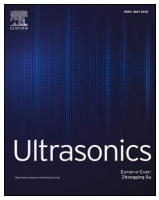
Ultrasonics 124 (2022) 106747

Contents lists available at ScienceDirect

Ultrasonics

journal homepage: www.elsevier.com/locate/ultras



# Solving ultrasonic ray tracing in parts with multiple material layers through Root-Finding methods

Carmelo Mineo [a,1,*], Donatella Cerniglia [b], Ehsan Mohseni [c]

[a] High Performance Computing and Networking Institute (ICAR), National Research Council (CNR), 90146 Palermo, Italy
[b] Department of Engineering, University of Palermo, Viale delle Scienze, Edificio 8, 90128 Palermo, Italy
[c] Department of Electronic & Electrical Engineering (EEE), University of Strathclyde, 204 George St, Glasgow G1 1XW, UK

ARTICLE INFO

*Keywords:*
Wave propagation
Refraction
Iterative root-finding methods
Multi-layered structures

ABSTRACT

Ultrasonic testing has been used for material analysis and inspection since 1930′s. Nevertheless, the applicability of ultrasonic waves to new complex cases is still growing, thanks to the availability of powerful electronics and software. However, the complication that slows down the deployment of ultrasonic inspection to geometric complex parts and structures arises from the wave refraction phenomenon. A clear understanding of the ultrasound wave propagation, impacted by refractions, is crucial to interpret the data obtained from the inspection of multi-layered/multi-medium test subjects as it is not always possible to assume that mechanical waves travel in straight lines. This work presents suitable approaches for solving the ray-tracing problem in multi-layered structures. Accurate benchmarking shows that the use of the Newton-Raphson root-finding method allows a threefold reduction of the computation time, when compared to the bisection-based root-finding methods. An effective combination of the Newton-Raphson methods with bisection-type iterations is also proposed and discussed. Although the work repeatedly refers to the field of ultrasonic inspection, the presented findings are relevant and applicable to areas beyond material inspection.

## 1. Introduction

The propagation of ultrasonic waves has been used for material inspection and characterization for many decades and ultrasonic testing (UT) has become one of the most established non-destructive testing methods. Very short ultrasonic pulse-waves are normally transmitted into manufactured parts, components, and systems, to detect discontinuities and to characterize material properties based on the information contained in the received echoes. Many industrial sectors use ultrasonic testing to inspect components made of natural (e.g. wood), metallic, plastic and composite materials [1–5]. In medical diagnostics, ultrasonic echography is greatly used to obtain non-invasive examination of human and/or animal tissues. Whereas, the UT inspection of flat components is typically unproblematic, the ultrasonic assessment of specimens with curved surfaces, such as aerofoils, weld-caps, pipes, and other complex structures of this type can be difficult and necessitates bespoke approaches. High frequency pressure waves generated by ultrasound transducers are strongly attenuated when they travel through air gaps; therefore, conventionally, coupling liquids and/or solid wedges, designed to match the shape of the components, are used to couple the ultrasonic transducer with the part under inspection [6–8]. Seeking advanced research solutions to the inspection of components with curved surfaces has been the focus of several works, which have proposed ways to couple the ultrasonic transducers with complex shaped components. From these possible solutions, including the deployment of flexible coupling material layer between the transducer and the part that conforms to the surface of the part [9], and the placement of the transducer on a nearby planar surface to reach the region of interest from the side either directly [10] or by using the waves reflected from the back wall of the component [11] can be named. Besides the need to address the coupling problem, the inspection of parts with curved geometries introduces great complications due to wave refraction. The ultrasonic signals acquired during the inspection phase are required to be processed to extract the information corresponding to the material properties and/or to the location and size of any potential flaws. Unfortunately, due to the wave refraction at the interface of two adjacent layers with different material properties it is not possible to assume mechanical waves travel in a straight line through multiple materials.

* Corresponding author.
*E-mail address:* carmelo.mineo@icar.cnr.it (C. Mineo).
[1] This work was performed at the University of Palermo.

https://doi.org/10.1016/j.ultras.2022.106747
Received 3 September 2021; Received in revised form 30 March 2022; Accepted 3 April 2022
Available online 6 April 2022

This problem has been tackled by previous works, through the investigation of different ways to compute the ultrasonic ray paths [12–14]. The RAYTRAIM method allows numerical discrete solution of the ray-tracing in anisotropic inhomogeneous elastic materials [15,16]. One solution consists of using the Fast-Marching Method (FMM) combined with Fermat's principle. It considers the domain of interest as a grid of connected points, and then uses Dijkstra's algorithm [17]. Tant *et al.* [18] extended FMM to heterogeneous materials, where each region can be anisotropic. A recent work by the authors of the present paper introduced a generalized analytic iterative method for the computation of ultrasonic ray paths when the ultrasonic source and target are separated by multiple complex material interfaces in the two and three-dimensional domains [19]. That was based on the iterative bisection root-finding algorithm. Despite the general applicability of this iterative method, further investigations highlighted that the solutions are reached after lengthier computations when compared to other applicable mathematical models. The present work aims to alleviate the lengthy computational time of the previous work through the development and testing of new optimized iterative solvers to achieve reduced solution time for ray tracing across several material interfaces. Throughout this paper, the assumption is that each interface between two material layers is described by a continuous and differentiable function, with a local radius of curvature that is greater than the wavelength of the mechanical wave. The interface curves are also supposed to not intersect, meaning that the ray always impinges on the $i^{th}$ interface curve before intersecting the $(i+1)^{th}$ interface. Under these conditions, a ray-based approach is valid and the wave propagation problem and its transmission across interfaces can be modelled through computing the correct ray path between source positions and target within the component. This work presents faster methods, based on the well-known classical Newton-Raphson root-finding algorithm, which can be readily employed to efficiently solve the ray-tracing problem in the two-dimensional domain. The manuscript starts with a description of the bisection method and of the Newton-Raphson method, in Section 2. Then, Section 3 explains how both methods can be utilized to compute the ultrasonic path in parts with multiple material layers, which are separated by curved interfaces. Crucially, a novel formulation for the definition of the initial guess interval (for bisection-based methods) and of the initial guess of the root (for Newton-based methods) and a rationale for the choice of suitable criteria to stop the iteration of the calculus is introduced. Section 4 presents the implementation and the performance assessment of the proposed method and of various variants, through quantitative comparisons with the existing approach, highlighting advantages and limitations. Section 5 provides the conclusions of this work and the future prospects.

## 2. Iterative root-finding methods

The iteration of specific computations is at the core of all root-finding algorithms. Each iterative root-finding method generally uses a distinctive type of computation. Typically, an auxiliary function (also referred to as objective function) is defined and iteratively used to get progressively better approximations of the root. The iterations stop when a fixed point (up to the desired precision) of the auxiliary function is reached, that is when the newly computed value is sufficiently close to the preceding ones, satisfying the error bounds stated by the user. This work models the ray tracing problem for the application of the bisection method and of the Newton-Raphson method, comparing the performance of the two methods. The first is the simplest and the most robust root-finding algorithm, but its convergence speed versus time shows a linear behaviour. The latter is faster, achieving a quadratic convergence speed. This work also highlights the applicability of hybrid methods, consisting of the combination of the Newton-Raphson method with the bisection method.

### 2.1. Bisection method

The bisection method applies to any continuous function for which two values with opposite signs are known [20]. The method consists of iteratively bisecting the interval bounded by these values and then successively selecting the subintervals in which the function changes sign and, therefore, must contain a root until a root estimation closer that the predefined error is reached. The method is applicable to solve the equation $f(x) = 0$ numerically for the real variable $x$, where $f(x)$ is a continuous function defined on an interval $[a, b]$ and where $f(a)$ and $f(b)$ have opposite signs. In this case, $a$ and $b$ are said to bracket a root since, by the intermediate value theorem, the continuous function $f(x)$ must have at least one root in the interval $(a, b)$. The interval halving method is an iterative method. At each iteration the method divides the interval in two by computing the midpoint, $c = (a+b)/2$, of the interval and the value of the function $f(c)$. Unless $c$ itself is a root (which is very unlikely, but possible), there are only two possibilities: I. either $f(a)$ and $f(c)$ have opposite signs and bracket a root, or II. $f(c)$ and $f(b)$ have opposite signs and bracket a root. The method selects the subinterval that is guaranteed to be a bracket as the new interval to be used in the next iteration. Explicitly, if case I occurs, then the method sets $c$ as the new value for $b$, and if II becomes correct, then the method sets $c$ as the new $a$. In both cases, the new $f(a)$ and $f(b)$ have opposite signs, so the method is applicable to this smaller interval. In this way, the interval that contains a zero of $f(x)$ is reduced in width by 50% at each iteration. The process is continued until the interval is sufficiently small or if $f(c) = 0$. Then $c$ is taken as the root of the function and the process stops. The bisection method is easily applicable to the ray-tracing problem by estimating the interval of the incidence angle that is expected to contain the optimum incidence angle value, for the first layer. Then, starting from an initial guess value for the first incidence angle (the mean value of the interval), the ultrasonic ray is propagated through all material layers. A new subinterval, to be used for the next iteration of the method, is selected according to the entity and the sign of the deviation between the position reached by the ray and the position of the target point at the last layer. The iterations stop as soon as the deviation result becomes lower than a set maximum acceptable value. This approach was demonstrated in [19], for two dimensional and three-dimensional domains.

