

Morphing for faster computations with finite difference time domain algorithms

Ronald Aznavourian^{1,4,*}, Sébastien Guenneau², Bogdan Ungureanu^{3,4}, and Julien Marot¹

¹ Aix-Marseille Université, CNRS, Centrale Marseille, Institut Fresnel, 13013 Marseille, France

² UMI 2004 Abraham de Moivre-CNRS, Imperial College London, London SW7 2AZ, UK

³ Department of Mathematics, Imperial College London, London SW7 2AZ, UK

⁴ BTU Trust, 77700 Magny le Hongre, France

Received: 23 July 2021 / Accepted: 13 December 2021

Abstract. In the framework of wave propagation, finite difference time domain (FDTD) algorithms, yield high computational time. We propose to use morphing algorithms to deduce some approximate wave pictures of their interactions with fluid-solid structures of various shapes and different sizes deduced from FDTD computations of scattering by solids of three given shapes: triangular, circular and elliptic ones. The error in the L^2 norm between the FDTD solution and approximate solution deduced via morphing from the source and destination images are typically less than 1% if control points are judiciously chosen. We thus propose to use a morphing algorithm to deduce approximate wave pictures: at intermediate time steps from the FDTD computation of wave pictures at a time step before and after this event, and at the same time step, but for an average frequency signal between FDTD computation of wave pictures with two different signal frequencies. We stress that our approach might greatly accelerate FDTD computations as discretizations in space and time are inherently linked via the Courant–Friedrichs–Lewy stability condition. Our approach requires some human intervention since the accuracy of morphing highly depends upon control points, but compared to the direct computational method our approach is much faster and requires fewer resources. We also compared our approach to some neural style transfer (NST) algorithm, which is an image transformation method based on a neural network. Our approach outperforms NST in terms of the L^2 norm, Mean Structural SIMilarity, expected signal to error ratio.

Keywords: Morphing / finite difference time domain / elastodynamic waves

1 Introduction

Numerical simulations for elastodynamic wave propagation in complex media require huge computing resources and even with parallel computing resources, the computations may last many hours (for 2D configurations) or many days (in 3D). Elastodynamic waves play a key role in well-established research topics including medical imaging [1] and site-city interactions [2], at small and large scales, respectively. Research in seismic metamaterials (SM) that started with large scale experiments on the control of surface Rayleigh waves in structured sedimentary soil near the French Alpine city of Grenoble in 2012 [3] has led to numerous numerical investigations of their properties with finite element commercial softwares in the frequency domain [4–6], and also with spectral finite element [7–9] and finite difference time domain (FDTD) [10] freewares in the time domain. Research in SM is based upon bold analogies with electromagnetic metamaterials [11], that

benefit from the invariance of Maxwell's equations under coordinate changes [12], that underpin perfectly matched layers (PMLs) [13]. PMLs are essential when dealing with scattering problems in unbounded domains. However, research advances in SM are hindered by the complex nature of soil (e.g. its heterogeneity) and the size of the computational domain required to accurately model the interaction of elastodynamic waves with structural elements added to the bare soil. The governing equations for linear elasticity are not formed invariant, which makes PMLs a more delicate matter than in electromagnetism [14]. Moreover, when designing an SM, one needs to play with the size and shape of inclusions in soil, as well as their spacing and other geometric and elastic parameters to achieve the required effect (shielding, absorption and deflection). However, for each parameter change, all the computations have to be redone. This is actually only the tip of the iceberg as SM genuinely undergo nonlinear effects that make the direct numerical simulations even more challenging. Bearing in mind the complexity of the direct problem, solving an optimization inverse problem seems thus hardly tractable. To circumvent this obstacle, our idea

* e-mail: ronald.aznavourian@fresnel.fr

is to deduce intermediate results between two calculated results without having to recalculate anything. In other words, our objective is not to obtain the most accurate result, but to obtain an acceptable result in a very short period of time and small computational resources, so that we can move toward the best setting of parameters for our numerical simulations, as quickly and efficiently as possible. Now the point is how to proceed to make this idea come true. Photonics has entered the era of deep learning, neural networks and artificial intelligence, see e.g. [15], and such tools can be easily translated to the realm of acoustics [16], due to the similar structure of the governing equations. However, correspondences between governing equations for electromagnetic and elastodynamic waves are less straightforward, so we will see and check which technologies may be used to reach our goal.

Metamaterials do not escape the need for optimization [15,17,18], in particular with the prowess achieved by artificial intelligence, in the broadest sense, and more precisely by neural networks [18]. Actually, only a few papers deal with the application of neural networks to metamaterials. For instance, Pomot et al. proposed a genetic algorithm to optimize the structure of an invisibility cloak in the framework of acoustic wave propagation in metamaterials [19], but this is another approach to the optimization of a metamaterial. Among the various kinds of neural networks, there is the technique of the Neural Style Transfer (NST) [20,21], which consists of the transformation of an image into another with an iterative process. In the present work, NST is used as a comparative method.

