mathematics

MDPI

Article

Electrostatic Capacity of a Metallic Cylinder: Effect of the Moment Method Discretization Process on the Performances of the Krylov Subspace Techniques

Mario Versaci [1,*] and Giovanni Angiulli [2]

[1] Dipartimento di Ingegneria Civile Energia Ambiente e Materiali, "Mediterranea" University, Via Graziella Feo di Vito, I-89122 Reggio Calabria, Italy.

[2] Dipartimento di Ingegneria dell'Informazione Infrastrutture Energia Sostenibile, "Mediterranea" University, Via Graziella Feo di Vito, I-89122 Reggio Calabria, Italy; giovanni.angiulli@unirc.it

* Correspondence: mario.versaci@unirc.it; Tel.: +39-0965-8752273

Received: 5 August 2020; Accepted: 24 August 2020; Published: 26 August 2020

Abstract: When a straight cylindrical conductor of finite length is electrostatically charged, its electrostatic potential ϕ depends on the electrostatic charge q_e, as expressed by the equation $\mathcal{L}(q_e) = \phi$, where $\mathcal{L}$ is an integral operator. Method of moments (MoM) is an excellent candidate for solving $\mathcal{L}(q_e) = \phi$ numerically. In fact, considering q_e as a piece-wise constant over the length of the conductor, it can be expressed as a finite series of weighted basis functions, $q_e = \sum_{n=1}^{N} \alpha_n f_n$ (with weights α_n and N, number of the subsections of the conductor) defined in the $\mathcal{L}$ domain so that ϕ becomes a finite sum of integrals from which, considering testing functions suitably combined with the basis functions, one obtains an algebraic system $L_{mn}\alpha_n = g_m$ with dense matrix, equivalent to $\mathcal{L}(q_e) = \phi$. Once solved, the linear algebraic system gets α_n and therefore q_e is obtainable so that the electrostatic capacitance $C = q_e/V$, where V is the external electrical tension applied, can give the corresponding electrostatic capacitance. In this paper, a comparison was made among some Krylov subspace method-based procedures to solve $L_{mn}\alpha_n = g_m$. These methods have, as a basic idea, the projection of a problem related to a matrix $A \in \mathbb{R}^{n \times n}$, having a number of non-null elements of the order of n, in a subspace of lower order. This reduces the computational complexity of the algorithms for solving linear algebraic systems in which the matrix is dense. Five cases were identified to determine L_{mn} according to the type of basis-testing functions pair used. In particular: (1) pulse function as the basis function and delta function as the testing function; (2) pulse function as the basis function as well as testing function; (3) triangular function as the basis function and delta function as the testing function; (4) triangular function as the basis function and pulse function as the testing function; (5) triangular function as the basis function with the Galerkin Procedure. Therefore, five L_{mn} and five pair q_e and C were computed. For each case, for the resolution of $L_{mn}\alpha_n = g_m$ obtained, GMRES, CGS, and BicGStab algorithms (based on Krylov subspaces approach) were implemented in the MatLab® Toolbox to evaluate q_e and C as N increases, highlighting asymptotical behaviors of the procedures. Then, a particular value for N is obtained, exploiting both the conditioning number of L_{mn} and considerations on C, to avoid instability phenomena. The performances of the exploited procedures have been evaluated in terms of convergence speed and CPU-times as the length/diameter and N increase. The results show the superiority of BcGStab, compared to the other procedures used, since even if the number of iterations increases significantly, the CPU-time decreases (more than 50%) when the asymptotic behavior of all the procedures is in place. This superiority is much more evident when the CPU-time of BicGStab is compared with that achieved by exploiting Gauss elimination and Gauss–Seidel approaches.

Keywords: electrostatic charge distribution and capacitance; MoMs; Krylov subspaces

1. Introduction

As it is well known, electrostatic problems are classified according to the equations that describe them. These equations could be differential (ordinary or partial derivative), integral, or both. Most electrostatic problems can be represented by an abstract equation [1,2]

$$\mathcal{L}F = h \tag{1}$$

in which $\mathcal{L}$ is an operator (differential, integral, or integro-differential), h is the known excitation or source, and F is the unknown function to be determined. For solving the problem (1), it is usually transformed into an equivalent algebraic problem

$$L\alpha = g, \tag{2}$$

in which L is an algebraic matrix [2–4]. As it is well known, modern computers have created the conditions for developing numerical resolution methods for a wide class of problems lacking an analytical solution. Among them is the MoM which, initially introduced by Harrington [1,2], solves (1), transforming it into an equivalent algebraic formulation.

With this aim, the unknown is expanded using a known set of functions (the basis functions), and Equation (1) is then projected on another set of functions (the testing functions). The basis functions can be the same as the testing functions (Galerkin method) or chosen differently. Specifically, the choice of basis and testing functions determine, on the one hand, the formulation of the elements of the algebraic matrix L and, on the other, influences the speed of convergence of the methods employed to solve (2) [3,5]. However, the MoM transforms (1) into (2), where L is usually dense [1,3,6,7].

With this purpose, the MoM has been used with excellent results [8–10]. Specifically, the pulse and delta functions, respectively, as the basis and testing functions, have been used to solve (1) according to the observation that the choice of the basis and testing functions substantially influences the desired level of accuracy [1,11,12].

The scientific literature is rich concerning the study of q_e distributions and C computations through the MoM in which different basis and testing functions are exploited [10,13–19]. For example, in [14], related to a single cylindrical conductor, the q_e distribution was studied, and the C were studied when different basis and testing functions were exploited (pulse, delta, triangular functions). However, in these works, the solutions were obtained by exploiting only direct techniques for solving the (2) [17,19].

In this paper, starting from [14], the problem of the distribution of q_e on a cylindrical metallic conductor of finite length as well as the calculation of the related C was addressed when an external electric voltage V was applied. In particular, once the geometry of the problem was defined and the source and observation points fixed, the problem was formulated in terms of electrostatic potential ϕ by means of an integral formulation. Then, the basis and testing functions were defined and q_e developed in terms of a finite set of basis functions f_n, $n = 1, \ldots, N$,

$$q_e = \sum_{n=1}^{N} \alpha_n f_n, \tag{3}$$

with weights α_n. Applying the MoM, ϕ provides an equation with unknown N. However, choosing N points of independent observations on the surface of the conductor, we arrive at a formulation of $N \times N$ algebraic system

$$L_{mn}\alpha_n = g_m \tag{4}$$

in which the elements of the dense matrix L_{mn}, characterized by integral formulations, depend on the choice of basis and testing functions f_n and w_m, with $n = 1, \ldots, N$ and $m = 1, \ldots, N$ and g_m depending on w_m. The following five pairs of basis and testing functions were used: pulse and delta functions; pulse and pulse functions; triangular and delta functions (point matching); triangular and pulse functions; and triangular and triangular functions. Once the five dense matrices L were obtained (one for each pair of basis and testing function) by means of the MoM application, the corresponding linear algebraic systems were solved by using different Krylov subspaces methods because they are the most suitable for solving algebraic problems such as (2) in which L is dense and characterized by a large dimension [20–23]. In the past, Krylov subspaces have been widely exploited in both theoretical and applied areas. In particular, in mathematics, excellent works have been produced through the application of Krylov subspaces for solving sparse linear systems [21,23], for solving sequences of linear systems [24], and for obtaining the solution in quadratic-bilinear differential systems [25,26]. Furthermore, excellent works have been produced by applying the Krylov subspace for the resolution of sequences of non-linear structural problems [27] and for the topological problems of optimization [28]. Finally, the interest of a lot of researches focused on the Krylov subspaces for linear infinite-dimensional systems [29]. In the physical domain, Krylov subspaces have been exploited for solving a lot important equations such as the Dirac equation [30] and problems with matrix exponential operators [31]. Moreover, several works exploiting Krylov subspaces have been produced for solving practical problems. For example, in the chemical domain, preconditioned Krylov subspaces methods have been applied to evaluate the variational density [32,33]. On the other hand, in engineering, in particular, electromagnetics, many papers have been published concerning Krylov subspaces [34,35]. In particular, many efforts have been made to solve integral equations for scattering electromagnetic fields [36], for solving a class of tensor equations [37], and for large Lyapunov equations [38]. Moreover, even in the theory of electrical circuits, Krylov subspaces have been applied to obtain good performance in terms of simulations [39,40]. Although the scientific literature in the electromagnetic field is very rich in works in which Krylov subspaces have been successfully used, it lacks information regarding the application of Krylov subspaces for the computation of the electric charge and electrostatic capacity on electrical conductors. This is mainly due to the fact that in the literature, the calculation of these parameters often takes place through the usual solvers of linear systems since online computing was often not required. However, nowadays, real-time applications are increasingly frequent so that the usual solvers of linear systems do not offer high performances in terms of time-saving. Therefore, it is now essential to use solvers with computational complexity compatible with the times of real-time applications. In this context, the idea of the present work matures, which aims to compute the distribution of both electrostatic charge and capacity of conductors subjected to an external potential difference V using the methods based on Krylov subspaces, which notoriously offer computational complexity compared to the usual solver [21,23].

In this work, the generalized minimal residual method (GMRES), conjugate gradient squared (CGS) and biconjugate gradient stabilized (BicGStab) techniques have been implemented on an Intel Core 2 CPU 1.45 GHz machine and MatLab® R2017a, to solve the linear systems with dense matrix obtained by the MoM application to the electrostatic problem mentioned above to evaluate their performances by calculating the time consumed and the number of necessary iterations. Finally, the comparison was carried out both with two usual solution techniques for linear algebraic systems (Gauss elimination and Gauss–Seidel). The results obtained highlighted the superiority of the BicGStab procedure compared to the other approaches used. In particular, as the size of the problem increases, the number of iterations required by BicGStab, compared to the other procedures used, increases, but the CPU-time considerably reduces. Specifically, the highest CPU-time obtained with BicGStab (25.9 s) is less than (a) 46.1% of the highest CPU-time obtained with the CGS procedure (48.0254 s); (b) 27.91% of the highest CPU-time obtained with the GMRES procedure (35.92 s); (c) 81.93% of the highest CPU-time obtained with Gauss elimination (143.34 s), and (d) 93.1% of CPU-time detected with the Gauss-Seidel procedure (377.7439 s). Finally, since L in all studied cases is ill-conditioned, this suggests the need to consider and develop suitable pre-conditioners.

The remaining part of the paper is structured as follows: Section 2 describes the electrostatic problem under study: once the geometry of the conductor is defined (Section 2.1), the formulation of the electrostatic problem is detailed in Section 2.2. Section 3 offers a brief overview of the MoM to transform (1) into (2) and presents the sets of functions exploited as basis functions and testing ones. Section 4 develops the expansion of the electrostatic charge density in terms of basis functions and formulates the algebraic matrix following the MoM approach. Sections 4.1–4.5 show how the individual elements of the algebraic matrix can be obtained when different pairs of basis and testing functions are exploited. Then, once the algebraic systems are obtained, Section 5 offers a detailed overview of the Krylov subspace approach for solving high-dimensional linear systems. Again, Sections 5.2–5.8 describe the characteristics of each procedure exploited in this work and based on Krylov subspaces. The most significant numerical results are described and commented in detail in Section 6. Finally, some conclusive observations and possible future developments conclude the work (see Section 7).

2. The Electrostatic Problem

2.1. Geometry

Let us consider a system of Cartesian axes $Oxyz$ wherein the x axis is represented by the longitudinal symmetry axis of a cylindrical conductor occupying a region of $\mathbb{R}^3$ (Ω is the source region).

As shown in Figure 1, a cylindrical circular conductor is characterized by the length $\overline{L}$ and radius a. On the surface (plane xz) are located the observation points pointed by $\mathbf{r} = (x, 0, a)$, while the source points, on axis x, are pointed by $\mathbf{r}' = (x', 0, 0)$ so that $|\mathbf{r} - \mathbf{r}'| = \sqrt{(x-x')^2 + a^2} \neq 0$.

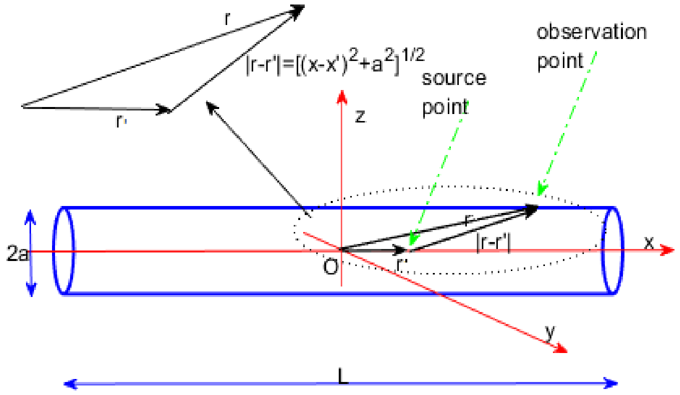


Figure 1. Cylinder with length $\overline{L}$ and radius a. $\mathbf{r}'$ and $\mathbf{r}$ are the position on which the potential ϕ (observation point) is evaluated and the position of the charge distribution on the cylinder, respectively (source point).

2.2. Formulation

As known, the electrostatic potential $\phi(x)$, computed at any point $\mathbf{r}$ due to any electrostatic charge q_e located at $\mathbf{r}'$ (see Figure 1), can be written as [1,2,5].

$$\phi(x) = \int_{\Omega} \frac{1}{4\pi\epsilon} \frac{q_e(x')}{|\mathbf{r} - \mathbf{r}'|} d\Omega = \int_{\Omega} \frac{1}{4\pi\epsilon} \frac{q_e(x')}{\sqrt{(x-x')^2 + a^2}} d\Omega. \tag{5}$$

Remark 1. *If we know q_e, we can compute the electric potential everywhere. Instead, if we know the electric potential but not the q_e, Equation (5) becomes an integral equation as $\mathcal{L}(q_e(x')) = \phi(x)$ for an unknown charge density so that $\mathcal{L}$ is an integral operator. Then, the problem must be solved numerically. The MoM is an excellent candidate for solving $\mathcal{L}(q_e(x')) = \phi(x)$ numerically [5,6].*

3. The Method of Moments: A Brief Overview

3.1. Backgrounds

Once (1) is obtained, the procedure for applying the MoM involves the following three steps:

1. the first step involves discretization of (1) into a matrix equation exploiting basis functions (or expansions) and weighting functions (or testing);
2. the evaluation of the matrix elements is obtained;
3. the matrix equation is solved to obtain the parameters of interest.

Concerning the expansions of f, a series of functions f_n defined in the $\mathcal{L}$ domain are considered so that

$$f \approx \sum_{n=1}^{N} \alpha_n f_n \tag{6}$$

where the α_n are the constants to be determined. We will call f_n the basis functions or expansion functions.

Remark 2. *To obtain the exact solution from (6), it is necessary to use an infinite sum and choose $f_n(x')$ as a complete set of basic functions. However, we will consider (6) with a finite summation, without being bound to choose a complete system for $f_n(x')$. On the other hand, $f_n(x')$ must be linearly independent and chosen so that (6) approximates $f(x')$.*

Substituting (6) into $\mathcal{L}f = g$, we get

$$g \approx \mathcal{L}\Big(\sum_{n=1}^{N} \alpha_n f_n\Big) = \sum_{n=1}^{N} \alpha_n \mathcal{L}(f_n) \tag{7}$$

where the residual R is

$$R = g - \sum_{n=1}^{N} \alpha_n \mathcal{L}(f_n). \tag{8}$$

We introduce a series of weighting functions (also called testing functions), $w_1, w_2, \ldots, w_N$, in the range of $\mathcal{L}$ and, assuming a suitable internal product of $\mathcal{L}f_n(x')$ with w_m, define it as follows:

Definition 1. *[Inner Product or Moment] Let us define an inner product or moment between a basis function $f_n(\mathbf{r}')$ and a testing or weighting function $w_m(\mathbf{r})$ as follows:*

$$< w_m, f_n >= \int_{w_m} w_m(\mathbf{r}) \cdot \int_{f_n} f_n(\mathbf{r}') d\mathbf{r}' d\mathbf{r} \tag{9}$$

in which the integral is surface, volume, or line depending on both the basis and testing functions.

It is required that the inner product of each w_m with the residual function be zero so that

$$\sum_{n=1}^{N} \alpha_n < w_m, \mathcal{L}(f_n) > \approx < w_m, g > \tag{10}$$

which results in the $N \times N$ matrix with element $< w_m, \mathcal{L}(f_n) >$. Then, the system can be matricially written as follows:

$$[L_{mn}][\alpha_n] = [g_m] \tag{11}$$

in which

$$[L_{mn}] = \begin{pmatrix} < w_1, \mathcal{L}(f_1) > & < w_1, \mathcal{L}(f_2) > & \dots \\ < w_2, \mathcal{L}(f_1) > & < w_2, \mathcal{L}(f_2) > & \dots \\ \dots & \dots & \dots \end{pmatrix}, \quad [\alpha_n] = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \dots \end{pmatrix}, \quad [g_m] = \begin{pmatrix} < w_1, g > \\ < w_2, g > \\ \dots \end{pmatrix}. \tag{12}$$

Obviously, if $[L_{mn}]$ is not singular, then the unknowns α_n are as follows:

$$[\alpha_n] = [L_{mn}]^{-1}[g_m], \tag{13}$$

in order that $f(x')$ can be reconstructed by Equation (6).