### 2.2. Newton-Raphson method

The Newton–Raphson method, named after Isaac Newton and Joseph Raphson [21], is a root-finding algorithm that produces a successively better approximation of the root (or zero) of a real-valued function $f(x)$ [22]. Having $f'(x)$ as the derivative function and $x_0$ as the initial guess for the root, the method suggests that $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$ is a better approximation of the root than $x_0$. Geometrically, $(x_1, 0)$ is the intersection of the x-axis and the tangent to the function at $(x_0, f(x_0))$. Therefore, the improved guess is the unique root of the linear approximation at the initial point. The process is repeated iteratively until the root is sufficiently accurately estimated, after the $j^{th}$ iteration through the formula:

$$x_{j+1} = x_j - \frac{f(x_j)}{f'(x_j)} \tag{1}$$

The main advantage of the method is that it can be readily generalized to higher-dimensional problems where it can be extended to systems of equations, with $n$ equations and unknown variables. The generalized form of the Newton-Raphson method for solving a system of equations, with $\boldsymbol{x}$ for the vector of unknown variables, $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$, and with $\boldsymbol{g}(\boldsymbol{x})$ for the vector of functions, can be formulated as:

$$\boldsymbol{x}_{j+1} = \boldsymbol{x}_j - \boldsymbol{g}(\boldsymbol{x}_j)\cdot\boldsymbol{J}(\boldsymbol{x}_j)^{-1} = \boldsymbol{x}_j - \boldsymbol{g}(\boldsymbol{x}_j)\cdot\begin{bmatrix} \frac{\partial g_1(\mathbf{x_j})}{\partial x_1} & \frac{\partial g_1(\mathbf{x_j})}{\partial x_2} & \cdots & \frac{\partial g_1(\mathbf{x_j})}{\partial x_n} \\ \frac{\partial g_2(\mathbf{x_j})}{\partial x_1} & \frac{\partial g_2(\mathbf{x_j})}{\partial x_2} & \cdots & \frac{\partial g_2(\mathbf{x_j})}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_n(\mathbf{x_j})}{\partial x_1} & \frac{\partial g_n(\mathbf{x_j})}{\partial x_2} & \cdots & \frac{\partial g_n(\mathbf{x_j})}{\partial x_n} \end{bmatrix}^{-1} \tag{2}$$

where $\boldsymbol{J}$ is the Jacobian matrix of $\boldsymbol{g}(\boldsymbol{x})$.

## 3. Problem modelling

Perfectly flat interfaces are only a theoretical abstraction since real interfaces always present some degree of roughness and deviation from flatness. Moreover, intentionally curved material interfaces are often present in parts that are designed to meet critical mechanical properties. In order to use the bisection and the Newton-Raphson root-finding methods to compute the optimum ultrasonic ray-path, a suitable set of equations is defined in this work.

Fig. 1 shows the schematic representation of an ultrasonic ray that travels through four material layers, which are connected through three curved interfaces. Taking $f_i(x)$ as the function that describes the interface between the $i^{th}$ and the $(i+1)^{th}$ material layer, it is possible to write the following $n$ valid equations, for the generic case with $n$ layers and $n$-1 interfaces:

$$\begin{cases} P_1^x = S^x + \left[S^y - f_1\left(P_1^x\right)\right]\cdot\tan(\theta_1) \\ P_2^x = P_1^x + \left[f_1\left(P_1^x\right) - f_2\left(P_2^x\right)\right]\cdot\tan(\theta_2) \\ \cdots \\ P_i^x = P_{i-1}^x + \left[f_{i-1}\left(P_{i-1}^x\right) - f_i\left(P_i^x\right)\right]\cdot\tan(\theta_i) \\ \cdots \\ P_{n-1}^x = P_{n-2}^x + \left[f_{n-2}\left(P_{n-2}^x\right) - f_{n-1}\left(P_{n-1}^x\right)\right]\cdot\tan(\theta_{n-1}) \\ T^x = P_{n-1}^x + \left[f_{n-1}\left(P_{n-1}^x\right) - T^y\right]\cdot\tan(\theta_n) \end{cases} \tag{3}$$

where $S \equiv (S^x, S^y)$ and $T \equiv (T^x, T^y)$ are, respectively, the source and the target points and $P_i \equiv \left(P_i^x, P_i^y\right)$ is the point of incidence with the interface $f_i(x)$. $\theta_i$ is the angle the ultrasonic ray forms with the vertical direction (the y-axis) in the $i^{th}$ layer.

Fig. 1b schematically illustrates the relationship between $\theta_i$, $\theta_{i+1}$, $\alpha_i$ and $\alpha_{i+1}$, where $\alpha_i$ is the incidence angle with respect to the interface normal at incidence point $P_i$ on the $i^{th}$ interface and $\alpha_{i+1}$ is the respective refraction angle. The relationship between these latter angles of $\alpha_i$ and $\alpha_{i+1}$ are described by the Snell's law which is conventionally stated in the form of the following equation:

$$\frac{\sin\alpha_i}{v_i} = \frac{\sin\alpha_{i+1}}{v_{i+1}} \tag{4}$$

where $v_i$ and $v_{i+1}$ are the speed of sound/ultrasound propagation for the $i^{th}$ and $(i+1)^{th}$ material layers, respectively. From Fig. 1b, it is clear that:

$$\alpha_i = \theta_i - \operatorname{atan}\left(f_i^{'}\left(P_i^x\right)\right) \tag{5}$$

$$\alpha_{i+1} = \theta_{i+1} - \operatorname{atan}\left(f_i^{'}\left(P_i^x\right)\right) \tag{6}$$

Replacing Eqs. (5) and (6) in Eq. (4) and reordering, it results:

$$\theta_{i+1} = asin\left\{\left(\frac{v_{i+1}}{v_i}\right)\cdot\sin\left[\theta_i - atan\left(f_i^{'}\left(P_i^x\right)\right)\right]\right\} + atan\left(f_i^{'}\left(P_i^x\right)\right) \tag{7}$$

Therefore, it is easy to understand that *n-1* additional equations, describing the dependence of every refraction angle on the originating incidence angle and the relative point of incidence at the interface, can be appended to the equations in Eq. (3). Thereby, the generalized form for the case with $n$ material layers has a system of equations consisting of *2n-1* equations and the same number of unknown values $\left(P_1^x, \cdots, P_{n-1}^x, \theta_1, \cdots, \theta_n\right)$. Making the elements of the vector function $\boldsymbol{g}(\boldsymbol{x})$ explicit and defining the vector of the unknown values as $\boldsymbol{x} = \left(P_1^x, \cdots, P_{n-1}^x, \theta_1, \cdots, \theta_n\right)$, it results:

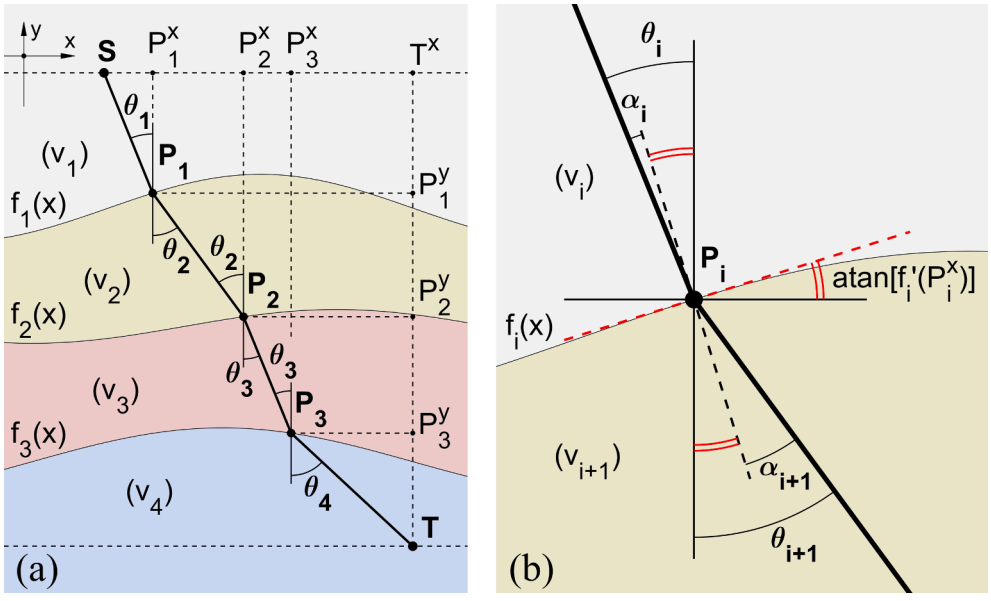



**Fig. 1.** Annotated ray path travelling through multiple material layers separated by curved interfaces (a) and illustration of relevant angles at the i[th] incidence point $(P_i)$ on the i[th] interface (b) [19].