In a recent work [22], some of us proposed to apply the computer graphic technique of morphing [23] – which consists in transforming an image into another with a series of intermediate images – to the field of photonics (in the case of cylindrical media allowing for a split of the vector problem into s and p polarizations). Governing equations for electromagnetic waves in such cylindrical media (i.e. invariant along x_3) have the following form:

$$\sum_{i=1}^2 \frac{\partial}{\partial x_i} \sum_{j=1}^2 \eta_{jk}^{-1}(x_1, x_2) \sum_{k=1}^2 \frac{\partial}{\partial x_k} u_3(x_1, x_2, t) = \kappa(x_1, x_2) \frac{\partial^2}{\partial t^2} u_3(x_1, x_2, t) + f(x_1, x_2, t) \quad (1)$$

with $u_3 = H_3$ the longitudinal magnetic (resp. $u_3 = E_3$ electric) field unknown, η_{jk} the anisotropic permittivity (resp. permeability) and κ the permeability (resp. permittivity) in s (resp. p) polarization. Moreover, $\partial/\partial x_i$ and $\partial/\partial t$ denote the partial space and time derivatives, respectively, and f is the source term. In [22], f had some time-harmonic dependence $\exp(i\omega t)$, with ω the angular wave frequency. Thus, the solution was assumed to be time-harmonic in (1) and the second derivative in time was replaced by $-\omega^2$.

Morphing was applied in a way that is similar to some techniques used in the transformational optics (TO) domain. In TO, a Cartesian grid is deformed to control in some desired manner the trajectory of light. In metamaterials, a grid or a Delaunay mesh is transformed

in a similar way [24]. In [22], we essentially worked with the time-harmonic Maxwell's system in a cylindrical setup whereby the governing partial differential equations (PDEs) have scalar-valued unknowns. In [22], we emphasized the potential use of morphing as a mean to speed up computations by deducing the wave pattern induced by the scattering by, say an elliptical obstacle of eccentricity $(a+b)/2$ from the knowledge of the wave patterns from elliptical obstacles of eccentricities a and b .

The FDTD method is an iterative process introduced by Kane Yee in 1966 in a seminal article [25] about the application of centered finite difference operators on staggered grids in space and time for each electric and magnetic vector field component in Maxwell's curl equations. The general principle of FDTD is that the wave field meshes, and the calculation of the wave propagation in each node is based on the wave propagation in the nearest nodes from the considered node. Following its success in computational electromagnetics, FDTD was later adapted to elastic wave propagation problems. The FDTD is completely different from the finite element method (FEM) which is based upon variational (or weak) forms of governing equations. Moreover, in stark contrast to FDTD, FEM originates in problems of elasticity [26,27] and it consists in applying a wave propagation function to all the nodes of the mesh at the same time. The traditional FEM leads to spurious modes in electromagnetics, and thus finite edge elements have been developed to circumvent such technical obstacles [28].

We note that results in [22] hold for acoustic waves, in which case η_{jk} in equation (1) stands for the anisotropic density and κ in the inverse of the bulk modulus. Equation (1) is also valid for anti-plane elastic waves in cylindrical media, in which case η_{jk} stands for the anisotropic shear modulus and κ for the density.

Here, we would like to demonstrate that results in [22] can be extended to PDEs with vector-valued unknowns such as in problems for in-plane coupled shear and pressure waves in cylindrical isotropic media. Such elastodynamic problems are modeled by the Navier equation

$$\rho(x) \frac{\partial^2 u_i}{\partial t^2}(x, t) = \sum_{j=1}^d \frac{\partial}{\partial x_j} \sum_{k=1}^d \sum_{l=1}^d C_{ijkl}(x) \frac{\partial u_k}{\partial x_l}(x, t) + f_i(x, t) \quad (2)$$

where \mathbb{C} is the rank-4 (symmetric) elasticity tensor with entries $C_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk})$, $i, j, k, l = 1, \dots, d$, λ and μ being the Lamé parameters and ρ the mass density.

Equation (2) reduces to (1) when we assume the displacement field has the form $(0, 0, u_3)(x_1, x_2, t)$. We note that these are two hyperbolic PDEs that become elliptic PDEs when a time-harmonic dependence is assumed, such as in [22]. Numerical schemes for solving hyperbolic and elliptic PDEs are very different, and thus it is far from obvious that morphing would help accelerate numerical simulations for the former, even if this was successful for the latter.

In the present paper, we want to focus on the in-plane case for which the displacement has the form $(u_1, u_2, 0)(x_1, x_2, t)$ and so $d = 2$ in equation (2). We will focus on the case

of hyperbolic PDEs, and so we face the double challenge of moving from scalar elliptic PDEs to vector hyperbolic PDEs.