Remark 3. *Usually, L is a large matrix so that the usual methods (direct or iterative) for solving linear algebraic systems struggle to obtain the solution in a reasonable time, requiring the use of alternative procedures for solving (13). The procedures for solving linear systems based on Krylov subspaces are a valid alternative to the classic resolution methods, as they ensure reduced computation times even in the presence of large matrices [21,23].*

3.2. Point Matching

We applied the boundary conditions by testing the integral equation in a series of discrete points on the conductor. This is equivalent to using a delta function as a weighting function $w_m(\mathbf{r}) = \Delta(\mathbf{r})$. This procedure (point matching or point collocation), in evaluating the elements of the matrix, does not require any integral in the range of the test function but only that related to the pulse function. However, the boundary conditions are met only in discrete locations across the domain of the solution, allowing it to assume a different value at points other than those used for testing.

3.3. Galerkin Procedure

The choice of the testing function is decisive for obtaining a good solution. According to the Galerkin approach, basis functions can also be used as testing functions with the definite advantage of applying the boundary conditions in the whole domain of the solution than in discrete points.

Remark 4. *It is worth noting that, as in the following, Galerkin testing refers to a testing procedure where the set of testing functions is the same as the set of basis functions. However, in this work, even when the same type of functions are used as basis and testing functions, the sets are different since the testing functions are translated with respect to the basis functions. Thus, in this work, Galerkin testing is not used. Then, we shall use the term "pseudo-Galerking procedure".*

3.4. The Set of Functions

3.4.1. Pulse Functions

Dividing the domain into $N+1$ points with N sections, the Pulse function is defined as follows:

$$\begin{cases} f_n(x) = 1 & x_n \leq x \leq x_{n+1}; \\ f_n(x) = 0 & \text{elsewhere.} \end{cases} \tag{14}$$

The pulse functions represent a simple approximation of the solution on each section (Figure 2a). However, they can only simplify the computation of the elements of the matrix $\mathcal{L}$. Since the derivative of the pulse functions is the delta function Δ, they cannot be used when $\mathcal{L}$ contains a derivative with respect to x.

3.4.2. Delta Function

As known

$$\begin{cases} \Delta(x) = \text{singular} \quad \text{if} \;\; x = 0; \\ \Delta(x) = 0 \qquad\quad \text{elsewhere}. \end{cases} \tag{15}$$

It is worth noting that $\Delta(x)$ is not a function but it represents a distribution.

3.4.3. Piecewise Triangular Functions

Triangular functions, unlike the pulse functions, extend over two adjacent segments and vary from zero in the external points while it is equal to the unit in the center (Figure 2b). Specifically, dividing the domain into $N+1$ points and N sections will result in an $N-1$ triangular basis function. Generally, a triangular function is defined as follows:

$$\begin{cases} f_n(x) = \frac{x - x_{n-1}}{x_n - x_{n-1}}; \quad x_{n-1} \leq x \leq x_n \\ f_n(x) = \frac{x_{n+1} - x}{x_{n+1} - x_n} \quad x_n \leq x \leq x_{n+1}. \end{cases} \tag{16}$$

These functions can also be used when $\mathcal{L}$ contains derivatives with respect to x.

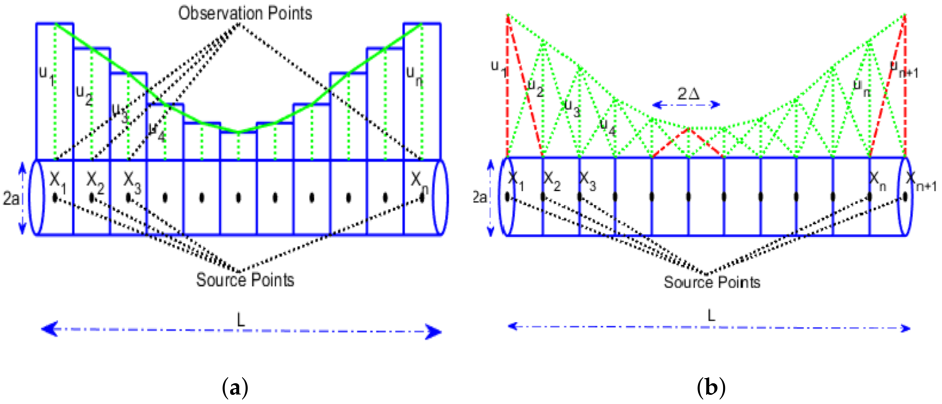


Figure 2. (**a**) Pulse functions and (**b**) triangular functions.

4. The $[L_{mn}]$ Formulations

Let us divide the conductor into $N \in \mathbb{N}$ sections with length Δx valuable as $\Delta x = \frac{L}{N}$. In addition, let us consider $N+1$ points $x_m = m\Delta x$, $m = 1, 2, \ldots, N$ at the ends of each section so that $\Omega = [0, L]$. We highlight that within each section, the charge density has a constant value. Then, $q_e(x')$ is the piece-wise constant over the length of the conductor. Therefore, according to (6), we write the following:

$$q_e(x') = \sum_{n=1}^{N} \alpha_n f_n(x') \tag{17}$$

Hence, substituting (17) into (5), we obtain

$$\phi(x) = \int_\Omega \frac{1}{4\pi\epsilon} \frac{q_e(x')}{\sqrt{(x-x')^2 + a^2}} dx' = \frac{1}{4\pi\epsilon} \sum_{n=1}^{N} \alpha_n \int_{(n-1)\Delta x}^{n\Delta x} \frac{f_n(x')}{\sqrt{(x-x')^2 + a^2}} dx' \tag{18}$$

from which

$$4\pi\epsilon\phi(x) = \alpha_1 \int_0^{\Delta x} \frac{f_1(x')}{\sqrt{(x-x')^2+a^2}}dx' + \alpha_2 \int_{\Delta x}^{2\Delta x} \frac{f_2(x')}{\sqrt{(x-x')^2+a^2}}dx' + \ldots + \\ +\alpha_{N-1} \int_{(N-2)\Delta x}^{(N-1)\Delta x} \frac{f_{N-1}(x')}{\sqrt{(x-x')^2+a^2}}dx' + \alpha_N \int_{(N-1)\Delta x}^{N\Delta x} \frac{f_N(x')}{\sqrt{(x-x')^2+a^2}}dx'. \quad (19)$$

Equation (19) defines an equation in N unknowns. To solve it, we must obtain N equations in N unknowns. Then, we need to choose N independent observation points x_m on the surface of the conductor (each one at the center of a conductor section). Therefore, if the boundary condition is $\phi = V$ = constant, we write the following:

$$\begin{cases} 4\pi\epsilon V = \alpha_1 \int_0^{\Delta x} \frac{f_1(x')}{\sqrt{(x_1-x')^2+a^2}}dx' + \ldots + \alpha_N \int_{(N-1)\Delta x}^{N\Delta x} \frac{f_N(x')}{\sqrt{(x_1-x')^2+a^2}}dx' \\ \ldots \\ 4\pi\epsilon V = \alpha_1 \int_0^{\Delta x} \frac{f_1(x')}{\sqrt{(x_N-x')^2+a^2}}dx' + \ldots + \alpha_N \int_{(N-1)\Delta x}^{N\Delta x} \frac{f_N(x')}{\sqrt{(x_N-x')^2+a^2}}dx' \end{cases} \quad (20)$$

which represents a system in the form (11). In addition, according to (9), the local weighted average is formed on each x_m by exploiting a local testing function w_m, which is centered at x_m so that the system (20), with $m = 1, \ldots, N$, can be rewritten, if V is not constant on x, in the following way:

$$\begin{cases} 4\pi\epsilon \int_0^{\Delta x} w_1(x)V(x)dx = \\ = \alpha_1 \int_0^{\Delta x} \int_{(m-1)\Delta x}^{m\Delta x} \frac{w_1(x)\cdot f_1(x')}{\sqrt{(x_1-x')^2+a^2}}dxdx' + \ldots + \alpha_N \int_0^{\Delta x} \int_{(m-1)\Delta x}^{m\Delta x} \frac{w_1(x)\cdot f_N(x')}{\sqrt{(x_1-x')^2+a^2}}dxdx' \\ \ldots \\ 4\pi\epsilon \int_{(N-1)\Delta x}^{N\Delta x} w_N(x)V(x)dx = \\ = \alpha_1 \int_0^{\Delta x} \frac{w_N(x)\cdot f_1(x')}{\sqrt{(x_N-x')^2+a^2}}dx' + \ldots + \alpha_N \int_{(N-1)\Delta x}^{N\Delta x} \frac{w_N(x)\cdot f_N(x')}{\sqrt{(x_N-x')^2+a^2}}dx' \end{cases} \quad (21)$$

in which the individual elements L_{mn} and the vector g_m are, respectively,

$$L_{mn} = \int_{(n-1)\Delta x}^{n\Delta x} \int_{(m-1)\Delta x}^{m\Delta x} \frac{w_m(x)\cdot f_n(x')}{\sqrt{(x_m-x')^2+a^2}}dxdx' \quad (22)$$

$$g_m = 4\pi\epsilon \int_{(m-1)\Delta x}^{m\Delta x} w_m(x)V(x)dx. \quad (23)$$

Remark 5. *Once q_e is obtained by* (17)*, C is easy to compute as follows:*

$$C = \frac{q_e}{V}. \quad (24)$$

In this work, the approach described above will be applied to exploit the following sets of basis and weighting functions:

1. In the first case, we present the study carried out that exploits the pulse function as the basis function, and as the testing function, the delta function will be considered as required in the point-matching approach;
2. In the second case, following the pseudo-Galerkin procedure, the pulse function will be exploited as the basis function as well as the testing function;
3. In the third case, the triangular function will be exploited as the basis function, while the delta function will be exploited as the testing function (point-matching approach);
4. In the fourth case, the triangular function will be considered the basis function while the pulse function will be exploited as the testing function;

5. Finally, in the fifth case, the triangular function will be considered the basis function as well as the testing function according to the pseudo-Galerkin approach.

4.1. First Case: Pulse Function as Basis Function and Delta Function as Testing Function (Point Matching)

Taking into account the pulse function defined in Section 4.5 (see Figure 2a) and the delta function defined in Section 3.4.2, the individual elements L_{mn}, according to (22), can be easily written as follows:

$$L_{mn} = \int_{(n-1)\Delta x}^{n\Delta x} \frac{1}{\sqrt{(x_m - x')^2 + a^2}} dx'. \tag{25}$$

After the numerical integration of (25), the following expression is determined [14]:

$$L_{mn} = \log \frac{(x_{n+1} - x_m) + \sqrt{(x_{n+1} - x_m)^2 - a^2}}{(x_{n-1} - x_m) + \sqrt{(x_n - x_m)^2 - a^2}}, \tag{26}$$

in which $x_{n+1} = x_n + \Delta x$ and $x_{n-1} = x_n - \Delta x$. Thus, according to (23), g_m can be written as

$$g_m = V_m = 4\pi\epsilon \int_{(m-1)\Delta x}^{m\Delta x} \Delta(x) V(x) dx = 4\pi\epsilon V. \tag{27}$$

Therefore, by means of (13), we obtain α_n so that $q_e(x')$ can be computed, exploiting (17). Finally, exploiting (24), the electrostatic capacity C is easily evaluated.

4.2. Second Case: Pulse Function as Basis Function as well as Testing Function

Exploiting the pulse function as the basis and testing functions as well, the individual elements L_{mn}, according to (22), can be easily computed as follows:

$$L_{mn} = \int_{(n-1)\Delta x}^{n\Delta x} \int_{(m-1)\Delta x}^{m\Delta x} \frac{f_n(x' - x) \cdot f_m(x' - x)}{\sqrt{(x - x')^2 + a^2}} dx dx'. \tag{28}$$

After performing the integration, we obtain [14]:

$$\begin{aligned} L_{mn} = {} & \sqrt{(x_{m+1} - x_{n+1})^2 - a^2} + (x_{m+1} - x_{n+1}) \cdot \log\left(\sqrt{x_{m+1} - x_{n+1} - a^2} - x_{m+1} + x_{n+1}\right) - \\ & - \sqrt{(x_m - x_{n+1})^2 - a^2} - (x_m - x_{n+1}) \cdot \log\left(\sqrt{(x_m - x_{n+1})^2 - a^2} - x_m + x_{n+1}\right) - \\ & - \sqrt{(x_{m+1} - x_{n-1})^2 - a^2} - (x_{m+1} - x_{n-1}) \cdot \log\left(\sqrt{(x_{m+1} - x_{n-1})^2 - a^2} - x_{m+1} + x_{n-1}\right) + \\ & + \sqrt{(x_{m-1} - x_{n-1})^2 - a^2} + (x_{m-1} + x_{n-1}) \cdot \log\left(\sqrt{(x_{m-1} - x_{n-1})^2 - a^2} - x_{m-1} + x_{n-1}\right). \end{aligned} \tag{29}$$

in which $x_{n+1} = x_n + \Delta x$, $x_{n-1} = x_n - \Delta x$, $x_{m+1} = x_m + \Delta x$ and $x_{m-1} = x_m - \Delta x$. Then, g_m, as required in (23), becomes

$$g_m = V_m = 4\pi\epsilon \int_{(m-1)\Delta x}^{m\Delta x} f_m(x' - x) V(x) dx = 4\pi\epsilon \Delta x V \tag{30}$$

in which $f_m(x' - x)$ is defined as in Section 4.5. Further, in this case, by exploiting (13), we easily obtain α_n, from which $q_e(x')$ is evaluable by means of (17). Finally, exploiting (24), C is easily obtainable.

4.3. Third Case: Triangular Function as Basis Function and Delta Function as Testing Function (Point Matching)

The triangular function spans over two adjacent sections and varies from 0 at the corner points of the section to 1 at the centre point of the section. Obviously, the function overlaps by one section so that the triangles give a piece-wise linear variation of the solution among sections (see Figure 2b).

Remark 6. *When the basis function is the triangular function, $q_e(x')$ expansion becomes*

$$q_e(x') = \sum_{n=1}^{NN} \alpha_n f_n(x') \tag{31}$$

with $f_n(x')$ defined as described in Section 3.4.3. Moreover, the charge distribution along the conductor, when the triangular basis functions are considered, has a span of $2\Delta x$ (twice as long as the Pulse case).

The cylindrical conductor, in this case, is divided into N sections whose width is e $\Delta x = L/N$ and $N+1$ points, $x_m = m\Delta x$, located at the end points of these sections (see Figure 2b). Obviously, the basis function at the first and last sections must be a half triangle because the charge density at the edges shoots up. For the triangular function as the basis function and the delta function as the testing function, the individual elements L_{mn}, according to (22), become:

$$L_{mn} = \int_{x_{n-1}}^{x_n} \left(\frac{x' - x_{n-1}}{\Delta x} \right) \frac{1}{\sqrt{(x_m - x')^2 + a^2}} dx' + \int_{x_n}^{x_{n+1}} \left(\frac{x_{n+1} - x'}{\Delta x} \right) \frac{1}{\sqrt{(x_m - x')^2 + a^2}} dx'. \tag{32}$$

After performing the integration of (32), L_{mn} assumes the following form [14]:

$$\begin{aligned} L_{mn} = {} & \frac{x_{n+1}}{\Delta x} \left[\log \frac{\left(\sqrt{a^2 + (x_{n+1} - x_m)^2} + x_{n+1} - x_m \right)}{\left(\sqrt{a^2 + (x_n - x_m)^2} + x_n - x_m \right)} \right] - \\ & - \frac{x_{n-1}}{\Delta x} \left[\log \frac{\left(\sqrt{a^2 + (x_n - x_m)^2} + x_n - x_m \right)}{\left(\sqrt{a^2 + (x_{n-1} - x_m)^2} + x_{n-1} - x_m \right)} \right] - \\ & - \frac{1}{\Delta x} \left[\frac{x_m \sqrt{a^2 + (x_{n+1} - x_m)^2} \log(2(\sqrt{a^2 + (x_{n+1} - x_m)^2} + x_{n+1} - x_m)) + a^2 + (x_{n+1} - x_m)^2}{\sqrt{a^2 + (x_{n+1} - x_m)^2}} \right] - \\ & - \frac{1}{\Delta x} \left[\frac{x_m \sqrt{a^2 + (x_{n-1} - x_m)^2} \log(2(\sqrt{a^2 + (x_{n-1} - x_m)^2} + x_{n-1} - x_m)) + a^2 + (x_{n-1} - x_m)^2}{\sqrt{a^2 + (x_{n-1} - x_m)^2}} \right] + \\ & + \frac{2}{\Delta x} \left[\frac{x_m \sqrt{a^2 + (x_n - x_m)^2} \log(2(\sqrt{a^2 + (x_n - x_m)^2} + x_n - x_m)) + a^2 + (x_n - x_m)^2}{\sqrt{a^2 + (x_n - x_m)^2}} \right] \end{aligned} \tag{33}$$

in which, as usual, $x_{n+1} = x_n + \Delta x$ and $x_{n-1} = x_n - \Delta x$. Finally, as required in (23), g_m becomes

$$g_m = V_m = 4\pi\epsilon V. \tag{34}$$

As explained in the previous sections, exploiting (13), we determine α_n so that $q_e(x')$, by means of (17), is computable. Therefore, by (24), C is computed.