$$
\begin{cases}
g_1\left(P_1^x, \cdots, P_{n-1}^x, \theta_1, \cdots, \theta_n\right) = S^x - P_1^x + \left[S^y - f_1\left(P_1^x\right)\right]\cdot\tan(\theta_1) \\
g_2\left(P_1^x, \cdots, P_{n-1}^x, \theta_1, \cdots, \theta_n\right) = P_1^x - P_2^x + \left[f_1\left(P_1^x\right) - f_2\left(P_2^x\right)\right]\cdot\tan(\theta_2) \\
\cdots \\
g_i\left(P_1^x, \cdots, P_{n-1}^x, \theta_1, \cdots, \theta_n\right) = P_{i-1}^x - P_i^x + \left[f_{i-1}\left(P_{i-1}^x\right) - f_i\left(P_i^x\right)\right]\cdot\tan(\theta_i) \\
\cdots \\
g_{n-1}\left(P_1^x, \cdots, P_{n-1}^x, \theta_1, \cdots, \theta_n\right) = P_{n-2}^x - P_{n-1}^x + \left[f_{n-2}\left(P_{n-2}^x\right) - f_{n-1}\left(P_{n-1}^x\right)\right]\cdot\tan(\theta_{n-1}) \\
g_n\left(P_1^x, \cdots, P_{n-1}^x, \theta_1, \cdots, \theta_n\right) = P_{n-1}^x - T^x + \left[f_{n-1}\left(P_{n-1}^x\right) - T^y\right]\cdot\tan(\theta_n) \\
g_{n+1}\left(P_1^x, \cdots, P_{n-1}^x, \theta_1, \cdots, \theta_n\right) = asin\left\{(v_2/v_1)\cdot\sin\left[\theta_1 - atan\left(f_1'\left(P_1^x\right)\right)\right]\right\} + atan\left(f_1'\left(P_1^x\right)\right) - \theta_2 \\
\cdots \\
g_{2n-1}\left(P_1^x, \cdots, P_{n-1}^x, \theta_1, \cdots, \theta_n\right) = asin\left\{(v_n/v_{n-1})\cdot\sin\left[\theta_{n-1} - atan\left(f_{n-1}'\left(P_{n-1}^x\right)\right)\right]\right\} + atan\left(f_{n-1}'\left(P_{n-1}^x\right)\right) - \theta_n
\end{cases}
\tag{8}
$$

### 3.1. Application of the bisection method

The ultrasonic ray is propagated through all material layers in the bisection method by making an initial guess for the interval of the first incidence angle and then, choosing the mean value of the interval as an initial approximation of the first incidence angle. This is done by solving the subsystems consisting of two equations, $g_i(\boldsymbol{x}) = 0$ and $g_{n+i}(\boldsymbol{x}) = 0$, with $1 \le i \le (n-1)$, sequentially. The solution of the $i^{th}$ subsystem allows computing the incidence point $P_i^x$ with the $i^{th}$ material layer interface and the refraction angle at that interface $\theta_{i+1}$, which are inserted in the two equations of the next $(i+1)^{th}$ subsystem. Eventually, the resulting values of $P_{n-1}^x$ and $\theta_n$ are inserted in the last unused function, $g_n(\boldsymbol{x})$, which gives the deviation between the target point and the point reached by the ray path on the horizontal line for the target. Therefore, the deviation from the target is used to determine a new subinterval to be used for the next iteration of the method. Furthermore, given that a new approximation of the ultrasonic path becomes available at the end of the calculations of each iteration of the algorithm, it is possible to choose suitable criteria to stop the iterations. For example, one could use the aforementioned deviation, given by the function $g_n(\boldsymbol{x})$, to stop the iterations as soon as such deviation result is lower than a set maximum acceptable value. However, this is not always the most efficient approach as it will be explained in Section 3.3.

It should be noted that, generally, the interface functions $(f_1(\boldsymbol{x}), \cdots, f_{n-1}(\boldsymbol{x}))$ are not linear. This may lead to nonlinear subsystems, which also require approximation methods to be solved. Therefore, there must be another approximation algorithm nested within the primary bisection algorithm. To assess the performance of different approaches, both the bisection and the Newton-Raphson (N-R) methods were used as the secondary approximation algorithm in this work. In the first case, the ray-tracing problem is herein said to be solved through a pure "bisection method". In the latter case, the approximation algorithm is referred to as "bisection with nested N-R" in the text.

### 3.2. Application of the Newton-Raphson method

Assuming to have an initial guess for the vector of the unknown values, $\boldsymbol{x}_0 = \left(P_1^x, \cdots, P_{n-1}^x, \theta_1, \cdots, \theta_n\right)$, the Newton-Raphson method can be applied to approximate the solution of the system of equations given in Eq. (8). The formula in Eq. (2) is used iteratively to refine the approximation of the solution. Although it is anticipated that the method reduces the number of iterations required to reach a solution at a given precision as compared to the bisection method, $k^2$ partial derivatives of the system functions should be computed to construct the Jacobian matrix, where $k$ is the number of functions in the multivariable system. Furthermore, the method needs to calculate the inverse Jacobian matrix at every iteration, which is a computationally expensive operation to perform for large matrices.

Looking at the system in Eq. (8), an attentive reader will notice that it is possible to reduce the number of functions. Indeed, the refraction angles $(\theta_2, \cdots, \theta_n)$ are not independent variables. Writing $\theta_2$ as a function of $\theta_1$ from $g_{n+1}(\boldsymbol{x})$ and replacing it in $g_{n+2}(\boldsymbol{x})$, it is possible to extract $\theta_3$ as a function of $\theta_1$. Likewise, replacing $\theta_3$ in $g_{n+3}(\boldsymbol{x})$, it is possible to extract $\theta_4$ as a function of $\theta_1$. Finally, by progressing with replacing $\theta_i$ as a function of $\theta_1$, obtained from $g_{n+i-1}(\boldsymbol{x})$, into $g_{n+i}(\boldsymbol{x})$, all angles can be expressed as functions of the first incidence angle $\theta_1$. This is formally giving:

$$
\begin{cases}
\theta_2 = asin\left\{\left(\frac{v_2}{v_1}\right)\cdot\sin\left[\theta_1 - atan\left(f_1'\left(P_1^x\right)\right)\right]\right\} + atan\left(f_1'\left(P_1^x\right)\right) \\
\theta_3 = asin\left\{\left(\frac{v_3}{v_2}\right)\cdot\sin\left[asin\left\{\left(\frac{v_2}{v_1}\right)\cdot\sin\left[\theta_1 - atan\left(f_1'\left(P_1^x\right)\right)\right]\right\} + atan\left(f_1'\left(P_1^x\right)\right) - atan\left(f_2'\left(P_2^x\right)\right)\right]\right\} + atan\left(f_2'\left(P_2^x\right)\right) \\
\theta_4 = asin\left\{\left(\frac{v_4}{v_3}\right)\cdot\sin\left[asin\left\{\left(\frac{v_3}{v_2}\right)\cdot\sin\left[asin\left\{\left(\frac{v_2}{v_1}\right)\cdot\sin\left[\theta_1 - atan\left(f_1'\left(P_1^x\right)\right)\right]\right\} + atan\left(f_1'\left(P_1^x\right)\right) - atan\left(f_2'\left(P_2^x\right)\right)\right]\right\} + atan\left(f_2'\left(P_2^x\right)\right) - atan\left(f_3'\left(P_3^x\right)\right)\right]\right\} + atan\left(f_3'\left(P_3^x\right)\right) \\
\cdots \\
\theta_n = \cdots
\end{cases}
\tag{9}
$$

