

PAPER • OPEN ACCESS

# Refraction-corrected ray-based inversion for three-dimensional ultrasound tomography of the breast

To cite this article: A Javaherian *et al* 2020 *Inverse Problems* **36** 125010

View the [article online](#) for updates and enhancements.



**IOP | ebooks™**

Bringing together innovative digital publishing with leading authors from the global scientific community.

Start exploring the collection—download the first chapter of every title for free.

# Refraction-corrected ray-based inversion for three-dimensional ultrasound tomography of the breast

A Javaherian<sup>1,\*</sup> , F Lucka<sup>2,3</sup>  and B T Cox<sup>1</sup> 

<sup>1</sup> Department of Medical Physics & Biomedical Engineering, University College London, WC1E 6BT, London, United Kingdom

<sup>2</sup> Centrum Wiskunde & Informatica (CWI), 1098 XG Amsterdam, The Netherlands

<sup>3</sup> Centre for Medical Image Computing, University College London, WC1E 6BT, London, United Kingdom

E-mail: [a.javaherian@ucl.ac.uk](mailto:a.javaherian@ucl.ac.uk)

Received 22 May 2020, revised 23 September 2020

Accepted for publication 14 October 2020

Published 14 December 2020



CrossMark

## Abstract

Ultrasound tomography (UST) has seen a revival of interest in the past decade, especially for breast imaging, due to improvements in both ultrasound and computing hardware. In particular, three-dimensional UST, a fully tomographic method in which the medium to be imaged is surrounded by ultrasound transducers, has become feasible. This has led to renewed attention on UST image reconstruction algorithms. In this paper, a comprehensive derivation and study of a robust framework for large-scale bent-ray UST in 3D for a hemispherical detector array is presented. Two ray-tracing approaches are derived and compared. More significantly, the problem of linking the rays between emitters and receivers, which is challenging in 3D due to the high number of degrees of freedom for the trajectory of rays, is analysed both as a minimisation and as a root-finding problem. The ray-linking problem is parameterised for a convex detection surface and two robust, accurate, and efficient derivative-free ray-linking algorithms are formulated and demonstrated and compared with a Jacobian-based benchmark approach. To stabilise these methods, novel adaptive-smoothing approaches are proposed that control the conditioning of the update matrices to ensure accurate linking. The nonlinear UST problem of estimating the sound speed was recast as a series of linearised subproblems,

\* Author to whom any correspondence should be addressed.



Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

each solved using the above algorithms and within a steepest descent scheme. The whole imaging algorithm was demonstrated to be robust and accurate on realistic data simulated using a full-wave acoustic model and an anatomical breast phantom, and incorporating the errors due to time-of-flight (TOF) picking that would be present with measured data. This method can be used to provide a low-artefact, quantitatively accurate, 3D sound speed maps. In addition to being useful in their own right, such 3D sound speed maps can be used to initialise full-wave inversion methods, or as an input to photoacoustic tomography reconstructions.

**Keywords:** ultrasound tomography, ray tracing, ray linking, breast imaging, time-of-flight, refraction

(Some figures may appear in colour only in the online journal)

## 1. Introduction

Ultrasound tomography (UST) has received growing interest in the past decade, especially for breast imaging [18, 23, 32, 51, 67, 68]. Despite first being proposed nearly 50 years ago [19], it was not until recently that improvements in both ultrasound (US) and computing hardware have begun to be exploited to bring UST to the point where it can potentially compete with, and complement, the more established medical imaging modalities. UST is a fully tomographic method in which the medium to be imaged is surrounded by transducers (emitters and receivers) and US is sequentially transmitted from the transmitters (or from small groups) into the imaging target. The US propagates through the tissue, is scattered, refracted and attenuated, and finally detected by the array of detectors. The inverse problem is to recover images of the acoustic properties of the target. These typically include images of sound speed or acoustic attenuation, which can be quantitative, and reflection images (which are usually qualitative but related somewhat to the gradient of the acoustic impedance). The hope is that this quantitative information about the tissue properties can be used to aid diagnosis [58]. (It should be noted that UST is very different from conventional ultrasound imaging as widely-used in clinical settings, which is a backward-mode imaging modality that uses a small-area probe to give qualitative reflection images in real time.) In addition, photoacoustic tomography, which has similarly been receiving a great deal of attention in the past decade for 3D breast imaging [38, 49], depends on quantitatively accurate knowledge of the sound speed distribution. Even a low resolution (smoothed) sound speed map, if quantitatively accurate, could significantly improve 3D photoacoustic tomography reconstructions in heterogeneous media such as breast, where the image extends to deeper than superficial regions.

The majority of UST approaches have been limited to 2D cases, and thus out-of-plane effects are neglected, leading to image artefacts. This study concentrates on 3D volumetric reconstruction of the sound speed distribution from measurements made on a hemispherical surface around the imaging target.

The UST inverse problem can be tackled using full-waveform inversion approaches, such as are used in seismic applications, which are based on a minimisation of the  $L_2$ -norm of the discrepancy between the measured acoustic pressure and that from a numerical model [6, 39, 43, 44, 70]. To do this, the acoustic pressure is modelled using the wave equation for heterogeneous media, and the unknown acoustic property (here the sound speed) is iteratively updated using a computation of gradient of the  $L_2$ -norm of the discrepancy [54]. This class of methods

can provide high spatial resolution images, are flexible, in the sense that different forward models can be used, and make use of all the information in the measured data. (See, for example, [43, 44, 70] for time-domain methods and [67, 68] for frequency-domain approaches.) However, one problem with these approaches is that they are very computationally expensive, and tackling 3D imaging scenarios remains challenging [43, 44, 70]. They are also vulnerable to uncertainties in the forward modelling, e.g., in the transducer properties [25], and the solution tends to converge to the nearest local minimum instead of the global minimum of the problem. As a result, minimisation-based full-wave approaches typically require a good initial guess. There is therefore still a great deal of interest in other approaches to UST, especially methods that are less intensive computationally.

Ray-based methods, which use a high-frequency approximation to reduce the wave propagation problem to a ray-propagation problem, are still popular. Indeed, ray-based methods are commonly used to provide the initial guess for full-wave inversions [39, 44, 67, 68, 70]. This approach, initially inspired by x-ray tomography, uses measurements of the time-of-flight (TOF) of the acoustic signals across the imaging target to reconstruct the slowness distribution (the reciprocal of the sound speed) using radon-type inversion techniques [3]. Typically, two sets of data are recorded, one from a known homogeneous reference medium such as water and another from the target. The inverse problem then becomes reconstructing the difference in the slowness distributions between the target and water from the differences in the TOFs measured between every pair of emitter and receiver [15]. To solve this inverse problem, a method of relating the TOFs to the slowness is required. This is usually achieved by computing the TOFs as the integrals of the slowness along the rays that link the pairs of emitters and receivers. The inverse problem now becomes a minimisation of the norm of the discrepancy between the measured and modelled TOFs via an iterative adjustment of the slowness distribution. This is often done using a successive enforcement of a radon-type forward operator and its adjoint using the trajectories of the rays [32]. This raises the question of how the ray trajectories are computed.

Early approaches to UST used an assumption that the acoustic waves propagate along straight lines [15], thus neglecting diffraction, refraction and scattering [14]. Refraction (bending) of the rays can be incorporated by basing the calculation of each ray trajectory on Fermat's principle, which states that the energy tends to propagate along the path with minimal acoustic length [4, 14, 16, 50]. However, because rays inherently assume a high-frequency approximation, diffraction and scattering effects are still neglected [14]. An approach was recently published that applies to weak heterogeneities and broadens the concept of a ray to a sensitivity kernel that includes these phenomena to some extent [42].

One class of ray-tracing approaches is based on tracking the wavefront from an emitter across the medium to all receivers simultaneously [29]. Among them, the fast marching method (FMM) [61, 64] has received much attention for UST [2, 37]. However, because this approach is based on a calculation of the wavefronts through the entire domain, it becomes computationally intensive for the 3D UST [13], and has predominantly been used for 2D studies [24]. In addition, this approach solves the trajectory of rays on the grid points, and therefore, the accuracy is limited by the grid spacing. For example, in [2], FMM was used to compute TOFs for a 2D medium encompassed by a circular detection surface. In [37], out-of-plane refraction is accounted for by applying the FMM to a slab-like volume including the target slice aligned with the circular transducer array. An application of FMM to 3D UST was also reported in [13].

Another class of ray-tracing approaches, *two-point ray tracing*, treats the emitters and receivers as ray-emission and ray-reception points and computes only the rays between them (for all emitter–receiver pairs). The numerical methods for calculating the trajectory of a ray

that links a pair of emission and reception points can be categorised into *bending* [27, 46, 48, 52, 66, 69] and *shooting* [4, 5, 50] approaches. In the bending method, the problem is framed as a two-point boundary value problem. By fixing both end-points of the ray, its trajectory is iteratively computed until the perturbation in the path becomes smaller than a tolerance [27, 46, 48, 52, 65, 66, 69]. In [27, 66], central finite difference approaches were used for solving a second-order differential equation derived from an Euler Lagrange equation describing the trajectory of the rays (see section 3.1).

In [52], a method for ray bending based on a reduction of the second-order ray equation to a set of first-order equations was proposed, and was numerically solved using adaptively-varying finite difference methods and varying meshes. It was shown that bending methods are more efficient than shooting methods for a simple distribution of the sound speed, but they are less efficient, or may fail for more complex media [11]. In the shooting method, the ray's path is solved as a sequence of initial value problems, given the starting position and an initial direction for the ray. With the initial position of the ray fixed at the emission point, the initial direction of the ray is iteratively controlled until the end point of the ray intercepts the reception point to within a small tolerance. This is known as the *ray-linking* problem [4, 56, 57]. Shooting methods have been widely used for 2D UST imaging [4, 5, 50], but little work has been done on their extension to 3D.

Both classes of two-point ray-tracing approaches, *shooting* and *bending* methods, are non-linear inverse problems, and are subject to instability or divergence, especially for 3D scenarios [52, 62]. For example, when the sound speed is non-differentiable or causes shadow zones [52, 57], the data may no longer depend continuously on the unknown function, leading to ill-posedness in the Hadamard sense. This is easier to deal with in shooting methods, as testing the convergence of (and setting a convergence criterion for) a shooting method is straightforward, because the error functional is a measure of the distance between the end point of the ray and the reception point.

One further comparison between full-wave and ray-based approaches relates to rotating measurement systems. The computational load of full-wave inversions, or ray-based approaches based on wavefront tracking, does not change as the number of receivers increases, as it only scales with the number of emitters. While this can be an advantage, it is not necessarily so. For example, consider a fixed array with  $N$  transducers, which act both as emitters and receivers. The number of unique emitter–receiver pairs will be  $N(N - 1)/2$ . For some applications, including breast imaging, it may not be possible to fit enough transducers in the array because of their finite size, and it is necessary to rotate or translate the system to obtain sufficient data [18]. In this case, the number of effective emitters is  $N_{\text{eff}} = N_{\text{rot}}N'$ , where  $N_{\text{rot}}$  is the number of rotations, and  $N' < N$  is the number of transducers. The number of unique emitter–receiver pairs is given in the rotating case by  $N_{\text{eff}}(N' - 1)/2$ , so for the rotating system to measure as much data as the equivalent fixed system, it is necessary for  $N_{\text{eff}} > N$ . In other words, data collected using a rotating array, as opposed to an equivalent fixed array, leads to an increased computational burden during the reconstruction when using full-wave or wavefront methods. For two-point ray tracing approaches, however, it will remain the same.

In this paper, a two-point ray-tracing approach for 3D transmission UST of the breast using a hemispherical detection surface will be described. Key to this is a novel technique to solve the problem of 3D ray-linking in this geometry. To the best of our knowledge, this manuscript presents the first refraction-corrected two-point ray-based inversion approach for full-3D transmission UST of the breast. Although, here, the ray-linking approach is applied to a hemispherical detection surface, it is parameterised in a general fashion and can be straightforwardly used for any arbitrary convex detection surface. In section 2, the inverse problem

of UST is defined, based on a refraction-corrected ray-based approach for a hemi-spherical detection surface. In section 3, the ray tracing approach used is introduced, and novel efficient approaches to solving the associated inverse problem of ray-linking are derived in section 4. In section 5, all aspects of the method—ray-tracing, ray-linking, and quantitative sound speed imaging—are numerically validated. Section 6 contains a discussion and the conclusions. In the appendix A, the approach used for calculation of the TOFs (a slight modification of [33]) is presented.

## 2. Inverse problem of ultrasound tomography

In this section, the forward operator under consideration and the associated inverse problem, are introduced.

### 2.1. Forward operator

Let  $\mathbf{x} = (x^1, \dots, x^d)$  denote a spatial position in  $\mathbb{R}^d$  with  $d$  the dimension. For UST,  $d$  can be either 2 or 3, but here is restricted to  $d = 3$ .  $\Omega \subset \mathbb{R}^3$  is an open bounded set given by the hemispherical volume, centred at the origin, which is bounded by the surface  $\mathbb{S} = (\mathbb{S}_1 \cup \mathbb{S}_2) \subset \mathbb{R}^2$ , where  $\mathbb{S}_1 = \{x^3 < 0 \cap |\mathbf{x}| = R\}$  is hemispherical surface of radius  $R$ , and  $\mathbb{S}_2 = \{x^3 = 0 \cap \sqrt{x^{12} + x^{22}} \leq R\}$  is a circular plane.  $\Omega$  contains the spatially-varying part of the refractive index distribution,  $n(\mathbf{x}) = c_w/c(\mathbf{x})$ , where  $c(\mathbf{x})$  denotes the spatially varying sound speed distribution, and  $c_w$  is a scalar value representing the reference sound speed (here the sound speed in water), i.e.  $(n(\mathbf{x}) - 1) \in C_0^\infty(\Omega)$ .

**Definition 1.** A ray is defined using  $f_{(n,e,p)}(\mathbf{x}) = 0$ , where  $e \in \mathbb{S}_1$  denotes an initial point for the ray, and  $p \in \mathbb{S}$  represents the interception point of the ray with surface  $\mathbb{S}$  after traveling through the medium.

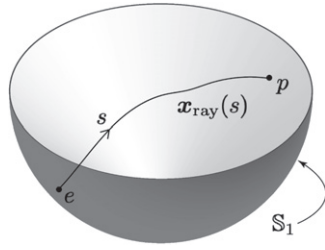
**Definition 2.** The continuous forward operator is defined as

$$\begin{aligned} \mathcal{A} : \mathbb{D}(\Omega) &\rightarrow \mathbb{S}_1 \times \mathbb{S} \\ L_{(e,p)} &= \mathcal{A}_{(n,e,p)}[n(\mathbf{x})] = \int_{\Omega} n(\mathbf{x}) \delta(f_{(n,e,p)}(\mathbf{x})) \, d\mathbf{x}, \end{aligned} \quad (1)$$

where the space  $\mathbb{D}$  is defined such that any function  $n(\mathbf{x}) \in \mathbb{D}$  satisfies the condition that  $(n(\mathbf{x}) - 1) \in C_0^\infty(\Omega)$ . Here,  $\delta$  denotes the Dirac delta function, and  $L_{(e,p)}$  is the acoustic length along a ray that links the point  $e$  to the point  $p$ , and is given by the path integral of  $n(\mathbf{x})$  along the ray  $\delta(f_{(n,e,p)}(\mathbf{x}))$ . (A ray can be parameterised by the distance along it,  $s$ , by describing it as the line of points  $\{\mathbf{x}_{\text{ray}}(s), s \in [0, L_{\text{ray}}]\}$  that satisfy  $f_{(n,e,p)}(\mathbf{x}) = 0$ , where  $L_{\text{ray}}$  is the physical length—not the acoustic length—of the ray; see figure 1.)

### 2.2. Discrete forward operator

In practice, the calculations are done on a grid of  $N_n$  discrete grid points.  $\bar{n} \in \mathbb{R}^{N_n}$  refers to the discretised refractive index field defined on the grid,  $\bar{e} \in \mathbb{S}_1$  and  $\bar{r} \in \mathbb{S}_1$  indicate the discretised emission and reception points, and  $N_e$  and  $N_r$  are the number of emitters and receivers, respectively. Note that the discretised emission and reception points do not necessarily lie on the grid points but rather correspond to the centres of actual transducer positions, as defined in the experiment. (In general, an overbar,  $\bar{\cdot}$  indicates a discretised variable.) The measurements consist of the times-of-flight  $\bar{T} \in \mathbb{R}^{N_e N_r}$  between every physical emitter and receiver.



**Figure 1.** A ray starts at the emission point  $e \in \mathbb{S}_1$  (also written as in  $\mathbf{x}_e \in \mathbb{R}^3$ ) and ends at the interception point  $p$  (or  $\mathbf{x}_p$ ). The ray consists of the points  $\mathbf{x}_{\text{ray}}(s)$ ,  $s \in [0, L_{\text{ray}}]$  that satisfy  $f_{(n,e,p)} = 0$  (see definitions 1 and 2).

**Definition 3.** A discretised variant of the forward operator can be defined as

$$\begin{aligned} A : \mathbb{R}^{N_n} &\rightarrow \mathbb{R}^{N_e N_r} \\ \bar{L} &= A[\bar{n}], \end{aligned} \quad (2)$$

where  $\bar{L} = c_w \bar{T}$  is the discretised acoustic length, and, from equation (1), the dependence of  $\bar{L}$  on  $\bar{n}$  is nonlinear. Also,  $\bar{L}_{(\bar{e}, \bar{r})}$  is used to denote the acoustic length between an emission point  $\bar{e}$  and reception point  $\bar{r}$ . (Note that a ray in the continuous domain links an arbitrary point  $e \in \mathbb{S}_1$  to an interception point  $p \in \mathbb{S}$ , and a ray in the discretised domain links a discretised emission point  $\bar{e} \in \mathbb{S}_1$  to a discretised reception point  $\bar{r} \in \mathbb{S}_1 \subset \mathbb{S}$ .) The emitters and receivers are assumed to be points in space and emit and receive rays in all directions with equal weighting.

**Definition 4.** A linearisation of  $A$  around a fixed refractive index distribution  $\bar{n}$  gives a Jacobian matrix of the form

$$\begin{aligned} J &\in \mathbb{R}^{N_e N_r \times N_n} \\ \delta \bar{L} &= J[\bar{n}] \delta \bar{n}, \end{aligned} \quad (3)$$

where  $\delta \bar{L}$  denotes the change in the acoustic length due to perturbation  $\delta \bar{n}$  in the refractive index field. The action of  $J$  on  $\delta \bar{n}$  for each pair of emitters and receivers is equivalent to an inner product of the corresponding row in  $J$  by  $\delta \bar{n}$ , and gives the integral of the refractive index distribution along a ray that links this pair. Further details are given in section 3.4.