4.4. Fourth Case: Triangular Function as Basis Function and Pulse Function as Testing Function

In this case, the expression provided by (22) is integrated considering the pulse testing function and replacing x_m with x on which the integration is performed. The expression for the individual L_{mn} can be written in the following compact form:

$$L_{mn} = \frac{x_{n+1}}{\Delta x}(A_{1mn} - A_{2mn}) - \frac{x_{n-1}}{\Delta x}(A_{2mn} - A_{3mn}) - \frac{1}{\Delta x} A_{4mn} - \frac{1}{\Delta x} A_{5mn} + \frac{2}{\Delta x} A_{6mn} \tag{35}$$

in which A_{imn}, for $i = 1, 2, \ldots, 6$, are computed as follows [14]:

$$\begin{aligned} A_{1mn} = {} & \sqrt{a^2 + (x_{m+1} - x_{n+1})^2} - \sqrt{a^2 + (x_{m-1} - x_{n+1})^2} + \\ & + \log \left[\frac{\left(\sqrt{a^2 + (x_{m+1} - x_{n+1})^2} - x_{m+1} + x_{n+1} \right)^{(x_{m+1} - x_{n+1})}}{\left(\sqrt{a^2 + (x_{m-1} - x_{n+1})^2} - x_{m-1} + x_{n+1} \right)^{(x_{m-1} - x_{n+1})}} \right] \end{aligned} \tag{36}$$

$$\begin{aligned} A_{2mn} = {} & \sqrt{a^2 + (x_{m+1} - x_{n})^2} - \sqrt{a^2 + (x_{m-1} - x_{n})^2} + \\ & + \log \left[\frac{\left(\sqrt{a^2 + (x_{m+1} - x_{n})^2} - x_{m+1} + x_{n} \right)^{(x_{m+1} - x_{n})}}{\left(\sqrt{a^2 + (x_{m-1} - x_{n})^2} - x_{m-1} + x_{n} \right)^{(x_{m-1} - x_{n})}} \right] \end{aligned} \tag{37}$$

$$\begin{aligned} A_{3mn} = {} & \sqrt{a^2 + (x_{m+1} - x_{n-1})^2} - \sqrt{a^2 + (x_{m-1} - x_{n-1})^2} + \\ & + \log \left[\frac{\left(\sqrt{a^2 + (x_{m+1} - x_{n-1})^2} - x_{m+1} + x_{n-1} \right)^{(x_{m+1} - x_{n-1})}}{\left(\sqrt{a^2 + (x_{m-1} - x_{n-1})^2} - x_{m-1} + x_{n-1} \right)^{(x_{m-1} - x_{n-1})}} \right] \end{aligned} \tag{38}$$

$$\begin{aligned} A_{4mn} = {} & \frac{1}{2} \log \left[\frac{\left(\sqrt{a^2 + (x_{m+1} - x_{n+1})^2 - x_{m+1} + x_{n+1}} \right)^{(x_{m+1}^2 - x_{n+1}^2)}}{\left(\sqrt{a^2 + (x_{m-1} - x_{n+1})^2 - x_{m-1} + x_{n+1}} \right)^{(x_{m-1}^2 - x_{n+1}^2)}} \right] + \\ & + \frac{1}{4} (3x_{m+1} + x_{n+1}) \left(\sqrt{a^2 + (x_{m+1} - x_{n+1})^2} \right) - \\ & - \frac{1}{4} (3x_{m-1} + x_{n+1}) \left(\sqrt{a^2 + (x_{m-1} - x_{n+1})^2} \right) + \\ & + \frac{a^2}{4} \log \left[\frac{\left(\sqrt{a^2 + (x_{m+1} - x_{n+1})^2 + x_{m+1} - x_{n+1}} \right)}{\left(\sqrt{a^2 + (x_{m-1} - x_{n+1})^2 + x_{m-1} - x_{n+1}} \right)} \right] \end{aligned} \tag{39}$$

$$\begin{aligned} A_{5mn} = {} & \frac{1}{2} \log \left[\frac{\left(\sqrt{a^2 + (x_{m+1} - x_{n-1})^2 - x_{m+1} + x_{n-1}} \right)^{(x_{m+1}^2 - x_{n-1}^2)}}{\left(\sqrt{a^2 + (x_{m-1} - x_{n-1})^2 - x_{m-1} + x_{n-1}} \right)^{(x_{m-1}^2 - x_{n-1}^2)}} \right] + \\ & + \frac{1}{4} (3x_{m+1} + x_{n-1}) \left(\sqrt{a^2 + (x_{m+1} - x_{n-1})^2} \right) - \\ & - \frac{1}{4} (3x_{m-1} + x_{n-1}) \left(\sqrt{a^2 + (x_{m-1} - x_{n-1})^2} \right) + \\ & + \frac{a^2}{4} \log \left[\frac{\left(\sqrt{a^2 + (x_{m+1} - x_{n-1})^2 + x_{m+1} - x_{n-1}} \right)}{\left(\sqrt{a^2 + (x_{m-1} - x_{n-1})^2 + x_{m-1} - x_{n-1}} \right)} \right] \end{aligned} \tag{40}$$

$$A_{6mn} = \frac{1}{2}\log\left[\frac{\left(\sqrt{a^2+(x_{m+1}-x_n)^2}-x_{m+1}+x_n\right)^{(x_{m+1}^2-x_n^2)}}{\left(\sqrt{a^2+(x_{m-1}-x_n)^2}-x_{m-1}+x_n\right)^{(x_{m-1}^2-x_n^2)}}\right]+ \tag{41}$$
$$+\frac{1}{4}(3x_{m+1}+x_n)\left(\sqrt{a^2+(x_{m+1}-x_n)^2}\right)-$$
$$-\frac{1}{4}(3x_{m-1}+x_n)\left(\sqrt{a^2+(x_{m-1}-x_n)^2}\right)+$$
$$+\frac{a^2}{4}\log\left[\frac{\left(\sqrt{a^2+(x_{m+1}-x_n)^2}+x_{m+1}-x_n\right)}{\left(\sqrt{a^2+(x_{m-1}-x_n)^2}+x_{m-1}-x_n\right)}\right]$$

where $x_{m+1} = x_m + \Delta x$ and $x_{m-1} = x_m - \Delta x$. Therefore, g_m is given by

$$g_m = V_m = 4\pi\epsilon\Delta x V \tag{42}$$

and, as usual, by (13), we obtain α_n so that $q_e(x')$, exploiting (17), is carried out. Finally, by (24), C is computed.

4.5. Fifth Case: Triangular Function as a Basis Function with the Galerkin Procedure

In the last case, the individual L_{mn} is determined as follows:

$$L_{mn} = \int_{x_{n-1}}^{x_n}\int_{x_{m-1}}^{x_m}\left(\frac{x'-x_{n-1}}{\Delta x}\right)\left(\frac{x-x_{m-1}}{\Delta x}\right)\frac{dx'dx}{\sqrt{a^2+(x-x')^2}}+ \tag{43}$$
$$+\int_{x_n}^{x_{n+1}}\int_{x_m}^{x_{m+1}}\left(\frac{x_{n+1}-x'}{\Delta x}\right)\left(\frac{x_{m+1}-x}{\Delta x}\right)\frac{dx'dx}{\sqrt{a^2+(x-x')^2}}.$$

The integration of (43) is quite complicated. However, after long mathematical steps, it is possible to express it in in the following compact form:

$$L_{mn} = A_{mn} - B_{mn} - C_{mn} + D_{mn}, \tag{44}$$

where

$$A_{mn} = \frac{x_{n+1}}{(\Delta x)^2}(A_{1mn}-A_{2mn}) - \frac{x_{n-1}}{(\Delta x)^2}(A_{2mn}-A_{3mn}) - \tag{45}$$
$$-\frac{1}{(\Delta x)^2}(A_{4mn}-A_{5mn}) + \frac{2}{(\Delta x)^2}A_{6mn},$$

$$B_{mn} = \frac{x_{n+1}}{(\Delta x)^2}(B_{1mn}-B_{2mn}) - \frac{x_{n-1}}{(\Delta x)^2}(B_{2mn}-B_{3mn}) - \tag{46}$$
$$-\frac{1}{(\Delta x)^2}(B_{4mn}-B_{5mn}) + \frac{2}{(\Delta x)^2}B_{6mn},$$

$$C_{mn} = \frac{x_{m-1}x_{n+1}}{(\Delta x)^2}(C_{1mn}-C_{2mn}) - \tag{47}$$
$$\frac{x_{m-1}x_{n-1}}{(\Delta x)^2}(C_{2mn}-C_{3mn})\frac{x_{m-1}}{(\Delta x)^2}(C_{4mn}+C_{5mn}-C_{6mn})$$

$$\begin{gathered} D_{mn} = \frac{x_{m-1}x_{n+1}}{(\Delta x)^2}(D_{1mn} - D_{2mn}) - \\ -\frac{x_{m-1}x_{n-1}}{(\Delta x)^2}(D_{2mn} - D_{3mn}) - \frac{x_{m+1}}{(\Delta x)^2(}D_{4mn} + D_{5mn} - D_{6mn}). \end{gathered} \tag{48}$$

Each term in (45)–(48) is specifiable as follows:

$$\begin{gathered} A_{1mn} = \frac{1}{4}(x_m + 3x_{n+1})\sqrt{a^2 + (x_m - x_{n+1})^2} - \\ -\frac{1}{4}a^2 \log\left[\frac{\sqrt{a^2 + (x_m - x_{n+1})^2} + x_m - x_{n+1}}{\sqrt{a^2 + (x_{m-1} + x_{n+1})^2} + x_{m-1} - x_{n+1}}\right] + \\ +\frac{1}{2}(x_m^2 - x_{n+1}^2)\log\left(\sqrt{a^2 + (x_m - x_{n+1})} - x_m + x_{n+1}\right) - \\ -\frac{1}{4}(x_{m-1} + 3x_{n+1})\sqrt{a^2 + (x_{m-1} - x_{n+1})^2} - \\ -\frac{1}{2}(x_{m-1}^2 - x_{n+1}^2)\log\left(\sqrt{a^2 + (x_{m-1} - x_{n+1})^2} - x_{m-1} + x_{n+1}\right) \end{gathered} \tag{49}$$

$$\begin{gathered} A_{2mn} = \frac{1}{4}(x_m + 3x_n)\sqrt{a^2 + (x_m - x_n)^2} - \\ -\frac{1}{4}a^2 \log\left[\frac{\sqrt{a^2 + (x_m - x_n)^2} + x_m - x_n}{\sqrt{a^2 + (x_{m-1} + x_n)^2} + x_{m-1} - x_n}\right] + \\ +\frac{1}{2}(x_m^2 - x_n^2)\log\left(\sqrt{a^2 + (x_m - x_n)} - x_m + x_n\right) - \\ -\frac{1}{4}(x_{m-1} + 3x_n)\sqrt{a^2 + (x_{m-1} - x_n)^2} - \\ -\frac{1}{2}(x_{m-1}^2 - x_n^2)\log\left(\sqrt{a^2 + (x_{m-1} - x_n)^2} - x_{m-1} + x_n\right) \end{gathered} \tag{50}$$

$$\begin{gathered} A_{3mn} = \frac{1}{4}(x_m + 3x_{n-1})\sqrt{a^2 + (x_m - x_{n-1})^2} - \\ -\frac{1}{4}a^2 \log\left[\frac{\sqrt{a^2 + (x_m - x_{n-1})^2} + x_m - x_{n+1}}{\sqrt{a^2 + (x_{m-1} + x_{n-1})^2} + x_{m-1} - x_{n-1}}\right] + \\ +\frac{1}{2}(x_m^2 - x_{n-1}^2)\log\left(\sqrt{a^2 + (x_m - x_{n-1})} - x_m + x_{n-1}\right) - \\ -\frac{1}{4}(x_{m-1} + 3x_{n-1})\sqrt{a^2 + (x_{m-1} - x_{n-1})^2} - \\ -\frac{1}{2}(x_{m-1}^2 - x_{n-1}^2)\log\left(\sqrt{a^2 + (x_{m-1} - x_{n-1})^2} - x_{m-1} + x_{n-1}\right) \end{gathered} \tag{51}$$

$$\begin{gathered} A_{4mn} = \frac{1}{3}\log\left[\frac{\left(\sqrt{a^2 + (x_m - x_{n+1})^2} - x_m + x_{n+1}\right)^{(x_m^3 - x_{n+1}^3)}}{\left(\sqrt{a^2 + (x_{m-1} - x_{n+1})^2} - x_{m-1} + x_{n+1}\right)^{(x_{m-1}^3 - x_{n+1}^3)}}\right] + \\ \frac{1}{9}\sqrt{a^2 + (x_m - x_{n+1})^2} \cdot (a^2 + 4x_m^2 + x_m x_{n+1} + 4x_{n+1}^2) - \\ -\frac{1}{9}\sqrt{a^2 + (x_{m-1} - x_{n+1})^2}(a^2 + 4x_{m-1}^2 + x_{m-1}x_{n+1} + 4x_{n+1}^2) \end{gathered} \tag{52}$$

$$\begin{aligned} A_{5mn} = \frac{1}{3}\log\left[\frac{(\sqrt{a^2+(x_m-x_{n-1})^2}-x_m+x_{n-1})^{(x_m^3-x_{n-1}^3)}}{(\sqrt{a^2+(x_{m-1}-x_{n-1})^2}-x_{m-1}+x_{n-1})^{(x_{m-1}^3-x_{n-1}^3)}}\right]+ \\ +\frac{1}{9}\sqrt{a^2+(x_m-x_{n-1})^2}\cdot(a^2+4x_m^2+x_mx_{n-1}+4x_{n-1}^2)- \\ -\frac{1}{9}\sqrt{a^2+(x_{m-1}-x_{n-1})^2}(a^2+4x_{m-1}^2+x_{m-1}x_{n-1}+4x_{n-1}^2) \end{aligned} \tag{53}$$

$$\begin{aligned} A_{6mn} = \frac{1}{3}\log\left[\frac{(\sqrt{a^2+(x_m-x_n)^2}-x_m+x_n)^{(x_m^3-x_n^3)}}{(\sqrt{a^2+(x_{m-1}-x_n)^2}-x_{m-1}+x_n)^{(x_{m-1}^3-x_n^3)}}\right]+ \\ \frac{1}{9}\sqrt{a^2+(x_m-x_n)^2}\cdot(a^2+4x_m^2+x_mx_n+4x_n^2)- \\ -\frac{1}{9}\sqrt{a^2+(x_{m-1}-x_n)^2}(a^2+4x_{m-1}^2+x_{m-1}x_n+4x_n^2) \end{aligned} \tag{54}$$

$$\begin{aligned} B_{1mn} = \frac{1}{4}(x_{m+1}+3x_{n+1})\sqrt{a^2+(x_{m+1}-x_{n+1})^2}- \\ -\frac{1}{4}a^2\log\left[\frac{\sqrt{a^2+(x_{m+1}-x_{n+1})^2}+x_{m+1}-x_{n+1}}{\sqrt{a^2+(x_m+x_{n+1})^2}+x_m-x_{n+1}}\right]+ \\ +\frac{1}{2}(x_{m+1}^2-x_{n+1}^2)\log\left(\sqrt{a^2+(x_{m+1}-x_{n+1})}-x_{m+1}+x_{n+1}\right)- \\ -\frac{1}{4}(x_m+3x_{n+1})\sqrt{a^2+(x_m-x_{n+1})^2}- \\ -\frac{1}{2}(x_m^2-x_{n+1}^2)\log\left(\sqrt{a^2+(x_m-x_{n+1})^2}-x_m+x_{n+1}\right) \end{aligned} \tag{55}$$

$$\begin{aligned} B_{2mn} = \frac{1}{4}(x_m+3x_n)\sqrt{a^2+(x_m-x_n)^2}- \\ -\frac{1}{4}a^2\log\left[\frac{\sqrt{a^2+(x_m-x_n)^2}+x_m-x_n}{\sqrt{a^2+(x_{m-1}+x_n)^2}+x_{m-1}-x_n}\right]+ \\ +\frac{1}{2}(x_m^2-x_n^2)\log\left(\sqrt{a^2+(x_m-x_n)}-x_m+x_n\right)- \\ -\frac{1}{4}(x_{m-1}+3x_n)\sqrt{a^2+(x_{m-1}-x_n)^2}- \\ -\frac{1}{2}(x_{m-1}^2-x_n^2)\log\left(\sqrt{a^2+(x_{m-1}-x_n)^2}-x_{m-1}+x_n\right) \end{aligned} \tag{56}$$

$$\begin{aligned} B_{3mn} = \frac{1}{4}(x_{m+1}+3x_{n-1})\sqrt{a^2+(x_{m+1}-x_{n-1})^2}- \\ -\frac{1}{4}a^2\cdot\log\left[\frac{\sqrt{a^2+(x_{m+1}-x_{n-1})^2}+x_{m+1}-x_{n+1}}{\sqrt{a^2+(x_m+x_{n-1})^2}+x_m-x_{n-1}}\right]+ \\ +\frac{1}{2}(x_{m+1}^2-x_{n-1}^2)\cdot\log\left(\sqrt{a^2+(x_{m+1}-x_{n-1})}-x_{m+1}+x_{n-1}\right)- \\ -\frac{1}{4}(x_m+3x_{n-1})\sqrt{a^2+(x_m-x_{n-1})^2}- \\ -\frac{1}{2}(x_m^2-x_{n-1}^2)\log\left(\sqrt{a^2+(x_m-x_{n-1})^2}-x_m+x_{n-1}\right) \end{aligned} \tag{57}$$