Replacing these angles, expressed as functions of $\theta_1$, in the system given in Eq. (8), it is possible to reduce the number of nonlinear equations from *2n-1* to *n*, with *n* being also the number of independent unknown values:

$$\begin{cases} g_1\left(P_1^x, \cdots, P_{n-1}^x, \theta_1\right) = S^x - P_1^x + \left[S^y - f_1\left(P_1^x\right)\right]\cdot\tan(\theta_1) \\ g_2\left(P_1^x, \cdots, P_{n-1}^x, \theta_1\right) = P_1^x - P_2^x + \left[f_1\left(P_1^x\right) - f_2\left(P_2^x\right)\right]\cdot\tan\left(asin\left\{\left(\frac{v_2}{v_1}\right)\cdot\sin\left[\theta_1 - atan\left(f_1^{'}\left(P_1^x\right)\right)\right]\right\} + atan\left(f_1^{'}\left(P_1^x\right)\right)\right) \\ \cdots \\ g_i\left(P_1^x, \cdots, P_{n-1}^x, \theta_1\right) = \cdots \\ \cdots \\ g_{n-1}\left(P_1^x, \cdots, P_{n-1}^x, \theta_1\right) = \cdots \\ g_n\left(P_1^x, \cdots, P_{n-1}^x, \theta_1\right) = \cdots \end{cases} \tag{10}$$

This minimises the size of the Jacobian matrix. Nevertheless, replacing the refraction angles with their mathematical dependence of the first incidence angle in the functions from $g_2(\boldsymbol{x})$ to $g_n(\boldsymbol{x})$, increases the length and complexity of those functions and, hence, can neutralize the advantage in terms of computation time. This work quantifies the performance of the different approaches, by testing the Newton-Raphson algorithm with the full system of equations (herein referred to as "N-R method") and the algorithm with the reduced system ("Reduced N-R method").

### 3.3. Stopping criteria

Specific criteria for halting the iterative algorithms should be defined, to stop the iteration of the calculus. The traditional approach to terminate an iterative root-finding algorithm is to monitor the difference between two consecutive approximations of the root and to stop the iteration as soon as the modulus of the difference becomes smaller than a threshold value. This could be easily done for the bisection method and for the Newton-based method, when the objective function has a single variable. Calculating the difference between two vectors, for multi-variable objective functions, is not straightforward. Indeed, the physical meaning of the difference of the two vectors is not always clear. For example, this is the case for vectors comprised of variables of different types (e.g. lengths and angles). As it was mentioned in Section 3.1, the deviation of the point reached by the ray path on the horizontal line for the target, from the target point, which becomes available at the end of each iteration, can be used to determine the input subinterval for the next iteration of the bisection method. This deviation is also readily computable at the end of each iteration of the Newton-Raphson method. Furthermore, it is a direct measure of how far the approximated ray path solution is from crossing the target point, its physical meaning is well understood and it is always expressed in a unit of length (e.g. millimetres, metres, etc). However, using such deviation as a stopping condition is not the only practicable approach. It must be considered that mechanical waves are typically generated through the use of physical probes. In this work, it has been decided to select the stopping criteria among parameters that can be easily controlled in practice. An inspector can only position a probe up to a certain level of accuracy either by using naked eyes or by seeking help from devices that augment human senses (e.g. callipers, goniometers, laser meters, etc). Similarly, a mechanical manipulator (e.g. a gantry system or an industrial robotic arm) can only achieve a certain positioning accuracy depending on its linear and/or angular resolution (stated by the manufacturer). Therefore, it is sensible to terminate the iteration of the root-finding algorithms as soon as the difference between one or more measurable parameters becomes smaller than what a human or a machine can achieve, in terms of positioning resolution. Indeed, further iterations would not translate into any practical benefit. Noteworthy, mechanical wave rays can also be generated using phased array probes [2]. They are an advanced method of transmitting/receiving ultrasound waves with a broad range of applications in medical imaging and industrial non-destructive testing. The beam from a phased array probe can be focused and swept electronically without moving the probe, whereas single-element (non-phased array) probes, known technically as monolithic probes, only can emit a beam in a fixed direction and must be physically moved to sweep the beam through a large area of interest. The beam direction

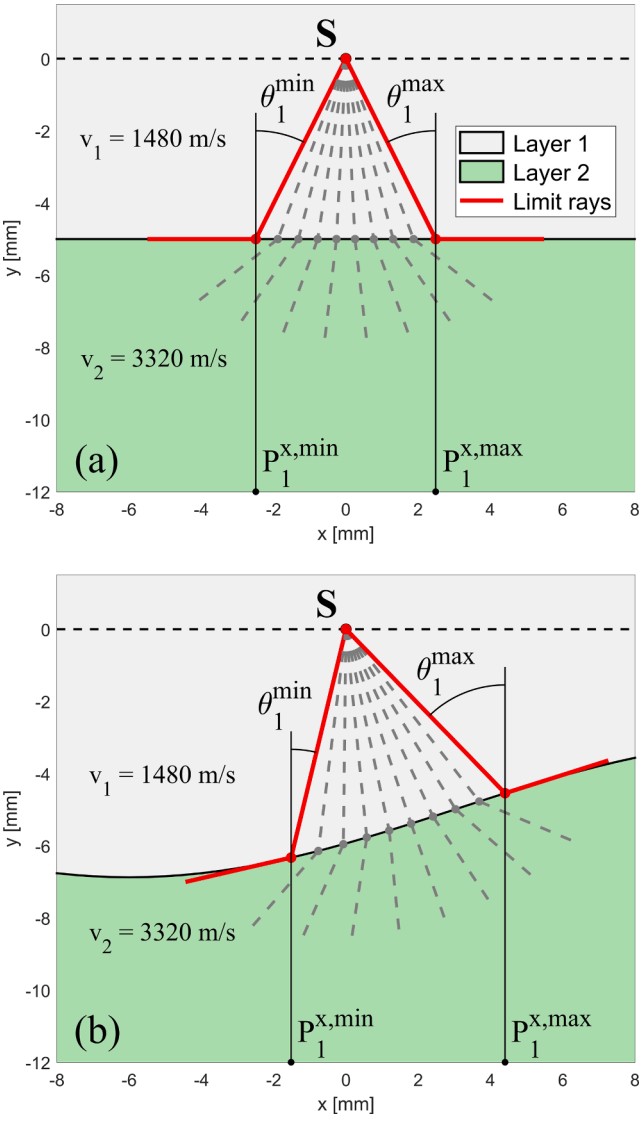



**Fig. 2.** Limit incidence points and angles for an example with a flat interface (a) and with a curved interface (b).

and focus are controllable because a phased array probe is made up of multiple small elements, each of which can be pulsed individually at a computer-calculated timing. In fact, the term "phased" refers to the transmit/receive timing, and the term "array" refers to the group of elements. Commercial phased array controllers have a limited temporal resolution, which limits the minimum time delay between consecutive pulses. The time-difference between the delays is typically of the order of nanoseconds [6]. This corresponds to quite high precision calculation for ultrasonic non-destructive testing applications, where the wave periods are most commonly at least 10 times larger, even for a very high frequency of 100 MHz. Thus, in this work, it has been decided to compute the travelling time of the approximated ray path at the end of each iteration of the iterative root-finding algorithm and stop the iterations as soon as the difference between consecutive travelling times falls below 1 ns ($\Delta t < 10^{-9} s$). In this work, it was decided to use this stringent criterion as stopping condition for all executed tests, whose results are discussed in Section 4.

### 3.4. Selection of initial guess

Both the bisection-based and the Newton-based methods require an initial guess for the root to be approximated by the iterative algorithms. The bisection-based methods also require an interval that is guaranteed to bracket the root. Therefore, it is crucial to establish a way to compute such preliminary values. The relationships between the angles, given in Eq. (9), are used to identify these preliminary values in this work. Combining those relationships together with the fact that $\mathrm{asin}(x)$ is only defined for $-1 \le x \le 1$, it is possible to define a set of inequalities that the unknown variables must satisfy:

$$\begin{cases} -1 \le \left(\frac{v_2}{v_1}\right)\cdot\sin\left[\theta_1 - atan\left(f'_1\left(P_1^x\right)\right)\right] \le 1 \\ -1 \le \left(\frac{v_3}{v_2}\right)\cdot\sin\left[asin\left\{\left(\frac{v_2}{v_1}\right)\cdot\sin\left[\theta_1 - atan\left(f'_1\left(P_1^x\right)\right)\right]\right\} + atan\left(f'_1\left(P_1^x\right)\right) - atan\left(f'_2\left(P_2^x\right)\right)\right] \le 1 \\ -1 \le \left(\frac{v_4}{v_3}\right)\cdot\sin\left[asin\left\{\left(\frac{v_3}{v_2}\right)\cdot\sin\left[asin\left\{\left(\frac{v_2}{v_1}\right)\cdot\sin\left[\theta_1 - atan\left(f'_1\left(P_1^x\right)\right)\right]\right\} + atan\left(f'_1\left(P_1^x\right)\right) - atan\left(f'_2\left(P_2^x\right)\right)\right]\right\} + atan\left(f'_2\left(P_2^x\right)\right) - atan\left(f'_3\left(P_3^x\right)\right)\right] \le 1 \\ \cdots \\ -1 \le \left(\frac{v_n}{v_{n-1}}\right)\cdot\sin\left[asin\left\{\left(\frac{v_{n-1}}{v_{n-2}}\right)\cdots\right\} + atan\left(f'_{n-2}\left(P_{n-2}^x\right)\right) - atan\left(f'_{n-1}\left(P_{n-1}^x\right)\right)\right] \le 1 \end{cases} \tag{11}$$