However, if morphing works it will allow us to overcome the convergence criterion for numerical methods such as for FDTD, which is known as Courant–Friedrichs–Lewy (CFL) condition [29]. CFL is a necessary condition for convergence of hyperbolic PDEs:

$$C = \frac{v_1 \Delta t}{\Delta x_1} + \frac{v_2 \Delta t}{\Delta x_2} \leq C_{\max} \quad (3)$$

where C is the Courant constant and C_{\max} is linked to the largest speed of elastic waves in the simulation. More precisely, Δt is the time step (whose dimension is time) and Δx is the length interval (whose dimension is length) in the FDTD scheme. Moreover, v_1 and v_2 are the velocity amplitudes associated with u_1 and u_2 . We note that C is dimensionless and so is C_{\max} . In most situations, the user has to choose C_{\max} as the exact value for the largest speed of sound present in the simulation. However, if a user wants to run several simulations with different media then things become more complicated.

In this 2D study, we need to fulfill the following CFL criterium

$$\Delta t < \frac{\Delta x}{\sqrt{2}C_{\max}} \quad (4)$$

with $C_{\max} = 1.5 \text{ mm}/\mu\text{s}$. We consider a grid step $\Delta x = 0.1$ and so the time step is set as $\Delta t = 0.99\Delta x/(\sqrt{2}C_{\max})$.

Clearly, morphing is foreign to the CFL condition in the case of a linear phenomenon. Indeed, morphing is just a linear interpolation on the shapes and the colors between two images computed *via* FDTD. Since they were computed *via* FDTD, these two images respect the CFL conditions, and since morphing transforms, in a linear way, the first image into the second one, by proportionality of the linear phenomenon it also respects the CFL conditions. So, for a linear wave propagation phenomenon, there is no stability rule to be applied on morphing as it is simply based on some image treatment. We keep in mind that if we were to study a non-linear problem, the proportionality rule would no longer apply and thus linear interpolation of two images with morphing would not be appropriate to accelerate numerical simulations. Thus, it is interesting to check whether or not one can interpolate elastic fields computed *via* FDTD at different time steps to almost immediately retrieve elastic fields at intermediate time steps with reasonable accuracy. Similarly, it is interesting to check if morphing can be used to interpolate elastic fields computed for sources at different frequencies.

First of all, we show how we have built our tests. Then, we have three sets of tests: the first one is based on isosceles triangles, the second one on convex lenses, and the last one on a Luneburg lens. In our first tests, we remind how we used the morphing applied in the space domain, with this small difference that now we are working on elastic waves (using FDTD computations) and no more in TO.

Here, we further show that morphing can be used in the time domain. However, one could also argue that the morphing technique can be applied not only to changes of shapes of physical domains but also to changes in time (in actuality, morphing came from the field of computer-aided animation, intermediate images being deduced by morphing to make movies more fluid [30]).

Then, we describe the morphing principle applied to the propagation of acoustic (pressure and shear) waves generated by a Gaussian pulse and scattered by aluminum lenses of varying thicknesses in water, at the same time-step. We also describe the application of morphing to the same kind of waves scattered by an aluminum lens of a given thickness at different time steps.

In the end, we go back to the frequency domain (time-harmonic elastic waves, i.e. an acoustic signal periodic in time), and we apply morphing to deduce the wave pattern at a given frequency sandwiched between two frequencies where the wave pattern has already been computed.

Note that for these last two applications (in the time domain and in the frequency domain), we also compare the style transfer method to the use of the morphing technique.

The outline of the paper is as follows. In Section 2, we explain what morphing is, how it works, we show how we have built our tests and how we proceeded with them. In Section 3, we show results achieved via morphing and we compare them against those obtained by computation. We measure the differences between these two methods (morphing-assisted computations versus fully numerical solutions). In Section 4, we discuss the strength and weakness of morphing as an acceleration tool for FDTD, and more generally for numerical simulations involving hyperbolic PDEs. In Section 5, we draw some concluding remarks.

2 Morphing algorithm: principles and illustrative examples

In what follows, we make use of two free-wares [31,32] for respectively morphing [31] and FDTD [32] algorithms, but of course, other softwares could equally well be used for our purpose. Note that we have used a desktop computer running under Windows 10 64 bits, with a processor at 3.70GHz and 64Go of RAM.

2.1 Definitions

We shall use the following two definitions to evaluate the difference between images. First, following [22], we introduce

$$L^2 \text{norm} = \sqrt{\frac{1}{N} \times \sum_{i=1}^N \left(\frac{P_i}{255}\right)^2} \quad (5)$$

where the sum is taken over all the N pixels $(P_i)_{i=1,\dots,N}$ in the grayscale image P , and each pixel P_i has a value between 0 and 255, see Figure A1 in Appendix.