$$\begin{aligned} B_{4mn} = \frac{1}{3}\log\left[\frac{(\sqrt{a^2+(x_{m+1}-x_{n+1})^2}-x_{m+1}+x_{n+1})^{(x_{m+1}^3-x_{n+1}^3)}}{(\sqrt{a^2+(x_m-x_{n+1})^2}-x_m+x_{n+1})^{(x_m^3-x_{n+1}^3)}}\right]+ \\ +\frac{1}{9}\sqrt{a^2+(x_{m+1}-x_{n+1})^2}\cdot(a^2+4x_{m+1}^2+x_{m+1}x_{n+1}+4x_{n+1}^2)- \\ -\frac{1}{9}\sqrt{a^2+(x_m-x_{n+1})^2}(a^2+4x_m^2+x_mx_{n+1}+4x_{n+1}^2) \end{aligned} \tag{58}$$

$$\begin{aligned} B_{5mn} = \frac{1}{3}\log\left[\frac{(\sqrt{a^2+(x_{m+1}-x_{n-1})^2}-x_{m+1}+x_{n-1})^{(x_{+1}m^3-x_{n-1}^3)}}{(\sqrt{a^2+(x_m-x_{n-1})^2}-x_m+x_{n-1})^{(x_m^3-x_{n-1}^3)}}\right]+ \\ +\frac{1}{9}\sqrt{a^2+(x_{m+1}-x_{n-1})^2}\cdot(a^2+4x_{m+1}^2+x_{m+1}x_{n-1}+4x_{n-1}^2)- \\ -\frac{1}{9}\sqrt{a^2+(x_m-x_{n-1})^2}(a^2+4x_m^2+x_mx_{n-1}+4x_{n-1}^2) \end{aligned} \tag{59}$$

$$\begin{aligned} B_{6mn} = \frac{1}{3}\log\left[\frac{(\sqrt{a^2+(x_{m+1}-x_n)^2}-x_{m+1}+x_n)^{(x_{m+1}^3-x_n^3)}}{(\sqrt{a^2+(x_m-x_n)^2}-x_m+x_n)^{(x_m^3-x_n^3)}}\right]+ \\ +\frac{1}{9}\sqrt{a^2+(x_{m+1}-x_n)^2}\cdot(a^2+4x_{m+1}^2+x_{m+1}x_n+4x_n^2)- \\ -\frac{1}{9}\sqrt{a^2+(x_m-x_n)^2}(a^2+4x_m^2+x_mx_n+4x_n^2) \end{aligned} \tag{60}$$

$$\begin{aligned} C_{1mn} = \sqrt{a^2+(x_m-x_{n+1})^2}-\sqrt{a^2+(x_{m-1}-x_{n+1})^2}+ \\ +\log\left[\frac{(\sqrt{a^2+(x_m-x_{n+1})^2}-x_m+x_{n+1})^{(x_m-x_{n+1})}}{(\sqrt{a^2+(x_{m-1}-x_{n+1})^2}-x_{m-1}+x_{n+1})^{(x_{m-1}-x_{n+1})}}\right] \end{aligned} \tag{61}$$

$$\begin{aligned} C_{2mn} = \sqrt{a^2+(x_m-x_n)^2}-\sqrt{a^2+(x_{m-1}-x_n)^2}+ \\ +\log\left[\frac{(\sqrt{a^2+(x_m-x_n)^2}-x_m+x_n)^{(x_m-x_n)}}{(\sqrt{a^2+(x_{m-1}-x_n)^2}-x_{m-1}+x_n)^{(x_{m-1}-x_n)}}\right] \end{aligned} \tag{62}$$

$$\begin{aligned} C_{3mn} = \sqrt{a^2+(x_m-x_{n-1})^2}-\sqrt{a^2+(x_{m-1}-x_{n-1})^2}+ \\ +\log\left[\frac{(\sqrt{a^2+(x_m-x_{n-1})^2}-x_m+x_{n-1})^{(x_m-x_{n-1})}}{(\sqrt{a^2+(x_{m-1}-x_{n-1})^2}-x_{m-1}+x_{n-1})^{(x_{m-1}-x_{n-1})}}\right] \end{aligned} \tag{63}$$

$$\begin{aligned} C_{4mn} = \frac{1}{2}\log\left[\frac{(\sqrt{a^2+(x_m-x_{n+1})^2}-x_m+x_{n+1})^{(x_m^2-x_{n+1}^2)}}{(\sqrt{a^2+(x_{m-1}-x_{n+1})^2}-x_{m-1}+x_{n+1})^{(x_{m-1}^2-x_{n+1}^2)}}\right]+ \\ +\frac{1}{4}(3x_m+x_{n+1})\sqrt{a^2+(x_m-x_{n+1})^2}- \\ -\frac{1}{4}(3x_{m-1}+x_{n+1})\sqrt{a^2+(x_{m-1}-x_{n+1})^2}+ \\ +\frac{a^2}{4}\log\left[\frac{\sqrt{a^2(x_m-x_{n+1})^2}+x_m-x_{n+1}}{\sqrt{a^2+(x_{m-1}-x_{n+1})^2}+x_{m-1}-x_{n+1}}\right] \end{aligned} \tag{64}$$

$$\begin{aligned} C_{5mn} = \frac{1}{2}\log\left[\frac{(\sqrt{a^2+(x_m-x_{n-1})^2}-x_m+x_{n-1})^{(x_m^2-x_{n-1}^2)}}{(\sqrt{a^2+(x_{m-1}-x_{n-1})^2}-x_{m-1}+x_{n-1})^{(x_{m-1}^2-x_{n-1}^2)}}\right]+ \\ +\frac{1}{4}(3x_m+x_{n-1})\sqrt{a^2+(x_m-x_{n-1})^2}- \\ -\frac{1}{4}(3x_{m-1}+x_{n-1})\sqrt{a^2+(x_{m-1}-x_{n-1})^2}+ \\ +\frac{a^2}{4}\log\left[\frac{\sqrt{a^2(x_m-x_{n-1})^2}+x_m-x_{n-1}}{\sqrt{a^2+(x_{m-1}-x_{n-1})^2}+x_{m-1}-x_{n-1}}\right] \end{aligned} \tag{65}$$

$$\begin{aligned} C_{6mn} = \frac{1}{2}\log\left[\frac{(\sqrt{a^2+(x_m-x_n)^2}-x_m+x_{n-1})^{(x_m^2-x_n^2)}}{(\sqrt{a^2+(x_{m-1}-x_n)^2}-x_{m-1}+x_n)^{(x_{m-1}^2-x_n^2)}}\right]+ \\ +\frac{1}{4}(3x_m+x_n)\sqrt{a^2+(x_m-x_n)^2}- \\ -\frac{1}{4}(3x_{m-1}+x_n)\sqrt{a^2+(x_{m-1}-x_n)^2}+ \\ +\frac{a^2}{4}\log\left[\frac{\sqrt{a^2(x_m-x_n)^2}+x_m-x_n}{\sqrt{a^2+(x_{m-1}-x_n)^2}+x_{m-1}-x_n}\right]. \end{aligned} \tag{66}$$

$$\begin{aligned} D_{1mn} = \sqrt{a^2+(x_{m+1}-x_{n+1})^2}-\sqrt{a^2+(x_m-x_{n+1})^2}+ \\ +\log\left[\frac{(\sqrt{a^2+(x_{m+1}-x_{n+1})^2}-x_{m+1}+x_{n+1})^{(x_{m+1}-x_{n+1})}}{(\sqrt{a^2+(x_m-x_{n+1})^2}-x_m+x_{n+1})^{(x_m-x_{n+1})}}\right] \end{aligned} \tag{67}$$

$$\begin{aligned} D_{2mn} = \sqrt{a^2+(x_{m+1}-x_n)^2}-\sqrt{a^2+(x_m-x_n)^2}+ \\ +\log\left[\frac{(\sqrt{a^2+(x_{m+1}-x_n)^2}-x_{m+1}+x_n)^{(x_{m+1}-x_n)}}{(\sqrt{a^2+(x_m-x_n)^2}-x_m+x_n)^{(x_m-x_n)}}\right] \end{aligned} \tag{68}$$

$$\begin{aligned} D_{3mn} = \sqrt{a^2+(x_{m+1}-x_{n-1})^2}-\sqrt{a^2+(x_m-x_{n-1})^2}+ \\ +\log\left[\frac{(\sqrt{a^2+(x_{m+1}-x_{n-1})^2}-x_{m+1}+x_{n-1})^{(x_{m+1}-x_{n-1})}}{(\sqrt{a^2+(x_m-x_{n-1})^2}-x_m+x_{n-1})^{(x_m-x_{n-1})}}\right] \end{aligned} \tag{69}$$

$$\begin{aligned} D_{4mn} = \frac{1}{2}\log\left[\frac{(\sqrt{a^2+(x_{m+1}-x_{n+1})^2}-x_{m+1}+x_{n+1})^{(x_{m+1}^2-x_{n+1}^2)}}{(\sqrt{a^2+(x_m-x_{n+1})^2}-x_m+x_{n+1})^{(x_m^2-x_{n+1}^2)}}\right]+ \\ +\frac{1}{4}(3x_{m+1}+x_{n+1})\sqrt{a^2+(x_{m+1}-x_{n+1})^2}- \\ -\frac{1}{4}(3x_m+x_{n+1})\sqrt{a^2+(x_m-x_{n+1})^2}+ \\ +\frac{a^2}{4}\log\left[\frac{\sqrt{a^2(x_{m+1}-x_{n+1})^2}+x_{m+1}-x_{n+1}}{\sqrt{a^2+(x_m-x_{n+1})^2}+x_m-x_{n+1}}\right] \end{aligned} \tag{70}$$

$$\begin{aligned} D_{5mn} = \frac{1}{2}\log\left[\frac{(\sqrt{a^2+(x_{m+1}-x_{n-1})^2}-x_{m+1}+x_{n-1})^{(x_{m+1}^2-x_{n-1}^2)}}{(\sqrt{a^2+(x_m-x_{n-1})^2}-x_m+x_{n-1})^{(x_m^2-x_{n-1}^2)}}\right]+ \\ +\frac{1}{4}(3x_{m+1}+x_{n-1})\sqrt{a^2+(x_{m+1}-x_{n-1})^2}- \\ -\frac{1}{4}(3x_m+x_{n-1})\sqrt{a^2+(x_m-x_{n-1})^2}+ \\ +\frac{a^2}{4}\log\left[\frac{\sqrt{a^2(x_{m+1}-x_{n-1})^2}+x_{m+1}-x_{n-1}}{\sqrt{a^2+(x_m-x_{n-1})^2}+x_m-x_{n-1}}\right] \end{aligned} \tag{71}$$

$$\begin{aligned} D_{6mn} = \frac{1}{2}\log\left[\frac{(\sqrt{a^2+(x_{m+1}-x_n)^2}-x_{m+1}+x_{n-1})^{(x_{m+1}^2-x_n^2)}}{(\sqrt{a^2+(x_m-x_n)^2}-x_m+x_n)^{(x_m^2-x_n^2)}}\right]+ \\ +\frac{1}{4}(3x_{m+1}+x_n)\sqrt{a^2+(x_{m+1}-x_n)^2}- \\ -\frac{1}{4}(3x_m+x_n)\sqrt{a^2+(x_m-x_n)^2}+ \\ +\frac{a^2}{4}\log\left[\frac{\sqrt{a^2(x_{m+1}-x_n)^2}+x_{m+1}-x_n}{\sqrt{a^2+(x_m-x_n)^2}+x_m-x_n}\right]. \end{aligned} \tag{72}$$

As usual, $x_{n+1} = x_n + \Delta x$, $x_{n-1} = x_n - \Delta x$, $x_{m+1} = x_m + \Delta x$ and $x_{m-1} = x_m - \Delta x$. Finally, g_m is computed as

$$g_m = V_m = 4\pi\epsilon V\left\{\int_{x_m-1}^{x_m}\left(\frac{x-x_{m-1}}{\Delta x}\right)dx + \int_{x_m}^{x_{m+1}}\left(\frac{x_{m+1}-x}{\Delta x}\right)dx\right\} \tag{73}$$

so that, after integration, we obtain

$$g_m = V_m = \frac{4\pi\epsilon}{\Delta x}V\left\{x_m^2 - \frac{x_{m-1}^2}{2} - \frac{x_{m+1}^2}{2} - x_{m-1}(x_m - x_{m-1}) + x_{m+1}(x_{m+1} - x_m)\right\}. \tag{74}$$

Therefore, once g_m is determined, by means of (13), we carry out α_n so that $q_e(x')$, exploiting (17), is computable. Finally, by (24), C is computed.

Remark 7. *As previously noted, whatever the basis function and the chosen test functions, the modeling of the problem under study leads to the formulation of the linear algebraic problem (11) whose resolution is imperative for the calculation of the electrostatic charge density (and therefore electrostatic capacity). The usual resolution methods, direct (such as the Gauss elimination method) or iterative (such as the Gauss–Seidel method), although applicable in all the cases studied, do not take into account the fact that the matrix L is large with high dimensions. Therefore, we will focus on some methods of solving large linear systems based on Krylov subspaces.*

5. High-Dimension Linear Systems Resolution: A Krylov Subspaces Approach

5.1. Iterations in Krylov Subspaces

Definition 2. *Let us consider $A \in \mathbb{R}^{N\times N}$ and $\mathbf{b} \in \mathbb{R}^N$. For $r = 1, 2, \dots,$ we define the Krylov subspace with index r generated by A and $\mathbf{b}$, the set*

$$\mathcal{K}_r := span(\mathbf{b}, A\mathbf{b}, A^2\mathbf{b}, ..., A^{r-1}\mathbf{b}) \subset \mathbb{R}^N. \tag{75}$$

Projection methods in Krylov subspaces consist of projecting a problem of size N into a smaller Krylov subspace [21,23]. They then determine, for each r, an approximate solution $\mathbf{x}^{(r)}$ of the linear system $A\mathbf{x} = \mathbf{b}$, which belongs to the subspace $\mathcal{K}_r$. Krylov methods can be divided into four different classes that lead to different algorithms and that represent the different optimality criteria to choose the vector $\mathbf{x}^{(r)}$:

(1) Ritz–Galerkin approach: construct the vector $\mathbf{x}^{(k)}$ such that the corresponding residual is orthogonal to the Krylov sub-space with index k, i.e.,

$$\mathbf{b} - A\mathbf{x}^{(k)} \perp \mathcal{K}_r; \tag{76}$$

(2) The residual minimum norm approach: identify $\mathbf{x}^{(k)}$ for which the Euclidean residual norm $\|\mathbf{b} - A\mathbf{x}^{(k)}\|_2$ is minimum on $\mathcal{K}_r$;

(3) Petrov–Galerkin approach: find $\mathbf{x}^{(k)}$ such that $\mathbf{b} - A\mathbf{x}^{(k)}$ is orthogonal to some subspace of size k;

(4) The least norm error approach: determine

$$\mathbf{x}^{(k)} \in A^T\mathcal{K}_k(A^T, \mathbf{b}) \tag{77}$$

(Krylov subspace with index k generated by A^T and $\mathbf{b}$) so the Euclidean norm of error $\|\mathbf{x}^{(k)} - \mathbf{x}\|_2$ is minimal.

It has been observed that these optimality criteria lead to different algorithms; in particular, the Ritz–Galerkin approach can lead to the conjugate gradient and Lanczos methods. The second approach leads to the meetings of generalized minimal residual (GMRES) and minimal residual (MINRES). However, these two approaches have the disadvantage of being very slow, especially in the case of non-symmetric systems, and computationally expensive to determine the approximate solution. Moreover, this problem is solved by considering other subspaces for the orthogonal condition (as in the Petrov-Galerkin method). The third approach leads to the bi-conjugate gradient (Bi-CG) and quasi-minimal residual (QMR) methods while the fourth is not immediate, but for $A = A^T$, it leads to the symmetric LQ (SYMMLQ) method. To identify the approximation $\mathbf{x}^{(l)}$ of the solution, it is necessary to determine a basis for the Krylov subspace of index l, which can be extended to subspaces of increasing size.

Remark 8. *We observe that the basis for the Krylov subspace l-dimensional $\mathcal{K}_l$ is*

$$\{\mathbf{b}, A\mathbf{b}, A^2\mathbf{b}, ..., A^{l-1}\mathbf{b}\}. \tag{78}$$

This base turns out to be numerically unstable because as l increases, the vectors $A^l\mathbf{b}$ tend to dominate the eigen vector and therefore tend to be linearly independent. One could think of considering this base non-orthogonal and orthogonalize it later, but this could lead to strong ill-conditioning. The problem, therefore, arises in replacing this unstable base with an orthogonal or orthonormal base that characterizes the Krylov procedure used.