Evidently, extrapolating the domain of validity for all unknown variables from this set of nonlinear inequalities is not an easy task. For this reason, the search of the domain of validity was restricted to the incidence angle ($\theta_1$), for which obtaining an approximation of the domain interval from the first inequality in the set of Eq. (11) can be simply calculated and written as:

$$\begin{cases} \left[\theta_1^{min}, \theta_1^{max}\right] = \left[atan\left(f'_1\left(P_1^x\right)\right) - asin\left(\frac{v_1}{v_2}\right),\ atan\left(f'_1\left(P_1^x\right)\right) + asin\left(\frac{v_1}{v_2}\right)\right], & \text{for: } v_2 > v_1 \\ \left[\theta_1^{min}, \theta_1^{max}\right] = \left[atan\left(f'_1\left(P_1^x\right)\right) - \frac{\pi}{2},\ atan\left(f'_1\left(P_1^x\right)\right) + \frac{\pi}{2}\right], & \text{for: } v_2 \le v_1. \end{cases} \tag{12}$$

It is clear that the extremities of the interval depend on the incidence point at the first interface, since the derivative of $f_1(x)$ is computed for $x = P_1^x$ in Eq. (12). In [19], the intersection between the segment $\overline{ST}$ and the first interface $f_1(x)$ was used to guide an initial guess for $P_1^x$. However, this approach had the disadvantage of leading to a local interval of validity for $\theta_1$, which was only accurate for the tentative value of $P_1^x$. Therefore, the present work proposes a different approach for a more robust definition of the interval of validity. This originates from the fact that, besides the formulas in Eq. (12), the formula in Eq. (13) is also valid, as the $P_1$ is connected with a straight line to S.

$$\theta_1 = atan\left(\frac{P_1^x - S^x}{S^y - f_1\left(P_1^x\right)}\right) \tag{13}$$

Hence, putting together Eq. (12) and Eq. (13), it is possible to write two equations with a single unknown value ($P_1^{x,min}$ or $P_1^{x,max}$). Such unknown values mark the extremities of the interval of validity for the incidence point of the ultrasonic ray emitted from S (the source point) with the first interface.

$$\text{for } (v_2 > v_1): \begin{cases} atan\left(\frac{P_1^{x,min} - S^x}{S^y - f_1\left(P_1^{x,min}\right)}\right) = atan\left(f'_1\left(P_1^{x,min}\right)\right) - asin\left(\frac{v_1}{v_2}\right) \\ atan\left(\frac{P_1^{x,max} - S^x}{S^y - f_1\left(P_1^{x,max}\right)}\right) = atan\left(f'_1\left(P_1^{x,max}\right)\right) + asin\left(\frac{v_1}{v_2}\right) \end{cases}$$

$$\text{for } (v_2 \le v_1): \begin{cases} atan\left(\frac{P_1^{x,min} - S^x}{S^y - f_1\left(P_1^{x,min}\right)}\right) = atan\left(f'_1\left(P_1^{x,min}\right)\right) - \frac{\pi}{2} \\ atan\left(\frac{P_1^{x,max} - S^x}{S^y - f_1\left(P_1^{x,max}\right)}\right) = atan\left(f'_1\left(P_1^{x,max}\right)\right) + \frac{\pi}{2} \end{cases} \tag{14}$$

Again, since these are non-linear equations, their roots can be approximated using the iterative bisection method. Therefore, once the approximated values of $P_1^{x,min}$ and $P_1^{x,max}$ are available, the relative extremities of the interval of validity of $\theta_1$ can be computed through Eq. (13) as: $\theta_1^{min} = atan\left(\frac{P_1^{x,min} - S^x}{S^y - f_1\left(P_1^{x,min}\right)}\right)$ and $\theta_1^{max} = atan\left(\frac{P_1^{x,max} - S^x}{S^y - f_1\left(P_1^{x,max}\right)}\right)$. The geometric significance of this is illustrated in Fig. 2, where the limit incidence points and angles are illustrated, for two examples: with a flat interface (Fig. 2a) and with a curved interface (Fig. 2b). It is clear that the effective domain interval for $\theta_1$, which would originate from the simultaneous solution of the inequalities in the system given in Eq. (11),

may be smaller than the interval $[\theta_1^{min}, \theta_1^{max}]$, since additional constraints would be applied to $\theta_1$. The approach adopted in this work, to refine the interval of validity, is based on a modified version of the iterative bisection method. It must be noted that the need for defining the initial guess (e.g. a search interval for the bisection method and an initial guess vector for the Newton-based methods) is the main drawback of all iterative root-finding algorithms. The approach proposed in this work is deemed to be robust when the layer interfaces are continuous and there exists an interval ($[a, b]$), where the deviation from the target point is always a real number. This is the assumption behind the use of the herein described approach. When propagating a ray through the bottom limit with the first incidence angle $\theta_1 = \theta_1^{min}$, the final deviation from the target point can be checked to see whether a real number or an imaginary number is obtained. If the former is the case, $\theta_1^{min}$ is confirmed to be the true bottom limit. Otherwise, the bottom limit should be found somewhere in between $\theta_1^{min}$ and $\theta_1^{max}$, truncating the interval of validity. The same logic applies for identification of the top limit. If the ray propagated by the first incidence angle, $\theta_1 = \theta_1^{max}$, generates an imaginary number as the final deviation from the target point, the true top limit is to be sought between $\theta_1^{min}$ and $\theta_1^{max}$.

As described in Section 2.1, the standard bisection method applies to any continuous functions for which one knows two values with opposite signs. However, it is easily adaptable to the situation where an objective function produces a real number at one end and an imaginary number at the other end of an interval. In this case, the method is used twice (for the bottom limit and the top limit) to repeatedly bisect the interval $[\theta_1^{min}, \theta_1^{max}]$ and to obtain the subinterval $[a, b]$, with $\theta_1^{min} \leq a \leq b \leq \theta_1^{max}$. The extremities of this subinterval are a numerical approximation of the true limits of validity for $\theta_1$ and are found by iterating the bisection of the original interval, $[\theta_1^{min}, \theta_1^{max}]$, at least K times, with K being defined upfront according to the width of the initial interval, to guarantee a given approximation accuracy. For example, if $(\theta_1^{max} - \theta_1^{min}) = \pi$ radians and it is desirable to guarantee an accuracy of circa $1.19^{-7}$ radians (e.g. the distance from 1.0 to the next larger single-precision floating-point number, according to the IEEE 754 standard [23]), the minimum number of iterations will be $K = \lceil log_2(\pi/1.19^{-7}) \rceil = 25$. Once the proper interval of validity ($[a, b]$) is approximated, the bisection method starts from the initial guess of the first incidence angle, equal to: $\theta_1^0 = \frac{a+b}{2}$. In order to compute the initial guess vector for the Newton-based methods ($\boldsymbol{x}^0$), the incidence ray striking the first interface with incidence angle $\theta_1^0$ is propagated through all layers, computing all subsequent incidence points and refraction angles.

**Table 1**
Number of iterations to converge to the solution.

| Method | $n$ = 2 | $n$ = 4 | $n$ = 8 | $n$ = 16 | $n$ = 32 | $n$ = 64 |
|---|---|---|---|---|---|---|
| **Bisection** | 9 | 10 | 11 | 18 | 13 | 14 |
| **Bisection with nested N-R** | 9 | 10 | 11 | 10 | 11 | 14 |
| **N-R** | 3 | NC* | NC | NC | NC | 5 |
| **Reduced N-R** | 3 | NC | NC | NP** | NP | NP |

* Non-Convergent.
** Not-Possible.

## 4. Implementation and results

The algorithms described in the previous section were implemented in MATLAB. Fig. 3 illustrates the schematic workflow for the solution of the ray-tracing problem, comprising all options investigated in this work.