### 2.3. Inverse problem

The UST inverse problem is to find the refractive index distribution of an object  $\bar{n}_{\text{object}}(\mathbf{x})$  from the discrepancies between the measured acoustic lengths across the object and across a reference medium (water), i.e. to find  $\bar{n}_{\text{object}}(\mathbf{x})$  from  $\Delta \bar{L} = c_w \Delta \bar{T}$ , where  $\Delta \bar{T} = \bar{T}_{\text{object}} - \bar{T}_w$  and  $\bar{T}_w$  are the times-of-flight through water with known sound speed  $c_w$ . This can be posed as a nonlinear minimisation problem of the form

$$\bar{n}_* = \arg \min_{\bar{n}} E[\bar{n}] = \arg \min_{\bar{n}} \|A[\bar{n}] - A[\bar{n}_w] - \Delta \bar{L}\|_2^2, \quad (4)$$

where the ‘object’ subscript has been dropped and from now on is implied. This nonlinear problem can be solved as a sequence of linearised problems. A linearisation of the forward operator  $A$  about a fixed  $\bar{n}_q$ , the  $q$ th update for the refractive index field, gives the



approximations

$$\begin{aligned} A[\bar{n}] &\approx A[\bar{n}_q] + J[\bar{n}_q] (\bar{n} - \bar{n}_q) \\ A[\bar{n}_w] &\approx A[\bar{n}_q] + J[\bar{n}_q] (\bar{n}_w - \bar{n}_q) \end{aligned} \quad (5)$$

Substituting equation (5) into equation (4) gives the  $q$ th linearised minimisation

$$\bar{n}_{q+1} = \arg \min_{\bar{n}} \tilde{E}_q[\bar{n}] = \arg \min_{\bar{n}} \|J[\bar{n}_q] \Delta \bar{n} - \Delta \bar{L}\|_2^2, \quad (6)$$

where  $\Delta \bar{n} = \bar{n} - \bar{n}_w$  with  $\bar{n}_w = 1$  for all grid points. Each linearised subproblem equation (6) is solved by setting  $\nabla \tilde{E}_q[\bar{n}] = 0$ , where the operator  $\nabla(\cdot)$  denotes the gradient. (By convention, the first iteration is  $q = 0$ .) As these are least squares problems, they lead to normal equations, which here are solved using a steepest descent algorithm with a fixed step length 1, and terminated after a fixed number of iterations. The sequence of linearised subproblems is recursively repeated until

$$1 - \tilde{E}[\bar{n}_{q+1}] / \tilde{E}[\bar{n}_q] < \varepsilon_n, \quad (7)$$

where  $\varepsilon_n$  is a user-defined stopping threshold.

**Remark 1.** Early-stopping of the steepest descent iterations in the solution of the subproblems can act implicitly as a regulariser, but an explicit regularisation term was not included in the objective function equation (4) so that the reconstructed image depends only on the proposed ray-based reconstruction algorithm and not on the value of a regularisation parameter. See, for example, [2, 32] for the application of regularisation to the UST inverse problem.

### 3. Ray tracing

Here, the method used for finding the rays  $f_{(n,e,p)}(\mathbf{x}) = 0$  (see definition 1) is described.

#### 3.1. Ray equation

A general form of the acoustic wave equation in free space is

$$u_{tt}(\mathbf{x}, t) - c(\mathbf{x})^2 u_{xx}(\mathbf{x}, t) = 0, \quad \mathbf{x} \in \mathbb{R}^3, \quad t > 0, \quad (8)$$

where  $u$  is the acoustic pressure, and  $t$  time. When  $c$  is constant, a harmonic wave solution of equation (8) can be written as  $u(\mathbf{x}, t) = U(\mathbf{x}) \exp(i(\mathbf{k} \cdot \mathbf{x} - \omega t))$ , where  $U$  is the amplitude,  $\omega$  is the angular frequency, and  $\mathbf{k}$  is a wavevector describing the direction of propagation.  $U$  and  $\omega$  are specified by the initial conditions [59]. For a spatially varying  $c(\mathbf{x})$ , this can be modified, under a high frequency approximation, into the form [31]

$$u(\mathbf{x}, t) \approx U(\mathbf{x}) e^{i\omega(W(\mathbf{x})/c_0 - t)}, \quad (9)$$

where  $\mathbf{k} \cdot \mathbf{x}$  has been replaced by  $(\omega/c_0)W(\mathbf{x})$ , the accumulated phase, and  $c_0$  is a reference sound speed. For the high frequency approximation to hold, the fractional change in sound speed gradient over the longest wavelength involved in the problem must be small compared to the sound speed [31]. Now, substituting equation (9) into equation (8) gives the two equations

$$\nabla^2 U - \frac{\omega^2}{c_0^2} U \nabla W \cdot \nabla W = -\frac{\omega^2}{c^2(\mathbf{x})} U, \quad 2\nabla W \cdot \nabla U + U \nabla^2 W = 0. \quad (10)$$



For the left-hand-side equation, neglecting the first term because of the same high frequency approximation gives [31]

$$\nabla W \cdot \nabla W = n^2(\mathbf{x}). \quad (11)$$

This is known as the *eikonal equation*. The surfaces on which  $W(\mathbf{x})$  is constant are surfaces of constant phase called *wavefronts*, and the lines which are always normal to these wavefronts are *rays*. The acoustic energy propagates along these rays in the direction defined by the unit vector  $d\mathbf{x}_{\text{ray}}/ds$ , where [4]

$$\frac{d\mathbf{x}_{\text{ray}}}{ds} = \frac{\nabla W(\mathbf{x})}{n(\mathbf{x})}. \quad (12)$$

As  $d/ds(n d\mathbf{x}_{\text{ray}}/ds) = d/ds(\nabla W) = \nabla(dW/ds) = \nabla n$ , this leads directly to another form of the eikonal equation which here is called the *ray equation*:

$$\frac{d}{ds} \left( n \frac{d\mathbf{x}_{\text{ray}}(s)}{ds} \right) = \nabla n. \quad (13)$$

In addition, the right-hand-side equation in equation (10) is called *transport equation*, and can be used for computation of the amplitude of the field along the ray [57, 59], although it is not used in this study.

### 3.2. Fermat's principle

The ray equation can also be found using Fermat's principle [22, 53], which states that the path between two points taken by a ray makes the acoustic length stationary under variations in a family of nearby paths. As above, let  $s$  denote the distance along a ray, and  $\mathbf{x}(s) \in \mathbb{R}^3$  and  $(d\mathbf{x}/ds)(s) \in S^2$ , denote the ray's position vector and (unit) direction vector, respectively. (The subscript on  $\mathbf{x}_{\text{ray}}(s)$  has been dropped for this section for conciseness.)

**Definition 5.** The acoustic length of a path  $\mathbf{x}(s)$  taken by a ray in passing from point  $p_1$  to  $p_2$  is defined by

$$L_{(p_1, p_2)} = \int_{p_1}^{p_2} n(\mathbf{x}(s)) ds, \quad (14)$$

where  $ds = |d\mathbf{x}(s)|$  is an infinitesimal distance along the ray (see equation (1)).

**Definition 6.** A family of  $C^2$  continuous paths  $\mathbf{x}(s, \epsilon) \in \mathbb{R}^3$ , which smoothly depends on  $\epsilon$ , gives a set of smooth transformations of the ray path  $\mathbf{x}(s)$  in an interval including  $\epsilon = 0$ , and satisfies [22]

$$\mathbf{x}(s_{p_1}, \epsilon) = \mathbf{x}(s_{p_1}), \quad \mathbf{x}(s_{p_2}, \epsilon) = \mathbf{x}(s_{p_2}). \quad (15)$$

An infinitesimal variation of the path  $\mathbf{x}(s)$  is defined in the form [22]

$$\delta\mathbf{x}(s) := \left. \frac{d}{d\epsilon} \right|_{\epsilon=0} \mathbf{x}(s, \epsilon), \quad (16)$$

where, from equation (15),  $\delta\mathbf{x}(s_{p_1}) = \delta\mathbf{x}(s_{p_2}) = 0$ .

**Definition 7.** Using definitions 5 and 6, together with the fact that  $\mathbf{dx}/ds$  is a unit vector, the stationary acoustic length under variation in a family of nearby paths satisfies [22]

$$\delta L_{(p_1, p_2)} = \delta \left[ \int_{s_{p_1}}^{s_{p_2}} n(\mathbf{x}(s)) \left( \frac{d\mathbf{x}}{ds} \cdot \frac{d\mathbf{x}}{ds} \right)^{1/2} ds \right] = 0. \quad (17)$$

**Lemma 1.** Using definition 5 and the boundary conditions in equation (15), the stationarity of the optical length defined in equation (17) yields a path  $\mathbf{x}(s)$  that satisfies equation (13).

**Proof.** Consider the integrand in equation (17) as a Lagrangian function in the form  $F(\mathbf{x}, \mathbf{x}', s) = n(\mathbf{x})(\mathbf{x}' \cdot \mathbf{x}')^{1/2}$ , where  $\mathbf{x}' = d\mathbf{x}/ds$  has been used for brevity. Using definition 6, a perturbation of the first order,  $\delta\mathbf{x}$ , is applied to the path  $\mathbf{x}(s)$  to give

$$\begin{aligned} \delta L_{(p_1, p_2)} &= \int_{s_{p_1}}^{s_{p_2}} [F(\mathbf{x} + \delta\mathbf{x}, \mathbf{x}' + \delta\mathbf{x}') - F(\mathbf{x}, \mathbf{x}')] ds \\ &= \int_{s_{p_1}}^{s_{p_2}} \left[ \delta\mathbf{x} \cdot \frac{\partial F}{\partial \mathbf{x}} + \delta\mathbf{x}' \cdot \frac{\partial F}{\partial \mathbf{x}'} \right] ds. \end{aligned} \quad (18)$$

Now, applying an integration by parts to the second term in the integrand, together with the boundary conditions in definition 6, gives

$$\delta L_{(p_1, p_2)} = \int_{s_{p_1}}^{s_{p_2}} \delta\mathbf{x} \cdot \left( \frac{\partial F}{\partial \mathbf{x}} - \frac{d}{ds} \frac{\partial F}{\partial \mathbf{x}'} \right) ds. \quad (19)$$

The proof is straightforwardly completed by substituting  $F$  into the above equation and setting the first order change  $\delta L_{(p_1, p_2)}$  to zero for all perturbations  $\epsilon$ .  $\square$

### 3.3. Numerical ray tracing

In this section, two numerical ray-tracing algorithms are derived.

**3.3.1. Dual-update algorithm.** Consider a ray starting from an emission point  $\mathbf{x} = \mathbf{x}_e$ . A Taylor series expansion of the ray's position vector  $\mathbf{x}(s)$  can be written

$$\mathbf{x}(s + \Delta s) = \mathbf{x}(s) + \left. \frac{d\mathbf{x}}{ds} \right|_s \Delta s + \frac{1}{2} \left. \frac{d^2\mathbf{x}}{ds^2} \right|_s \Delta s^2 + O(\Delta s^3), \quad (20)$$

where  $d\mathbf{x}/ds|_s$  and  $d^2\mathbf{x}/ds^2|_s$  denote the first-order and second-order derivatives of the position vector at point  $s$ , respectively. The second derivative can be found by expanding equation (13) to get,

$$\left. \frac{d^2\mathbf{x}}{ds^2} \right|_s = \frac{1}{n} \left[ \nabla n - \left( \nabla n \cdot \left. \frac{d\mathbf{x}}{ds} \right|_s \right) \left. \frac{d\mathbf{x}}{ds} \right|_s \right], \quad (21)$$

which is orthogonal to  $d\mathbf{x}/ds|_s$ . For the first step, the first-order derivative, the ray tangent, is given by the user but for subsequent steps it needs to be found. A Taylor series expansion for  $d\mathbf{x}/ds$  can be written as

$$\left. \frac{d\mathbf{x}}{ds} \right|_{s+\Delta s} = \left. \frac{d\mathbf{x}}{ds} \right|_s + \left. \frac{d^2\mathbf{x}}{ds^2} \right|_s \Delta s + O(\Delta s^2). \quad (22)$$

**Algorithm 1.** Ray tracing: dual-update approach.

---

```

1: input:  $\mathbf{x}_{\bar{e}} \in \mathbb{R}^3, \mathbf{d}_{\bar{e}} \in S^2$                                 ▷ Input initial ray position and tangent vectors
2: initialise:  $\mathbf{x}(0) = \mathbf{x}_{\bar{e}}, \mathbf{d}(0) = \mathbf{d}_{\bar{e}}$                         ▷ Set initial ray position and tangent vectors
3: while  $\mathbf{x}(s)$  is inside  $\Omega$  do                                ▷ Run the loop while the ray remains in the domain
4:    $\mathbf{h} = (\nabla n - (\nabla n \cdot \mathbf{d})\mathbf{d})/n$                         ▷ Compute the second derivative from equation (21)
5:    $\mathbf{g} = \mathbf{d} + \mathbf{h}\Delta s/2$                                     ▷ Calculate the ray position update direction
6:    $\mathbf{g} \leftarrow \mathbf{g}/|\mathbf{g}|$                                     ▷ Make the update direction a unit vector
7:    $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{g}\Delta s$                                 ▷ Update the ray position vector
8:    $\mathbf{d} \leftarrow \mathbf{d} + \mathbf{h}\Delta s$                                 ▷ Update the ray tangent vector using equation (22)
9:    $\mathbf{d} \leftarrow \mathbf{d}/|\mathbf{d}|$                                     ▷ Make the tangent vector a unit vector
10: end while

```

---

The two Taylor series, equations (20) and (22) can be combined with equation (21) into a dual-update iterative scheme to calculate the next step along the ray  $\mathbf{x}(s + \Delta s)$  from the current point  $\mathbf{x}(s)$ . This is written below as algorithm 1. The normalisation of the update vector in line 6 is included because it makes numerical integration along the ray easier if the ray is defined at evenly spaced points. The normalisation of the ray tangent vector,  $\mathbf{d}$ , in line 9 is there to ensure that it, like  $d\mathbf{x}/ds$  that it is an approximation to, is a unit vector. In numerical tests in which  $n$  and  $\nabla n$  are known exactly, the error in the path of the ray as calculated by algorithm 1 converges at a rate of  $O(\Delta s^2)$ , see section 5.1.3. In many practical scenarios, such as in the inverse problem considered in this paper, the error in the ray path will, however, be dominated by error in the approximations of  $n$  and  $\nabla n$  (section 3.4). Furthermore, it is often the acoustic length—the integral of  $n$  along the ray—that is the salient quantity, not the ray path *per se*, and that will include errors due to the numerical approximation of the path integral as well. It can therefore be beneficial to trade a reduction in the formal rate of convergence of an algorithm for an increase in its computational efficiency, particularly when the former has little detrimental effect on the accuracy in the overall problem and the latter can significantly reduce the computational burden. With this in mind, a second ray-tracing algorithm was derived.

**3.3.2. Mixed-step algorithm.** The ray tracing algorithm derived in this section and used in the examples below, sometimes known as the mixed-step algorithm, [8, 14, 26] is computationally more efficient than algorithm 1, although converges more slowly with  $\Delta s$ . Starting from the Taylor series expansion for the ray position vector, equation (20), and replacing the first derivative with the central difference approximation  $d\mathbf{x}/ds|_s = (\mathbf{x}(s + \Delta s) - \mathbf{x}(s - \Delta s))/(2\Delta s)$ , gives a result which can be rearranged into the form

$$\mathbf{x}(s + \Delta s) = \mathbf{x}(s) + \mathbf{d}|_s \Delta s + \frac{d^2 \mathbf{x}}{ds^2} \bigg|_s \Delta s^2, \quad (23)$$

where  $\mathbf{d}|_s := (\mathbf{x}(s) - \mathbf{x}(s - \Delta s))/\Delta s$  is a backward difference. Equation (23) can be rewritten as

$$\mathbf{x}(s + \Delta s) = \mathbf{x}(s) + \mathbf{d}|_{s+\Delta s} \Delta s, \quad (24)$$

where the ray-direction update is given by

$$\mathbf{d}|_{s+\Delta s} = \mathbf{d}|_s + \frac{d^2 \mathbf{x}}{ds^2} \bigg|_s \Delta s, \quad (25)$$

**Algorithm 2.** Ray tracing: mixed-step approach.

---

```

1: input:  $\mathbf{x}_{\bar{e}} \in \mathbb{R}^3, \mathbf{d}_{\bar{e}} \in S^2$                                 ▷ Input initial ray position and tangent vector
2: initialise:  $\mathbf{x} = \mathbf{x}_{\bar{e}}, \mathbf{d} = \mathbf{d}_{\bar{e}}$                                 ▷ Set initial ray position and tangent vector
3:  $\mathbf{h} = (\nabla n - (\nabla n \cdot \mathbf{d})\mathbf{d})/n$                                 ▷ Compute the initial second derivative using equation (21)
4:  $\mathbf{d} \leftarrow \mathbf{d} + \mathbf{h}\Delta s/2$                                 ▷ Update the ray direction based on equation (20)
5:  $\mathbf{d} \leftarrow \mathbf{d}/|\mathbf{d}|$                                 ▷ Make the update direction a unit vector
6:  $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{d}\Delta s$                                 ▷ Update ray position vector
7: while  $\mathbf{x}(s)$  is inside  $\Omega$  do                                ▷ Run the loop while the ray remains in the domain
8:    $\mathbf{h} = (\nabla n - (\nabla n \cdot \mathbf{d})\mathbf{d})/n$                                 ▷ Compute the second derivative using equation (21)
9:    $\mathbf{d} \leftarrow \mathbf{d} + \mathbf{h}\Delta s$                                 ▷ Update the ray direction using equation (25)
10:   $\mathbf{d} \leftarrow \mathbf{d}/|\mathbf{d}|$                                 ▷ Make the update direction a unit vector
11:   $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{d}\Delta s$                                 ▷ Update the ray position vector
12: end while

```

---

(see the Taylor series, equation (22)). Note that the ray-direction update  $\mathbf{d}|_{s+\Delta s}$  must be normalised to ensure that Fermat's principle holds (see equation (21)) and it gives steps along the ray of equal length  $\Delta s$ . Using equation (24) to calculate the next position along the ray requires  $\mathbf{d}|_{s+\Delta s}$  to be calculated from equation (25), which raises the question of how to compute  $d^2\mathbf{x}/ds^2$ . This must be done using the ray equation, equation (21), as it is via this equation that the refractive index affects the ray path, but  $d\mathbf{x}/ds|_s$  in equation (21) is not known. In deriving equation (23) from the Taylor series,  $d\mathbf{x}/ds|_s$  was approximated using a central difference. However, here the backward difference  $\mathbf{d}|_s$  is preferred for reasons of computational efficiency (and for Fermat's principle to hold it must be normalised, as mentioned above). The use of a backward difference raises one issue, which is that it is necessary to know  $\mathbf{x}(s - \Delta s)$  at each step, and for the first step this is not known. This can be overcome by defining  $\mathbf{d}|_0$  as the user-defined initial ray tangent, and using equations (20) and (21) to compute  $\mathbf{x}(\Delta s)$  given  $\mathbf{x}(0)$ . The resulting algorithm is shown in algorithm 2. This algorithm can be coded more efficiently than algorithm 1 as it contains fewer steps and fewer stored variables; the trade-off is a reduction in the convergence rate due to the first-order backward difference approximation used in the ray equation. However, as mentioned above, this is rarely a problem in practice as typically greater errors arise from the approximation of the refractive index gradient and the numerical path integration (described in sections 3.4 and 3.4.3 below).

### 3.4. Grid-to-ray interpolation

As mentioned in section 2.2, the refractive index is defined on a grid of points. In contrast, the points along the ray computed using the algorithms above can lie anywhere in  $\Omega$  and are not restricted to grid points. An interpolation map from the grid to the rays is therefore necessary for approximating  $n$  and  $\nabla n$  in the ray tracing algorithms, as well as for forming the Jacobian matrix  $J[\bar{n}_q]$  (cf definition 4) used for solving the inverse problem in equation (6).

**3.4.1. Numerical approximation of refractive index.** The refractive index field was discretised on a mesh of points  $x_i$  with  $i \in \{1, \dots, N_n\}$ . For simplicity, a rectilinear grid was used with grid points indexed with the multi-index  $i = (i^1, i^2, i^3) \in \{1, \dots, N_n^1\} \times \{1, \dots, N_n^2\} \times \{1, \dots, N_n^3\}$  with  $N_n = \prod_{j=1}^3 N_n^j$  and an equal grid spacing  $\Delta x$  along all Cartesian coordinates  $j$ . Also,  $x_{ij}$  is used to indicate the position of grid point  $i$  along Cartesian coordinate  $j$ . (Recall that  $x^j$  denotes the position in continuous Euclidean space along the Cartesian coordinate  $j$ .)