5.2. Conjugate Gradient (CG) as Krylov Method

It is possible to demonstrate that the CG method is characterized by an iteration in Krylov subspaces and therefore can be considered as a Krylov method [21,23]. In fact, the following theorem holds:

Theorem 1. *Let us suppose to apply the CG method to the linear system $A\mathbf{x} = \mathbf{b}$, with A symmetrical and positive defined. If the initial vector is $\mathbf{x}^{(0)} = 0$, then the method proceeds until the residue $\mathbf{r}^{(k)}$ is canceled and the following identities hold, for $l = 1, \ldots, k$, for the Krylov subspaces:*

$$\begin{aligned}\mathcal{K}_l := span(\mathbf{b}, A\mathbf{b}, \ldots, A^{l-1}\mathbf{b}) = span(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(l)}) = \\ = span(\mathbf{p}^{(0)}, \ldots, \mathbf{p}^{(l-1)}) = span(\mathbf{r}^{(0)}, \ldots, \mathbf{r}^{(l-1)}).\end{aligned} \tag{79}$$

Furthermore, the residuals are orthogonal for every l $(\mathbf{r}^{(t)})^T \mathbf{r}^{(j)} = 0,\ j = 0, 1, \ldots, l-1$ *and the research directions are A-conjugated,* $(\mathbf{p}^{(l)})^T A \mathbf{p}^{(l)} = 0,\ j = 0, 1, \ldots, l-1.$

As can be seen from Theorem 1, the CG method constitutes the base of $\mathcal{K}_l$ numerically unstable (78) with the base constituted by the A-orthogonal directions $\{\mathbf{p}^{(0)}, \ldots, \mathbf{p}^{(l-1)}\}$.

Let us introduce the following definition:

Definition 3. *Under the assumption that* A *is a symmetric and definite positive matrix, and therefore, the eigenvalues of* A *are all positive or equivalently* $\mathbf{x}^T A \mathbf{x} \geq 0$ *for each vector* $\mathbf{x} \in \mathbb{R}^N$ *not null, the function* $\| \cdot \|_A$ *defined by*

$$\| \cdot \|_A = \sqrt{\mathbf{x}^T A \mathbf{x}} \tag{80}$$

is the so-called $A - norm$*.*

It can, therefore, be shown that the generic iteration $\mathbf{x}^{(l)}$ of the CG method has an important optimality property. The following result holds:

Theorem 2. *The vector* $\mathbf{x}^{(l)}$ *minimizes in Krylov subspace* $\mathcal{K}_l$ *the A-norm of the error, i.e., it turns out that*

$$\mathbf{x}^{(l)} = \arg \min_{\mathbf{x} \in \mathcal{K}_l} \|\mathbf{x}^* - \mathbf{x}\|_A, \tag{81}$$

where $\mathbf{x}^*$ *is the exact solution of the linear system. Moreover, if*

$$\mathbf{e}^{(l)} = \mathbf{x}^* - \mathbf{x}^{(l)} \tag{82}$$

then we have

$$\|\mathbf{e}^{(l)}\|_A \leq \|\mathbf{e}^{(l-1)}\|_A, \quad \forall l. \tag{83}$$

5.3. The Arnoldi Method

The Arnoldi Iteration

The Arnoldi method is a Krylov subspace projection method for non-Hermitian matrices. This procedure, introduced as a method for reducing a dense matrix in the form of Hessenberg, allows to obtain a matrix factorization. Arnoldi iteration allows us to build a base $\{\mathbf{q}_1, \ldots, \mathbf{q}_l\}$ for the Krylov subspace $\mathcal{K}_l$ consisting of vectors orthonormal to each other, replacing the numerically unstable base (78). This can be done by applying the Gram–Schmidt orthogonalization method, obtaining a factorization of $A \in \mathbb{R}^{N \times N}$ in the form

$$AQ_l = Q_{l+1} \hat{H}_l, \tag{84}$$

where

$$Q_l \in \mathbb{R}^{N \times l}, \quad |_{l+1} \in \mathbb{R}^{N \times (l+1)} \tag{85}$$

and $\hat{H}_l \in \mathbb{R}^{(l+1) \times l}$. The procedure can be summarized in the following steps:

1. The algorithm starts by means of an arbitrary vector $\mathbf{q}_1$ whose norm is 1;
2. For $k = 2, 3, \ldots$ $\mathbf{q}_k = A\mathbf{q}_{k-1}$; in addition, for $j = 1, \ldots, k-1$

   $$h_{j,k-1} = \mathbf{q}_j^* \mathbf{q}_k \tag{86}$$

   and

   $$\mathbf{q}_k = \mathbf{q}_k - h_{j,k} - \mathbf{q}_j; \tag{87}$$

3. Then,
$$h_{k,k-1} = \|\mathbf{q}_k\| \tag{88}$$
and
$$\mathbf{q}_k = \frac{\mathbf{q}_k}{h_{k,k-1}}. \tag{89}$$
4. The algorithm breaks down when $\mathbf{q}_k$ is a null vector or, alternatively, each of its components is less than a fixed tolerance. This occurs when the minimal polynomial of A is of degree k.

Remark 9. *It is worth noting that the j-loop projects the components of* $\mathbf{q}_k$ *on the directions of* $\mathbf{q}_1, \ldots, \mathbf{q}_{k-1}$ *ensuring the orthogonality of all the generated vectors.*

5.4. GMRES Method

This method consists of iteratively constructing an orthonormal base, through the Arnoldi method, for the Krylov space $\mathcal{K}_l$, with $l = 1, \ldots, n$, approximating at each step the solution of the system $\mathbf{x}^*$ with the vector $\mathbf{x}^{(l)} \in \mathcal{K}_l$ to minimize the 2-norm of the residual $\mathbf{r}^{(l)} = \mathbf{b} - A\mathbf{x}^{(l)}$. Practically, if we set

$$\mathbf{x}^{(l)} = \arg \min_{\mathbf{x} \in \mathcal{K}_l} \|\mathbf{b} - A\mathbf{x}\|_2, \tag{90}$$

the solution is projected into the Krylov $\mathcal{K}_l$ subspace and is determined by solving a least squares problem. We observe that for the properties of the Arnoldi iteration, the method, in many cases, makes it possible to obtain a sufficiently accurate approximation of the system solution in a number of steps much lower than N, despite being a method that ends in at most N steps. Let us see how we proceed in an operational way, remembering that the matrix in the form of Hessenberg

$$H_l = Q_l^T A Q_l \tag{91}$$

is the representation in the orthonormal base $\mathbf{q}_1, \ldots, \mathbf{q}_l$ of the orthogonal projection A on the $\mathcal{K}_l$ subspace. In fact, taking a generic vector $\mathbf{y} \in \mathbb{R}^l$, the product $Q_l\mathbf{y}$ expresses a generic vector of $\mathcal{K}_l$ as a linear combination of the elements of the orthonormal base considered. Therefore, one has

$$\|\mathbf{b} - A\mathbf{x}\| = \|AQ_l\mathbf{y} - \mathbf{b}\|. \tag{92}$$

Taking into account that $AQ_k = Q_{k+1}\hat{H}_k$, we can write

$$\|AQ_l\mathbf{y} - \mathbf{b}\| = \|Q_{l+1}\hat{H}_l\mathbf{y} - \mathbf{b}\|, \tag{93}$$

from which, being $Q_{l+1}^T Q_{l+1} = I_{l+1}$ and observing that $Q_{l+1}^T\mathbf{b} = \|\mathbf{b}\|\mathbf{e}_1$, we obtain

$$\|\mathbf{b} - A\mathbf{x}\| = \left\|\hat{H}_l\mathbf{y} - \|\mathbf{b}\|\mathbf{e}_1\right\|. \tag{94}$$

Thus, to solve the least squares problem $\mathbf{x}^{(l)} = \arg\min_{\mathbf{x}\in\mathcal{K}_l} \|\mathbf{b} - A\mathbf{x}\|_2$, it is possible to calculate the solution $\mathbf{y}^{(l)}$ of the minimization problem

$$\min_{\mathbf{y}\in\mathbb{R}^l} \left\|\hat{H}_l\mathbf{y} - \|\mathbf{b}\|\mathbf{e}_1\right\|, \tag{95}$$

and calculate $\mathbf{x}^{(l)} = Q_l\mathbf{y}^{(l)}$.

Remark 10. *The cost of the algorithm is that due to the Arnoldi method. Furthermore, it should be noted that to calculate a new solution, it is necessary to memorize all the vectors of the base. Hence, the GMRES procedure becomes too expensive and inefficient if the size of the Krylov subspace grows too much. A possible solution to the problem is represented by restarting. It consists of stopping the algorithm when a certain size m of the Krylov subspace is reached; we then calculate* $\mathbf{x}_m$ *and start again with the algorithm with initial approximation* $\mathbf{x}_m$ *above. Furthermore, it is worth noting that restarting while increasing the number of iterations considerably reduces the CPU-time.*

5.5. The Biconjugate Gradient Method (BiCG)

CG is not suitable for non-symmetric systems because the residual vectors cannot be made orthogonal with short recurrences [21,23]. While the GMRES method preserves the orthogonality of the residues by exploiting extensive occurrences with a large storage demand, the BiCG procedure replaces the orthogonal sequence of the residues with two further mutually-orthogonal sequences without the need for minimization [23]. Together with the orthogonal residuals generated by the CG procedure, the BiCG method constructs a second set of residuals in which the initial residual can be chosen without particular constraints. Thus, the BiCG procedure requires two matrix-by-vector products by A and by A^T. The residuals form biorthogonal bases. Moreover, the corresponding direction form biconjugate bases. Regarding convergence, the BiCG method provides the same results as the CG method when positive defined systems are considered but with a considerable increase in the computational cost for each iteration. If the matrices are not symmetrical in cases where there is a significant reduction in the residual norm, the BiCG method is comparable, in terms of the number of iterations, with the GMRES method. Furthermore, the convergence can be irregular and the procedure could face breakdown phenomena which, among other things, can be avoided by means of appropriate restart operations.

Remark 11. *However, it is not always guaranteed that restarting the BiCG procedure improves performance; in fact, while GMRES is guaranteed to converge in n steps, it is not ensured that the method is not subject to stagnation using this technique.*

Remark 12. *To compare the performances obtained with the GMRES and BiCG is difficult. While GMRES minimizes a residue by increasing the work to keep all the orthogonal residues with a consequent increase in the demand for space in memory, the BiCG method does not minimize any residue, but its precision is comparable to that obtained with the GMRES method at the cost of twice the quantity of matrix products per vector for each iteration. However, no special memory efforts are required to generate the base vectors because this procedure is based on the Lanczos biorthogonalization cheaper from the point of view of memory demand than the orthogonalization of Arnoldi [21,23].*

To increase the effectiveness of the BiCG method, several variants have been proposed, including the CGS and the BiCGStab methods.

5.6. The CGS Method

This approach, proposed by Sonneveld [23], replaces the multiplication with A^T in BiCG by a second one with A and writes the residual by a square polynomial [21,23]. At each iteration, the dimension of the Krylov subspace is increased by two units. Concerning the convergence, it is almost twice as fast as BiCG, but even more erratic. If the computation wit hA^T is impractical, CGS may be attractive.

Remark 13. *Increase in rounding errors and possible overflow could take place. This occurrence leads to a preference for the BiCStab method.*

5.7. The BicGStab Method

Developed by Van der Vorst [21,23], this method avoids the often irregular convergence of CGS by means of combining the CGS sequence with the steepest descent update, performing some local optimization and smoothing. The procedure is based on writing the residual by a product between two polynomials and formally constituted by the following steps:

(1) Firstly, we compute $\mathbf{r}_0 := \mathbf{b} - A\mathbf{x_0}$;

(2) Then, we set $\mathbf{p}_0 := \mathbf{r}_0$;

(3) For $j = 0, 1, \ldots$ until convergence we compute the following quantities

$$\alpha_j = \frac{(\mathbf{r}_j, \mathbf{r}_0)}{(A\mathbf{p}_j, \mathbf{r}_0)}; \quad \mathbf{s}_j = \mathbf{r}_j - \frac{\alpha_j}{A\mathbf{p}_j}; \quad \omega_j = \frac{(A\mathbf{s}_j, \mathbf{s}_j)}{(A\mathbf{s}_j, A\mathbf{s}_j)}; \tag{96}$$
$$\mathbf{x}_{j+1} = \mathbf{x}_j + \alpha_j \mathbf{p}_j + \omega_j \mathbf{s}_j; \quad \mathbf{r}_{j+1} = \mathbf{s}_j - \omega_j A\mathbf{s}_j;$$
$$\beta_j = \frac{(\mathbf{r}_{j+1}, \mathbf{r}_0^*)}{(\mathbf{r}_j, \mathbf{r}_0)} \frac{\alpha_j}{\omega_j}; \quad \mathbf{p}_{j+1} = \mathbf{r}_{j+1} + \beta_j(\mathbf{p}_j - \omega_j A\mathbf{p}_j).$$

5.8. Comparison between GMRES and BicGStab

It is not generally possible to say which of the two methods is more efficient. However, it is possible to formulate the following hypothesis: as the number of elements in the matrix decreases, the BicGStab algorithm should be faster. In fact, although there are two matrix-carrier products in BiCGStab, these should have a limited cost in relation to the products performed many times and to the high storage cost of GMRES. The time taken by the BicGStab algorithm is almost always shorter. However, the GMRES algorithm performs fewer iterations (despite taking longer) than the BicGStab algorithm as the number of non-zero elements increases. Finally, to influence the computational cost of the GMRES algorithm, in addition to the products performed a large number of times, it could be the high number of elements that must be stored.

Remark 14. *It is worth noting that the matrices are often ill-conditioned. A pre-condition must therefore exist: the results depend very much on the accuracy of the pre-condition.*

6. Numerical Results

L_{mn} have been computed by (26), (29), (33), (35), and (44) exploiting, as described above, pulse–delta, pulse–pulse, triangular–delta, triangular–pulse, and triangular–triangular basis-testing functions, respectively. Once the matrix L was formed, the corresponding linear system was solved by GMRES, CGS, and BicGStab procedures, comparing the results with those determined by Gauss elimination and Gauss–Seidel methods to obtain the unknown q_e on each section of the conductor. Then, substituting the value of charge density from Equation (13) and the basis function, as described in Sections 4.5 or 3.4.3, into Equations (17) and (24), q_e and C are obtained. The following remark justifies the use of the Gauss–Seidel iterative procedure.

Remark 15. *We observe that the Gauss–Seidel iterative method is applicable in our cases because all the matrices L_{mn} are all with dominant diagonal by rows. In other words:*

$$|L_{ii}| > \sum_{j=1, j\neq i}^{n} |L_{ij}|. \tag{97}$$

This condition implies the non-singularity of the matrices, from which follows the uniqueness of the solution [23].

6.1. Results Concerning the Values of Capacitance C & Instability Phenomena

The numerical values obtained for C for unit length (using different basis and testing functions) exploiting the above-mentioned procedures are displayed in Figures 3a–c and 4a,b, respectively. The trends highlight that C, for each procedure applied, assumes regular behavior when $\frac{\overline{L}}{d} < 20$ while a slight instability occurs when $20 < \frac{\overline{L}}{d} < 40$. This is due to the fact that the distribution of q_e on the conductor generates an electric dipole vector $\mathbf{p}$ which forms an angle θ with the electric field vector $\mathbf{E}$. It is proven that when $\theta \neq 0$ instabilities occur until θ, due to the forces generated by $\mathbf{E}$, becomes null. This condition of instability occurs when $20 < \frac{\overline{L}}{d} < 40$ [2]. When $\frac{\overline{L}}{d} < 20$ and $\frac{\overline{L}}{d} > 40$, $\mathbf{p}$ and $\mathbf{E}$ are almost parallel for which the instability is extremely reduced. However, the instability displayed in each simulation has a limited amplitude and it is such as not to cause concern.

However, it is worth noting that whatever the numerical procedure used to solve the dense linear system and whatever the basis-testing functions pair, for high values of the $\overline{L}/d$ ratio, C tends to assume the same value. This is due to the fact that for increasing $\overline{L}/d$, the mesh becomes denser, and therefore, asymptotically, the numerical procedures applied provide the same results.

Regarding the convergence of the numerical data related to the electrostatic capacitance C are displayed in Figures 5a,b and 6a–c. In particular, the trends of C versus N, when $\frac{\overline{L}}{d}$ ratio is equal to 400 for each basis testing functions pairs, are highlighted for each linear system resolution procedure used.

It should be noted that the Gauss elimination method provides values of C very close to each other regardless of the basis-testing functions pair used, while the trend differs considerably with the increase of the subsections while highlighting an obvious saturation behavior as N increases. This is due to the fact that beyond 30 sections, the curve assumes a straight horizontal trend so that the increase in the number of subsections affects the value of C. Similar behavior is exhibited by the performance of the Gauss–Seidel method, even if saturation is obtained with fewer subsections. Furthermore, by increasing the number of subsections, the mesh thickens considerably so that the phenomenon of asymptoticity is felt according to which, whatever the numerical procedure adopted, the values obtained tend to coincide.