The links between the blocks of the diagram represent the sequence of the operations and the exchange of information. The process starts

**Table 2**
Number of iterations to reach convergence in hybrid methods.

| Method | Type of iteration | $n$ = 2 | $n$ = 4 | $n$ = 8 | $n$ = 16 | $n$ = 32 | $n$ = 64 |
|---|---|---|---|---|---|---|---|
| **Hybrid N-R** | Bisection | 0 | 2 | 2 | 2 | 1 | 0 |
| | N-R | 3 | 4 | 4 | 4 | 4 | 5 |
| | Total | 3 | 6 | 6 | 6 | 5 | 5 |
| **Hybrid Reduced N-R** | Bisection | 0 | 2 | 2 | NP | NP | NP |
| | N-R | 3 | 5 | 4 | NP | NP | NP |
| | Total | 3 | 7 | 6 | NP | NP | NP |

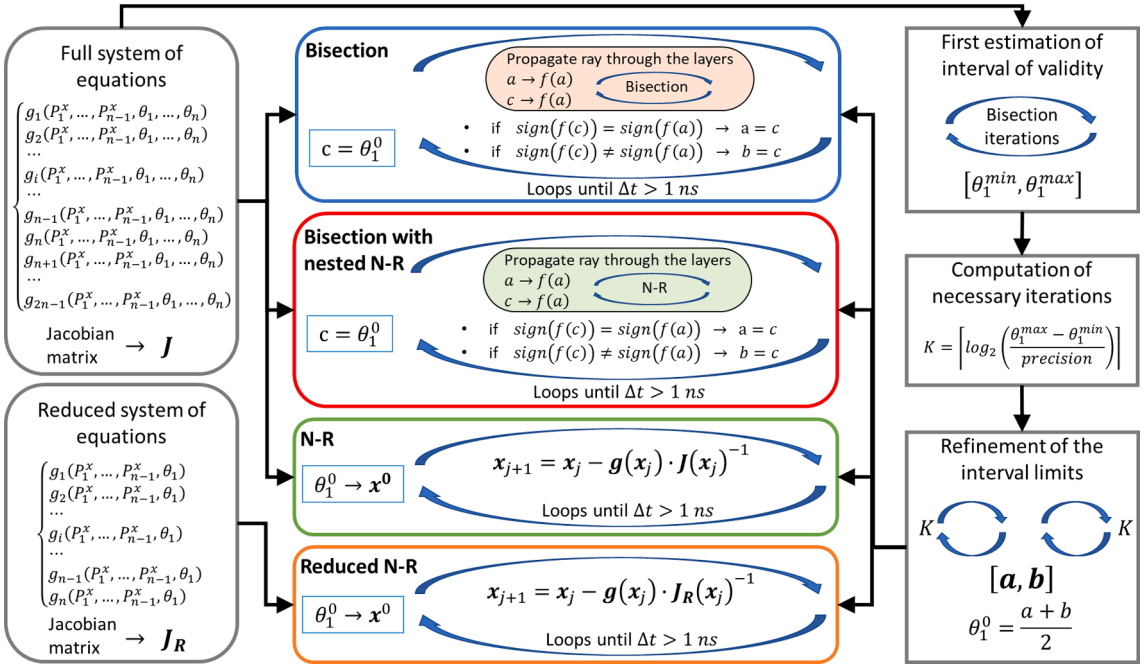



**Fig. 3.** Ray tracing workflow, with all investigated options.

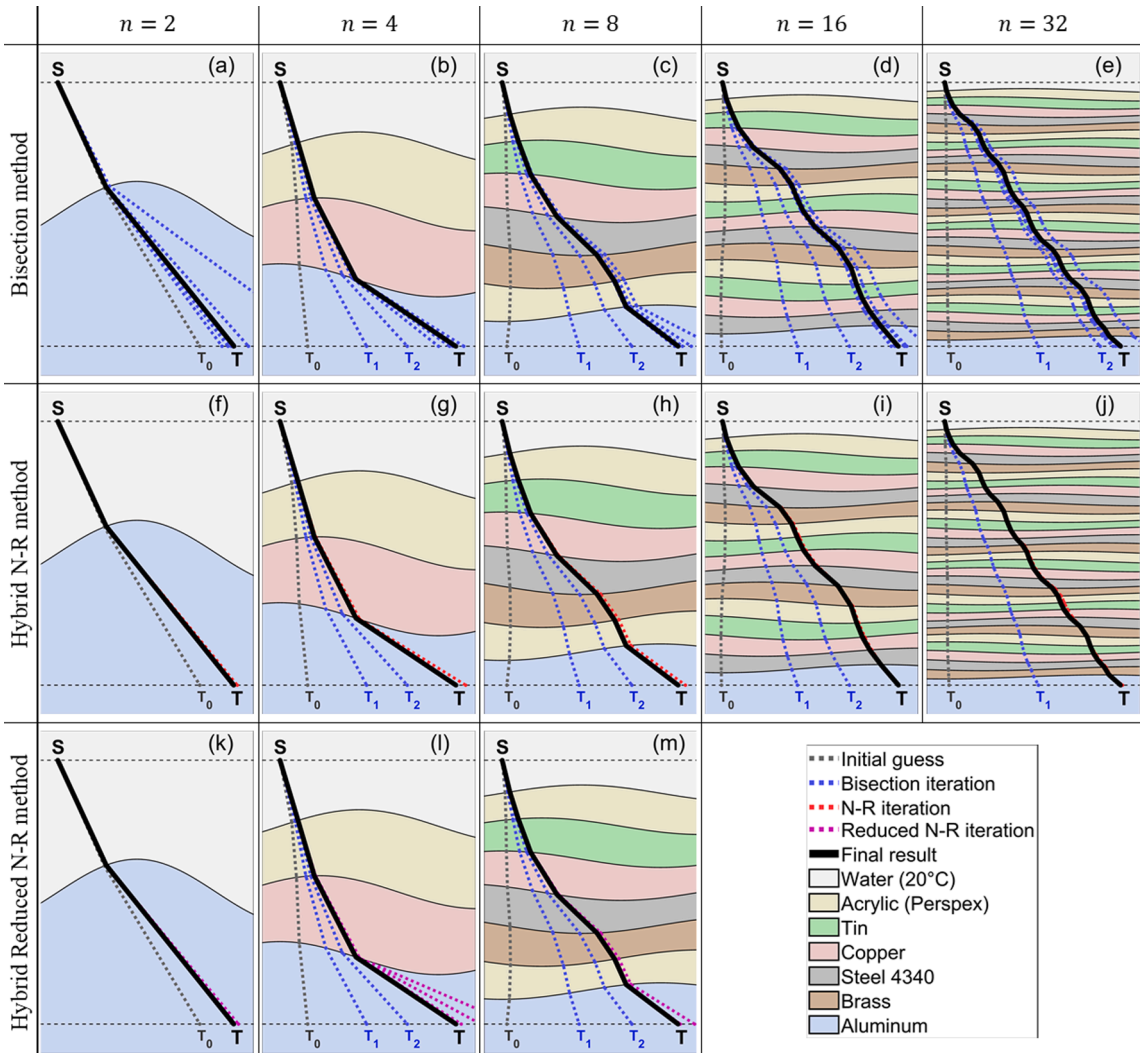



**Fig. 4.** Resulting ray paths obtained with bisection (a-e), hybrid N-R (f-j) and hybrid reduced N-R (k-m) for: $n = 2$, $n = 4$, $n = 8$, $n = 16$ and $n = 32$.

with the preparation of the symbolic system of equations and the relative symbolic Jacobian matrix ($J$), shown on the left-hand side of the diagram. Then, the full system of equations is used by the blocks on the right-hand side of the diagram, which are related to the definition of the interval of validity of the first incidence angle and the initial guess. The full system of equations, the interval of validity and the initial guess are all that is needed to solve the ray-tracing problem through the bisection method, the bisection with nested N-R or the N-R method. The Reduced N-R method requires also the preparation of the reduced system and its Jacobian matrix ($J_R$). The MATLAB-based toolbox developed in this work is openly accessible at: https://doi.org/10.5281/zenodo.5026763.