Let  $\{\phi_i(\mathbf{x}), i = 1, \dots, N_n\}$  denote a set of basis functions for which basis function  $\phi_i(\mathbf{x})$  is related to grid point  $i$ . An arbitrary continuous scalar field  $z(\mathbf{x})$  (this could be the refractive index or a component of its gradient) can be approximated as a linear combination of these basis functions

$$z(\mathbf{x}) \approx \hat{z}(\mathbf{x}) = \sum_{i=1}^{N_n} \bar{z}_i \phi_i(\mathbf{x}). \quad (26)$$

Here, trilinear basis functions defined on a regular grid were used, which take the form [14]

$$\phi_i(\mathbf{x}) = \prod_{j=1}^3 \left(1 - |u_i^j(\mathbf{x})|\right), \quad (27)$$

where

$$u_i^j(\mathbf{x}) = \begin{cases} (x^j - x_{ij}) / \Delta x, & x_{(i-1)^j} < x^j < x_{(i+1)^j} \\ 0, & \text{otherwise.} \end{cases} \quad (28)$$

Here,  $x_{(i\pm 1)^j}$  denotes the two grid points adjacent to the grid point  $x_i$  along the Cartesian coordinate  $j$ . The basis function  $\phi_i(\mathbf{x})$  has a pyramidal shape with the vertex on point  $i$ , and it vanishes on the neighboring grid points. Because this basis function is a polynomial of first order along each Cartesian coordinate  $j$ , the interpolated function  $\hat{z}$  is continuous but its first derivative is not continuous [14]. Therefore, for an approximation of the directional gradients of the refractive index, we follow [4, 5], where the discretised directional gradients are first calculated from the values of field at the grid nodes using finite differences, and are then approximated for off-grid points using the same interpolation as for the field itself. (Using this approach, the need for different interpolation operators for the refractive index and each of the three gradient components is avoided.) It has been shown that this approach gives an approximation that is sufficiently accurate for weakly heterogeneous media [4, 5, 14], which is the case here. We will show in section 5 that this approach provides a good trade-off between accuracy and computational cost.

**3.4.2. Interpolation operator.** This section describes the map used for interpolating from the discretised field on the grid,  $\bar{z}$ , to an arbitrary point along the ray using equation (26) [28].

**Definition 8.** The positions of the first and last vertices of each cubic voxel  $v$  along each Cartesian coordinate  $j$  are given by  $x_{i'j}^v$ , where  $i' \in \{0, 1\}$  denotes the first and last vertices. Any arbitrary point  $\mathbf{x} = \{x^j, j = 1, 2, 3\}$  contained in  $v$  satisfies  $x_{0j}^v \leq x^j < x_{1j}^v$  for all Cartesian coordinates  $j$ . The values of the discretised field  $\bar{z}$  associated with the vertices of voxel  $v$ , containing  $\mathbf{x}$ , are notated  $\bar{z}^v$  and defined using the map

$$\begin{aligned} J^v : \mathbb{R}^{N_n} &\rightarrow \mathbb{R}^8 \\ \bar{z}^v[\mathbf{x}] &= J^v[\mathbf{x}] \bar{z}. \end{aligned} \quad (29)$$

Correspondingly, the elements of  $\bar{z}^v$  are written as

$$\bar{z}^v = [\bar{z}_{(0,0,0)}^v, \bar{z}_{(0,0,1)}^v, \bar{z}_{(0,1,0)}^v, \bar{z}_{(0,1,1)}^v, \bar{z}_{(1,0,0)}^v, \bar{z}_{(1,0,1)}^v, \bar{z}_{(1,1,0)}^v, \bar{z}_{(1,1,1)}^v]^T. \quad (30)$$

We will also use

$$\mathcal{Q}^v[\mathbf{x}] = [1, l^{(v,1)}, l^{(v,2)}, l^{(v,3)}, l^{(v,1)}l^{(v,2)}, l^{(v,2)}l^{(v,3)}, l^{(v,1)}l^{(v,3)}, l^{(v,1)}l^{(v,2)}l^{(v,3)}], \quad (31)$$

where the dependence of  $l^{(v,j)}$  on  $\mathbf{x}$  is neglected for brevity, but is given by

$$l^{(v,j)}(\mathbf{x}) = \frac{\mathbf{x} - \mathbf{x}_{0j}^v}{\mathbf{x}_{1j}^v - \mathbf{x}_{0j}^v} = \frac{\mathbf{x} - \mathbf{x}_{0j}^v}{\Delta \mathbf{x}}. \quad (32)$$

**Definition 9.** The trajectory of a ray-linking an emission point  $\bar{e}$  to a reception point  $\bar{r}$  is defined by the points  $s_m$   $m \in \{0, \dots, M_{(\bar{e}, \bar{r})}\}$ , where the initial point  $s_0$  matches the emission point  $\bar{e}$ , the final point  $s_{M_{(\bar{e}, \bar{r})}}$  matches the reception point  $\bar{r}$ , and the number of the sampling points along the ray is  $M_{(\bar{e}, \bar{r})} + 1$ . For ray tracing algorithms that take equal sized steps along the ray, as is the case for algorithms 1 and 2, the points  $s_m$  must satisfy

$$s_m = \begin{cases} m\Delta s, & m \in \{0, \dots, M_{(\bar{e}, \bar{r})} - 1\} \\ (m-1)\Delta s + \Delta s', & m = M_{(\bar{e}, \bar{r})}. \end{cases} \quad (33)$$

Also, the second line in equation (33) is used in order to indicate that the last point of the ray must be matched to the reception point  $\bar{r}$ , and thus  $\Delta s' = s_{M_{(\bar{e}, \bar{r})}} - s_{M_{(\bar{e}, \bar{r})}-1}$  with  $\Delta s' \leq \Delta s$ . This is achieved using ray-linking, described in section 4.

**Lemma 2.** Using the trilinear basis function, equation (27), an interpolation map (operator) from a discretised field  $\bar{z}$  to sampling points  $s_m$  on a ray is defined as

$$\begin{aligned} J_{(s_m; \bar{e}, \bar{r})}^{\text{int}} : \mathbb{R}^{N_n} &\rightarrow \mathbb{R}^{M_{(\bar{e}, \bar{r})}+1} \\ \hat{z}[\mathbf{x}_{(s_m; \bar{e}, \bar{r})}] &= J_{(s_m; \bar{e}, \bar{r})}^{\text{int}} \bar{z} = Q^v[\mathbf{x}_{(s_m; \bar{e}, \bar{r})}] C J^v[\mathbf{x}_{(s_m; \bar{e}, \bar{r})}] \bar{z}, \end{aligned} \quad (34)$$

where, the matrix  $C \in \mathbb{R}^{8 \times 8}$  is in the form [28]

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 & 1 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 \\ -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \end{bmatrix}. \quad (35)$$

**Proof.** The proof is obtained by forming  $\hat{z}$  in equation (26) using equations (27) and (28), and then re-arranging into the formula equation (34) using equations (31) and (32).  $\square$

**Remark 2.** The interpolation operator  $J_{(\bar{e}, \bar{r})}^{\text{int}}$  is used for two purposes: for approximating  $n$  and  $\nabla n$  along the rays in the ray tracing algorithm above, and in constructing the Jacobian matrix (definition 4), described below.

**3.4.3. Jacobian matrix.** This section follows the definition for the Jacobian matrix  $J$  in definition 4, and gives further details about its action on a perturbation field  $\delta \bar{n}$ . It is worth mentioning that while  $J$  is dependent on  $\bar{n}$ , it acts on  $\Delta \bar{n}$  in our iterative inversion algorithm (see equation (6)). From equation (34),  $J_{(\bar{e}, \bar{r})}^{\text{int}}$  maps  $\delta \bar{n}$  on the grid points to  $\delta \hat{n}$  on the sample points along one ray, the one between the emission point  $\bar{e}$  and reception point  $\bar{r}$ . Given  $\delta \hat{n}$ , the

perturbation in the acoustic length along that ray can be calculated using the map

$$\begin{aligned}
 J_{(\bar{e}, \bar{r})}^l &: \mathbb{R}^{M(\bar{e}, \bar{r})+1} \rightarrow \mathbb{R} \\
 \delta \bar{L}_{(\bar{e}, \bar{r})} &= J_{(\bar{e}, \bar{r})}^l \delta \hat{n} \\
 &= \frac{1}{2} \delta \hat{n} [\mathbf{x}_{(0; \bar{e}, \bar{r})}] \Delta s + \sum_{m=1}^{M(\bar{e}, \bar{r})-2} \delta \hat{n} [\mathbf{x}_{(m \Delta s; \bar{e}, \bar{r})}] \Delta s \\
 &\quad + \frac{1}{2} \delta \hat{n} [\mathbf{x}_{((M(\bar{e}, \bar{r})-1) \Delta s; \bar{e}, \bar{r})}] (\Delta s + \Delta s') \\
 &\quad + \frac{1}{2} \delta \hat{n} [\mathbf{x}_{((M(\bar{e}, \bar{r})-1) \Delta s + \Delta s'; \bar{e}, \bar{r})}] \Delta s',
 \end{aligned} \tag{36}$$

which is a numerical path integration of the  $\delta \hat{n}$  along the ray using the trapezoidal rule. Using  $J_{(\bar{e}, \bar{r})}$  to denote the row of the full Jacobian  $J$  corresponding to an emission point  $\bar{e}$  and a reception point  $\bar{r}$ , and considering definition 4, equations (34) and (36), the action of  $J_{(\bar{e}, \bar{r})}$  on a field  $\delta \bar{n}$  can be written as

$$\begin{aligned}
 J_{(\bar{e}, \bar{r})} &: \mathbb{R}^{N_n} \rightarrow \mathbb{R} \\
 \delta \bar{L}_{(\bar{e}, \bar{r})} &= J_{(\bar{e}, \bar{r})} \delta \bar{n} = J_{(\bar{e}, \bar{r})}^l J_{(\bar{e}, \bar{r})}^{\text{int}} \delta \bar{n}.
 \end{aligned} \tag{37}$$

#### 4. Ray-linking (two-point ray tracing)

The ray tracing algorithms derived in section 3 solve a path for the rays  $f_{(n; \bar{e}, p)} = 0$ , where  $\bar{e} \in \mathbb{S}_1$  is the emission point on the hemispherical surface  $\mathbb{S}_1$ , and  $p \in \mathbb{S}$  is the interception point of the ray with the surface  $\mathbb{S}$  after traveling through the medium. From definitions 3 and 4, the forward operator and Jacobian matrix are given by the path integral of the refractive index along a set of rays that link the emission points  $\bar{e}$  to the reception points  $\bar{r}$ . These rays are defined using  $f_{(n; \bar{e}, \bar{r})} = 0$ . Accordingly, for each pair of points  $\bar{e}$  and  $\bar{r}$ , one seeks to find the ray's path  $f_{(n; \bar{e}, p)} = 0$  such that  $p$  matches  $\bar{r}$ . Applying this condition leads to a boundary value problem called *ray-linking*. This boundary value problem can be solved as a sequence of initial value problems by adjusting the initial direction of the ray until the ray intercepts  $\mathbb{S}$  sufficiently close to the reception point  $\bar{r}$  [4, 5, 56, 57, 60]. This approach fits into a class of methods for solving boundary value problems known as *shooting methods*.

An alternative class of methods for two-point ray tracing is those based on *ray bending*. These kinds of approaches follow directly the Fermat's principle, and are based on fixing the two end points of the ray at the required emission and reception points, and perturbing the ray trajectory until the acoustic length along the ray becomes stationary under these perturbations [30, 65, 69]. In this study, we use the former approach, ray-linking, and calculate the ray path between each pair of emitters and receivers by solving the inverse problem of finding the initial ray direction given the reception point through successive implementations of the ray-tracing algorithm.

##### 4.1. Ray-linking inverse problem

In the shooting method, the task of finding the ray that links the emission point  $\bar{e}$  to the reception point  $\bar{r}$  has been recast as the inverse problem of finding the initial ray-direction that results in the ray reaching the  $\bar{r}$  (or close enough to it). This must be solved for all pairs of emission and reception points  $(\bar{e}, \bar{r})$ . Here, this is solved for each ray by iteratively adjusting the initial ray



direction  $\mathbf{d}_{\bar{e}}$  until the interception point  $p$  matches the reception point  $\bar{r}$ . To facilitate this, we first parameterise the problem in a way that can be adapted to any convex detection surface. Accordingly, the unknown initial direction of the ray  $\mathbf{d}_{\bar{e}} = [\theta_{\bar{e}}, \varphi_{\bar{e}}]^T \in \mathbb{R}^2$  is written in terms of the azimuthal and polar angles. Then for each ray, the direction from the emission point  $\bar{e}$  to the end point of the ray, the interception point  $p$ , is defined as a unit vector in spherical-coordinates centered on the emission point  $\bar{e}$ , i.e.,

$$\begin{aligned} \mathbb{R}^2 &\rightarrow [0, 2\pi) \times [0, \pi) \\ \gamma_{(\bar{e}, p)}[\mathbf{d}_{\bar{e}}] &= \gamma_{(\bar{e}, p)}[\theta_{\bar{e}}, \varphi_{\bar{e}}] = \frac{\mathbf{x}_p[\mathbf{d}_{\bar{e}}] - \mathbf{x}_{\bar{e}}}{|\mathbf{x}_p[\mathbf{d}_{\bar{e}}] - \mathbf{x}_{\bar{e}}|}, \end{aligned} \quad (38)$$

where, from algorithm 2, the dependence of  $\mathbf{x}_p$ , and therefore  $\gamma_{(\bar{e}, p)}$ , on the unknown initial direction vector  $\mathbf{d}_{\bar{e}}$  is nonlinear. Here,  $\theta_{\bar{e}} \in [0, 2\pi)$  and  $\varphi_{\bar{e}} \in [0, \pi)$  are respectively the azimuthal and polar angles of the reception point  $p$  with respect to the emission point  $\bar{e}$ . In the same way, the position of the reception point  $\bar{r}$  is defined in terms of a unit direction vector  $\gamma_{(\bar{e}, \bar{r})}$ ,

$$\gamma_{(\bar{e}, \bar{r})} = (\theta_{(\bar{e}, \bar{r})}, \varphi_{(\bar{e}, \bar{r})})^T = \frac{\mathbf{x}_{\bar{r}} - \mathbf{x}_{\bar{e}}}{|\mathbf{x}_{\bar{r}} - \mathbf{x}_{\bar{e}}|}. \quad (39)$$

Using these definitions, the ray-linking problem for the pair of points  $\bar{e}$  and  $\bar{r}$  can be expressed either as a problem of solving for the root of the residual function  $F: \mathbb{R}^2 \rightarrow [-\pi, \pi) \times [-\pi, \pi)$ :

$$F_{(\bar{e}, \bar{r})}(\mathbf{d}_{\bar{e}}) = \gamma_{(\bar{e}, p)}[\mathbf{d}_{\bar{e}}] - \gamma_{(\bar{e}, \bar{r})} = 0, \quad (40)$$

or alternatively as a minimisation of the functional  $\mathcal{E}: \mathbb{R}^2 \rightarrow \mathbb{R}$ :

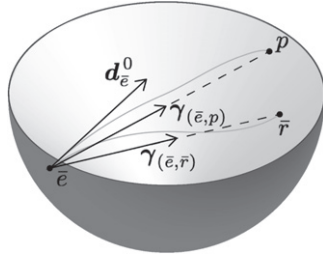
$$\arg \min_{\mathbf{d}_{\bar{e}}} \mathcal{E}(\mathbf{d}_{\bar{e}}) := \frac{1}{2} \|F_{(\bar{e}, \bar{r})}(\mathbf{d}_{\bar{e}})\|_2^2. \quad (41)$$

The problem equation (41) is equivalent to solving for the root of the gradient of the residual function. (Note that the two elements of the residual function are wrapped to their corresponding intervals.) A schematic for this ray-linking inverse problem is shown in figure 2. Using the above parameterisation, this inverse problem can be thought of as an iterative adjustment of the initial direction  $\mathbf{d}_{\bar{e}}$  until the unit vector  $\gamma_{(\bar{e}, p)}$  becomes aligned to the required unit vector  $\gamma_{(\bar{e}, \bar{r})}$  (see figure 2). In this way, the algorithms described below for solving the ray-linking inverse problem become independent from the geometry, and can be adapted to any arbitrary convex detection surface.

**Remark 3.** A wide variety of standard solvers were applied to this problem but a significant portion of the rays always failed to link to the reception points. The reason for this is that the problem can become ill-posed in some cases. Adaptive-smoothing schemes are proposed to manage the ill-posedness, and are incorporated into three classical methods (Gauss–Newton, BFGS, and Broyden) in the three subsections below. The three methods are compared for accuracy as well as computational cost.

#### 4.2. Initial ray direction

As described in section 2.3, the refractive index map  $\bar{n}_q$  is updated at every iteration  $q$  of the UST reconstruction. The ray-linking problem must be solved for every pair of emitters and receivers  $(\bar{e}, \bar{r})$  for every iteration  $q$ , as the updated refractive index will change the ray trajectories. For each iteration  $q > 0$  the initial direction of the ray,  $\mathbf{d}_{(\bar{e}, q)}^0$ , is chosen to be the optimal



**Figure 2.** The initial ray (grey line), with initial direction given by the unit vector  $\mathbf{d}_{\bar{e}}^0$ , leaves the emission point  $\bar{e} \in \mathbb{S}_1$  and arrives at the interception point  $p \in \mathbb{S}$  (which is in a direction  $\gamma_{(\bar{e}, p)}$  from  $\bar{e}$ ). Following the ray-linking, the ray leaves  $\bar{e}$  with a different initial direction and arrives at the desired reception point  $\bar{r} \in \mathbb{S}_1$  (which is in a direction  $\gamma_{(\bar{e}, \bar{r})}$  from  $\bar{e}$ ). The ray-linking method works by iteratively adjusting the initial direction  $\mathbf{d}_{\bar{e}}$  until  $\gamma_{(\bar{e}, p)} = \gamma_{(\bar{e}, \bar{r})}$ .

ray from the previous iteration  $\mathbf{d}_{(\bar{e}, q-1)}^*$ . For the first iteration it is chosen to be the unit vector in the direction of  $\bar{r}$ ,  $\gamma_{(\bar{e}, \bar{r})}$ :

$$\mathbf{d}_{(\bar{e}, q)}^0 = \begin{cases} \mathbf{d}_{(\bar{e}, q-1)}^*, & q > 0, \\ \gamma_{(\bar{e}, \bar{r})}, & q = 0. \end{cases} \quad (42)$$

For  $q = 0$ , because  $\bar{n}_0 = 1$ , the Jacobian matrix  $J[\bar{n}_0]$  in equation (6) is also formed using  $\gamma_{(\bar{e}, \bar{r})}$ . As shown in the numerical results in section 5, this choice provides very good initial guesses for the ray-linking problems. For the case of multiple linking paths, which will occur if the solution of the ray-linking problem is non-unique [4], this choice leads to the shortest path. An alternative way to choose the initial guesses is to shoot many initial rays from the emission point  $\bar{e}$  then choose, for each reception point  $\bar{r}$ , the closest ray as the initial guess. However, our experience was that this approach worked less well than the proposed approach above, perhaps due to the existence of multiple linking paths.