Remark 16. *As known, the capacitance of the thin straight wire (finite length) is given by [41]:*

$$C = \frac{2\pi\epsilon_0 \overline{L}}{\Lambda}\left\{1 + \frac{1}{\Lambda}(1-\ln 2) + \frac{1}{\Lambda^2}\left[1 + (1-\ln 2)^2 - \frac{\pi^2}{12}\right] + o\left(\frac{1}{\Lambda^3}\right)\right\} \tag{98}$$

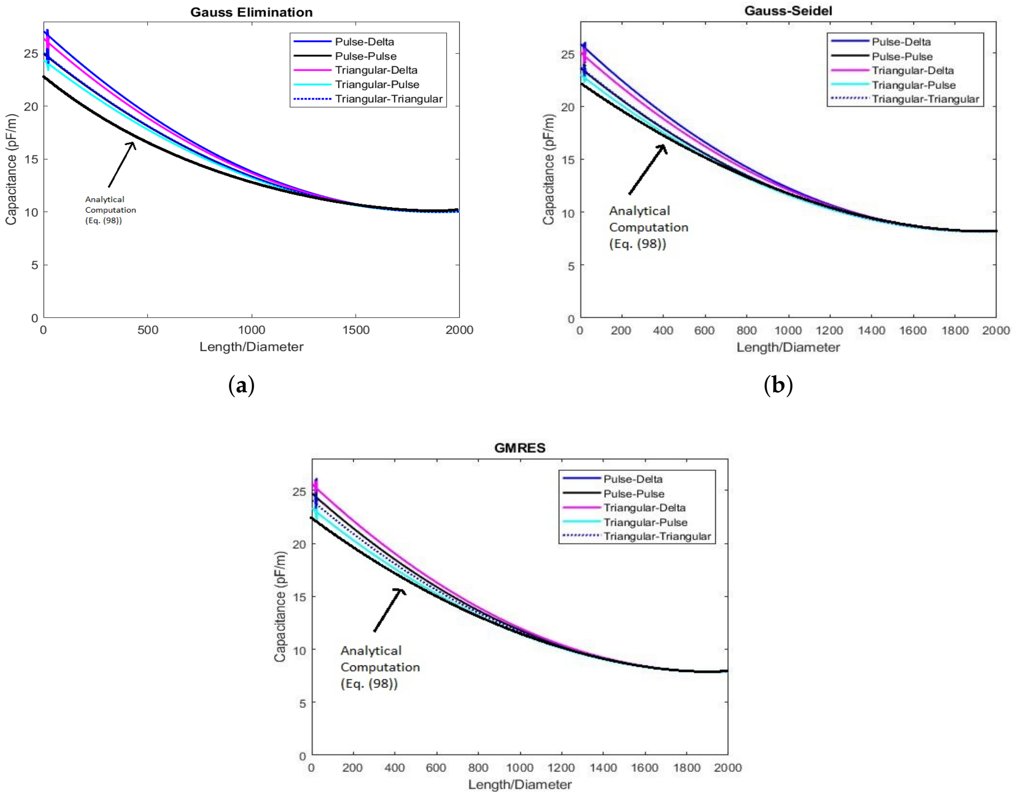
or, numerically,

$$C = 4\pi\epsilon_0 \frac{2\overline{L}}{\Lambda}\left\{1 + \frac{0.6137056}{\Lambda} + \frac{1.086766}{\Lambda^2} + o\left(\frac{1}{\Lambda^3}\right)\right\} \tag{99}$$

where C is the capacitance in (F), $\epsilon_0 = \frac{1}{36\pi}10^{-9}$ is the permittivity of the free space(F/m) and

$$\Lambda = \ln\left(\frac{2\overline{L}}{d}\right). \tag{100}$$

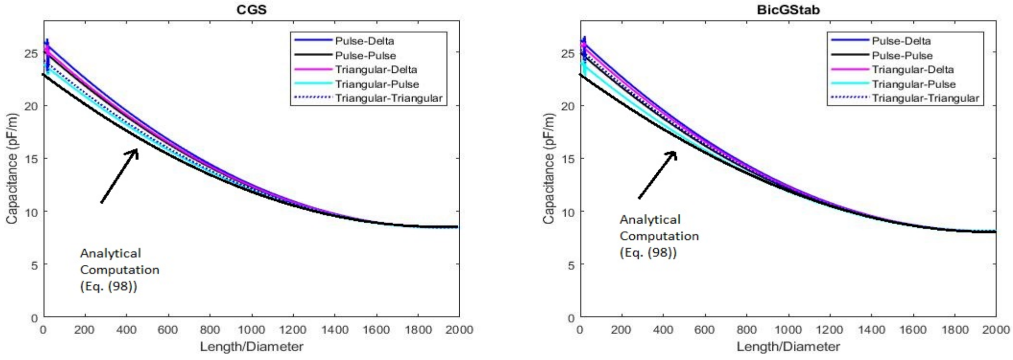
Generally, the values obtained by (98) is an underestimated value of the real value (because there are no infinitesimals of order higher than Λ^2); while the values obtained with the procedures based on the Krylov subspaces appear to be slightly overestimated. However, as can be seen from the Figures 3a–c and 4a,b, as the $\overline{L}/d$ ratio increases all the values obtained converge. However, even if the convergence is ensured for high values of $\overline{L}/d$ (and therefore for large system dimensions) triangular–triangular function pairs converge with the analytical values of capacitance obtained by Equation (98) at a smaller value of $\overline{L}/d$ making this pair of functions more attractive than the other pairs. It is worth noting that, even if this pair has a higher computational effort than the other pairs of functions, it provides values that are closer to real values, making this pair of functions more attractive than the other pairs. It is worth noting that, even if this pair has a higher computational cost than the other pairs of functions, it provides values of capacitance that are closer to real values starting from a smaller *value of $\overline{L}/d$. Therefore, the higher computational time required for performance with this pair of functions is largely justified by the fact that the obtained capacitance values are more reliable.*



(**a**) (**b**)

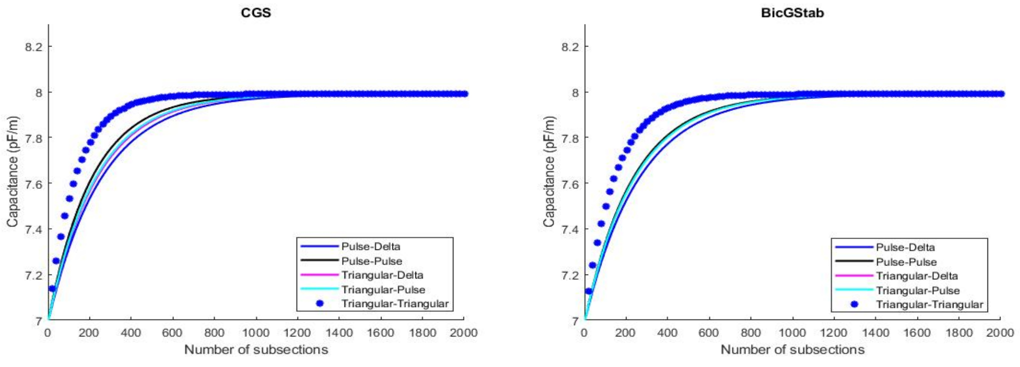
(**c**)

Figure 3. C vs. $\overline{L}/d$ ratio for each basis-testing functions with the (**a**) Gauss elimination procedure, (**b**) Gauss–Seidel method and (**c**) GMRS algorithm.



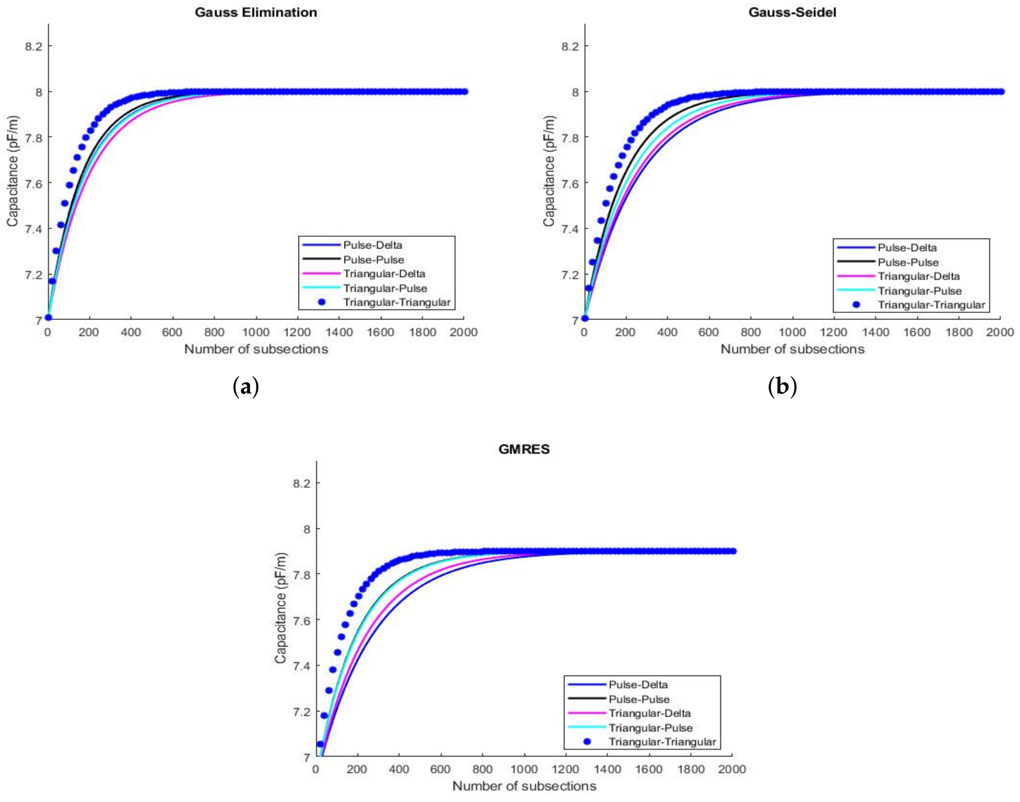
(**a**) (**b**)

Figure 4. C vs. $\overline{L}/d$ ratio for each basis-testing functions with the (**a**) CGS algorithm and (**b**) BicGStab procedure.



(**a**) (**b**)

Figure 5. C vs. N when $\overline{L}/d$ ratio is equal to 400 for each basis and testing functions: (**a**) CGS and (**b**) BicGStab algorithms.



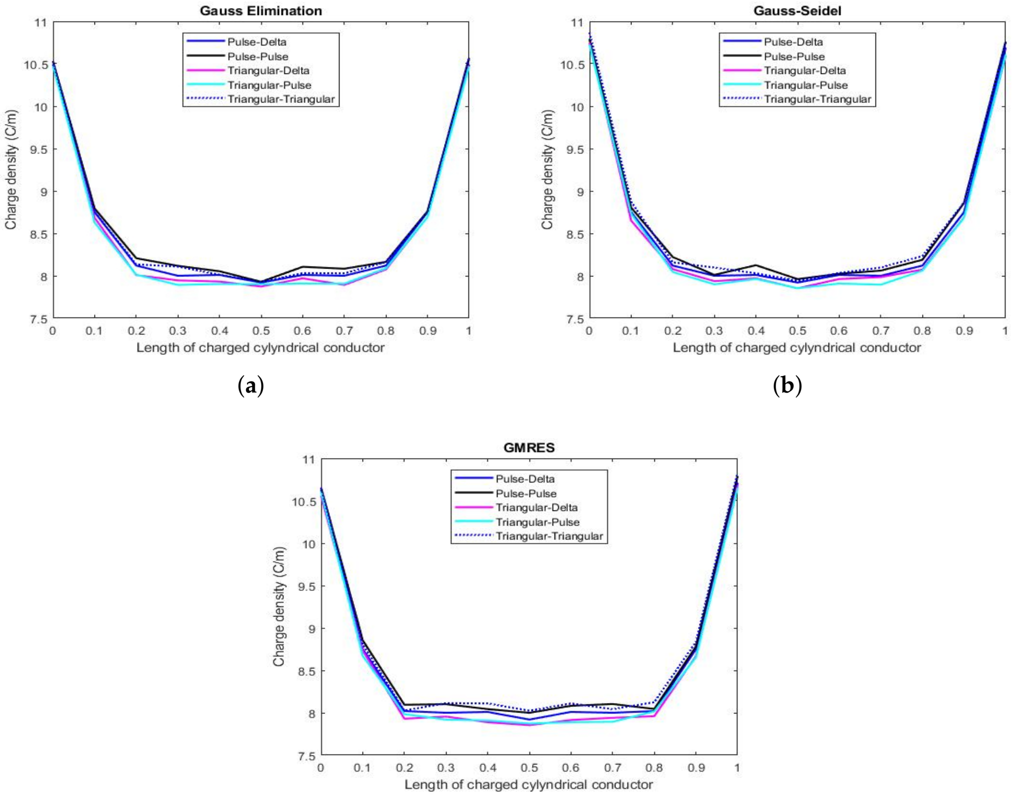
(**a**) (**b**)

(**c**)

Figure 6. C vs. N when $\overline{L}/d$ ratio is equal to 400 for each basis and testing functions: (**a**) Gauss elimination; (**b**) Gauss–Seidel procedures; (**c**) GMRES algorithm.

6.2. q_e & Performance of the Procedures

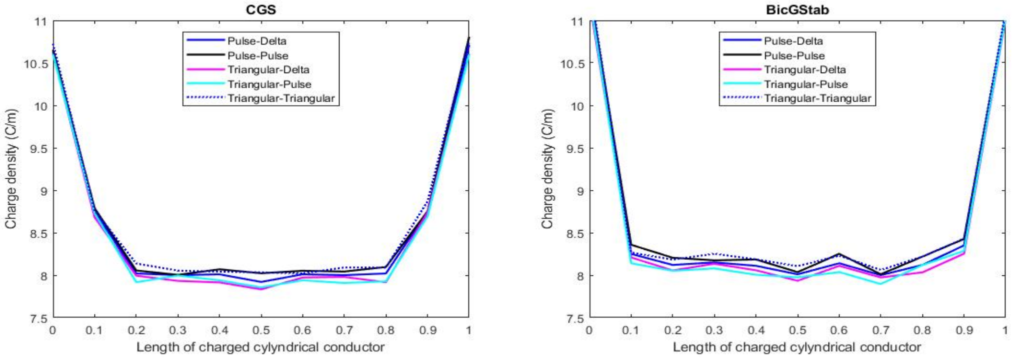
Figures 7a–c and 8a,b show the trend of the electrostatic charge density on the conductor when the $\overline{L}/d$ ratio is equal to 400 (when $N = 10$). There are no strong differences regarding the trend of the charge density for unit length when different procedures for solving algebraic system are considered. However, a slight instability is felt in the central area of the conductor, which is most evident when the BicGStab algorithm is used. In contrast, the same methodology provides linear results at the ends of the conductor.



(**a**) (**b**)

(**c**)

Figure 7. Charge density vs. $\overline{L}$ for $\frac{L}{d} = 400$ when (**a**) Gauss elimination procedure; (**b**) Gauss–Seidel methods and (**c**) GMRES algorithm are exploited.



(**a**) (**b**)

Figure 8. Charge density vs. $\overline{L}$ for $\frac{L}{d} = 400$ when (**a**) CGS and (**b**) BicGStab algorithms are exploited.

Remark 17. *The procedure based on Krylov subspaces, however, even if they exhibit the same behavior as the Gauss elimination procedure and Gauss–Seidel method with an increase of N (for the phenomenon of asymptoticity), show a more regular behavior before saturation, regardless of the pairs of basis and testing functions used. Furthermore, the distance between the curves for small N are not evident, especially when the BicGStab algorithm is applied, which provides very similar values of C. This is due to the fact that BicGStab, although requiring a higher number of iterations than the other procedures, converges more quickly, as evidenced by the CPU-time obtained. From this analysis, there is no doubt that a key role is played by the number of subsections N that determine the performance of individual procedures as well as trends in the values of C. Hence, the need to formulate appropriate selection criteria for N.*

6.3. Selection Criteria for N

Let us introduce the following two definitions.

Definition 4 (norm matrix)**.** *As known, a matrix norm is a norm on the vector space $K^{m\times n}$ of all matrices of size $m \times n$ with entries in a field K. In particular, it is a function [23]*

$$\| \cdot \| : K^{m\times n} \to \mathbb{R} \tag{101}$$

such that it satisfies the following properties:

1. $\|A\| \geq 0, \forall A \in K^{m\times n}$*;*
2. $\|A\| = 0$ *if only if* $A = 0^{m\times n}$*;*
3. $\|\alpha A\| = |\alpha|\ \|A\|, \alpha \in K, \forall A \in K^{m\times n}$*;*
4. $\|A + B\| \leq \|A\| + \|B\|, \forall A, B \in K^{m\times n}$*.*

Definition 5 (matrix norm induced by vector norms)**.** *Let us consider a vector norm $\|\dot{\|}$ on K^m. Then, any $A \in K^{m\times n}$ induces a linear operator from K^m to K^n with respect to the standard bases. So, the corresponding induced norm on $K^{m\times n}$ of $A \in K^{m\times n}$ is defined as follows [23]:*

$$\|A\| = \sup\{\|Ax\|\ :\ x \in K^n,\ \ \|x\| = 1\} = \sup\left\{\frac{\|Ax\|}{\|x\|}\ :\ x \in K^n,\ x \neq 0\right\}. \tag{102}$$

For $1 \leq p \leq +\infty$, the induced norm is as follows [23]:

$$\|A\|_p = \sup_{x\neq 0} \frac{\|Ax\|_p}{\|x\|_p}. \tag{103}$$

In particular, if $p = 2$, it is proved that [23]:

$$\|A\|_2 = \sqrt{\lambda_{\max}(A^*A)} = \sigma_{\max}(A), \tag{104}$$

in which $\sigma_{\max}(A)$ represents the largest singular value of matrix A (the square root of the largest eigenvalue of the matrix A^*A in which A^* is the conjugate transpose of A).

