 1-9, Jun. 2012.

[11] J. M. Johnson and Y. Rahmat-Samii, "Genetic algorithm optimization and its application to antenna design," in *Antennas and Propagation Society International Symposium, 1994. AP-S. Digest*, vol. 1, pp. 326-329, Jun. 1994.

[12] P. Nayeri, F. Yang, and A. Z. Elsherbeni, "Design of single-feed reflectarray antennas with asymmetric multiple beams using the particle swarm optimization method," *IEEE Trans. Antennas Propag.*, vol. 61, no. 9, pp. 4598-4605, Sep. 2013.

[13] M. Arrebola, J. A. Encinar, and M. Barba, "Multifed printed reflectarray with three simultaneous shaped beams for LMDS central station antenna," *IEEE Trans. Antennas Propag.*, vol. 56, no. 6, pp. 1518-1527, Jun. 2008.

[14] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., Springer, 2006.

[15] G. A. F. Seber and C. J. Wild, *Nonlinear Regression*, John Wiley & Sons, 2003.

[16] O. M. Bucci, G. D'Elia, G. Mazzarella, and G. Panariello, "Antenna pattern synthesis: a new general approach," *Proc. IEEE*, vol. 82, no. 3, pp. 358-371, Mar. 1994.

[17] O. M. Bucci, G. D'Elia, and G. Romito, "Power synthesis of conformal arrays by a generalized projection method," *IEE Proc. Microw. Antennas Propag.*, vol. 142, no. 6, pp. 467-471, Dec. 1995.

[18] D. W. Boeringer and D. H. Werner, "Particle swarm optimization versus genetic algorithms for phased arrays synthesis," *IEEE Trans. Antennas Propag.*, vol. 52, no. 3, pp. 771-779, Mar. 2004.

[19] I. Lopez, J. R. Pérez, and J. Basterrechea, "An approach for the design of reflectarrays using CG-FFT and PSO," in *Proceedings of the Fourth*

*European Conference on Antennas and Propagation (EuCAP)*, Barcelona, Spain, pp. 1-5, Apr. 12-16, 2010.

[20] J. A. Zornoza and J. A. Encinar, "Efficient phase-only synthesis of contoured-beam patterns for very large reflectarrays," *Int. J. RF Microw. Comput. Eng.*, vol. 14, no. 5, pp. 415-423, Sep. 2004.

[21] Y.-T. Lo and S.-W. Lee, Eds., *Antenna Handbook Vol. 1*, Van Nostrand Reinhold, 1993.

[22] J. A. Encinar, L. S. Datashvili, J. A. Zornoza, M. Arrebola, M. Sierra-Castaner, J. L. Besada-Sanmartin, H. Baier, and H. Legay, "Dual-polarization dual-coverage reflectarray for space applications," *IEEE Trans. Antennas Propag.*, vol. 54, no. 10, pp. 2827-2837, Oct. 2006.

[23] J. A. Encinar and M. Barba, "Design, manufacture and test of a Ka-band reflectarray antenna for transmitting and receiving in orthogonal polarization," in *14th International Symposium on Antenna Technology and Applied Electromagnetics (ANTEM) and the American Electromagnetics Conference (AMEREM)*, Ottawa, Canada, pp. 1-4, Jul. 5-8, 2010.

[24] J. A. Encinar, M. Arrebola, L. F. de la Fuente, and G. Toso, "A transmit-receive reflectarray antenna for direct broadcast satellite applications," *IEEE Trans. Antennas Propag.*, vol. 59, no. 9, pp. 3255-3264, Sep. 2011.

[25] R. Florencio, J. Encinar, R. R. Boix, and G. Perez-Palomino, "Dual-polarisation reflectarray made of cells with two ortogonal sets of parallel dipoles for bandwidth and cross-polarisation improvement," *IET Microw. Antennas Propag.*, pp. 1-9, Aug. 2014.

[26] P. Nayeri, A. Z. Elsherbeni, and F. Yang, "Design, full-wave analysis, and near-field diagnostics of reflectarray antennas," *Appl. Comp. Electro. Society (ACES) Journal*, vol. 28, no. 4, pp. 284-292, Apr. 2013.

[27] C. Wan and J. A. Encinar, "Efficient computation of generalized scattering matrix for analyzing multilayered periodic structures," *IEEE Trans. Antennas Propag.*, vol. 43, no. 11, pp. 1233-1242, Nov. 1995.

[28] R. Florencio, R. R. Boix, and J. A. Encinar, "Enhanced MoM analysis of the scattering by periodic strip gratings in multilayered substrates," *IEEE Trans. Antennas Propag.*, vol. 61, no. 10, pp. 5088-5099, Oct. 2013.

[29] D. M. Pozar and T. A. Metzler, "Analysis of a reflectarray antenna using microstrip patches of variable size," *Electron. Lett.*, vol. 29, no. 8, pp. 657-658, Apr. 1993.

[30] J. Álvarez, R. G. Ayestarán, G. León, L. F. Herrán, A. Arboleya, J. A. López-Fernández, and F. Las-Heras, "Near field multifocusing on antenna arrays via non-convex optimisation," *IET Microw. Antennas Propag.*, vol. 8, no. 10, pp. 754-764, Jul. 2014.

[31] S. E. Nai, W. Ser, Z.-L. Yu, and H. Chen, "Beampattern synthesis for linear and planar arrays with antenna selection by convex optimization," *IEEE Trans. Antennas Propag.*, vol. 58, no. 12, pp. 3923-3930, Dec. 2010.

[32] M. Sato, "OpenMP: parallel programming API for shared memory multiprocessors and on-chip multi-processors," in *15th International Symposium on System Synthesis*, Kyoto, Japan, pp. 109-111, Oct. 2-4, 2002.

[33] Z. Xianyi, W. Qian, and Z. Yunquan, "Model-driven level 3 BLAS performance optimization on Loongson 3a processor," in *IEEE 18th International Conference on Parallel and Distributed Systems (ICPADS)*, Singapore, pp. 684-691, Dec. 17-19, 2012.

[34] *Intel® Math Kernel Library Reference Manual*, Intel Corporation, Aug. 2008.

[35] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed., Cambridge University Press, 2013.

[36] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 4th ed., The Johns Hopkins University Press 2013.

[37] M. Arrebola, Y. Álvarez, J. A. Encinar, and F. Las-Heras, "Accurate analysis of printed reflectarrays considering the near field of the primary feed," *IET Microw. Antennas Propag.*, vol. 3, no. 2, pp. 187-194, Mar. 2009.