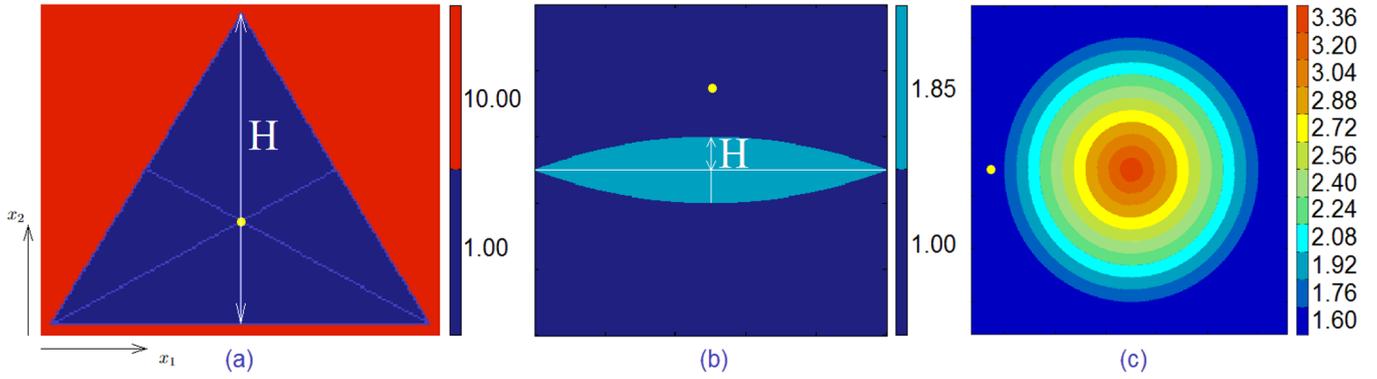


Fig. 1. Shapes of obstacles: (a) For each isosceles triangle, the point force (directed along the x_2 axis) is located at the center of gravity of the triangle and the density outside the triangle is ten times larger than inside the triangle. Moreover, starting from this equilateral triangle of base 20 cm, and a height H equal to $10\sqrt{3}$ cm, we have built three isosceles triangles keeping always the same base of 20 cm but increasing each time the height H as follows: four-thirds of H for the first isosceles triangle, five-thirds of H for the second isosceles triangle, and six-thirds of H for the last isosceles triangle. (b) A convex lens (of maximum half-width H) with a point force (directed along the x_2 axis) above it. (c) A layered Luneburg lens (a disk surrounded by 10 concentric layers) with a point force (directed along x_1) on its left side. The density of disk, layers, and the surrounding medium, appears on the color scale (in units of 10^3 kg.m^{-1}).

Next, starting from all three components (red, green and blue) of all N pixels both in the colored image of the calculation result (noted R) and in the colored image (noted D) of the difference between the calculation result and an estimated result (noted E), such that $D = R - E$, we define the calculation of the expected signal to error ratio (ESER) as follows:

$$\text{ESER} = 10 \times \log \left(\sum_{l=1}^3 \sum_{i=1}^N \frac{R_{il}^2}{D_{il}^2} \right) \quad (6)$$

where the sum is taken over the squares of all three components (noted in 'l') and all N pixels (noted in 'i'), of R (i.e. R_{il}) and D (i.e. D_{il}), by dividing R by D .

2.2 Principle

In [22], we explained how morphing works: one must place some control points, both in the two computation result images, and launch the morphing software which creates a mesh, starting from the control points, for each of these two images. Then, one must proceed with a transformation of the mesh of the first image to make it match the mesh of the second image. Once the morphing process is over, we can choose and pick one of the intermediate results of the morphing, for instance at 50% of the process, and compare this result with the computation result for an average value of parameters.

2.3 Exemplification

In this subsection, we remind how we have proceeded to use the morphing in the spatial domain, even if it was in TO in what we have already published [22], but the point is to understand well how the morphing works and how we have used it.

2.3.1 Creating different obstacles

For the spatial tests, we need different shapes of obstacles.

As a first illustrative example, we show in Figure 1 the wave pattern of a point force-generating elastic waves in a low-density elastic medium surrounded by an elastic medium with a density that is ten times higher. To avoid reflections on the boundaries of the computational domain, we set some perfectly matched layers [33]. The low-density medium is of an equilateral triangular shape, see Figure 1, and the point force is located at the barycenter (i.e. center of gravity) of the triangle.