Once the matrix norm has been introduced in Definition 4 , two criteria are exploited to choose N. The first criterion acts on the conditioning number of L, K. In particular, for each value of the $\overline{L}/d$ ratio and for each pair of basis and testing functions,

$$K(N) = \|L(N)\| \cdot \|L^{-1}(N)\| \geq 1, \tag{105}$$

depending on N, is computed. Thus, setting

$$\begin{cases} \left\{ \max_N \left[(K(N))_{\frac{\overline{L}}{d}=15}, \ (K(N))_{\frac{\overline{L}}{d}=20}, \ (K(N))_{\frac{\overline{L}}{d}=25}, \ (K(N))_{\frac{\overline{L}}{d}=30}, \ (K(N))_{\frac{\overline{L}}{d}=60} \right]_j \right\} \\ j \in \{\text{'Pulse–Delta'}, \text{'Pulse–Pulse'}, \text{'Triangular–Delta'}, \\ \text{'Triangular–Pulse'}, \text{'Triangular–Triangular'}\} \end{cases} \tag{106}$$

in stability conditions gives us the first choice of N. For example, Figure 9 depicts the trend of $K(N)$ depending on N for each basis and testing functions when $\overline{L}/d = 60$. In this case, the value 41 represents the lower value for N obtained by this criterion when $\overline{L}/d$ ratio varies, highlighting that, sometimes, instability arises even if N is a small value. However, strong instability is more evident when N increases.

Remark 18. *We observe that increasing N also increases the number of unknowns and therefore also increases the degree of freedom of the system. This means that $\lambda_{\max}$ in (104) becomes more and more extreme. This leads to a trend of $K(N)$ which, starting from 1 when $N = 1$, grows with evident instability when $N \geq 42$.*

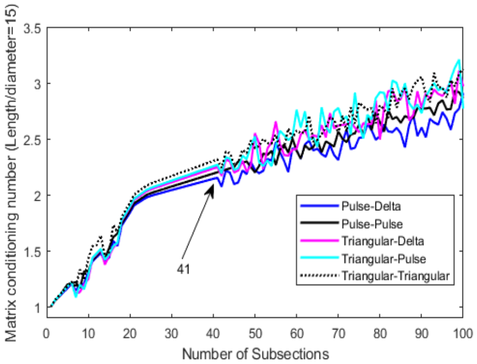


Figure 9. $K(L)$ vs. N with different B/T functions.

The second criterion acts on C. Specifically, for each procedure exploited (Gauss elimination, Gauss–Seidel, GMRES, CGS and BicGStab) and for each $\overline{L}/d$ ratio, C is evaluated when N increases. Then,

$$\begin{cases} \min\left\{ \max_N \left[\left\{ (C(N))_{\frac{L}{d}=15}, \ (C(N))_{\frac{L}{d}=20}, \ (C(N))_{\frac{L}{d}=25}, \ (C(N))_{\frac{L}{d}=30}, \ (C(N))_{\frac{L}{d}=60} \right\}_k \right]_j \right\} \\ k \in \{\text{'Gauss Elimination'}, \text{'Gauss–Seidel'}, \text{'GMRES'}, \text{'CGS'}, \text{'BicGStab'}\} \\ j \in \{\text{'Pulse–Delta'}, \text{'Pulse–Pulse'}, \text{'Triangular–Delta'}, \\ \text{'Triangular–Pulse'}, \text{'Triangular–Triangular'}\} \end{cases} \tag{107}$$

in the stability condition, gives us the second choice of N. Finally, the quantity obtained taking the minimum value between (106) and (107) gives us the definitive number of subsection N. In our study, we obtained $N \cong 41$.

Remark 19. *We note that the choice of N, understood as a minimum between the values obtained by applying the two criteria described above, is also dictated by the fact that for high values of N, the values of C do not undergo appreciable variations.*

6.4. Convergence Speed & CPU-Time

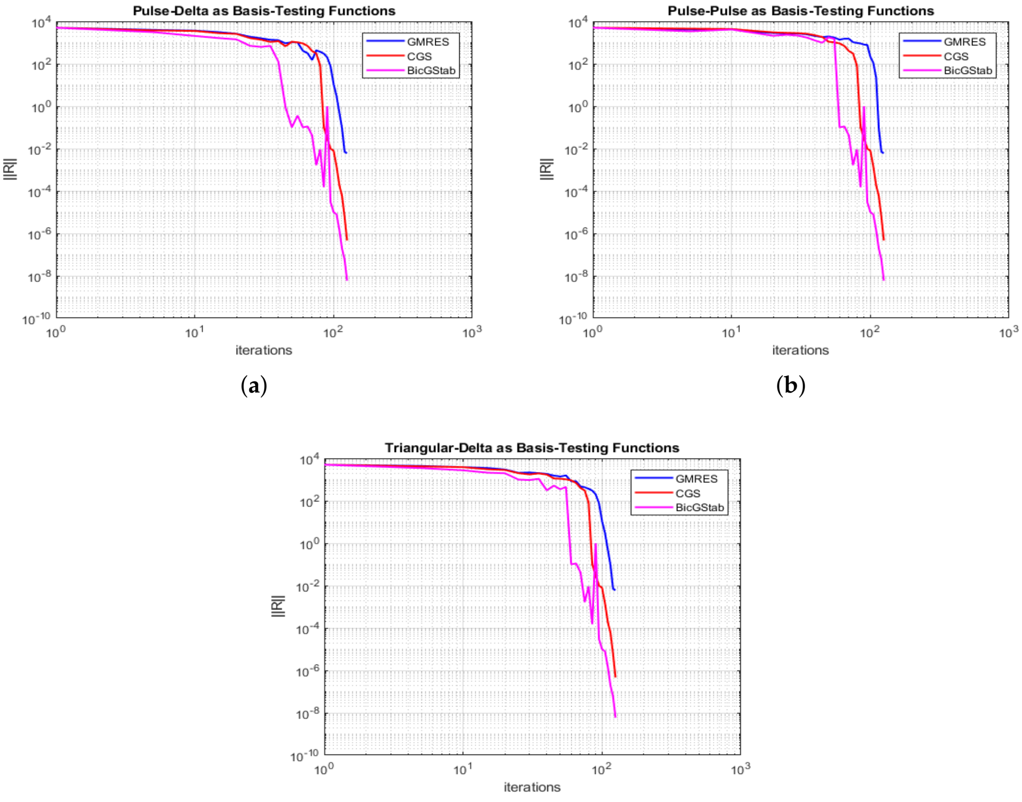
The performance of each procedure was also assessed by the convergence speed computation expressed in terms of $\|R\|$ as the number of iterations increased. In particular, for each $\frac{\overline{L}}{d}$ considered, exploiting an Intel Core 2 CPU 1.45 GHz machine and MatLab® R2017a, the trends of $\|R\|$ were plotted as a function of the number of iterations as both the subspace-based approach of Krylov used (GMRES, CGS, and BicGStab) and for each pair of basis-testing functions. The obtained results, as shown in Figures 10a–c and 11a,b, relating to $\frac{\overline{L}}{d} = 2000$, showed, in logarithmic scale and in all the cases studied, a higher convergence speed of the BicGStab procedure compared to the other approaches.

Remark 20. *It is worth noting that in the case of Triangular–Delta as basis-testing functions and for a low number of iterations, the performances of the three approaches are to be considered equivalent (see Figure 10c). This is due to the fact that in this specific case, the number of subsections is higher providing a better approximation for $\|R\|$ without the need, for the GMRES procedure, to resort to restarting.*

However, to get complete assessments about the convergence speed of each exploited procedure, the results obtained in terms of $\|R\|$ versus the number of iterations must be compared with the CPU-times obtained. Therefore, for each $\frac{\overline{L}}{d}$, the CPU-times and number iterations have been computed for GMRES, CGS, and BicGStab and for each pair basis-testing function. In particular, Tables 1, 2, 3, 4 and 5 display the performance of GMRES in terms of CPU-times as $\frac{\overline{L}}{d}$ increases, for each pair of basis-testing functions, respectively. Analyzing Table 1, relating to the Pulse–Delta pair, it can be seen that, with the variation of $\frac{\overline{L}}{d}$, the restarting procedure allows to obtain more CPU-time content even, if the number of iterations increases. However, for high values of $\frac{\overline{L}}{d}$, the CPU-time increases considerably, but in any case, the values obtained are significantly below the values obtained through the methods of Gauss elimination and Gauss–Seidel (see Tables 6 and 7) in which the calculation times for high values of $\frac{\overline{L}}{d}$ are extremely dilated so that the performance of the Gauss elimination and Gauss–Seidel methods are not suitable for any real-time applications. The same observations can be extended to the case of the pulse–pulse, triangular–delta, triangular–pulse, and triangular–triangular pairs (see Tables 2, 3, 4 and 5) where, even if the CPU-time increases for high values of $\frac{\overline{L}}{d}$ and becomes more evident, they are still significantly below the values obtained through the Gauss elimination and Gauss–Seidel methods. The performance of the CGS procedure can be considered similar to that of GMRES, as can be seen from the comparison between Tables 1, 2, 3, 4, 5 and 8 for which the GMRES and CGS procedures, depending on $\frac{\overline{L}}{d}$, can be considered equivalent. Both approaches appear slow and expensive in terms of CPU-time since they are applied to linear systems in which the matrices, in addition to being dense, are also non-symmetric. Furthermore, in the GMRES approach, both the Arnoldi iteration and the memorization of all the vectors of the base for the calculation of a new solution worsen the performance. However, the restarting operation partly overcomes this drawback and, even if the number of iterations increases, the CPU-time decreases. It is worth noting that, in these cases, convergence is always guaranteed [23]. Even if the number of iterations increases, BicGStab performs better than the other approaches by providing $\|R\|$ below tolerance by an higher number of iterations but with a very lower CPU-time (see Table 9). In particular, as the elements of the matrix increase, the BicGStab procedure is faster than the GMRES procedure. Furthermore, by increasing the number of non-null elements of the matrix, GMRES presents a lower number of iterations but is more time-consuming than the iterations required by the BicGStab procedure which, although being in large numbers, require a lot less time to run. Finally, the performance obtained showed, whatever $\frac{\overline{L}}{d}$, an increase in calculation times when the pair triangular–triangular functions as basis-testing functions are considered compared to the other pair basis-testing functions. However, the increase in CPU-times, for these pairs of basis-testing functions, slightly affect performance providing the best performance, taking into account both the CPU-times and the number of iterations to obtain $\|R\|$ below the tolerance threshold especially with regard to the BicGStab procedure, which requires half of the iterations to break down $\|R\|$ below the tolerance threshold.

Remark 21. *We observe that GMRES minimizes the residual at each iteration. However, as displayed in Figures 10a–c and 11a,b, sometimes the residual increases from one iteration to the next. This is due to the restarting procedure [23].*

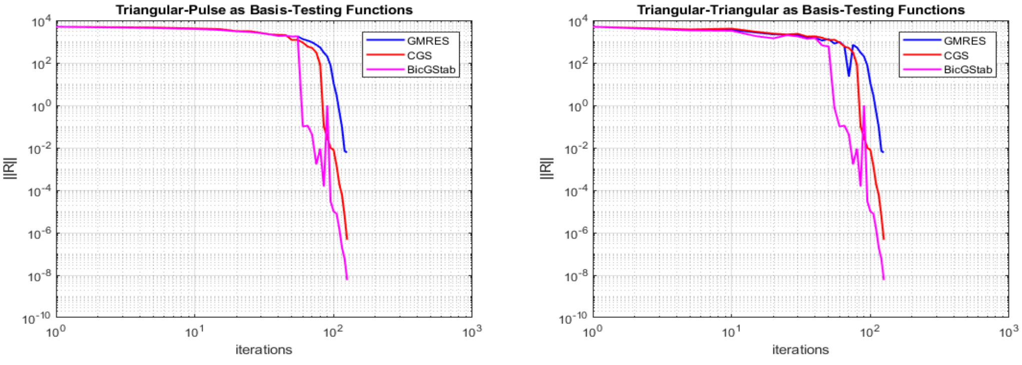
Finally, the application of the Gauss elimination method and the Gauss–Seidel algorithm did not offer performances comparable with the methods based on Krylov subspaces since the CPU times obtained were much higher, highlighting the non-competitiveness. This is essentially due to the fact that the Gauss elimination method and the Gauss–Seidel algorithm do not take into account the dimension of the matrix of each algebraic system.



(**a**) (**b**)

(**c**)

Figure 10. $\|R\|$ vs. number of iterations for GMRES. CGS and BicGStab algorithms: (**a**) pulse–delta (**b**) pulse–pulse (**c**) triangular–delta functions as basis-testing functions.



(**a**) (**b**)

Figure 11. $\|R\|$ vs. number of iterations for GMRES. CGS and BicGStab algorithms: (**a**) triangular–pulse (**b**) triangular–triangular functions as basis-testing functions.

Table 1. GMRES: CPU-time (s) and #iterations for pulse–delta basis-testing functions as $\frac{\overline{L}}{d}$ changes.

		GMRES		
		Pulse–Delta		
$\overline{\mathrm{L}}/\mathrm{d}$	**Iterations**	**CPU-Time**	**Restarting Iterations**	**CPU-Time**
6	2	2.0140	3	2.0090
10	2	2.1400	3	2.0190
15	2	2.2813	3	2.2217
20	2	2.2501	3	2.2307
25	3	2.1875	5	2.1755
30	3	2.3281	5	2.1875
60	3	2.6719	5	2.5313
100	5	2.5469	7	2.3906
250	6	2.7969	8	2.6875
500	6	4.0938	8	3.9063
1000	7	11.0000	8	10.2656
2000	8	48.6719	9	32.2188

Table 2. GMRES: CPU-time (s) and #iterations for each pair pulse–pulse basis-testing functions as $\frac{\overline{L}}{d}$ changes.

		GMRES		
		Pulse-Pulse		
$\overline{\mathrm{L}}/\mathrm{d}$	**Iterations**	**CPU-Time**	**Restarting Iterations**	**CPU-Time**
6	2	2.7969	3	2.6063
10	2	2.8438	3	2.6094
15	2	3.0156	3	3.0125
20	2	2.6875	3	2.6399
25	3	2.3906	3	2.2501
30	3	2.0910	3	2.0154
60	3	2.6719	4	2.5750
100	4	2.6999	4	2.8910
250	4	3.1406	4	3.0694
500	4	4.1875	5	3.9219
1000	5	9.3438	6	7.2344
2000	6	34.3281	7	32.7344

Table 3. GMRES: CPU-time (s) and #iterations for triangular–delta basis-testing functions as $\frac{\overline{L}}{d}$ changes.

		GMRES		
		Triangular–Delta		
$\overline{\mathrm{L}}/\mathrm{d}$	**Iterations**	**CPU-Time**	**Restarting Iterations**	**CPU-Time**
6	2	2.7344	3	2.5781
10	2	2.7656	3	3.0156
15	2	2.5625	3	2.4844
20	3	2.7188	3	2.2500
25	3	2.7969	4	2.3281
30	3	2.7813	4	2.2344
60	3	2.6406	4	2.1875
100	3	2.6719	4	2.6250
250	4	3.4625	4	3.3906
500	4	4.3281	5	3.4375
1000	5	8.6250	6	8.5938
2000	6	42.7813	7	37.3750

Table 4. GMRES: CPU-time (s) and #iterations for triangular–pulse basis-testing functions as $\frac{\overline{L}}{d}$ changes.

		GMRES		
		Triangular-Pulse		
$\overline{L}/d$	**Iterations**	**CPU-Time**	**Restarting Iterations**	**CPU-Time**
6	2	2.8281	3	3.0625
10	2	3.1250	3	2.9844
15	2	2.5313	3	2.2344
20	3	2.5938	3	2.2031
25	3	2.2384	4	2.1875
30	3	3.0000	4	2.1563
60	3	2.4375	4	2.3594
100	4	3.2344	5	2.9688
250	4	2.7344	5	2.7188
500	5	3.5156	6	3.4250
1000	6	9.7969	7	9.1250
2000	6	34.2500	7	31.0587

Table 5. GMRES: CPU-time (s) and #iterations for triangular–triangular basis-testing functions as $\frac{\overline{L}}{d}$ changes.

		GMRES		
		Triangular–Triangular		
$\overline{L}/d$	**Iterations**	**CPU-Time**	**Restarting Iterations**	**CPU-Time**
6	2	2.0589	3	2.0358
10	2	2.0825	3	1.9875
15	2	2.1875	3	2.0987
20	3	2.5156	3	2.1719
25	3	2.6250	4	2.2656
30	3	2.1406	4	2.3594
60	3	2.4063	4	2.4001
100	4	2.7656	5	24844
250	4	2.4219	5	2.4089
500	5	4.3125	6	3.8281
1000	6	8.5156	7	7.7188
2000	6	35.9219	7	33.4688

Table 6. Gauss elimination: CPU-time (s) for each pair basis-testing functions as $\frac{\overline{L}}{d}$ changes.