The execution of all methods was tested through MATLAB 2020b, running in a computer with an Intel® i7-6820HQ CPU (2.70 GHz, 4 Cores) and 32 Gb of Random-Access Memory. The tests aimed at measuring the computation time of each method. It was decided to apply the bisection method, the bisection with nested N-R, the N-R method and the Reduced N-R to solve the ray-tracing problem for an increasing number of material layers ($n = 2, n = 4, n = 8, n = 16, n = 32$ and $n = 64$). Source (S) and target (T) were kept at constant positions: (0, 0) and (20, −30), respectively. The function $f_i(P_i^x)$ of the $i^{th}$ interface was set equal to the following trigonometric function:

$$f_i(P_i^x) = a \cdot \sin\left[10 \cdot P_i^x + (i-1) \cdot \frac{\pi}{4}\right] - i \cdot \left(\frac{30}{n}\right) \tag{15}$$

with the amplitude ($a$) of the sinusoidal part being set equal to $a = 30/4n$. The first layer material was always set to water (at 20℃), with a speed of sound of $v_1 = 1480 m/s$. The last layer was always set to aluminium ($v_n = 6320 m/s$). In the case with four layers, the second layer was set to acrylic (Perspex material, $v = 2730 m/s$) and the third

**Table 3**
Average time taken for the preparation of the symbolic system of equations relative to the full system and the reduced system (times are given in milliseconds).

| Method | $n = 2$ | $n = 4$ | $n = 8$ | $n = 16$ | $n = 32$ | $n = 64$ |
|---|---|---|---|---|---|---|
| **Full system** | 401 | 1510 | 6063 | 24,321 | 97,280 | 401,571 |
| **Reduced system** | 295 | 1057 | 58,125 | NP | NP | NP |

**Table 4**
Average execution time of each iteration for every method (times are given in milliseconds).

| Method | $n = 2$ | $n = 4$ | $n = 8$ | $n = 16$ | $n = 32$ | $n = 64$ |
|---|---|---|---|---|---|---|
| **Bisection** | 134 | 429 | 1066 | 2350 | 4785 | 9691 |
| **Bisection with nested N-R** | 165 | 515 | 1259 | 2753 | 5640 | 11,548 |
| **N-R** | 0.052 | 0.063 | 0.104 | 0.206 | 0.419 | 0.970 |
| **Reduced N-R** | 0.044 | 0.100 | 0.643 | NP | NP | NP |

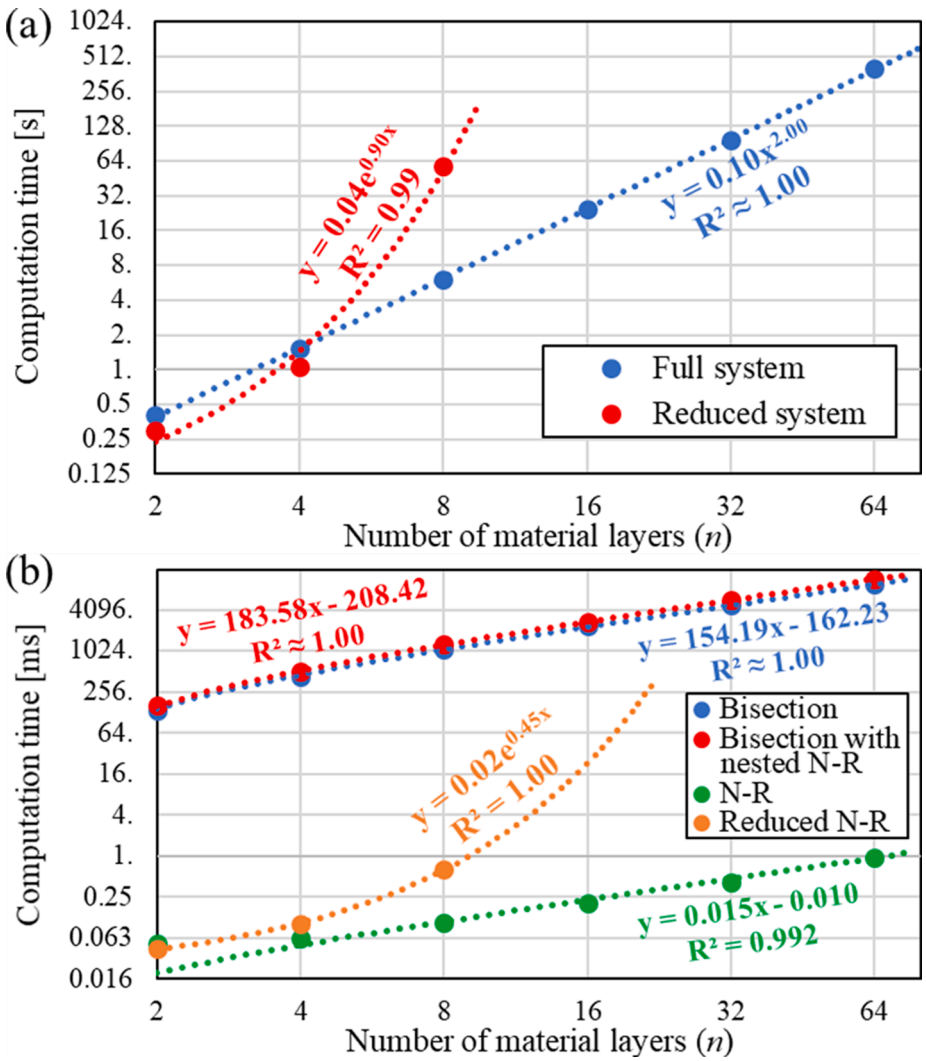



**Fig. 5.** Computation time relative to the preparation of the symbolic system of equations (a) and computation time of each iteration for every method (b).

layer was set to copper ($v = 4660m/s$). For the cases with more than four layers, other three material types are used: tin ($v = 3320m/s$), steel 4340 ($v = 5850m/s$) and brass ($v = 4430m/s$). Between the first and the last layer, the sequence of acrylic, tin, copper, steel 4340 and brass was repeated the necessary number of times to fill all layer positions. Table 1 reports the number of iterations necessary for the proposed methods to converge to the solution, with the defined stopping criterion ($\Delta t > 10^{-9}s$). As it was expected, the number of iterations for the bisection method and the bisection with nested N-R are similar. They are equal for $n = 2$, $n = 4$, $n = 8$ and $n = 64$. Interestingly, the bisection with nested N-R converges with 8 fewer iterations for $n = 16$ and with 2 less iterations for $n = 32$. This is thought to be the result of a more accurate estimation of the intersection points between the ray path and the interfaces, produced by the nested Newton-based root-finding algorithm, which reduced the propagation of numerical errors in the computation.

The N-R method reaches convergence with fewer iterations than the bisection-based algorithms. It solved the ray tracing problem with just 3 iterations for $n = 2$ and 5 iterations for $n = 64$. However, the main drawback of the Newton-Raphson method is that it is not guaranteed to converge, particularly when the initial guess is not sufficiently close to the root of the problem. In fact, the N-R method did not converge for $n = 4$, $n = 8$, $n = 16$ and $n = 32$. Table 1 reports the acronym "NC" for "Non-Convergent". The Reduced N-R method also reached convergence with 3 iterations for $n = 2$. However, beside not converging for $n = 4$ and $n = 8$, it was not possible to run the method for $n = 16$, $n = 32$ and $n = 64$. Table 1 reports the acronym "NP" for "Not-Possible". The inability to run these tests was due to a limit related to symbolic computation in MATLAB 2020b. Due to the replacement of the refraction angles with their mathematical dependence from the first incidence angle, into the functions from $g_2(x)$ to $g_n(x)$, the implementation of Reduced N-R method leads to very long functions that reach the limits of MALTAB 2020b, which is only able to manage symbolic functions up to a certain length. Such length is exceeded for $n = 16$, $n = 32$ and $n = 64$.

In order to bypass the non-convergence state of the Newton-based methods, this work tested the combination of them with the bisection method, resulting in a "hybrid N-R method" and in a "hybrid reduced N-R method". This is simply achieved by attempting to use the Newton-based computation and checking the values contained in the Jacobian matrix and the new estimation of the root at each iteration. If these values contain imaginary numbers, the root estimation is replaced with that calculated through the bisection-based computation. Therefore, the values of the Jacobian matrix and the estimated root at each iteration are the basis for deciding whether the iteration should be carried out with the Newton-based or the bisection-based computation. Table 2 reports the number of iterations that were necessary to reach convergence for hybrid methods.

It is clear that the combination of N-R and Reduced N-R methods with the bisection-type iterations allows the hybrid methods to overcome the non-convergence problem. It was observed that a certain

number of bisection-type iterations are used at first to get sufficiently close to the root. Then, Newton-type iterations are used to refine the approximation of the root until meeting the stopping criteria.

Fig. 4 illustrates the ray paths computed with bisection, hybrid N-R and hybrid reduced N-R method for: $n = 2$, $n = 4$, $n = 8$, $n = 16$ and $n = 32$. It can be noted how all methods started solving the ray tracing problem from the same initial guess, which was calculated through the approach described in Section 3.4. Whereas the intersection point between the initial guess path and the horizontal line for the target ($T$) is marked with $T_0$, the intersection point corresponding to the approximation of the $i^{th}$ iteration is marked with $T_i$ (only if sufficiently far from the target point, for the sake of graphical clarity).