For our tests, we stretch the height H , see Figure 1a, to create three different isosceles triangles which all have the same base of 20 cm. So, starting from an equilateral triangle of base 20 cm, where all the heights are, of course, $10\sqrt{3}$ cm long, we create three isosceles triangles with the same base of 20 cm but increasing each time the height H as follows: four-thirds of H for the first isosceles triangle, which means $\frac{4}{3} \times 10\sqrt{3}$ cm height, five-thirds of H for the second isosceles triangle, which means $\frac{5}{3} \times 10\sqrt{3}$ cm height, and six-thirds of H for the last isosceles triangle, which means $\frac{6}{3} \times 10\sqrt{3}$ cm height.

For the second set of tests, we start from this lens with this kind of shape, see Figure 1b, then we have built three lenses only by increasing the size of H as follows: H equal to 2.5 cm for the first lens, H equal to 5 cm for the second lens and H equal to 7.5 cm for the last lens. The three lenses are 50 cm wide.

As a second illustrative example, we show in Figure 1b the wave pattern of a point force-generating elastic waves in a low-density elastic medium near an elastic medium with a density 1.85 times larger (aluminum). To avoid reflections on the boundaries of the computational domain, we set some perfectly matched layers [33]. For all our tests, the point force is located above the lens, at 12.5 cm from the top and 25 cm from the left or the right side.

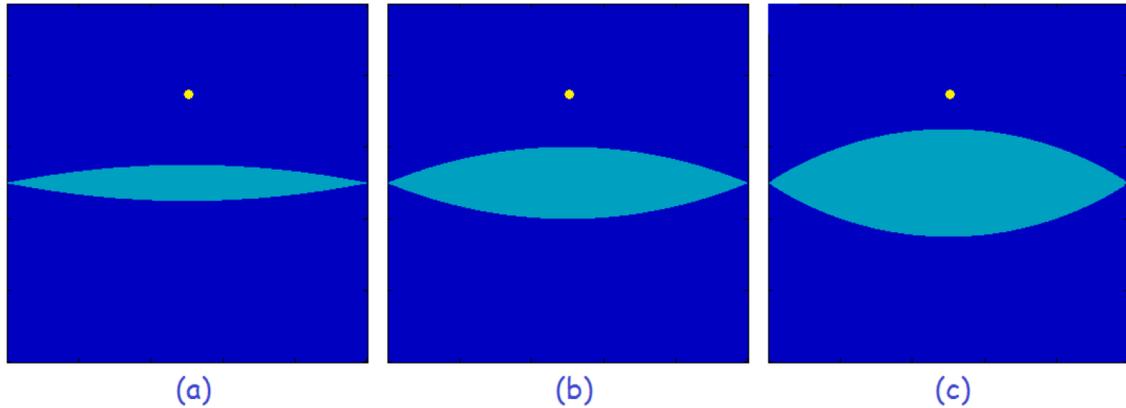


Fig. 2. Three convex lenses of increasing width $2H$ at their center: (a) $H = 2.5$ cm for Lens 1; (b) $H = 5$ cm for Lens 2 and (c) $H = 7.5$ cm for Lens 3. All three lenses are 50 cm wide (from left to right); the point force marked by the yellow dot is located on the orthogonal bisector at a distance 12.5 cm from the top of the lens. The mass density is 1.85 larger inside the lens than in the surrounding medium.

As shown in Figure 2, in our tests we use three kinds of lenses:

⇒ the first, Figure 2a, which we will call “Lens 1”, with $H = 2.5$ cm,

⇒ the second, Figure 2b, which we will call “Lens 2”, with $H = 5.0$ cm,

⇒ the third, Figure 2c, which we will call “Lens”, with $H = 7.5$ cm.

where H is the half-width of the lens at its center, see Figure 1b.

All these lenses are tested in a computational domain of 50 cm by 50 cm, where the lenses are placed in the middle of the field. Note that the width of the lenses is 50 cm.

2.3.2 Applying the FDTD computation to our obstacles

For each triangle, we throw a signal for the radiation pattern of a point force of central frequency 10 kHz located at the gravity point of an isosceles triangle. The FDTD snapshots have been taken at the time $t = 710\mu\text{s}$.

For each lens, we throw a signal just above the lens (located at $x_1 = 25$ cm and $x_2 = 12.5$ cm) with a frequency of 250 kHz. Then we generate the FDTD snapshots and keep the snapshot taken at the time $t = 140\mu\text{s}$.

2.3.3 Applying the morphing computation between some FDTD computation results

For a given snapshot time, we use the morphing computation between the smallest isosceles triangle, see Figure 3a, and the highest isosceles triangle results, see Figure 3b, which leads to Figure 3d.

This takes 20 s once we have placed the control points, which has been done manually, but some algorithms based on convolutional experts network [34–37] could be implemented to avoid any human intervention.