			Gauss Elimination		
	Pulse–Delta	**Pulse–Pulse**	**Triangular–Delta**	**Triangular–Pulse**	**Triangular–Triangular**
$\overline{L}/d$	**CPU-Time**	**CPU-Time**	**CPU-Time**	**CPU-Time**	**CPU-Time**
6	0.0313	0.0156	0.0156	0.0156	0.0156
10	0.0156	0.0163	0.0169	0.0173	0.0174
15	0.0434	0.0436	0.0433	0.0469	0.0521
20	0.0625	0.0691	0.0729	0.0785	0.0873
25	0.0794	0.0749	0.0757	0.0789	0.0798
30	0.0813	0.0836	0.0849	0.0875	0.0899
60	0.2313	0.2156	0.2515	0.2790	0.2873
100	0.3290	0.3334	0.3512	03586	0.3599
250	0.6250	0.6094	0.6406	0.6563	0.7344
500	2.0781	2.0625	2.4375	2.3125	2.4219
1000	15.7188	17.0625	17.9063	17.6563	18.2031
2000	115.6875	117.3281	117.8125	122.5781	143.3438

Table 7. Gauss–Seidel: CPU-time (s) and number of iterations for each pair basis-testing functions as $\frac{\overline{L}}{d}$ changes.

	Gauss–Seidel				
	Pulse–Delta	**Pulse–Pulse**	**Triangular–Delta**	**Triangular–Pulse**	**Triangular–Triangular**
$\overline{\mathbf{L}}/\mathbf{d}$	**CPU-time**	**CPU-time**	**CPU-time**	**CPU-time**	**CPU-time**
6	0.0078	0.0093	0.0099	0.0089	0.0080
10	0.0145	0.0130	0.0096	0.0117	0.0159
15	0.0192	0.0195	0.0148	0.0171	0.0216
20	0.0245	0.0279	0.0314	0.0301	0.0317
25	0.0305	0.0377	0.0288	0.0313	0.0399
30	0.0443	0.0437	0.0534	0.0829	0.0827
60	0.1476	0.2309	0.2284	0.2378	0.1849
100	0.4046	0.5806	0.4640	0.5668	0.5505
250	3.0992	4.4617	4.9277	5.4185	6.4172
500	16.0497	18.5411	21.9452	20.6219	20.8443
1000	63.5523	74.6598	71.0213	80.4083	72.5813
2000	326.7238	311.5719	321.5325	352.0632	377.7439

Table 8. CGS: CPU-time (s) and number of iterations for each pair basis-testing functions as $\frac{\overline{L}}{d}$ changes.

	CGS									
	Pulse-Delta		**Pulse-Pulse**		**Triangular-Delta**		**Triangular-Pulse**		**Triangular-Triangular**	
$\overline{\mathbf{L}}/\mathbf{d}$	**Iterations**	**CPU-Time**	**Iterations**	**CPU-Time**	**Iterations**	**CPU-Time**	**Iterations**	**CPU-Time**	**Iterations**	**CPU-Time**
6	2	2.0151	2	2.7841	2	2.6999	2	2.8101	2	2.5147
10	2	2.1235	2	2.8891	2	2.7391	2	3.1341	2	2.0799
15	2	2.2789	2	3.0045	2	2.5701	2	2.5401	2	2.1798
20	2	2.2124	2	2.6587	2	2.7089	2	2.5899	2	2.5124
25	3	2.1822	3	2.3924	3	2.7899	3	2.2322	3	2.6212
30	3	2.3214	3	2.0892	3	2.7796	3	2.9971	3	2.1427
60	3	2.6745	3	2.6752	3	2.6498	3	2.4301	3	2.3983
100	5	2.5402	5	2.6905	5	2.6746	5	3.2314	5	2.7691
250	6	2.7912	6	3.1382	6	3.4601	6	2.7391	6	2.4290
500	6	4.1598	6	4.1926	6	4.3301	6	3.5149	6	4.3259
1000	7	11.0548	7	9.3875	7	8.6254	7	9.9782	7	8.5203
2000	8	48.0254	8	34.0369	8	42.9816	8	34.9311	8	35.9315

Table 9. BicGStab: CPU-time (s) and number of iterations for each pair basis-testing functions as $\frac{\overline{L}}{d}$ changes.

	BicGStab									
$\overline{\mathrm{L}}/\mathrm{d}$	**Pulse-Delta Iterations**	**CPU-Time**	**Pulse-Pulse Iterations**	**CPU-Time**	**Triangular-Delta Iterations**	**CPU-Time**	**Triangular-Pulse Iterations**	**CPU-Time**	**Triangular-Triangular Iterations**	**CPU-Time**
6	5	0.2188	5	0.3438	5	0.4531	5	0.3987	5	0.4193
10	12	0.2656	11	0.5156	14	0.3125	16	0.2188	16	0.2982
15	28	0.2969	20	0.4844	21	0.2656	26	0.2656	27	0.0938
20	48	0.1563	48	0.1496	38	0.1094	67	0.2656	63	0.2188
25	59	0.2500	64	0.2500	93	0.2188	65	0.2969	90	0.1875
30	134	0.1250	50	0.1406	181	0.4219	197	0.2656	141	0.1406
60	130	0.500	868	0.6094	286	0.3125	135	0.3750	344	0.3281
100	189	0.3438	307	0.3906	204	0.3281	191	0.4063	179	0.5625
250	98	0.5313	851	0.7969	95	0.4219	449	0.7969	115	0.3750
500	317	0.8906	251	0.5156	325	0.6875	416	0.7656	235	0.5625
1000	304	1.3125	320	1.3906	143	0.8906	395	1.5938	113	0.5781
2000	468	24.8281	799	25.8439	924	24.4219	1013	25.5000	1023	25.9243

7. Conclusions and Perspectives

In this paper, an in-depth comparison among some Krylov subspace method procedures for solving the algebraic linear systems with dense matrix deriving from the application of the MoM for calculating the distribution of the electrostatic charge and the related capacitance of a straight cylindrical conductor of finite length have been presented. Five cases have been identified according to the type of basis-testing functions pair used: (a) pulse function as the basis function and delta function as the testing function (point matching); (b) pulse function as the basis function as well as testing function; (c) triangular function as the basis function and delta function as the testing function (point matching); (d) triangular function as the basis function and pulse function as the testing function; (e) triangular function as the basis function as well as the testing function. For the resolution of the five algebraic linear systems thus obtained, GMRES, CGS, and BicGStab algorithms (all based on Krylov subspaces approaches) have been exploited to achieve the electrostatic charge distribution on the surface of the conductor and capacitance. Numerical results demonstrate the superiority of the BicGStab algorithm in terms of reducing the number of iterations required to reduce the residual norm below the selected tolerance against a reduced increase in CPU-time for each iteration compared to the other considered procedures. In particular, as the size of the matrix increases (due to the increase in the number of subsections of the mesh), the number of iterations required by the BicGStab procedure, compared to the other procedures used, increases significantly but with a significantly reduced CPU-time consumed. In fact, the CPU-time value obtained with BicGStab (25.9 s) is 46.1% lower than the highest CPU-time obtained with the CGS procedure (which is around 48 s) while it is reduced by 27.91% of the highest CPU-time value obtained with the GMRES method (which is around 36 s). Again, it is reduced by 81.93% of the CPU-time value obtained with the Gauss elimination method (around 143 s), and is reduced by 93.1% compared to the higher CPU-time obtained by the Gauss–Seidel iterative procedure. This confirms the indisputable superiority of the BicGStab procedure compared to the others used because, even if it requires a greater number of iterations, it performs them in a considerably more contained time. This research can be considered as an important step especially for a realtime application that requires quick approaches. Finally, the fact that L matrices suffer from the problem of poor conditioning in all cases, in the future, it is desirable to use "ad-hoc" preconditions capable of eliminating (or at least reducing) the effects of this problem.

Author Contributions: conceptualization, M.V. and G.A.; methodology, G.A.; software, M.V.; validation,M.V. and G.A.; formal analysis, M.V. and G.A.; investigation, M.V.; resources, G.A.; writing—original draft preparation, M.V.; writing—review and editing, G.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

EM	electromagnetic problems
$\mathcal{L}$	differential, integral, or integro-differential operator
h	excitation source
f	unknown function to be determined
L	algebraic sparse matrix
MoM	method of moments
q_e	electrostatic charge distribution
C	electrostatic capacitance
GMRES	generalized minimal residual method
CGS	conjugate gradient squared
BicGStab	biconjugate gradient Stabilized
$Oxyz$	system of Cartesian axes
x	longitudinal symmetry axis of the cylindrical conductor
Ω	source region
$\overline{L}$	length of the conductor
a	radius of the conductor
$\mathbf{r}'$	vector pointing the observation points
$\mathbf{r}$	vector pointing the source points
$\phi(x)$	electrostatic potential
ϵ	permittivity of the free space
α_n	constants to be determined
f_n	basis functions or expansion functions
R	residual
N	number of sections of the conductor
$w_m, m = 1, \ldots, N$	weighting or testing functions
$\Delta(x)$	Dirac Delta
V	electrical voltage
L_{mn}	individual elements of the matrix L
$\mathcal{K}_r$	Krylov subspace with index r
$\mathcal{K}_k(A^T, \mathbf{b})$	Krylov subspace with index k generated by A^T and $\mathbf{b}$
K	conditioning number

References

1. Harrington, R.F. *Field Computation by Moment Methods*; MacMillan: New York, NY, USA, 1968.
2. Papas, C.H. *Theory of Electromagnetic Wave Propagation*; Dove Publications, Inc.: New York, NY, USA, 2005.
3. Angiulli, G.; Tringali, S. Stabilizing the E-Field Integral Equation at the Internal Resonances through the Computation of Its Numerical Null Space. *Int. J. Appl. Electromagn. Mech.* **2010**, *32*, 63–72.
4. Dault, D.; Nair, N.V.; Li, J.; Shanker, B. The Generalized Method of Moments for Electromagnetic Boundary Integral Equations. *IEEE Trans. Antennas Propag.* **2014**, *62*, 3174–3188.
5. Jonassen, N. *Electrostatics*; Springer Nature Switzerland AG: Cham, Switzerland, 2019.
6. Luek, A.; Wolf, V. Generalized Method of Moments for Estimating Parameters of Stochastic Reaction Networks. *Syst. Biol.* **2016**, *10*, 98.
7. Gibson, W.C. *The Method of Moments in Electromagnetics*; Chapman & Hall/CRC, Taylor & Francis Group: Boca Raton, FL, USA; London, UK; New York, NY, USA, 2008.
8. Chakrabarty, S.B. Capacitance of Dielectric Coated Cylinder of Finite Axial Length and Truncated Cone Isolated in Free Space. *IEEE Trans. Electromagn. Compat.* **2002**, *44*, 394–398.
9. Angiulli, G.; Cacciola, M.; Calcagno, S.; De Carlo, D.; Morabito, F.C.; Sgró, A.; Versaci, M. A Numerical Study on the Performances of the flexible BicGStab to Solve the Discretized E-Field Integral Equation. *Int. J. Appl. Electromagn. Mech.* **2014**, *46*, 547–553.
10. Alad, R.H.; Chakrabarty, S.B. Capacitance and Surface Charge Distribution Computations for a Satellite Modeled as a Rectangular Cuboid and Two Plates. *J. Electrost.* **2013**, *71*, 1005–1010.
11. Cacciola, M.; Morabito, F.C.; Polimeni, D.; Versaci, M. Fuzzy Characterization of Flawed Metallic Plates with Eddy Current Tests. *Prog. Electromagn. Res.* **2007**, *72*, 241–252.
12. Burrascano, P.; Callegari, S.; Montisci, A.; Ricci, M.; Versaci, M. *Ultrasonic Nondestructive Evaluation Systems: Industrial Application Issues*; Springer International Publishing: Cham, Switzerland, 2015.
13. Mishra, M.; Gupta, N.; Dubey, A.; Shekhar, S. A New Algorithm for Method of Moments Solution of Static Charge Distribution. In Proceedings of the 2008 IEEE Regium 10 Colloquium amd the Third International Conference on Industrial and Information Systems, Kharagpur, India, 8–10 December 2008; pp. 1–3.
14. Alad, R.H.; Chakrabarty, S.B. Investigation of Different Basis and Testing Functions in Method of Moments for Electrostatic Problems. *Prog. Electromagn. Res. B* **2012**, *44*, 31–52.
15. Barrett, J.; Formaggio, J.; Corona, T. A Method to Calculate the Spherical Multipole Expansion Electrostatic Charge Distribution on a Triangular Boundary. *Prog. Electromagn. Res. B* **2015**, *63*, 123–143.
16. Hamada, S. Electrostatic Field Analysis of Anisotropic Conductive Media Using Voxe-Based Static Method of Moments with Fast Multipole Method. *Eur. J. Comput. Mech.* **2016**, *25*, 54–70.
17. Maity, K. Capacitance Computation of a Charge Conducting Plate Using Mehod of Moments. *Int. J. Res. Sci. Innov.* **2018**, *V*, 7–11.
18. Salama, F.; Siwinski, A.; Atieh, K.; Mehrani, P. Investigation of Electrostatic Charge Distribution Within the Reactor Wall Fouling and Bulk Regions of a Gas-Solid Fluidized Bed. *J. Electrost.* **2013**, *71*, 21–27.
19. Ahmed, S.; Sharma, D.; Chaturved, S. Finite-Difference Time-Domain Analysis of Parallel-Plate Capacitors. *Int. J. Comput. Methods Eng. Sci. Mech.* **2005**, *6*, 105–114.
20. Chen, J.; McInnes, L.C.; Xhang, H. Analysis and Practical Use of Flexible BicGStab. *J. Sci. Comput.* **2016**, *68*, 803–825.
21. Van Der Vorst, H. *Iterative Krylov Methods for Large Linear Systems*; Cambridge University Press: Cambridge, UK, 2003.
22. Simoncini, V.; Szyld, D.B. Recent Computational Developments in Krylov Subspace Methods for Linear Systems. *Numer. Linear Algebra Appl.* **2007**, *14*, 112–138.
23. Trefethen, L.N.; Bau, D. *Numerical Linear Algebra*; SIAM: Philadelphia, PA, USA, 2018.
24. Parks, M.L.; de Sturler, E.; Mackey, G. Recycling Krylov Subspace for Sequences of Linear Systems. *SIAM J. Sci. Comput.* **2004**, *28*, 1–24.
25. Yang, J.M.; Jiang, Y.L. Krylov Subspace Approximation for Quadratic-Bilinear Differential System. *Int. J. Syst. Sci.* **2018**, *49*, 1950–1963.
26. Tranquilli, P.; Sandu, A. Rosenbrock-Krylov Methods for Large Systems of Differential Equations. *SIAM J. Sci. Comput.* **2014**, *36*, A1313–A1338.

27. Gosselet, P.; Rey, C.; Pebrel, J. Total and Selective Reuse of Krylov Subspaces for the Resolution of Sequences of Non-Linear Structural Problems. *Int. J. Numer. Methods Eng.* **2013**, *94*, 60–83.
28. Wang, S.; de Sturler, E.; Paulino, G.H. Large-Scale Topology Optimization using Preconditioned Krylov Subspace Methods with Recycling. *Int. J. Numer. Methods Eng.* **2007, 69, 2441–2468**.
29. Harkort, C.; Deutscher, J. Krylov Subspace Methods for Linear Infinite-Dimensional Systems. *IEEE Trans. Autom. Control* **2011**, *56*, 441–447.
30. Beerwerth, R.; Bauke, H. Krylov Subspace Methods for the Dirac Equation. *Comput. Phys. Commun.* **2015**, *188*, 189–197.
31. Saad, Y. Analysis of Some Krylov Subspace Approximations to the Matrix Exponential Operator. *SIAM J. Nunerical Anal.* **2002**, *29*,209–228.
32. Pedroza-Montero, J.N.; Morales, J.L.; Geudtner, G.; Alvarez-Ibarra, A.; Calaminici, P.; Koester, M. Variational Density Fitting with a Krylov Subspace Method. *J. Chem. Theory Comput.* **2020**, *16*, 2965–2974.
33. Sou-Cheng, T.; Christopher, C.; Saunders, M.A.; Koester, M. MINRES-QLP: A Krylov Subspace Method for Indefinite or Singular Symmetric Systems. *SIAM J. Sci. Comput.* **2011**, *33*, 1810–1836.
34. Puzyrev, V.; Cela, J.M. A Review of Block Krylov Subspace Methods for Multisource Electromagnetic Modelling. *Geophys. J. Int.* **2015**, *202*, 1241–1252.
35. Balidemaj, E.; Remis, R.F. A Krylov Subspace Approach to Parametric Inversion of Electromagnetic Data Based on Residual Minimization. *PIERS Online* **2010**, *6*, 773–777.
36. Oscar, B.; Elling, T.; Paffenroth, R.; Catalin, T. Electromagnetic Integral Equations Requiring Small Number of Krylov-Subspace Iterations. *J. Comput. Phys.* **2009**, *228*, 6169–6183.
37. Huang, B.; Xie, Y.; Ma, C. Krylov Subspace Methods to Solve a Class of Tensor Equations via the Einstein Product. *Numer. Linear Algebra Appl.* **2019**, *26*, 122–139.
38. Jaimoukha, I.M.; Kasenally, E.M. Krylov Subspace Methods for Solving Large Lyapunov Equations. *SIAM J. Numer. Anal.* **2004**, *31*, 227–251.
39. Freund, R.W. Krylov Subspace Methods for Reduced-Order Modeling in Circuit Simulation *J. Comput. Appl. Math.* **2000**, *123*, 395–421.
40. Heres, P.; Schilders, W. Krylov Subspace Methods in the Electronic Industry. In *Progress in Industrial Mathematics at ECMI 2004*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 8, pp. 227–239.
41. Jackson, J.D. Charge Density on Thin Straight Wire, Revisited. *Am. J. Phys.* **2000**, *68*, 789–799.