#### 4.3. Ray-linking using functional minimisation: damped Gauss–Newton

In the geophysics literature, a popular approach for ray-linking is solving the minimisation problem equation (41) based on an iterative linearisation of the functional  $\mathcal{E}$  using finite differences, and solving a sequence of linearised minimization problems using Newton-type methods [27, 56, 57, 60]. Correspondingly, the  $k$ th linearisation of  $\gamma_{(\bar{e}, p)}$  around  $\mathbf{d}_{\bar{e}}^k$  gives an approximation

$$\gamma_{(\bar{e}, p)}[\mathbf{d}_{\bar{e}}] \approx \gamma_{(\bar{e}, p)}[\mathbf{d}_{\bar{e}}^k] + \mathcal{J}[\mathbf{d}_{\bar{e}}^k](\mathbf{d}_{\bar{e}} - \mathbf{d}_{\bar{e}}^k). \quad (43)$$

Here, the superscript  $k$  denotes the iteration number (the number of the linearised subproblem), and it is assumed that  $\mathbf{d}_{\bar{e}}$  is sufficiently close to  $\mathbf{d}_{\bar{e}}^k$ . Also,  $\mathcal{J}[\mathbf{d}_{\bar{e}}^k] \in \mathbb{R}^{2 \times 2}$  denotes the partial derivative (Jacobian) matrix at  $\mathbf{d}_{\bar{e}}^k$ , and is calculated using finite differences. ( $\mathcal{J}$  should not be confused with its counterpart  $J$  that stands for the Jacobian matrix (definition 4) for the UST inverse problem.) Using equation (43), the minimisation problem equation (41) can be modified into the form

$$\mathbf{p}^k = \arg \min_p \tilde{\mathcal{E}}[\mathbf{d}_{\bar{e}}^k] = \arg \min_p \|\mathcal{J}[\mathbf{d}_{\bar{e}}^k] \mathbf{p} - (\gamma_{(\bar{e}, \bar{r})} - \gamma_{(\bar{e}, p)}[\mathbf{d}_{\bar{e}}^k])\|_2^2, \quad (44)$$

where  $\tilde{\mathcal{E}}[\mathbf{d}_e^k]$  is the  $k$ -linearised functional, and  $\mathbf{p}^k$  denotes the search direction. (The factor of a half has been dropped as it does not affect the minimiser.) The problem equation (44) is solved for  $\mathbf{p}^k$  by setting the gradient of the linearised functional to zero by forming the normal equations

$$\nabla \tilde{\mathcal{E}}[\mathbf{d}_e^k] = \mathcal{H}^k \mathbf{p}^k - (\mathcal{J}^k)^T (\gamma_{(\bar{e}, \bar{p})} - \gamma_{(\bar{e}, p)}[\mathbf{d}_e^k]) = 0, \quad (45)$$

where  $\mathcal{J}^k := \mathcal{J}[\mathbf{d}_e^k]$ , and  $\mathcal{H}^k = (\mathcal{J}^k)^T \mathcal{J}^k$  is an approximation of the Hessian matrix in which the term of second order derivatives is neglected.

**4.3.1. Partial derivative (Jacobian) matrix.** The Jacobian matrix is in the form

$$\mathcal{J}[\mathbf{d}_e^k] = \left[ \frac{\partial \gamma_{(\bar{e}, p)}}{\partial \theta_e^k} [\theta_e^k, \varphi_e^k], \quad \frac{\partial \gamma_{(\bar{e}, p)}}{\partial \varphi_e^k} [\theta_e^k, \varphi_e^k] \right]. \quad (46)$$

Here, the partial derivatives are calculated using finite differences

$$\begin{aligned} \frac{\partial \gamma_{(\bar{e}, p)}}{\partial \theta_e^k} [\theta_e^k, \varphi_e^k] &= \frac{\gamma_{(\bar{e}, p)}[\theta_e^k + \Delta \theta_e^k, \varphi_e^k] - \gamma_{(\bar{e}, p)}[\theta_e^k, \varphi_e^k]}{\Delta \theta_e^k} \\ \frac{\partial \gamma_{(\bar{e}, p)}}{\partial \varphi_e^k} [\theta_e^k, \varphi_e^k] &= \frac{\gamma_{(\bar{e}, p)}[\theta_e^k, \varphi_e^k + \Delta \varphi_e^k] - \gamma_{(\bar{e}, p)}[\theta_e^k, \varphi_e^k]}{\Delta \varphi_e^k}, \end{aligned} \quad (47)$$

where  $\Delta \theta_e^k$  and  $\Delta \varphi_e^k$  are perturbations enforced on the two elements of  $\mathbf{d}_e^k$ . How these perturbations are chosen is important. It has been shown that the inverse problem of ray-linking is subject to instability or divergence because of its highly nonlinear nature, especially for 3D cases [57]. One important factor that affects the convergence is the accuracy of the Jacobian matrix [57], which is affected by the perturbations to the finite differences. A small perturbation gives a more accurate Jacobian matrix when  $\gamma_{(\bar{e}, p)}$  is differentiable with respect to the initial direction  $\mathbf{d}_e$ , but the Jacobian matrix can become singular or close to singular when  $\gamma_{(\bar{e}, p)}$  is nondifferentiable with respect to  $\mathbf{d}_e$  or very sensitive to changes in  $\mathbf{d}_e$ . This may occur, for example, in the presence of singularities, such as when the trajectory of a ray is tangent to inter-medium boundaries, or when a ray is perpendicular to  $\nabla n$ , and thus a small change in  $\mathbf{d}_e$  will lead to a large change in the position of the interception point of the ray with the detection surface  $\mathbb{S}$ . This issue, together with the low-dimensional nature of the ray-linking inverse problem, motivates using an adaptive approach for calculation of the Jacobian matrix. To facilitate this, each iteration of the ray-linking algorithm starts with a scaled perturbation in the form

$$\begin{aligned} \Delta \theta_e^k &= \tau \operatorname{sign}(\theta_e^k) \max \left( \operatorname{abs}(\theta_e^k), \frac{1}{2} \|\mathbf{d}_e^k\|_1 \right), \\ \Delta \varphi_e^k &= \tau \operatorname{sign}(\varphi_e^k) \max \left( \operatorname{abs}(\varphi_e^k), \frac{1}{2} \|\mathbf{d}_e^k\|_1 \right), \end{aligned} \quad (48)$$

where  $\tau$  is a small scalar. The perturbation is then increased recursively by controlling  $\tau$  until the singular values of  $\mathcal{H}^k$ ,  $\Lambda^k \in \mathbb{R}^2$ , satisfy the two conditions

$$\max(\Lambda^k) / \min(\Lambda^k) < \vartheta \quad \text{and} \quad \min(\Lambda^k) > \min(|\nabla \tilde{\mathcal{E}}[\mathbf{d}_e^k]|, \varsigma). \quad (49)$$

Here,  $\vartheta$  and  $\varsigma$  are large and small scalar values, respectively. The former condition enforces a

**Algorithm 3.** Adaptive smoothing calculation of well-conditioned Hessian and Jacobian matrices,  $\mathcal{J}^k$  and  $\mathcal{H}^k$ , for the *damped Gauss–Newton* approach to the ray-linking problem.

---

```

1: input:  $N_h$                                 ▷ Maximum number of iterations
2: input:  $\vartheta, \varsigma$                             ▷ Scalars in the conditions in equation (49)
3: input:  $\eta$                                 ▷ Integer factor > 1 by which  $\tau$  is recursively increased
4: initialise:  $\tau$                             ▷ Controls the perturbation, equation (48)
5: initialise:  $k_h = 0$                         ▷ Set the counter
6: calculate  $\mathcal{J}^k$  and  $\mathcal{H}^k$  using equations (46),
   (47), (48) and algorithm 2                ▷ Form Jacobian & Hessian
7: calculate the singular values  $\Lambda^k$  of  $\mathcal{H}^k$ 
8: while  $\Lambda^k$  does not satisfy equation (49)
   and  $k_h < N_h$  do                        ▷ Iterate until Hessian is well-conditioned
9:    $\tau \leftarrow \eta \tau$                       ▷ Increase the perturbation size
10:  calculate  $\mathcal{J}^k$  and  $\mathcal{H}^k$  using equations (46),
   (47), (48) and algorithm 2                ▷ Form Jacobian & Hessian
11:  calculate the singular values  $\Lambda^k$  of  $\mathcal{H}^k$ 
12:   $k_h \leftarrow k_h + 1$                     ▷ Increment the counter
13: end while
14: output:  $\mathcal{J}^k$  and  $\mathcal{H}^k$                 ▷ Return well-conditioned Jacobian and
                                                    Hessian matrices

```

---

bound on the ill-conditioning of the Hessian matrix, and the latter prevents very large search directions (see section 4.1 in [47]).

This scheme, here referred to as *adaptive smoothing*, is outlined in algorithm 3. The parameters  $\eta$  and  $N_h$  are user-adjusted parameters, but our numerical experience shows that the optimal values are independent of the particular object being imaged.

**4.3.2. Backtracking line search.** Having defined the Jacobian matrix  $\mathcal{J}^k$  and Hessian matrix  $\mathcal{H}^k$ , equation (45) can be solved for the search direction  $\mathbf{p}^k$ . Once this is done, a search is conducted along  $\mathbf{p}^k$  for a point that gives a sufficient descent step for  $\mathcal{E}[\mathbf{d}_e^k]$ :

$$\mathbf{d}_e^{k+1} = \mathbf{d}_e^k + \alpha^k \mathbf{p}^k, \quad (50)$$

where  $\alpha^k$  is the step length. Choosing  $\alpha^k$  is the *damping process* of the Gauss–Newton method, and is done here using a backtracking line search technique. The step length starts at  $\alpha = 1$  and is reduced recursively by a factor  $\beta \in (0, 1)$  until the following *Armijo* condition is satisfied:

$$\mathcal{E}(\mathbf{d}_e^k + \alpha^k \mathbf{p}^k) \leq \mathcal{E}(\mathbf{d}_e^k) + \mu \alpha^k \mathbf{p}^k \nabla \mathcal{E}(\mathbf{d}_e^k), \quad (51)$$

where  $\mu \in (0, 0.1]$  is a user-adjusted parameter. In addition, a limit was set on the maximum number of recursions in the backtracking loop. (The full Wolfe conditions also require a recursive calculation of  $\nabla \mathcal{E}(\mathbf{d}_e^k + \alpha^k \mathbf{p}^k)$  for each iteration  $k$ , which is costly, so only the Armijo rule was used here.)

Algorithm 3 can now be used to compute the initial ray direction to ensure ray-linking from the emission point  $\bar{e}$  to the reception point  $\bar{r}$ . The whole ray-linking procedure using this functional minimisation approach, equation (41), is outlined in algorithm 4. In this algorithm,  $\mathbf{d}_{(\bar{e},q)}^0$  and  $\mathbf{d}_{(\bar{e},q)}^*$  denote the initial guess and an optimal solution respectively for the initial direction of the ray for linearisation  $q$  of the UST inverse problem (section 2.3). The choice of initial direction  $\mathbf{d}_{(\bar{e},q)}^0$  is described above in section 4.2.

**Algorithm 4.** Ray-linking from emission point  $\bar{e}$  to reception point  $\bar{r}$  using a damped Gauss–Newton method (for linearisation  $q$  of the UST inverse problem).

---

```

1: input:  $\gamma_{(\bar{e},\bar{r})}$  ▷ Unit vector pointing to  $\bar{r}$  from  $\bar{e}$ 
2: input:  $N_{\text{link}}$  ▷ Maximum number of iterations
3: input:  $\varepsilon_{\text{link}}$  ▷ Minimum error threshold
4: initialise:  $d_e^k = d_{(\bar{e},q)}^0$  ▷ Set the initial ray direction
5: initialise:  $k = 0$  ▷ Set the counter
6: calculate  $\mathcal{E}(d_e^k)$  using equation (41) ▷ Evaluate the error functional
7: while  $\mathcal{E}(d_e^k) > \varepsilon_{\text{link}}$  and  $k < N_{\text{link}}$  do ▷ Iterate initial ray direction until  $\gamma_{(\bar{e},p)} = \gamma_{(\bar{e},\bar{r})}$ 
8:   calculate  $\mathcal{J}^k$  and  $\mathcal{H}^k$  using algorithm 3 ▷ Form Jacobian and Hessian matrices
9:   calculate  $p^k$  using equation (45) ▷ Find descent direction
10:  choose  $\alpha^k$  that satisfies equation (51) ▷ Backtracking line search to find step size
11:   $d_e^{k+1} \leftarrow d_e^k + \alpha^k p^k$  ▷ Update initial ray direction
12:  calculate  $\mathcal{E}(d_e^{k+1})$  using equation (41) ▷ Evaluate the error functional
13:   $k \leftarrow k + 1$  ▷ Increment the counter
14: end while
15: output:  $d_{(\bar{e},q)}^*$  ▷ Return the optimal initial ray direction

```

---

#### 4.4. Ray-linking using root-finding: quasi-Newton with box constraints

An analogous approach to solving equation (41) using the Gauss–Newton approach would be to solve the system of nonlinear equations (40) using the *Newton–Raphson* method [17]. This approach gives a sequence of linear equations

$$\mathcal{J}^k p^k + F[d_e^k] = 0, \quad (52)$$

where, compared to equation (40), the subscripts  $(\bar{e}, \bar{r})$  for the function  $F$  has been neglected for brevity. The Newton–Raphson method, like the Gauss–Newton approach, requires frequent calculation of the Jacobian, which is computationally costly. Each iteration of a Newton-type method (or a trust region method [7]), used for solving either equations (40) or (41), requires at least two additional function evaluations per iteration to calculate the Jacobian. (Note that if a higher order finite difference scheme is used to increase the accuracy over formulae equation (47), the number of function evaluations will increase accordingly.) Fortunately, there is a class of approaches for solving nonlinear equations such as equation (52)—*quasi-Newton* methods—that have the great advantage of not requiring Jacobian calculations. For nonlinear equations, quasi-Newton methods suggest a replacement of the sequence of equations equation (52) by derivative-free linear equations of the form

$$\mathcal{B}^k p^k + F[d_e^k] = 0, \quad (53)$$

where  $\mathcal{B}^k$  is a matrix that approximates  $\mathcal{J}^k$  using the two last updates. There are different formulas for forming the approximate Jacobian  $\mathcal{B}^k$ . Here, constrained variants of the BFGS formula and a Broyden-like scheme will be described, as they have been found to give good results.

**4.4.1. BFGS-like formula.** For updating the matrix  $\mathcal{B}$ , one way is to use a modified variant of the BFGS formula in the form

$$\mathcal{B}^{k+1} = \mathcal{B}^k + \tau^k \left( -\frac{\mathcal{B}^k s^k s^{kT} \mathcal{B}^k}{s^{kT} \mathcal{B}^k s^k} + \frac{y^k y^{kT}}{y^{kT} s^k} \right), \quad (54)$$

**Algorithm 5.** Adaptive smoothing calculation of the approximate Jacobian matrix,  $\mathcal{B}^k$ , for the *BFGS-like* approach to the ray-linking problem.

---

```

1: input:  $\mathcal{B}^k$                                 ▷ Latest Jacobian approximation
2: input:  $N_h$                                 ▷ Maximum number of iterations
3: input:  $\vartheta, \varsigma$                             ▷ Scalars in the conditions in equation (49)
4: input:  $\eta \in (0, 1)$                         ▷ Factor by which  $\tau$  is recursively decreased
5: initialise:  $\tau = 1$                           ▷ Weights the update to  $\mathcal{B}^k$ , equation (54)
6: calculate the singular values  $\Lambda^k$  of  $\mathcal{B}^k$ 
7: while equation (49) is not satisfied and  $k_h < N_h$  do  ▷ Iterate until Jacobian  $\mathcal{B}$  is well-conditioned
8:    $\tau \leftarrow \eta \tau$                                 ▷ Reduce the update weight
9:   calculate  $\mathcal{B}^{k+1}$  using equation (54)            ▷ Update the Jacobian approximation
10:  calculate the singular values  $\Lambda^k$  of  $\mathcal{B}^{k+1}$ 
11:   $k_h \leftarrow k_h + 1$                             ▷ Increment the counter
12: end while
13: output:  $\mathcal{B}^{k+1}$                                 ▷ Return the updated Jacobian approximation

```

---

where  $s^k = \mathbf{d}_{\bar{e}}^{k+1} - \mathbf{d}_{\bar{e}}^k$  and  $y^k = F(\mathbf{d}_{\bar{e}}^{k+1}) - F(\mathbf{d}_{\bar{e}}^k)$ .  $\mathbf{d}_{\bar{e}}^{k+1}$  is calculated using the update formula equation (50) and then projected onto a set of box constraints, as described in section 4.4.3 below. The two terms in the parentheses are two symmetric rank-one matrices, but their summation provides a rank-two update for  $\mathcal{B}$ . Setting the parameter  $\tau^k \in \{0, 1\}$  to 1 for all  $k$  gives the standard BFGS formula [21, 34].  $\mathcal{B}^k$ , in equation (53), needs to be invertible, so that we can solve for  $\mathbf{p}^k$ . For unconstrained optimisation, the Wolfe conditions may be used to ensure  $\mathcal{B}^k$  is positive definite, and therefore invertible, but they are not applicable to the root-finding problem here [34]. Modified BFGS approaches have been proposed that ensure positive definiteness [21, 72] but these approaches require at least one more function evaluation per iteration than vanilla BFGS, and in any case are applicable only to symmetric nonlinear equations, so not relevant here. A BFGS trust-region method has also been proposed for nonlinear equations [71], but it relies on the Jacobian matrix which is what we are trying to avoid.

Here the nonsingularity of the matrix  $\mathcal{B}$  is ensured using adaptive smoothing, i.e. iterative adjustment of  $\tau$  as described in algorithm 5 below. For the Broyden-like method below, section 4.4.2, the parameter  $\tau$  has been included in previous theoretical work [35, 45, 55]. For the BFGS-like scheme, to the best of our knowledge, this is not the case but in our experience the method works well numerically. (This may be because for the optimisation case, for which BFGS is more commonly used, the positive definiteness of the update of  $\mathcal{B}$  can be ensured using the Wolfe conditions.) Indeed, it is shown in the results section below that this BFGS-like formula, when combined with a box constraint on the sequence of initial directions, is much more efficient for ray-linking than the Gauss–Newton method above, which relies on the costly calculation of the Jacobian matrix.

**4.4.2. Broyden-like formula.** Another way to approximate the Jacobian matrix  $\mathcal{B}^k$ , which is popular for solving nonlinear systems, is to use a Broyden-like formula [34, 35, 45, 55]. Initially,  $\mathcal{B}^0 = \mathcal{J}^0$ , where  $\mathcal{J}^0$  is calculated using equation (47), with a large enough perturbation to ensure nonsingularity, then for  $k \geq 1$  this approximation is updated using the formula

$$\mathcal{B}^{k+1} = \mathcal{B}^k + \tau^k \frac{(y^k - \mathcal{B}^k s^k) s_k^T}{s_k^T s^k}, \quad (55)$$

where  $|\tau^k - 1| \leq \bar{\tau}$  with  $\bar{\tau} \in (0, 1)$  a fixed scalar [34, 35, 45, 55]. When  $\tau^k = 1$  for all  $k$ , (55) becomes *Broyden's rank one* formula [9], and in this case a nonsingular  $\mathcal{B}^k$  does not necessarily ensure a nonsingular  $\mathcal{B}^{k+1}$ . For our application, this can occur when a small change in the initial

**Algorithm 6.** Adaptive smoothing calculation of the approximate Jacobian matrix,  $\mathcal{B}^k$ , for the *Broyden-like* approach to the ray-linking problem.