Crucially, beside noting the number of iterations necessary to reach convergence, this work has also accurately measured the time taken for the preparation of the symbolic system of equations relative to the full system and the reduced system (see Table 3) and the execution time of each iteration for every method (see Table 4).

The quantitative results reported in Table 3 and Table 4 are illustrated in graphical form in Fig. 5a and Fig. 5b, respectively. The diagrams are presented with a logarithmic scale for the horizontal and vertical axis. It can be noted how the preparation time is typically three orders of magnitude larger than the time taken by the single iterations. The measured points have been fitted with analytical curves to fully investigate the dependence of the computation times from the complexity of the problem (the number of material layers). $R^2$ (R-squared) quantifies the goodness of the fitted curves, and it is close to 1 when there is a good correlation between the actual points and the points predicted by the fit, which is the case for all the fits presented in the diagrams. Therefore, according to Fig. 5a, it is possible to conclude that the preparation time of the full system has a linear dependence from the number of layers (it increases proportionally). On the other hand, the preparation time of the reduced system appears to increase exponentially with respect to the number of material layers. As suggested by Fig. 5b plot, whereas the average computation time of each iteration increases linearly with the number of layers for the bisection, the bisection with nested N-R and the N-R method, the trend is exponential for the Reduced N-R method. The bisection with nested N-R does not provide any advantage in terms of iteration time, compared to the standard bisection method. The computation times of the iterations of the N-R method are always at least three orders of magnitude smaller than the computation times of the bisection-based methods. Finally, solving the ray-tracing problem with the Reduced N-R method may only be convenient for a number of material layers smaller than 4, which does not make it particularly attractive for complex ray-tracing problems.

## 5. Conclusions

Ultrasound waves have been used for inspecting components, medical diagnosis and analysing materials for many decades. The ultrasound wave propagation problems could become complicated when the wave impinges an interface between two adjacent different media at an oblique angle, which gives rise to wave refraction at that boundary. The refraction phenomenon at a boundary is normally described through Snell's law and causes the wave propagation path to deflect from the straight travelling line that it follows in a uniform and monolithic material domain. Accordingly, the change in the wave path across a boundary or several boundaries should be computed and the wave path should be traced to be able to account for it in the interpretation of data obtained from ultrasonic inspections. To this end, the present work investigated approaches based on iterative root-finding algorithms, suitable for solving the ray-tracing problem in multi-layered structures. For this purpose, the work introduced a new pivotal formulation for the definition of the initial guess interval (for bisection-based methods) and of the initial guess of the root (for Newton-based methods) and a rationale for the choice of suitable criteria to stop the iteration of the calculus. This novel approach used for making the best initial guess has proved to be a key step in reaching a solution using any of the ray-tracing methods investigated here. The presented findings are relevant for areas beyond material inspection and have the potential to be deployed within applications involving any kind of wave undergoing refraction. The performance and the limitations of all usable algorithms, which emerged through this work, were investigated quantitatively and rigorously. A series of models for multilayer domains were designed with an increasing number of material layers (from 2 to 64). Then, bisection-based and Newton-based methods were applied to solve the ray-tracing problem. The computation times were benchmarked through a computer with a known hardware configuration. It was found that the iterations of the Newton-Raphson method are always at least three orders of magnitude faster than the iterations of the bisection-based methods, which allow for exceptionally faster computations. Nevertheless, the Newton-based methods are not guaranteed to converge, if the initial guess is not sufficiently close to the root of the problem. This issue was also addressed by combining the Newton-Raphson methods with bisection-type iterations, to form hybrid iterative root-finding methods. The MATLAB-based implementation of all investigated methods is made publicly available, at https://doi.org/10.5281/zenodo.5026763, and can be used by the research community for future developments.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 835846.

## References

[1] L. Mordfin, Handbook of reference data for nondestructive testing, ASTM (2002).
[2] B.W. Drinkwater, P.D. Wilcox, Ultrasonic arrays for non-destructive evaluation: A review, NDT and E Int. 39 (7) (2006) 525–541.
[3] L. Espinosa, F. Prieto, L. Brancheriau, P. Lasaygues, Effect of wood anisotropy in ultrasonic wave propagation: A ray-tracing approach, Ultrasonics 91 (2019) 242–251.
[4] H. Du, Characterization of microstructural anisotropy using the mode-converted ultrasonic scattering in titanium alloy, Ultrasonics 119 (2022), 106633.
[5] F. Honarvar, A. Varvani-Farahani, A review of ultrasonic testing applications in additive manufacturing: Defect evaluation, material characterization, and process control, Ultrasonics 108 (2020), 106227.
[6] B.W. Drinkwater, A.I. Bowler, Ultrasonic array inspection of the Clifton Suspension Bridge chain-links, Insight-Non-Destructive Testing Condition Monitor. 51 (9) (2009) 491–498.
[7] J. Russell, R. Long, D. Duxbury, P. Cawley, Development and implementation of a membrane-coupled conformable array transducer for use in the nuclear industry, Insight-Non-Destructive Testing Condition Monitor. 54 (7) (2012) 386–393.
[8] C. Mineo, S.G. Pierce, B. Wright, I. Cooper, P.I. Nicholson, PAUT inspection of complex-shaped composite materials through six DOFs robotic manipulators, Insight-Non-Destructive Testing Condition Monitor. 57 (3) (2015) 161–166.
[9] A.J. Hunter, B.W. Drinkwater, P.D. Wilcox, Autofocusing ultrasonic imagery for non-destructive testing and evaluation of specimens with complicated geometries, NDT and E Int. 43 (2) (2010) 78–85.
[10] S. Mahaut, O. Roy, C. Beroni, B. Rotter, Development of phased array techniques to improve characterization of defect located in a component of complex geometry, Ultrasonics 40 (1-8) (2002) 165–169.
[11] J. Zhang, B.W. Drinkwater, P.D. Wilcox, A.J. Hunter, Defect detection using ultrasonic arrays: The multi-mode total focusing method, NDT E Int. 43 (2) (2010) 123–133.
[12] J. Zhang, B.W. Drinkwater, P.D. Wilcox, Efficient immersion imaging of components with nonplanar surfaces, IEEE Trans. Ultrason. Ferroelectr. Freq. Control 61 (8) (2014) 1284–1295.
[13] M. Sutcliffe, M. Weston, P. Charlton, K. Donne, B. Wright, I. Cooper, Full matrix capture with time-efficient auto-focusing of unknown geometry through dual-layered media, Insight-Non-Destructive Testing Condition Monitor. 55 (6) (2013) 297–301.

[14] J. Ye et al., Development of an ultrasonic ray model for phased array ultrasonic testing in austenitic weldments, in: 17th World Conference on Non-Destructive Testing, Shanghai, 2008, pp. 25-28.
[15] J.A. Ogilvy, A layered media model for ray propagation in anisotropic inhomogeneous materials, Appl. Math. Model. 14 (5) (1990) 237–247.
[16] B. Hawker, S. Burch, A. Rogerson, Application of the RAYTRAIM ray tracing program and RAYKIRCH post-processing program to the ultrasonic inspection of austenitic stainless-steel welds in nuclear power plant, in: Proc. 1st Int. Conf. NDE in Relation to Struct. Integrity for Nuclear and Pressure Components, 1998, vol. 2, pp. 899-911.
[17] O. Nowers, D.J. Duxbury, J. Zhang, B.W. Drinkwater, Novel ray-tracing algorithms in NDE: Application of Dijkstra and A* algorithms to the inspection of an anisotropic weld, NDT E Int. 61 (2014) 58–66.
[18] K.M.M. Tant, E. Galetti, A.J. Mulholland, A. Curtis, A. Gachagan, Effective grain orientation mapping of complex and locally anisotropic media for improved imaging in ultrasonic non-destructive testing, Inverse Prob. Sci. Eng. 28 (12) (2020) 1694–1718.
[19] C. Mineo, D. Lines, D. Cerniglia, Generalised bisection method for optimum ultrasonic ray tracing and focusing in multi-layered structures, Ultrasonics 111 (2021), 106330.
[20] R.L. Burden, J.D. Faires, 2.1 The bisection algorithm, Num. Anal., 1985.
[21] T.J. Ypma, Historical development of the Newton-Raphson method, SIAM Rev. 37 (4) (1995) 531–551.
[22] J.-F. Bonnans, J.C. Gilbert, C. Lemaréchal, C.A. Sagastizábal, Numerical optimization: theoretical and practical aspects. Springer Science & Business Media, 2006.
[23] B. Mathis, J. Stine, A Novel Single/Double Precision Normalized IEEE 754 Floating-Point Adder/Subtracter, in: 2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2019: IEEE, pp. 278-283.