We compare the latter to the average isosceles triangle result, see Figure 3c. Note the strong similarities between Figures 3c and 3d. Figure 3e is Figure 3c transformed in a grayscale image. For this we take the mean value, for each pixel, of the three R, G, and B channels. The same process will be performed in the rest of the paper we turning an RGB image into a gray level one. Figure 3f is Figure 3d transformed in a grayscale image, and Figure 3g is the absolute value of the difference between Figures 3e and 3f. When we compute the L^2 norm for Figure 3g, we get a result of 0.0729, which means a difference of 2% between Figures 3c and 3d (Cf. Fig. A1 in Appendix). Note that the size of the field is 80 cm by 80 cm.

For a given snapshot time, we use the morphing computation between Lens 1, see Figure 4a, and Lens 3, see Figure 4b, which leads to Figure 4d, which we compare to the Lens 2 result, see Figure 4c. Figure 4e is Figure 4c transformed in a grayscale image, Figure 4f is Figure 4d transformed in a grayscale image, and Figure 4g is the absolute value of the difference between Figures 4e and 4f. When we compute the L^2 norm for Figure 4g, we get a result of 0.0473, which means a difference of 1% between Figures 4c and 4d, but note that nothing happens at the bottom of the images while this part is counted for the L^2 norm computation. It explains such a good result.

2.3.4 Applying the NST computation between the same FDTD computation results

Style transfer manipulates images to adapt for these images the appearance or visual style of another image. To perform this kind of trick, the style transfer technique relies on artificial intelligence, and more precisely, on neural networks.

The seminal work of Gatys et al. [20] demonstrated the power of convolutional neural networks (CNNs) in