---

```

1: input:  $\mathcal{B}^k$                                 ▷ Latest Jacobian approximation
2: input:  $N_h$                                 ▷ Maximum number of iterations
3: input:  $\vartheta, \varsigma$                             ▷ Scalars in the conditions in equation (49)
4: input:  $\eta_c \in (0, \bar{\tau})$                     ▷ Magnitude of increment to  $\eta$ 
5: input:  $\bar{\tau}$                                 ▷ Limits size of weight  $\tau$ 
6: initialise:  $\eta = 0$                         ▷ Sets initial size of weight update
7: initialise:  $\eta_s = 1$                     ▷ Sign of weight update
8: initialise:  $\tau = 1$                         ▷ Weights the update to  $\mathcal{B}^k$ , equation (55)
9: calculate the singular values  $\Lambda^k$  of  $\mathcal{B}^k$ 
10: while  $\Lambda^k$  does not satisfy equation (49)
    and  $|\tau - 1| \leq \bar{\tau}$  and  $k_h < N_h$  do    ▷ Iterate until Jacobian  $\mathcal{B}$  is well-conditioned
11:   if  $\eta_s > 0$  then
12:      $\eta \leftarrow \eta + \eta_c$                 ▷ Increase weight update magnitude
13:   end if
14:    $\eta_s \leftarrow -\eta_s$                     ▷ Alternate weight update sign
15:    $\tau = 1 + \eta_s \eta$                         ▷ calculate the weight
16:   calculate  $\mathcal{B}^{k+1}$  using equation (55)    ▷ Update the Jacobian approximation
17:   calculate the singular values  $\Lambda^k$ 
18:    $k_h \leftarrow k_h + 1$                     ▷ Increment the counter
19: end while
20: output:  $\mathcal{B}^{k+1}$                         ▷ Return the updated Jacobian approximation

```

---

angle leads to a large change in the end point of the ray [57]. To avoid this issue, *adaptive smoothing* is used, i.e. the singular values  $\Lambda$  of the matrix  $\mathcal{B}$  are controlled using a scalar  $\tau$  initialised as 1, and gradually moved away up to a neighborhood of radius  $\bar{\tau}$  until the conditions in equation (49) are satisfied. (In the second condition the gradient is replaced by  $\mathcal{E}[\mathbf{d}_e^k]$ . See section 4.1 in [47].) This is outlined in algorithm 6. Note that adaptive smoothing via a recursive adjustment of  $\tau$ , as used in algorithms 5 and 6, is very cheap compared to the approach used in algorithm 3, as it does not require any additional function evaluations.

**4.4.3. Box constraints.** The reason for choosing the Gauss–Newton method to solve the minimisation problem equation (41) in section 4.3 is the fast convergence rate of Newton-type methods for unconstrained minimisation problems. Similarly, the convergence of quasi-Newton methods for solving equation (41) is well-established [34], but for nonlinear equations such as equation (40), less progress has been made [34, 72], a significant challenge is the lack of line-search techniques for algorithms that avoid costly derivative calculations [34]. A good review of derivative-free line search techniques is given in [34] and the reader is also referred to [12, 20, 36]. However, these algorithms are typically conservative, and the high number of function evaluations required per iteration risks losing the computational advantage of not calculating gradients. Here, to achieve higher efficiency, an undamped variant of the update formula equation (50) was used, with  $\alpha^k = 1$  for all  $k$ . If the search direction  $\mathbf{p}^k$  obtained from equation (53) is not a descent direction for  $\mathcal{E}$  at  $\mathbf{d}_e^k$ , then the sequence  $\mathbf{d}_e^k, k = 0, 1, \dots$  is likely to diverge. If the update direction  $\mathbf{p}^k$  is a descent direction, then, with an undamped scheme, it is still possible that the update step is large enough for the sequence to diverge. To ameliorate this, a box constraint of the form  $\Gamma = \{\mathbf{d}_e \in \mathbb{R}^2 \mid \mathbf{l} \leq \mathbf{d}_e \leq \mathbf{u}\}$ , was enforced, where  $\mathbf{l} \in \mathbb{R}^2$  and  $\mathbf{u} \in \mathbb{R}^2$  are specified lower and upper bounds, and the inequalities are component-wise. The bounds are chosen as angular intervals containing an initial guess. Since  $\mathbf{p}^k$  does not guar-



**Algorithm 7.** Ray-linking from emission point  $\bar{e}$  to reception point  $\bar{r}$  using a quasi-Newton method (for linearisation  $q$  of the UST inverse problem).

---

```

1: input:  $\gamma_{(\bar{e},\bar{r})}$  ▷ Unit vector pointing to  $\bar{r}$  from  $\bar{e}$ 
2: input:  $l, u$  ▷ Lower and upper bounds on the direction vector  $d_e^k$ 
3: input:  $N_{\text{link}}$  ▷ maximum number of iterations
4: input:  $\varepsilon_{\text{link}}$  ▷ Minimum error threshold
5: initialise:  $d_e^k = d_{(\bar{e},q)}^0$  ▷ Set the initial ray direction
6: initialise:  $k = 0$  ▷ Set the counter
7: calculate  $B^0 = J^0$  using equation (47) ▷ Form the approximate Jacobian
8: calculate  $\mathcal{E}(d_e^k)$  using equation (41) ▷ Evaluate the error functional
9: while  $\mathcal{E}(d_e^k) > \varepsilon_{\text{link}}$  and  $k < N_{\text{link}}$  do ▷ Iterate initial ray direction until  $\gamma_{(\bar{e},p)} = \gamma_{(\bar{e},\bar{r})}$ 
10:   calculate  $p^k$  using equation (53) ▷ Update to direction vector
11:   calculate  $\bar{p}^k$  using  $l$  and  $u$  ▷ Box constraints
   using equations (56) and (57)
12:   update  $d_e^{k+1} \leftarrow d_e^k + \bar{p}^k$  ▷ Apply direction vector update
13:   calculate  $s^k = d_e^{k+1} - d_e^k$  ▷ Difference used in Jacobian-update formula
14:   calculate  $F(d_e^{k+1})$  using equation (40) ▷ Difference between  $\gamma_{(\bar{e},\bar{r})}$  and  $\gamma_{(\bar{e},p)}$ 
15:   calculate  $y^k = F(d_e^{k+1}) - F(d_e^k)$  ▷ Difference used in Jacobian-update formula
16:   calculate  $B^{k+1}$  using ▷ Update approximation Jacobian
   algorithms 5 (BFGS) or 6 (Broyden)
17:   calculate  $\mathcal{E}(d_e^{k+1})$  using equation (41) ▷ Evaluate the error functional
18:    $k \leftarrow k + 1$  ▷ Increment the counter
19: end while
20: output:  $d_{(\bar{e},q)}^*$  ▷ Return the optimal initial ray direction

```

---

antee that the update  $d_e^k + p^k$  is a feasible point,  $d_e^k + p^k$  is projected onto the set  $\Gamma$ . To do this, a vector  $\psi$  is defined as

$$\psi_i^k = \begin{cases} \zeta \left( \frac{l_i - d_{(\bar{e},i)}^k}{p_i^k} \right), & \text{if } d_{(\bar{e},i)}^k + p_i^k < l_i, \\ \zeta \left( \frac{u_i - d_{(\bar{e},i)}^k}{p_i^k} \right), & \text{if } d_{(\bar{e},i)}^k + p_i^k > u_i, \\ 1, & \text{otherwise,} \end{cases} \quad (56)$$

where  $\zeta \in (0, 1)$  is a scalar, and  $i \in \{1, 2\}$  denotes the components. The update direction is then modified using

$$\bar{p}_i^k = \text{sign}(\psi_i^k) \left( \max \left( \text{abs}(\psi_i^k), \kappa \right) \right) p_i^k, \quad (57)$$

where  $\kappa$  is a very small scalar, used to prevent the direction becoming stuck at one or other of the limits. Note that because this minimisation is low-dimensional (two dimensions), a violation of the box constraint for even one component means that  $p^k$  may not be a descent direction, or may be a descent direction towards a local minimum far from the initial guess. In these cases, the update is projected to the interior of the feasible set, rather than the active set, using the parameter  $\zeta$ , which is heuristically chosen to be 0.5. The whole ray-linking procedure using the Quasi-Newton approaches for solving the root-finding problem, equation (40), is outlined in algorithm 7.

## 5. Numerical results

Having laid out our approach to the UST inverse problem of estimating the refractive index in section 2.3, and, above, the details of the ray tracing and ray-linking algorithms that will be used to achieve this, this section will describe numerical experiments demonstrating the effectiveness of this approach to 3D UST.

First, in section 5.1, the accuracy of the ray tracing algorithms, algorithms 1 and 2, will be demonstrated on both continuous (analytically-known gradient) and discretised refractive index maps based on the Maxwell fish-eye phantom. In section 5.1 these algorithms are used in calculations of acoustic length (the integral of the refractive index along a ray), and it is shown that these calculations are dominated by the grid-to-ray interpolation error and therefore algorithm 2 is more efficient for solving the UST problem.

Section 5.2 describes the more complex breast phantom that was used in the subsequent UST study, and how the measured time series were simulated. (The TOF picking method used to calculate the first-arrival of the signals from the simulated time series is described in an appendix A.) Finally, in section 5.3, all the previous work is brought together to demonstrate how to solve the problem of estimating the refractive index (sound speed) in 3D UST of the breast. Our bent ray approach is compared to an approach using straight-rays, and the different ray-linking algorithms are compared with respect to accuracy and computational time.

### 5.1. Numerical validation of ray tracing algorithms

**5.1.1. Maxwell's fish-eye lens phantom.** The numerical validation of the ray tracing algorithms was performed using a refractive index field for which ray paths across it are known analytically. A well-known refractive index function for this purpose is Maxwell's 'fish-eye lens', which is defined as

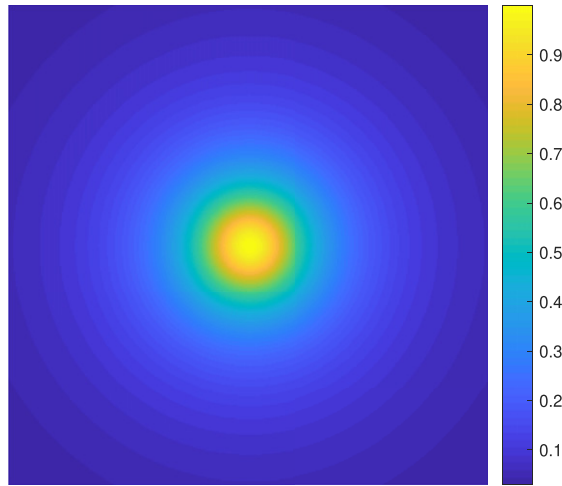
$$n(\mathbf{x}) = \frac{n_o}{1 + (\frac{\mathbf{x}}{a})^2}, \quad (58)$$

where  $n_o$  denotes the refractive index at the origin of the Cartesian coordinates ( $\mathbf{x}_o = [0, 0, 0]^T$ ). (Here, we set  $n_o = 1$  and  $a = 1$ .) [4, 26, 50]. This phantom has two interesting and useful properties which will be used below to test the ray tracing algorithms' accuracy (figure 3).

A ray starting from a point  $p_1$  satisfying  $|\mathbf{x}_{p_1}| = a$  will travel along a circular path including the mirror point  $p_2$  with respect to  $o$ , i.e.,  $\mathbf{x}_{p_2} = -\mathbf{x}_{p_1}$ , and will return to the initial point on completion of the circle. Then it is clear that:

- (a) The particular circular path followed is tangent to the initial ray direction, and
- (b) The acoustic length along the ray on completion of the corresponding circle will equal twice the acoustic length along the line segment  $p_1 o p_2$ .

**5.1.2. Ray-path convergence.** In this section, the first property of rays propagating through the fish-eye phantom will be exploited. A family of 100 rays were initialised at position  $\mathbf{x}_{p_1} = [0, 0, a]^T$  with initial directions tangent to a sphere centred at  $\mathbf{x}_{o'} = [a, a, 0]^T$ . The initial directions of the rays were chosen to lie in a plane normal to a line segment  $p_1 o'$ , and were spaced evenly to form a complete Rodriguez rotation with respect to the radial line  $p_1 o'$ . (In other words, 100 rotations with angles between 0 and  $2\pi$  were applied to a reference initial direction  $(-1/\sqrt{3})[1/\sqrt{2}, 1/\sqrt{2}, \sqrt{2}]^T$ , for which the axis of rotation was the line segment  $p_1 o'$ .) Analytically, all the rays will travel along the surface of the sphere and finally intercept the point  $p_1$ . The extent to which the paths calculated using the ray tracing algorithms differ from these paths can therefore be used as a metric to test the accuracy of the algorithms.



**Figure 3.** Maxwell's fish-eye lens, used as a phantom for the numerical validation of the ray tracing algorithms. The phantom, which has rotational symmetry, is shown for slice  $z = 0$ .

The rays with the above initialisations were propagated using algorithms 1 and 2 and terminated when they reached a neighborhood of  $p_1$  with a radius  $\Delta s$ , where  $\Delta s$  is the step size. The mean radial deviation of the calculated ray points from the sphere with radius  $p_1 o'$  is then computed:

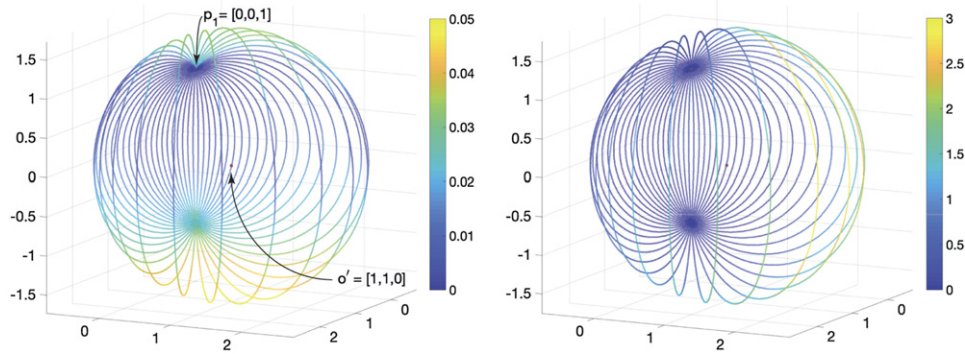
$$\text{RE}_{\text{rd}}(p_1) = \text{mean} \left( \frac{1}{M_{(p_1, \tilde{p}_1)}} \sum_{m=1}^{M_{(p_1, \tilde{p}_1)}} \frac{\text{abs}(|\mathbf{x}_{(s_m; p_1, \tilde{p}_1)} - \mathbf{x}_{o'}| - \sqrt{3}a)}{\sqrt{3}a} \right) \times 100. \quad (59)$$

Note that  $|\mathbf{x}_{p_1} - \mathbf{x}_{o'}| = \sqrt{3}a$  and the mean operator indicates averaging across the family of 100 rays.

First, the rays were traced using analytical  $n$  and  $\nabla n$  (calculated from equation (58)) in order to remove grid-to-ray interpolation effects associated with discretisation. The rays were traced for step sizes  $\Delta s = 2^{(-4.5:0.5:3)}$  times the reference length (in Matlab notation). The reference length was chosen to be  $2 \times \pi a / 360$ , which is equivalent to 1 degree arc length along a circle centred at  $o$ .

Figure 5(a) shows the convergence of the error  $\text{RE}_{\text{rd}}$  as the step size  $\Delta s$  is decreased. It is clear that the dual-update ray tracing algorithm, algorithm 1, exhibits a quadratic convergence rate for  $\text{RE}_{\text{rd}}$ , and the traditional mixed-step algorithm, algorithm 2, provides a linear convergence rate, as expected given the accuracy of the respective estimates of the direction vector updates (section 3.3).

Second, to demonstrate the effect of the grid-to-ray interpolation on the convergence, i.e. the effects using a discretised  $n$  and  $\nabla n$ , the same experiment was repeated on a computational grid with spacing equal to the reference length. Figure 4 shows the analytical ray paths through the fish-eye lens phantom, and the color indicates the deviation of the computed ray paths from these exact solutions. For this illustration, the step length  $\Delta s$  was set equal to the reference length (grid spacing).



**Figure 4.** Testing ray tracing algorithms using Maxwell's fish-eye lens phantom. The analytical ray paths with the color indicating the deviation from it,  $\text{abs}(|\mathbf{x}_{(sm;p_1,\hat{p}_1)} - \mathbf{x}_{o'}| - \sqrt{3}a) / \sqrt{3}a \times 100$  with  $a = 1$ , for the two algorithms, (a) dual-update algorithm (b) mixed-step algorithm. Note the different color scales. Rays for only half of the ray launch angles are shown for clarity. The step length  $\Delta s$  was set equal to the reference length (grid spacing) for this illustration.

Figure 5(a) shows the convergence of the error  $\text{RE}_{\text{rd}}$  in this case as the step size  $\Delta s$  is decreased. As expected, for larger step sizes the discretisation effects are negligible and the algorithms provide the same convergence rate as using a continuous  $n$ . However, the discretisation errors become more important as  $\Delta s$  decreases, eventually clearly dominating the error for algorithm 1 for small step sizes.

**5.1.3. Acoustic-length convergence.** In this section, the second property of the fish-eye phantom will be exploited. A family of 100 rays were initialised at point  $p_1$  with initial directions set as described above. Therefore, in the ideal case, all these rays would travel along the surface of the sphere, pass through the mirror point  $p_2$ , and finally intercept point  $p_1$ , as in figure 4. Numerically, when the rays were propagated using algorithms 1 and 2, they passed close to  $p_2$  and then back to close to  $p_1$ . They were terminated when they reached a neighborhood of  $p_1$  with a radius  $\Delta s$ , where  $\Delta s$  is the step size. This final point was then discarded (as it may be beyond  $p_1$ ) and the path was completed by linking the previous ray point to  $p_1$  using a straight line. The mean error in the acoustic length was calculated using

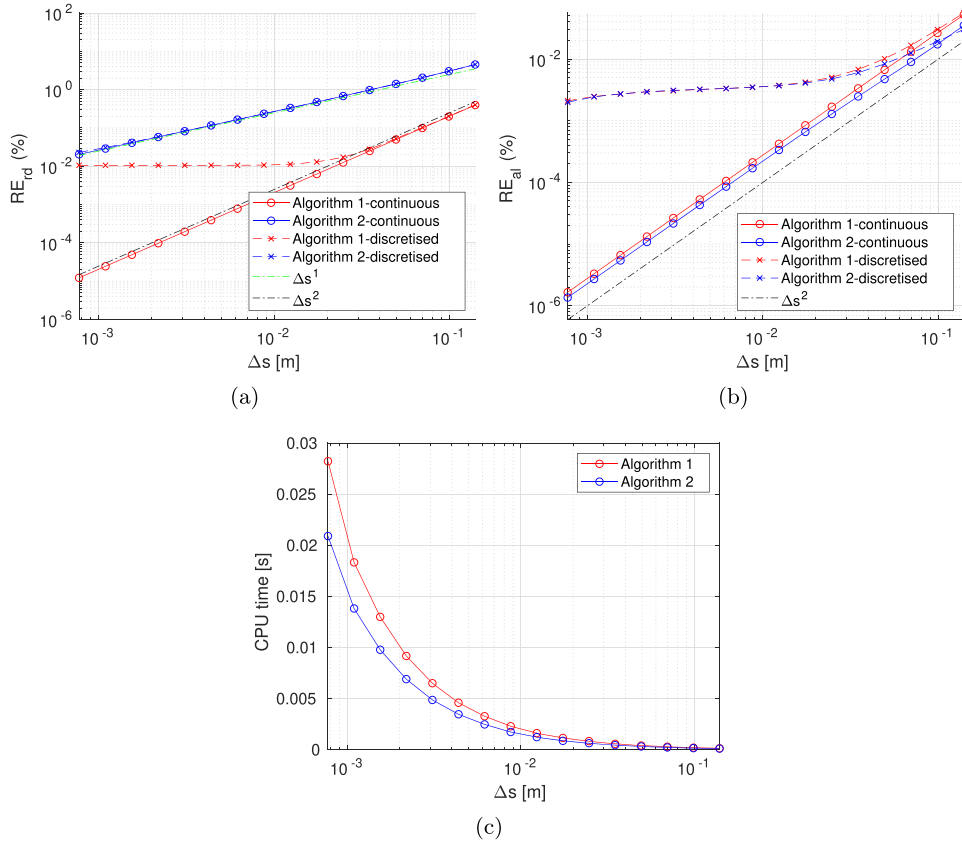
$$\text{RE}_{\text{al}}(p_1) = \text{mean} \left( \frac{\text{abs}(\bar{L}_{p_1} - L_{\text{true}})}{L_{\text{true}}} \right) \times 100, \quad (60)$$

where the acoustic length along the rays  $\bar{L}_{p_1}$  was calculated using equation (36), and also  $L_{\text{true}}$ , which is twice the acoustic length between  $p_1$  and  $p_2$  calculated analytically.

Figure 5(b) shows the error  $\text{RE}_{\text{al}}$  as the step size  $\Delta s$  is decreased for analytical and discretised  $n$ . Both ray tracing algorithms exhibit a quadratic convergence rate for  $\text{RE}_{\text{al}}$ , dominated by the error in the numerical integration equation (36). Figure 5(b) also shows that for larger  $\Delta s$  the grid-to-ray interpolation effects due to a discretisation of  $n$  is negligible, but the interpolation errors dominate for small  $\Delta s$ , and lead to a saturation in the reduction of  $\text{RE}_{\text{al}}$  with a decrease in  $\Delta s$ .

## 5.2. Ultrasound tomography: data simulation

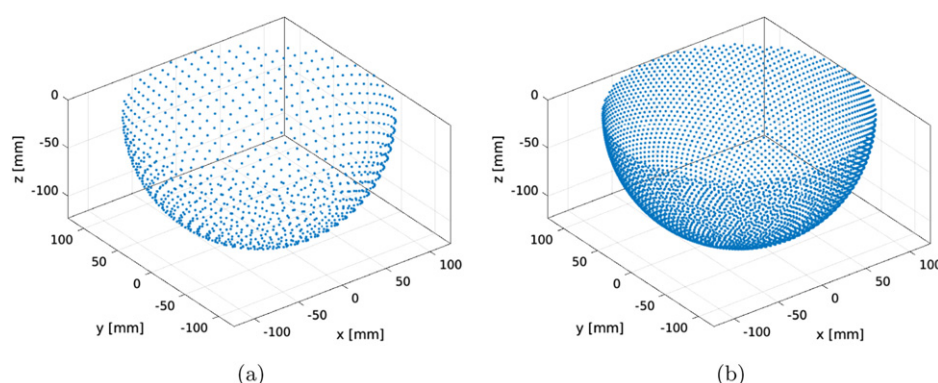
**5.2.1. Imaging system.** Ultrasound tomography data was simulated for an imaging system consists of 1024 emitters and 4048 receivers uniformly distributed over a hemispherical



**Figure 5.** Comparison of ray-tracing algorithms 1 and 2. (a) Mean error in ray-path (radius) deviation for continuous (analytical) and discretised refractive index. (b) Mean error in the acoustic length of the ray paths (integral of refractive index). (c) Mean CPU time per ray using continuous (analytical)  $n$ . Algorithm 2 was used in the image reconstructions in section 5.3 because it is faster for the same error in the acoustic length.

surface (bowl) of radius  $R = 12.35$  cm using a golden section method. The emitters and receivers were simulated as points as shown in figures 6(a) and (b), respectively. (Some practical systems use a coarser distribution of transducers in combination with translations and rotations of the bowl to provide an equivalent amount of data [18], as mentioned in the introduction. Because our proposed image reconstruction approach does not consider the correlation between neighboring transducers, it would still be applicable when the transducers are more coarsely spaced. Note that, unlike with fixed emitters, in the rotating setting the spacing between emitters can be varied by changing the rotation angle.

**5.2.2. Digital breast phantom.** A 3D digital phantom mimicking the anatomical properties of the breast [40] was used in this study. The sound speed was set to a range between  $1470 \text{ m s}^{-1}$  and  $1580 \text{ m s}^{-1}$ . A frequency-dependent absorption of the form  $\alpha = \alpha_0 f^y \text{ dB cm}^{-1}$  was included, with the coefficient  $\alpha_0 = 0.75 \text{ dB MHz}^{-y} \text{ cm}^{-1}$  for the breast tissue and  $\alpha_0 = 0.6 \text{ dB MHz}^{-y} \text{ cm}^{-1}$  for other tissue types. Here, the frequency power  $y$  was set to 1.4.  $f$  here is the frequency in MHz. The computational grid consisted of  $522 \times 522 \times 272$  grid points ( $26.1 \times 26.1 \times 13.6 \text{ cm}^3$ ) and a grid spacing of  $5 \times 10^{-2} \text{ cm}$  along all the Cartesian



**Figure 6.** The hemispherical imager bowl with (a) 1024 emitters (b) 4048 receivers.

coordinates. With this grid spacing and sound speed distribution, the maximum frequency supported by the grid is 1.47 MHz. Several slices through the phantom are shown in the left columns in figures 10 and 11.

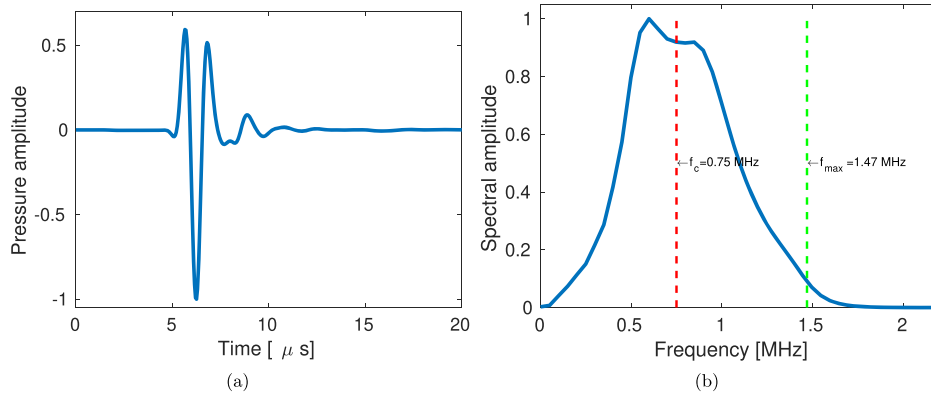
**5.2.3. Simulating time series data and time-of-flight picking.** A  $k$ -space pseudospectral method ( $k$ -wave) was used for simulation of the acoustic pressure time series data [63]. The emission and reception points were placed on the computational grid using nearest-neighbor interpolation. Two sets of simulations were performed, the first with the bowl filled with just water (sound speed  $1500 \text{ m s}^{-1}$ ), and the second with the breast phantom in the water. To simulate the data, each emitter was individually driven by an excitation pulse, and the set of acoustic pressure time series induced at the receivers were recorded simultaneously. This was repeated for each emitter. The pressure time series were recorded at 4245 time points with a sampling rate of 20 MHz, and additive white Gaussian noise was added to give a 40 dB signal-to-noise ratio. The computational time for a simulation of the two sets of data using  $k$ -Wave's C++ GPU code [63] on eight NVIDIA Tesla P40 Pascal GPUs was about 4.5 d.

The output of practical ultrasound transducers covers a finite (and typically quite limited) bandwidth, and the field produced becomes more directional at high frequencies. In choosing transducers for an imager, therefore, a trade-off must be made between the range of frequencies in the excitation pulse and the directionality of the detectors. In practical breast imaging systems, because acoustic absorption attenuates the higher frequencies preferentially, lower frequencies tend to be preferred, (typically below 5 MHz). The system modelled here uses relatively low frequencies (0.75 MHz centre frequency) and assumes the emitters and detectors are omni-directional. The excitation pulse and its spectrum are shown in figures 7(a) and (b).

There are three principal, practical, reasons for choosing this frequency range for the simulations. First, to match an experimental imager that is currently under construction. Second, to limit the time taken to simulate the 3D data to about few days given the available hardware. Third, when TOF images, such as we are producing here, are used as a starting point for full-wave inversions, they must be able to use the time series recorded for the full-wave inversions. When performing 3D image reconstruction with full-wave methods it is currently impractical to use higher frequencies over volumes of this size as the forward and adjoint calculations are too computationally demanding. Lower frequencies, however, can be modelled.

It is immediately clear that the effect of the chosen frequency range is that the pulse is not tightly confined in time. When it has propagated through the heterogeneous breast medium from the emitter to the detector, the long pulsewidth makes it harder to pick the time-of-arrival





**Figure 7.** (a) The excitation pulse, and (b) its frequency spectrum. The red dash line shows the centre frequency, and the green dash line shows the maximum frequency supported by the computational grid. (Any energy at frequencies above the maximum supported frequency is lost and does not propagate in the model.)

of the pulse at the detector. This is a non-trivial challenge in practice, and there are many methods devoted to accurate TOF picking. The method used here, and described in the appendix A, was found to be robust for this simulated breast imaging time series data.

It is important to note that the performance of ray-based approaches relying on first-arrival picking depends strongly on the properties of the excitation pulse. The decision to use lower frequency excitation pulses therefore has the effect of introducing more uncertainty into the TOF data that is used as the input to the image reconstruction. When it is possible to use excitation pulses with higher frequencies and wider bandwidths, the TOF picking would be expected to be more precise, having a knock-on beneficial effect on the reconstructed images.

### 5.3. Ultrasound tomography: image reconstruction procedure

The grid used for reconstruction consisted of  $262 \times 262 \times 138$  grid points with a spatial spacing of 1 mm along all the Cartesian coordinates. (This is different from the grid used to simulate the data to avoid the *inverse crime*, although this study does not use the same model for data generation and image reconstruction anyway.)

Prior to the image reconstructions, a region of interest (ROI) that encompassed the heterogeneous breast region was found, and the refractive index outside this region was set to 1. To define the ROI, a rough image was computed using straight rays and a binary image segmentation scheme based on an edge-based active contour method [10] was applied on the reconstructed low-contrast image to separate the breast region (ROI) from non-breast.

The initial guess for the refractive index was set to  $\bar{n} = 1$ , which corresponds to the known sound speed in water ( $1500 \text{ m s}^{-1}$ ). For each emitter, rays are computed only for receivers that are at least 8 cm from the emitter, based on the idea that a ray linking closer points will not hit the breast region, and is thus not useful in the image reconstruction. The image reconstructions were performed in Matlab R2019a (The MathWorks, Inc.) using one 8-core Xeon E5-2620 v4 2.1 GHz CPU.

**5.3.1. Straight-ray approach.** For comparison with the bent-ray approach, the breast phantom was reconstructed using straight rays, i.e. assuming that the medium is non-refracting and so the rays travel along straight lines between the emission and reception points. To implement



this, the initial direction  $\mathbf{d}_e$  in algorithm 2 was set to the unit vector pointing from the emission point to the reception point, and the ray's path was calculated using  $\nabla n = 0$  and  $\Delta s = 1$  mm, the grid spacing for image reconstruction. Because for this case, the forward operator  $A$  is linear, the refractive index was found by minimising the least squares problem equation (4) using a steepest descent algorithm, as explained in section 2.3. The sequence of reconstructed images were calculated using a constraint for  $\bar{n}$  outside the ROI, and the difference from the true refractive index was recorded after every 40 steepest descent iterations for comparison with the bent-ray approaches. The straight-ray inversion algorithm was terminated when the stopping condition in equation (7) was satisfied.

**5.3.2. Bent-ray approaches.** In the bent-ray approaches, the UST inversion for the refractive index was solved as a sequence of linearised minimisation problems, equation (6), as described in section 2.3. For each linearised subproblem  $q$ , the Jacobian matrix  $J$  was formed as described in section 3.4.3 using the ray paths between all sufficiently separated emitter–receiver pairs. Each linear subproblem was terminated after 400 steepest descent iterations.

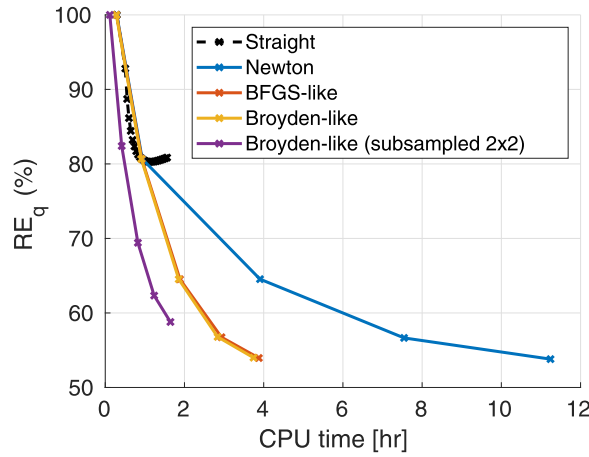
Algorithm 2 was used with  $\Delta s = 1$  mm to trace rays between each pair of emitters and receivers. This was implemented with  $\nabla n$  zero outside the ROI, assuming the medium outside the ROI is homogeneous, to reduce the computational cost. For the first iteration in the UST problem, iteration  $q = 0$ , the refractive index was set to be homogeneous ( $\bar{n} = 1$ ) and thus the first traced ray reaches the corresponding reception point directly (as in the straight-ray case). For subsequent iterations, the discretised refractive index was smoothed by convolving with a cube with sidelength equaling five grid points in all Cartesian directions.

For each ray, a ray-linking algorithm was then used to match the end point of the rays to the corresponding reception points. Images were reconstructed for all three ray-linking approaches. The ray-linking performances of the three ray-linking approaches are compared in section 5.4 below, and the final reconstructed images using the fastest approach are given in section 5.5.

#### 5.4. Comparison of ray-linking algorithms

The image reconstructions described above were performed for all three ray-linking algorithms, and the accuracy and convergence rates with which the algorithms honed in on the target reception points were compared. All the rays were traced as described in section 5.3.2 above, and, for each ray, one of the ray-linking algorithms was used to match the end point of the ray to its corresponding reception point. All the ray-linking algorithms were terminated when the residual function  $\mathcal{E}[\mathbf{d}_e^k]$  reached a tolerance  $\varepsilon_{\text{link}} = 1 \times 10^{-6}$ , or if the number of ray-linking iterations  $k$  exceeded  $N_{\text{link}} = 100$ .

**5.4.1. Gauss–Newton.** Ray-linking with the Gauss–Newton approach was performed using algorithm 4. For each iteration  $k$ , algorithm 3 was used to adjust the perturbations enforced on the initial angles for calculation of the finite differences, equation (47), in the Jacobian matrix  $\mathcal{J}$  via controlling the parameter  $\tau$ . The algorithm was initialised by  $\tau = 1 \times 10^{-15}$  for all  $k$ , and was recursively increased by a factor  $\eta = 10$  until the conditions in equation (49) were satisfied, with  $\vartheta = 1 \times 10^4$  and  $\varsigma = 1 \times 10^{-4}$ . This loop was unconditionally terminated when the associated iteration number  $k_h$  became greater than  $N_h = 5$ , i.e., when an increase of  $\tau$  up to  $10^{-10}$  still does not sufficiently reduce the ill-conditioning of  $\mathcal{H}^k$ . In these rare cases, the ray-linking problem was restarted using an initial  $\tau$  for all  $k$  10 times greater than the previous attempt. (For evaluation purposes, the iteration number  $k$  was not reset.) After calculation of a well-conditioned Hessian matrix  $\mathcal{H}^k$ , the initial direction was updated using equation (50) with a step length  $\alpha^k$  satisfying equation (51) with  $\mu = 1 \times 10^{-4}$ .



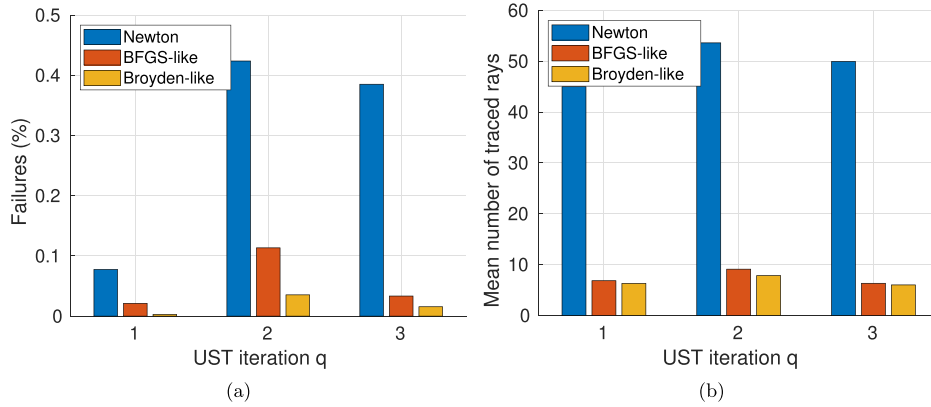
**Figure 8.** Accuracy of the reconstructed images against the computation time (hours) for the various reconstruction approaches, the relative error at each iteration  $RE_q$ .

**5.4.2. Quasi-Newton.** The quasi-Newton ray-linking approaches were performed using algorithm 7. The matrix  $\mathcal{B}^0$  was calculated using equation (47) with a fixed perturbation of  $1 \times 10^{-6}$  for both components. The update directions were bounded using upper and lower bounds  $\mathbf{l} = \mathbf{d}_{(0,q)}^0 - 0.2$  radians and  $\mathbf{u} = \mathbf{d}_{(0,q)}^0 + 0.2$  radians for all  $q$ , and  $\zeta = 0.5$ ,  $\kappa = 1 \times 10^{-6}$ . For the *BFGS-like* approach, the approximate derivative matrix  $\mathcal{B}^k$  was updated using algorithm 5 with the parameter  $\tau$  initialised by 1 and recursively reduced by a factor  $\eta = 0.5$  until  $\mathcal{B}^k$  satisfied the conditions in equation (49). (The conditions used the same parameters as for the Gauss–Newton method except that the objective function value  $\mathcal{E}[\mathbf{d}_{\bar{e}}^k]$  was replaced by its gradient.) The smoothing loop was unconditionally terminated if the iteration number  $k_h$  exceeded  $N_h = 20$ .

For the *Broyden-like* approach, the derivative matrix  $\mathcal{B}^k$  was updated using algorithm 6. The parameter  $\tau^k$  was initialised as 1 and changed within a neighborhood  $\bar{\tau} = 0.1$  with an increment parameter  $\eta_c = \pm 0.01$  until the conditions in equation (49) were satisfied with the same parameters as used for the BFGS-like approach. Figure (8) shows the accuracy of the UST reconstruction algorithms for the iterations  $q = -1$  used for defining the ROI,  $q = 0$  which applies straight rays on initial guess  $\bar{n}_0 = 1$ , and  $q = 1, 2, 3$  using different ray-linking approaches.

**5.4.3. Numerical results.** Figure 9(a) shows the fraction of the *refracted* rays for which the end point failed to reach sufficiently close to the reception point within the maximum permissible number of iterations. The failures were calculated as a percentage of the number of rays that hit the breast phantom and were therefore refracted, not as a percentage of all emitted rays. (If the non-refracted rays passing through the water only were included, this percentage would be much lower.) The results are shown for iterations  $q = 1, 2, 3$ . Subproblem  $q = 0$ , in which all the rays are straight rays, does not appear in figure 9(a), and the stopping criterion of our UST inverse problem, equation (7), was satisfied at iteration  $q = 3$ .