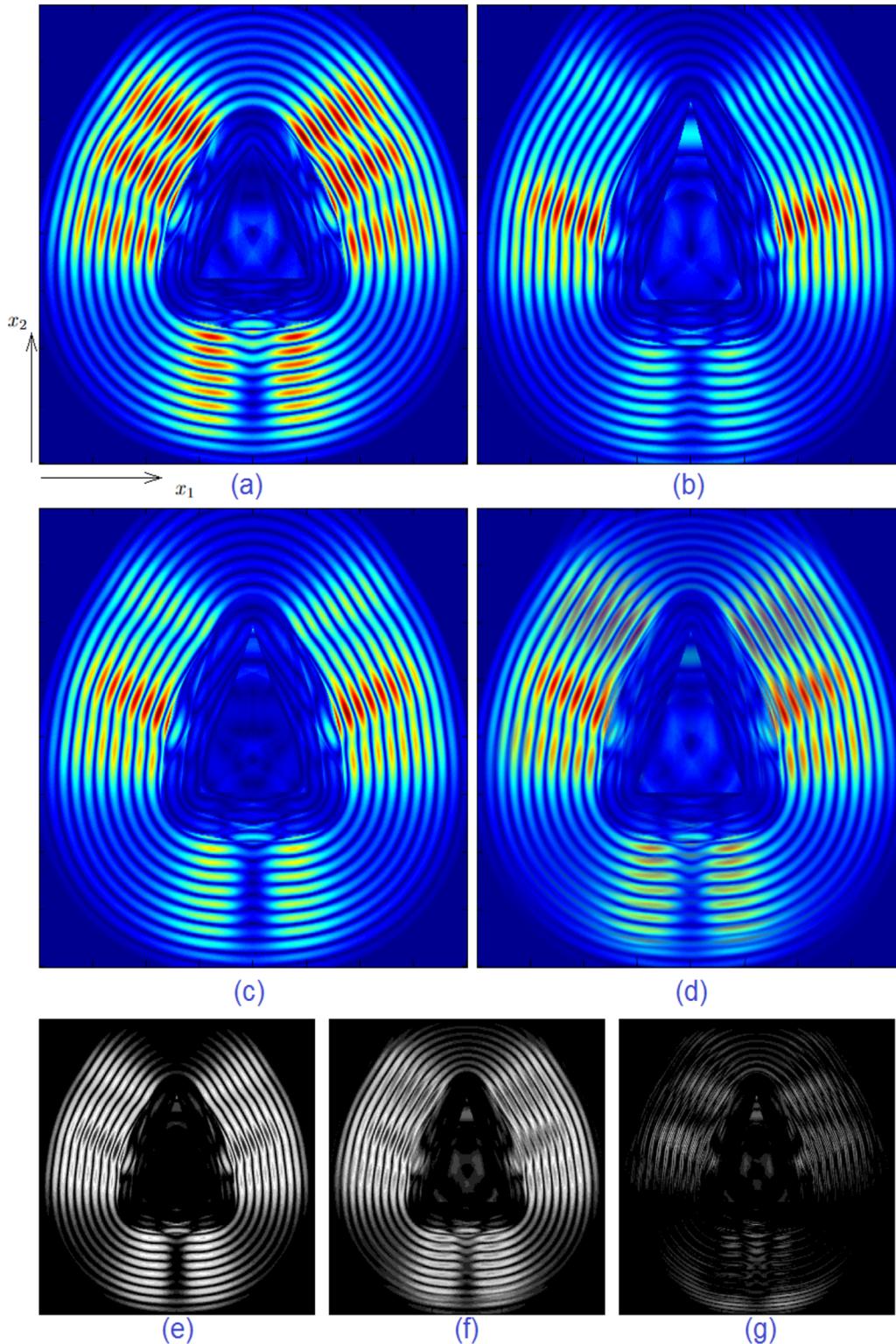


Fig. 3. Snapshot time $t = 710 \mu\text{s}$ for the radiation pattern of a point force (directed along x_2) vibrating at frequency 10 kHz located at the gravity point of an isosceles triangle of height $\frac{4}{3} \times 10\sqrt{3}$ cm and base 20 cm (a), isosceles triangle of height $\frac{6}{3} \times 10\sqrt{3}$ cm and base 20 cm (b), isosceles triangle of height $\frac{5}{3} \times 10\sqrt{3}$ cm and base 20 cm (c); morphing result (d) between (a) and (b). Note the strong similarities between (c) and (d). (e) is (c) transformed in a grayscale image, (f) is (d) transformed in a grayscale image, and (g) is the absolute value of the difference between (e) and (f). Note that the size of the field is 80 cm by 80 cm.