At the start of each iteration  $q$ , the rays need to be initialised. Rays previously successfully linked (on iteration  $q - 1$ ) used the same initial direction for iteration  $q$ . For rays that failed to link previously, a decision was made between using the last initial direction reached on iteration  $q - 1$ ,  $\mathbf{d}_{(\bar{e},q-1)}^{\text{link}}$ , and the first one,  $\mathbf{d}_{(\bar{e},q-1)}^0$ , the ray giving the smaller value of the functional  $\mathcal{E}$ ,



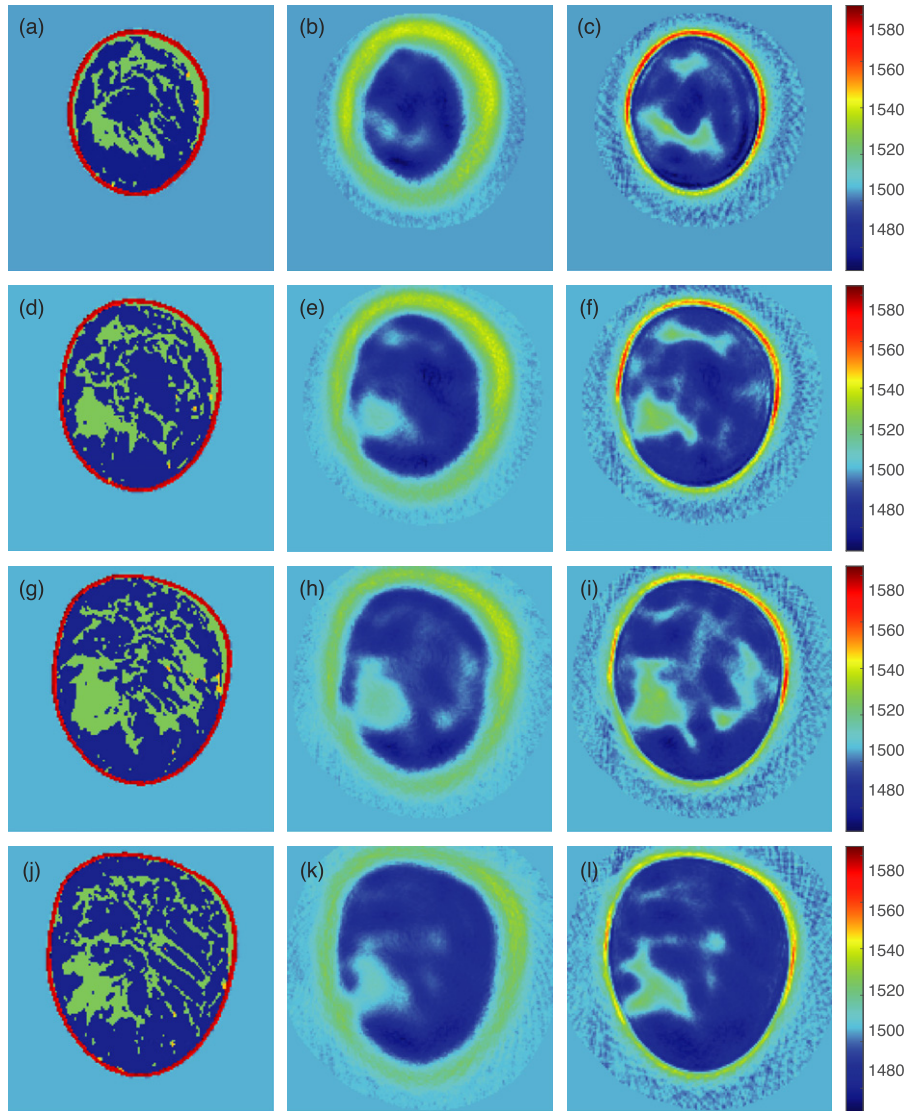
**Figure 9.** Ray-linking comparisons: (a) failure rate (% of refracted rays that failed to reach to within the required tolerance of the reception point within the maximum number of iterations), (b) mean number of traced rays for each emission–reception pair.

equation (41), being used. The reason for not discarding outright the ray that failed to link is that, typically, the algorithm has not diverged (because the singular values are controlled) but is stuck in a local minimum often, but not always, close to the reception point. As shown in figure 9(a) for all ray-linking algorithms, the fraction of rays that failed to reach the termination tolerance was very low at less than 0.5%. Among these algorithms, the Broyden-like approach performed best with fraction of failures on the order of  $10^{-4}$  for all UST iterations. Note that it would be possible to re-attempt the failed ray-linking problems after changing the size of the smoothing window enforced on the refractive index distribution (just for those specific rays). Because the portion of failures is very low, the increase in the computational cost of the whole image reconstruction algorithm would be negligible. Nevertheless, this approach was not taken in this work.

Figure 9(b) shows the mean number of traced rays to reach sufficiently close to the reception point for each emission–reception pair. This includes the rays that failed to reach the stopping threshold after  $N_{\text{link}}$  iterations, but not the unrefracted rays that reach the reception points directly, as these rays do not require ray-linking. Note that for the Gauss–Newton method, the number of traced rays is larger because each ray-linking iteration required at least three rays to be traced, two for the Jacobian and one for the residual, and maybe more in cases when smoothing loop and the backtracking loop require additional function evaluations. For the quasi-Newton methods, in contrast, each iteration was equivalent to tracing only one single ray. The mean number of iterations for ray-linking using the *Broyden-like* approach was about six, plus an additional single ray for initialisation of the derivative matrix. If the unrefracted rays were also included the mean number of traced rays would be significantly lower.

### 5.5. Ultrasound tomography: results

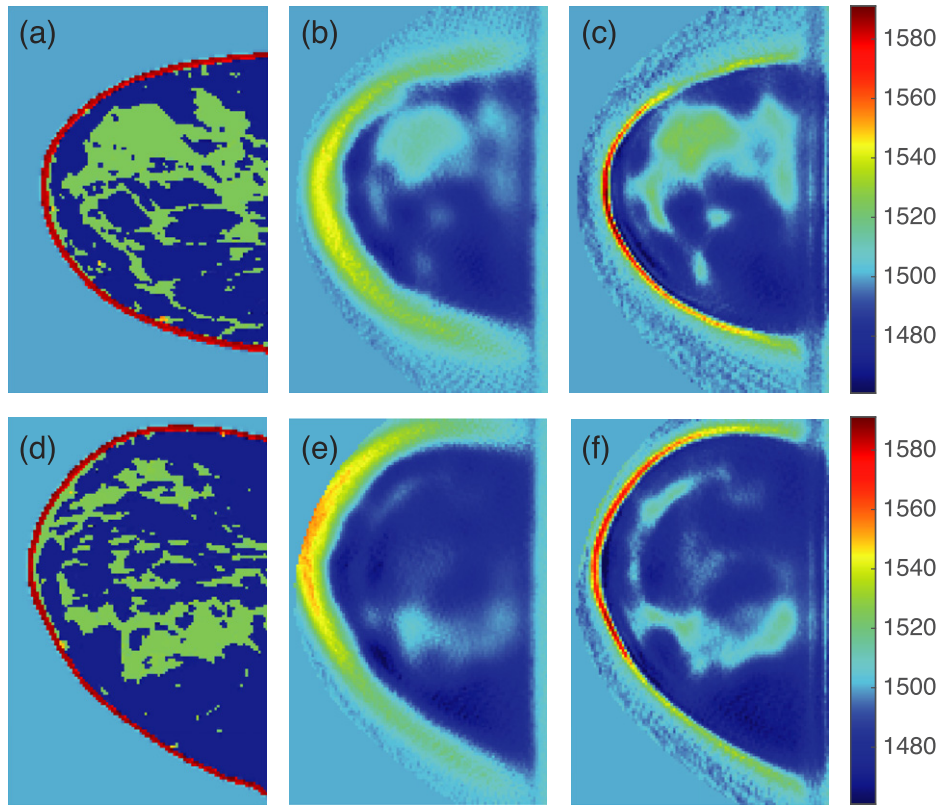
**5.5.1. Reconstructed sound speed (refractive index) images.** Figures 10 and 11 show slices through the 3D image volumes of the reconstructed breast phantom. The left column in figure 10 shows slices through the phantom at depths of  $z = -7, -6, -5, -4$  cm; the left column in figure 11 shows slices through the phantom at  $x = 0$  and  $y = -1$  cm. The centre columns show the images reconstructed using straight rays (the images for which  $RE_q$  was minimum are shown), and the right columns show the images reconstructed using the bent-ray code. (Only the images reconstructed using the fastest ray-linking approach, *Broyden-like*, are



**Figure 10.** Left: true sound speed distribution, centre: reconstructed using straight-ray approach,  $RE_* = 80.27\%$ , right: reconstructed using bent-ray approach (algorithms 2, 6 and 7),  $RE_* = 54.00\%$ . \* denotes the optimal iteration. Slices through the 3D images at (a)–(c):  $z = -7$  cm, (d)–(f):  $z = -6$  cm, (g)–(i):  $z = -5$  cm, (j)–(l):  $z = -4$  cm.

shown as all three approaches gave very similar images). All the figures are shown on the same colour scale. A metric was used to quantifying the accuracy of the reconstructed images, the squared relative error,

$$RE_q = \frac{\|\bar{c}_q - \bar{c}_{\text{breast}}\|_2^2}{\|\bar{c}_w - \bar{c}_{\text{breast}}\|_2^2} \times 100\%, \quad (61)$$

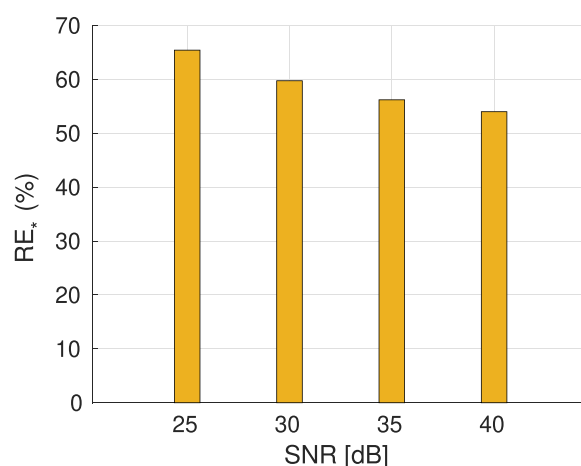


**Figure 11.** Left: true sound speed distribution, centre: reconstructed using straight-ray approach,  $RE_* = 80.27\%$ , right: reconstructed using bent-ray approach (algorithms 2, 6 and 7),  $RE_* = 54.00\%$ . Slices through the 3D images at (a)–(c):  $x = 0$  cm, (d)–(f):  $y = -1$  cm.

where  $\bar{c}_q$  is the reconstructed images of the sound speed after the subproblem  $q$  and  $\bar{c}_{\text{breast}}$  is the sound speed of the breast phantom interpolated to the image reconstruction grid. Figure 8 shows  $RE_q$  against the elapsed computation time for the whole image for each approach tested: straight rays and the three bent-ray methods (Gauss–Newton and the two quasi-Newton methods). The Broyden-like quasi-Newton method was also used using data evenly subsampled by a factor of 2 for both emitters and receivers (a four-fold reduction in emitter–receiver pairs). The computation time in this plot includes the calculation of the ROI at iteration  $q = -1$ , the construction of the Jacobian matrices, and the steepest descent iterations for all subproblems  $q$ , but not the TOF picking. Because an accelerated data processing approach was used (see appendix A) the time for calculation of the times-of-flight was just a few minutes using two CPUs. The results in figures 10 and 11 show the clear improvement that the bent-ray approach has over the straight ray approach, both in terms of image resolution and quantitative accuracy. Furthermore, as figure 8 shows, the sub-sampled bent-ray approach has reduced the squared error, compared to the straight ray case, by about 25% for about the same computation time and with four times fewer transducers.

**5.5.2. Effect of signal-to-noise ratio on the image reconstruction.** In this section, the ray-linking method based on the Broyden-like approach, which was the fastest approach, was used for image reconstruction from data with different signal-to-noise ratios (SNR). Figure 12





**Figure 12.** Accuracy of the reconstructed images for different SNR in the data, the relative error at the optimal iteration  $RE_*$ . The Broyden-like approach was used for ray linking.

shows the relative error (RE) of the optimal reconstructed images, showing—by comparing to figure 8—that the images reconstructed using the bent-ray approach from the data sets with  $SNR > 25$  dB are more accurate than the image reconstructed using the straight approach from the data sets with 40 dB SNR.

## 6. Discussion and conclusions

### 6.1. Contributions of this work

This paper presents a comprehensive derivation and study of a robust framework for large-scale bent-ray UST in 3D for a hemispherical detector array. While ray-based methods have been studied for decades, in geophysics in particular [57], most previous work for biological tissue has been restricted to 2D geometries or to obtaining reflection images. In particular,

- **Ray-tracing.** It was shown that a *dual-update* approach (algorithm 1) to ray-tracing provided more accurate ray-paths than the traditional *mixed-step* approach (algorithm 2), but for calculation of the acoustic length, which is dominated by the step size along the ray, the algorithms performed similarly. Algorithm 2 was more computationally efficient.
- **Ray-linking.** The ray-linking problem in 3D is challenging due to the high number of degrees of freedom for the trajectory of rays [57]. An approach to ray-linking was proposed that is novel in the way the problem is parameterised and solved for a hemi-spherical detection surface, which could be straightforwardly generalised to any convex detection surface. This problem was then analysed both as a minimisation, equation (41), and as a root-finding problem, equation (40). A *Gauss–Newton* approach was proposed to solve the former, and two derivative-free *quasi-Newton* approaches (*Broyden-like* and *BFGS-like*) were proposed to solve the latter. A limitation for these *quasi-Newton* approaches is the lack of efficient line search techniques. Here, a simple box constraint for mitigating nonmonotone behaviour and avoiding divergence was proposed and found to work well.

- **Adaptive smoothing.** To overcome ill-posedness in the ray-linking problem, novel adaptive smoothing schemes were proposed. These controlled the conditioning of the Hessian in the *Gauss–Newton* scheme, and the approximate derivative matrix in the *quasi-Newton* methods to ensure successful ray-linking. The fraction of ray-linking failures using the *Broyden-like* approach is about 0.05% (figure 9). (This neglects the pairs for which the first traced ray hits the reception point directly.) The mean number of rays needed for ray-linking between an emitter–receiver pair was only about 7.
- **Realistic simulation design.** The approach and algorithms presented here are intended to be practically useful, hence the emphasis is on robustness and efficiency at each stage of the algorithm design. In order to ascertain their likely performance with real measured data from a breast imager, a highly realistic scenario was simulated. First, the bowl array and number of transducers were chosen to be of comparable size and number to those used in practical breast imaging systems. Second, the phantom used was realistic of breast tissue [40]. Third, the frequency spectrum of the excitation pulse was bandlimited, mimicking the frequency range of detected signals in practice, with a relatively low centre frequency. (This was partly for data generation reasons—see below—and partly so that the same data could later be used for 3D full-wave inversion, which is also restricted in the usable frequency range for computational reasons.) Fourth, the data was simulated using a broadband full-wave model [63] and a realistic level of noise was added to the time series. Fifth, a commonly-used TOF-picking algorithm (with a minor modification for speed-up and reduction of mispicks) was used, so the TOF values contained a realistic level of variance—see appendix A.
- **Image reconstruction.** The nonlinear UST problem of estimating the sound speed was solved as a series of linearised subproblems, each solved using a steepest descent scheme. (Other methods, e.g. based on conjugate gradients or LSQR, could also be used but in our experience the images produced by these approaches are similar, and, for practical cases, they are more sensitive to the stopping criterion used for termination of each linearised subproblem.) Figure 8 shows that in about the same computational time as the straight ray case, a sub-sampled *Broyden-like* bent-ray approach reduced the squared error by about 25% with just 25% of the transducers. A comparison of the images in figures 10 and 11 shows that the bent-ray approach offers a significant improvement over the straight-ray approach.

## 6.2. Limitations

As explained in section 3, the theory of ray tracing is based on a high-frequency approximation. UST systems designed for use with ray-based image reconstructions therefore often use excitation pulses with a central frequency of about 2.5 MHz [15, 18, 58]. (The limit is the preferential high frequency attenuation of tissue.) Here, an excitation pulse with a much lower central frequency of 0.75 MHz was used in order to manage the computational cost of running full-wave acoustic simulations for 1024 emitters. Even so, these 1024 3D simulations took about 4.5 d on an 8-GPU cluster (see section 5.2) with a fast C++ full-wave simulation code [63]. The TOF-picking, and therefore the accuracy of the reconstructed images, would be significantly improved were a more broadband excitation pulse with a higher central frequency used (as may be the case with experimentally measured data). Unusually, therefore, for a simulation study, the images shown here are unlikely to represent the best performance of the algorithms, but are limited by the 3D data simulation. The TOF-picking will also be affected by the signal-to-noise ratio of the data. Our numerical experience shows that gradual increases in the noise level lead to a gradual degradation in image quality.



The number of emitters was also limited by the computational demands, and so it was not practical to study a scenario with a greater number of emitters, which are the case for example in a translational-rotational setting geometry [18], to see how it would affect image accuracy. By studying what happens when the number of emitters is reduced, it seems likely that there would be some improvement with more emitters but not a dramatic change.

There are several steps in the algorithms where approximations are made. Any of these could, under various circumstances, become the limiting factor in the image accuracy, although for the simulations shown here the variance in the TOF estimates dominates the error. These include the window used to smooth  $\bar{n}$ , grid spacing and grid-to-ray interpolation effects (the shape function used for trilinear interpolation is non-differentiable, but represents a trade-off between accuracy and speed), spacing along the ray, and the stopping criteria.

### 6.3. Summary

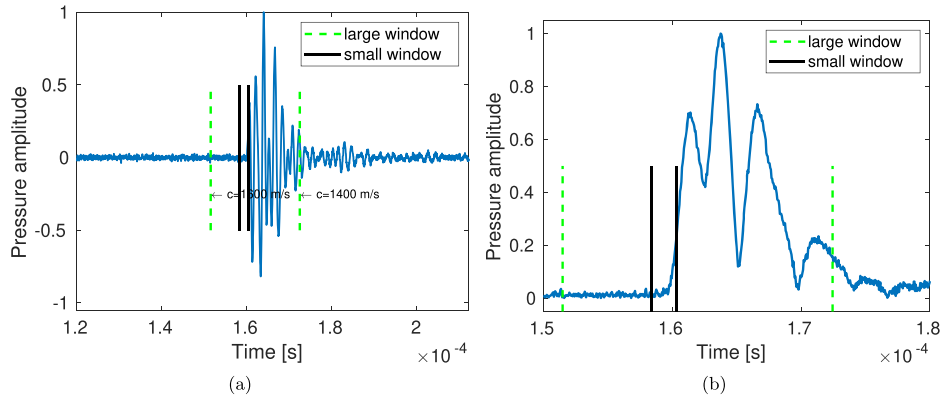
A set of robust and efficient algorithms have been designed for 3D bent-ray UST of the breast (ray tracing, ray-linking, and solving for the sound speed). These have been validated and tested with realistic simulated data. Because of the computational demands in generating the data for the simulation, the algorithms could not be tested to the limit of their performance. Nevertheless, the results demonstrate the usefulness of these algorithms for breast tomography and the technique could be used either as an adjunct to a full-wave inversion or another modality such as photoacoustic tomography, or as an efficient image reconstruction approach in its own right.

### Acknowledgments

Paper contributions: AJ designed and coded the algorithms, and wrote the paper. FL contributed to the experiment design, wave simulation codes and supervision. BC supervised the work, contributed to the algorithms and aided with the paper writing. All authors were involved in the conceptualisation of the work. This work was funded by the European Union's Horizon 2020 Research and Innovation program H2020 ICT 2016–2017 under Grant agreement No. 732411, which is an initiative of the Photonics Public Private Partnership. The authors would like to thank Michael Jaeger, and others in the Pammoth consortium, <https://pammoth-2020.eu>, as well as Marta Betcke, Francesc Rul-lan, Bradley Treeby and UCL's Biomedical Ultrasound Group for helpful discussions on various aspects of ray-based inversions and numerical modelling.

### Appendix A. Time-of-flight picking algorithm

This appendix describes the method used for picking, from the measured time series, the time of the first arrival of the ultrasound pulse after it has travelled through just water,  $\bar{T}_w$ , and after it has travelled through the object in water,  $\bar{T}_{\text{object}}$  (see section 2.3). In this approach, based on [33], the time at which the Akaike information criterion (AIC) is minimum is calculated in a time window that is expected to include the first arrival of the signal. This method is called *modified AIC*. This paper is primarily concerned with the inverse problem of reconstructing the sound speed when given TOF data, and not with the TOF picking itself. Many picking algorithms have been proposed, and we do not present here an analysis of which picking algorithm is optimal. Nevertheless, our experience with signals simulated from the breast phantom, section 5.2.2, and real signals measured in a 2D UST experiment, is that this approach



**Figure 13.** Large (green) and small (black) time windows. The small window has a size  $3 \mu\text{s}$ , which is almost equal to the temporal duration of the main lobe of the excitation pulse. (a) Measured signal (b) the envelope of the measured signal. The SNR of the signal is 40 dB.

outperforms many other first arrival picking approaches found in the geophysical literature, for example those reviewed in [1]. We also found that the accuracy of the *modified AIC* approach depends on the chosen time window.

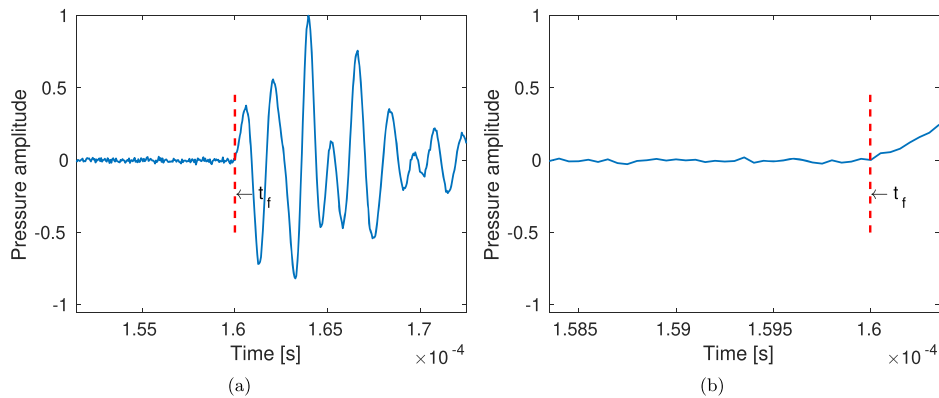
Here, we propose an approach for selecting an appropriate time window by including information about the amplitude of the envelope of the measured signal. We confirmed the effectiveness of our approach on measured 2D UST data (not shown). Each measured signal is defined using  $\mathbf{y}$  with indices  $i \in \{1, \dots, N_t\}$ , and  $t_i$  denoting the measurement time instant associated with the index  $i$ .

**Step 1: envelope.** Normalise the amplitude of the signal, and then calculate the envelope (absolute Hilbert transform).

**Step 2: large window.** For each measured signal, based on the distance between the relevant emitter and receiver pair together with an assumption for the minimum and maximum sound speed, choose a *large time window*. Here,  $1400$  and  $1600 \text{ m s}^{-1}$  were used for the minimum and maximum values for the sound speed. (In practical cases it might be possible to reduce the difference between these values; here the extremes are used, i.e. the minimum and maximum values across the whole domain.)

Figure 13(a) shows a measured signal that is normalised in amplitude. The chosen large window is shown by the green dash lines.  $\mathbf{y}_{w_l}$  is used to indicate the portion of the signal that inside this large window.

**Step 3: small window.** Another window, the *small time window*, that is expected to include the first-arrival of the signal is then chosen. The last point of the small window is chosen as the first time index in the large window for which the amplitude of the envelope of  $\mathbf{y}_{w_l}$  is greater than a threshold  $a_{th} \in (0, 1)$ . Here,  $a_{th}$  is a user-adjusted parameter, and is set to 0.5. For choosing the first time index for the small window, we move backward for a time equal to the duration of the main lobe of the excitation pulse. Using the excitation pulse shown in figure 7(a), the size of the small window was chosen to be  $3 \mu\text{s}$ , and is shown by the black lines in figures 13(a) and (b). The measured signal inside the small window is denoted  $\mathbf{y}_{w_s}$ , and  $i_l$  and  $i_r$  denote the indices representing the left and right edges, respectively. Also, the number of time indices in this window is represented by  $N_{w_s}$ .



**Figure 14.** The chosen first arrival time  $t_f$  from view of (a) the large window and (b) the small window.

**Step 4: modified Akaike information criterion (AIC).** The *modified AIC* approach is then applied to the portion of the signal that is confined to our small window,  $y_{sw}$ , in order to calculate the first arrival of the signal. Using AIC approach, Maeda's formula is used to calculate the AIC of the signal at time index  $i \in \{i_l, \dots, i_r\}$  [41]

$$\text{AIC}(i') = i \log(\sigma^2(y(1:i'))) + (N_t - i - 1) \log(\sigma^2(y(i' + 1:N_t))), \quad (62)$$

where  $\sigma^2$  denotes the variance. Following an approach proposed in [33], we calculate the minimum of AIC in our chosen small time window, and then choose another smaller window containing the minimum, here referred to as the *AIC window*, with time indices  $i'$ ,  $i' \in \{i'_l, \dots, i'_r\}$  with  $1 \leq i'_l < i'_r \leq N_{sw}$ . Here, we choose  $N_{\text{AIC}}$  as the closest integer to  $0.25N_{ws}$ . We then calculate the weights

$$\Upsilon(i') = \frac{e^{-\varrho(i')/2}}{\sum_{i'=1}^{N_{\text{AIC}}} e^{-\varrho(i')/2}}, \quad (63)$$

where  $\varrho(i') = \text{AIC}(i') - \text{AIC}_{\min}$ . The first-arrival time  $t_f$  is then calculated using

$$t_f = \sum_{i'=1}^{N_{\text{AIC}}} \Upsilon(i') t_{i'}, \quad (64)$$

where  $t_{i'}$  denotes the time corresponding to the index  $i'$ . Figures 14(a) and 14(b) show the calculated first arrival times within the large and small windows respectively. The approach explained above is applied separately to the two sets of data, collected from water alone and with the object present, and the discrepancy between the first arrivals  $\Delta \bar{T}$  is calculated and used as data for solving the UST inverse problem (see section 2.3).

## ORCID iDs

A Javaherian  <https://orcid.org/0000-0003-0798-3196>

F Lucka  <https://orcid.org/0000-0002-8763-5177>

B T Cox  <https://orcid.org/0000-0001-7296-4093>

## References

- [1] Akram J and Eaton D W 2019 A review and appraisal of arrival-time picking methods for downhole microseismic data *Geophysics* **81** KS67–87
- [2] Ali R, Hsieh S and Dahl J 2019 Open-source Gauss–Newton-based methods for refraction-corrected ultrasound computed tomography *Proc. SPIE 10955, Medical Imaging 2019: Ultrasonic Imaging and Tomography* vol 1095508
- [3] Anderson A H and Kak A C 1984 Simultaneous algebraic reconstruction techniques (SART): a superior implementation of the ART algorithm *Ultrason. Imaging* **6** 81–94
- [4] Anderson A H and Kak A C 1982 Digital ray tracing in two-dimensional refractive fields *J. Acoust. Soc. Am.* **72** 1593–606
- [5] Anderson A H 1990 A ray tracing approach to restoration and resolution enhancement in experimental ultrasound tomography *Ultrason. Imaging* **12** 268–91
- [6] Bachmann E and Tromp J 2020 Source encoding for viscoacoustic ultrasound computed tomography *J. Acoust. Soc. Am.* **147** 3221–35
- [7] Bellavia S, Macconi M and Pieraccini S 2012 Constrained Dogleg methods for nonlinear systems with simple bounds *Comput. Optim. Appl.* **53** 771–94
- [8] Bold G E J and Birdsall T G 1986 A top-down philosophy for accurate numerical ray tracing *J. Acoust. Soc. Am.* **80** 656–60
- [9] Broyden C G 2000 On the discovery of the good Broyden method *Math. Program.* **87** 209–13
- [10] Caselles V, Kimmel R and Sapiro G 1997 Geodesic active contours *Int. J. Comput. Vis.* **22** 61–79
- [11] Červený V 1987 Ray tracing algorithms in three-dimensional laterally varying layered structures *Seismic Tomography in Seismology and Exploration Geophysics* ed G Nolet (Berlin: Springer) pp 99–133
- [12] Cruz W, Martinez J and Raydan M 2006 Special residual method without gradient information for solving large scale nonlinear systems of equations *Math. Comput.* **75** 1429–48
- [13] Dapp R, Gemmeke H and Ruiter N 3D refraction-corrected transmission reconstruction for 3D ultrasound computer tomography *Proc. SPIE 8320 Medical Imaging 2012: Ultrasonic Imaging, Tomography, and Therapy* vol 832014
- [14] Denis F, Basset O and Gimenez G 1995 Ultrasonic transmission tomography in refracting media: reduction of refraction artefacts by curved-ray techniques *IEEE Trans. Med. Imaging* **14** 173–88
- [15] Duric N, Littrup P, Poulo L, Babkin A, Pevzner R, Holsapple E, Rama O and Glide C 2007 Detection of breast cancer with ultrasound tomography: first results with the computed ultrasound risk evaluation (CURE) prototype *Med. Phys.* **34** 773–85
- [16] Farra V 1993 Ray tracing in complex media *J. Appl. Geophys.* **30** 55–73
- [17] Galanti A 2000 The theory of Newton’s method *J. Comput. Appl. Math.* **124** 25–44
- [18] Gemmeke H, Hopp T, Zapf M, Kaiser C and Ruiter N V 2017 3D ultrasound computer tomography: hardware setup, reconstruction methods and first clinical results *Nucl. Instrum. Methods Phys. Res. A* **873** 59–65
- [19] Greenleaf J F, Johnson S A, Lee S L, Herman G T and Woo E H 1974 Algebraic reconstruction of spatial distributions of acoustic absorption within tissue from their two-dimensional acoustic projection *Acoustical Holography* ed P S Green (Boston, MA: Springer) pp 591–603
- [20] Griewank A 1986 The global convergence of Broyden-like methods with suitable line search *J. Aust. Math. Soc. B* **28** 75–92
- [21] Gu G Z, Li D H, Qi L and Zhou S Z 2003 Descent directions of quasi-Newton methods for symmetric nonlinear equations *SIAM J. Numer. Anal.* **40** 1763–74
- [22] Holm D D 2011 *Geometric Mechanics Part I: Dynamics & Symmetry* 2nd edn (London: Imperial College Press) pp 1–97
- [23] Hopp T, Ruiter N, Bamber J C, Duric N and van Dongen K W A (ed) 2017 *Int. Workshop on Medical Ultrasound Tomography* (Speyer Germany)
- [24] Hopp T, Zuch F, Comby P-A and Ruiter N V Fat ray ultrasound transmission tomography: preliminary experimental results with simulated data *Proc. SPIE 11319, Medical Imaging 2020: Ultrasonic Imaging and Tomography* vol 113190M
- [25] Huthwaite P and Simonetti F 2011 High-resolution imaging without iteration: a fast and robust method for breast ultrasound tomography *J. Acoust. Soc. Am.* **130** 1721–34
- [26] Johnson S A, Greenleaf J F, Samayoa W F, Duck F A and Sjostrand J 1975 Reconstruction of three-dimensional velocity fields and other parameters by acoustic ray tracing *IEEE Proc. Ultrason. Sympos* pp 46–51

- [27] Julian B R and Gubbins D 1977 Three-dimensional seismic ray tracing *J. Geophys.* **43** 95–113
- [28] Kang H R 2006 *Computational Color Technology* (Bellingham, WA: SPIE Optical Engineering Press) ch 9 (<https://doi.org/10.1117/3.660835.fm>)
- [29] Klimeš L 1996 Grid travel-time tracing: second-order method for the first arrivals in smooth media *Pure Appl. Geophys.* **148** 539–63
- [30] Koketsu K and Sekine S 1998 Pseudo-bending method for three-dimensional seismic ray tracing in a spherical earth with discontinuities *Geophys. J. Int.* **132** 339–46
- [31] Lay T and Wallace T C 1995 *Modern Global Seismology* (New York: Academic)
- [32] Li C, Duric N, Littrup P and Huang L 2009 *In vivo* breast sound-speed imaging with ultrasound tomography *Ultrasound Med. Biol.* **35** 1615–28
- [33] Li C, Huang L, Duric N, Zhang H and Rowe C 2009 An improved automatic time-of-flight picker for medical ultrasound tomography *Ultrasonics* **49** 61–72
- [34] Li D-H and Cheng W 2007 Recent progress in the global convergence of quasi-Newton methods for nonlinear equations *Hokkaido Math. J.* **36** 729–43
- [35] Li D-H and Fukushima M 2000 A derivative-free line search and global convergence of Broyden-like method for nonlinear equations *Optim. Methods Softw.* **13** 181–201
- [36] Li D and Fukushima M 1999 A globally and superlinearly convergent Gauss–Newton-based BFGS method for symmetric nonlinear equations *SIAM J. Numer. Anal.* **37** 152–72
- [37] Li S, Jackowski M, Dione D P, Varslot T, Staib L H and Mueller K 2010 Refraction corrected transmission ultrasound computed tomography for application in the breast imaging *Med. Phys.* **37** 2233–46
- [38] Lin L, Hu P, Shi J, Appleton C M, Maslov K, Li L, Zhang R and Wang L V 2018 Single-breath-hold photoacoustic computed tomography of the breast *Nat. Commun.* **9** 2352
- [39] Pérez-Liva M, Herraiz J L, Udías J M, Miller E, Cox B T and Treeby B E 2017 Time domain reconstruction of sound speed and attenuation in ultrasound computed tomography using full wave inversion *J. Acoust. Soc. Am.* **141** 1595–604
- [40] Lou Y, Zhou W, Matthews T P, Appleton C M and Anastasio M A 2017 Generation of anatomically realistic numerical phantoms for photoacoustic and ultrasonic breast imaging *J. Biomed. Opt.* **22** 041015
- [41] Maeda N 1985 A method for reading and checking phase time in auto-processing system of seismic wave data *JSSJ* **38** 365–79
- [42] Martiartu N K, Boehm C and Fichtner A 2020 3D Wave-equation-based finite-frequency tomography for ultrasound computed tomography TUFFC. **67** 1332–43
- [43] Matthews T P, Wang K, Li C, Duric N and Anastasio M A 2017 Regularized dual averaging image reconstruction for full-wave ultrasound computed tomography *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* **64** 811–25
- [44] Matthews T P and Anastasio M A 2017 Joint reconstruction of the initial pressure and speed of sound distributions from combined photoacoustic and ultrasound tomography measurements *Inverse Problems* **33** 124002
- [45] Moré J J and Trangenstein J A 1976 On the global convergence of Broyden’s method *Math. Comput.* **30** 523–40
- [46] Moser T J, Nolet G and Snieder R 1992 Ray bending revisited *Bull. Seismol. Soc. Am.* **82** 259–88
- [47] Cochran J J, Cox L A Jr, Keskinocak P, Kharoufeh J P, Smith J C and Murray W 2011 Newton-type methods In *Wiley encyclopedia of operations research and management science* ed J J Cochran, L A Cox, P Keskinocak, J P Kharoufeh and J C Smith (<https://doi.org/10.1002/9780470400531.eorms0569>)
- [48] Norton S and Linzer M 1982 Correcting for ray refraction in velocity and attenuation tomography: a perturbation approach *Ultrason. Imaging* **4** 201–33
- [49] Nyayapathi N and Xia J 2019 Photoacoustic imaging of breast cancer: a mini review of system design and image features *J. Biomed. Opt.* **24** 121911
- [50] Oliveira J A R, Balatazar A and Castelán M 2019 On ray tracing for sharp changing media *J. Acoust. Soc. Am.* **146** 1595–604
- [51] Opielinski K J, Pruchnicki P, Szymanowski P, Szepieniec W K, Szweda H, Swis E, Jozwik M, Tenderenda M and Bułkowskif M 2018 Multimodal ultrasound computer-assisted tomography: an approach to the recognition of breast lesions *Comput. Med. Imaging Graph.* **65** 102–14
- [52] Pereyera V, Lee W H L and Keller H B 1980 Solving two-point seismic ray tracing problems in a heterogeneous medium *Bull. Seismol. Soc. Am.* **70** 79–99

- [53] Pierce A D 1981 *Acoustics: An Introduction to its Physical Principles and Applications* 3rd Edition (Berlin: Springer and ASA Press) (<https://doi.org/10.1007/978-3-030-11214-1>)
- [54] Plessix R-E 2006 A review of the adjoint-state method for computing the gradient of a functional with geophysical applications *Geophys. J. Int.* **167** 495–503
- [55] Powell M J D 1970 A FORTRAN subroutine for solving systems of nonlinear algebraic equations *Numerical Methods for Nonlinear Algebraic Equations* ed P Rabinowitz (London: Gordon and Breach) pp 115–61
- [56] Rawlinson N, Houseman G A and Collins C D N 2001 Inversion of seismic refraction and wide-angle reflection traveltimes for three-dimensional layered crustal structure *Geophys. J. Int.* **145** 381–400
- [57] Rawlinson N, Hauser J and Sambridge M 2008 Seismic ray tracing and wavefront tracking in laterally heterogeneous media *Adv. Geophys.* **49** 203–73
- [58] Ruiter N V, Zapf M, Hopp T, Dapp R, Kretzek E, Birk M, Kohout B and Gemmeke H 2012 3D ultrasound computer tomography of the breast: a new era? *Eur. J. Radiol.* **81** S133–4
- [59] Runborg O 2007 Mathematical models and numerical methods for high frequency waves *Commun. Comput. Phys.* **2** 827–80
- [60] Sambridge M S and Kennett B L N 1990 Boundary value ray tracing in a heterogeneous medium: a simple and versatile algorithm *Geophys. J. Int.* **101** 157–68
- [61] Sethian J A 1996 A fast marching level set method for monotonically advancing fronts *Proc. Natl Acad. Sci.* **93** 1591–5
- [62] Thurber C H and Kissling E 2000 Advances in travel-time calculations for 3D structures *Advances in Seismic Event Location* (Dordrecht: Kluwer) pp 71–99
- [63] Treeby B E and Cox B T 2010 k-wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields *J. Biomed. Opt.* **15** 021314
- [64] Tsitsiklis J N 1995 Efficient algorithms for globally optimal trajectories *IEEE Trans. Control Syst.* **40** 1528–38
- [65] Um J and Thurber C 1987 A fast algorithm for two-point seismic ray tracing *Bull. Seismol. Soc. Am.* **77** 972–86
- [66] Wesson R L 1971 Travel-time inversion for laterally inhomogeneous crustal velocity models *Bull. Seismol. Soc. Am.* **61** 729–46
- [67] Wiskin J, Borup D T, Johnson S A and Berggren M 2012 Non-linear inverse scattering: high resolution quantitative breast tissue tomography *J. Acoust. Soc. Am.* **131** 3802–13
- [68] Wiskin J W, Borup D T, Iuanow E, Klock J and Lenox M W 2017 3D nonlinear acoustic inverse scattering: algorithm and quantitative results *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* **64** 1161–74
- [69] Xu T, Li F, Wu Z, Wu C, Gao E, Zhou B, Zhang Z and Xu G 2014 A successive three-point perturbation method for fast ray tracing in complex 2D and 3D geological models *Tectonophysics* **627** 72–81
- [70] Wang K, Matthews T, Anis F, Li C, Duric N and Anastasio M A 2015 Waveform inversion with source encoding for breast sound speed reconstruction in ultrasound computed tomography *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* **62** 475–93
- [71] Yuan G, Wei Z and Lu X 2011 A BFGS trust-region method for nonlinear equations *Computing* **92** 317–33
- [72] Zhou W-J and Li D-H 2008 A globally convergent BFGS method for nonlinear monotone equations without any merit functions *Math. Comput.* **77** 2231–40