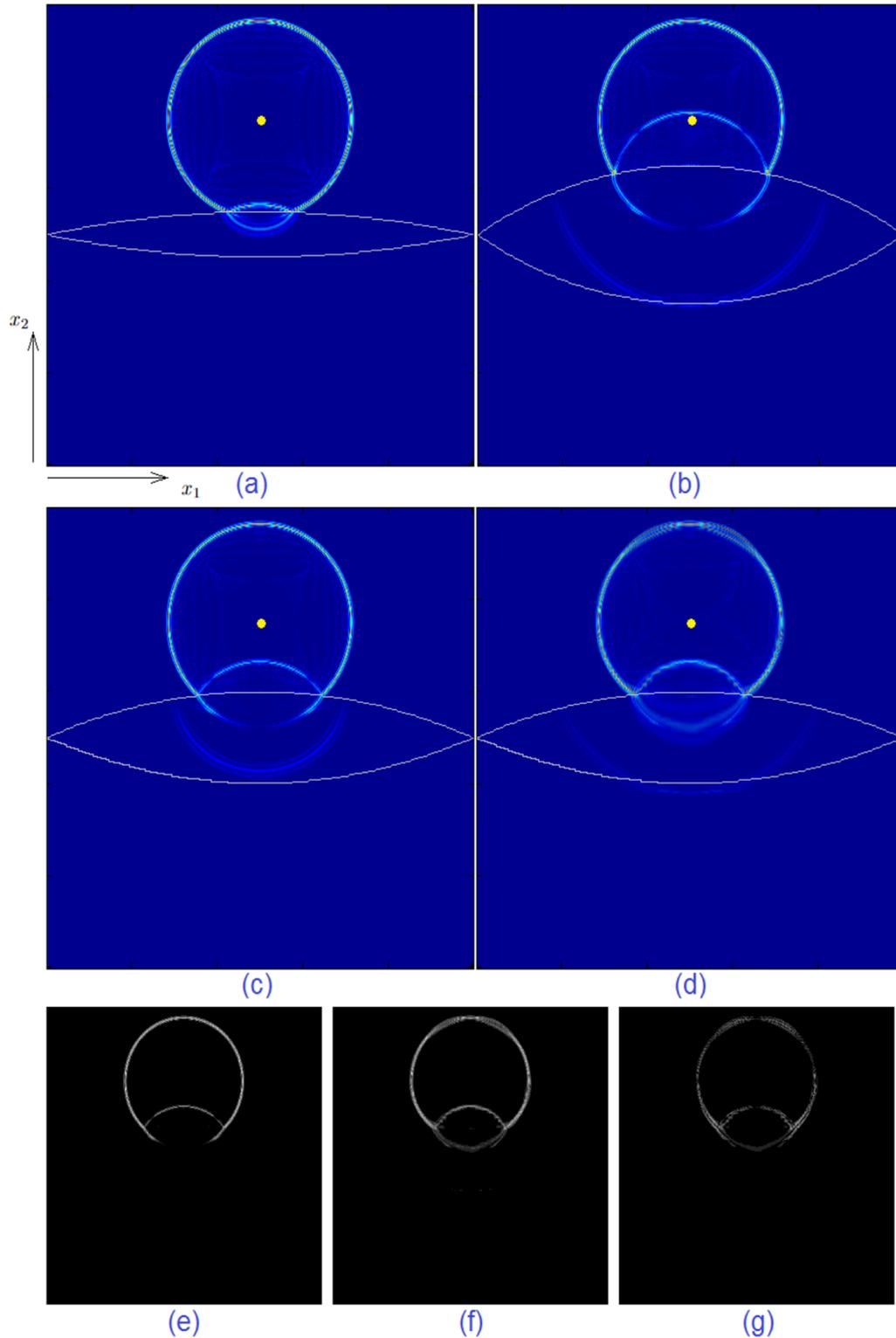


Fig. 4. Snapshot time $t = 140 \mu\text{s}$ for the radiation pattern of a point force (directed along x_2) generating a Ricker pulse with central frequency 250 kHz and spectral bandwidth 10 kHz, which is located at 12.5 cm from the top and 25 cm from the left and the right edge of a field of 50 cm by 50 cm, applied to three cases. (a) Applied to the “Lens 1”, (b) applied to the “Lens 3”, (c) applied to the “Lens 2”; morphing result (d) between (a) and (b); (e) is (c) transformed in a grayscale image; (f) is (d) transformed in a grayscale image and (g) is the absolute value of the difference between (e) and (f).

creating artistic imagery by separating and recombining image content and style. This process of using CNNs to render a content image in different styles is referred to as NST.

In a few words, this technique will first try to find specific shapes in the first image which could match with these kinds of shapes in the second image, and then, replace in the first image the content of these shapes by the corresponding content in the second image. We adapt the implementation proposed in [38], which is based on the work concerning large-scale machine learning problems and presented in [39]. In [39], a linear optimization method is studied, namely the limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm [39]. Its almost sure global convergence is established. In our examples, we have run the algorithm proposed in [38] with 10 epochs and 100 steps per epoch. For a given snapshot time, we use the NST computation between the smallest isosceles triangle, see Figure 5a, and the highest isosceles triangle results, see Figure 5b, which gives us a result, see Figure 5d we compare to the average isosceles triangle result, see Figure 5c. Note the weak similarities between Figures 5c and 5d. Figure 5e is Figure 5c transformed in a grayscale image, Figure 5f is Figure 5d transformed in a grayscale image, and Figure 5g is the absolute value of the difference between Figures 5e and 5f. When we compute the L^2 norm for Figure 5g, we get a result of 0.2598, which means a difference of 6.3% between Figures 5c and 5d. Note that the size of the domain over which we analyze the field is 80 cm by 80 cm. Moreover, the total computation duration is 170 min on our desktop computer.

For a given snapshot time, we use the NST computation between the results for Lens 1, see Figure 6a, and Lens 3, see Figure 6b. This gives us a result, see Figure 6d, that we compare to the Lens 2 result, see Figures 6c. Figure 6e is Figure 6c transformed in a grayscale image; Figure 6f is Figure 6d transformed in a grayscale image, and Figure 6g is the absolute value of the difference between Figure 6e and 6f. When we compute the L^2 norm for Figures 6g, we get a result of 0.0427, which means a difference of 1% between Figures 6c and 6d, but note that nothing happens at the bottom of the images while this part is accounted for the L^2 norm computation. This explains the remarkably good result. Figure 6d has nothing to do with Figure 6c. Figure 6d looks much more like Figure 6a, all the more so since the reflected wave does not correspond to the reflection of Lens 2, but rather to that of Lens 1.

3 Results

We explore the abilities of the morphing to face setting modifications. We no longer modify the spatial structure of the obstacle. We consider two different time intervals or two different values of the frequency of the wave.

3.1 Two new applications

For the first time in this paper, instead of creating different obstacles, we will keep the same obstacles, but we will work

with different snapshot times, for what we call the “time domain” use, and we will work with the same obstacle at the same snapshot time, but with different frequencies, for what we call the “frequency domain” use. For both “time domain” and “frequency domain” use, we will still apply the FDTD computations on our obstacles, and then, we will still apply the morphing computation between some FDTD computations results. To go further, we will also apply the style transfer computation between some FDTD computation results. For our tests in “time domain”, we will use the “Lens 2”, as described in Figure 2b, at different snapshots time, at $t=190\ \mu\text{s}$, at $t=195\ \mu\text{s}$ and at $t=200\ \mu\text{s}$, with a signal just above the lens (based at $x_1=25\ \text{cm}$ and $x_2=12.5\ \text{cm}$, still in a field of 50 cm by 50 cm) at a frequency of 250 kHz. For our tests in the frequency domain, we have designed a Luneburg lens with a radius of 5 cm based in the center of a domain of 50 cm by 50 cm. As shown in Figure 1c, this lens is made of eleven layers, or let us say ten circular layers plus a central disc. The central disc has the highest density, that is $3360\ \text{kg m}^{-3}$, and each time we cross a layer, starting from the center of the lens to the outside the lens, the density decreases by $160\ \text{kg m}^{-3}$ (see the density scale in Fig. 1c). Note that the medium outside the lens has a density of $1600\ \text{kg m}^{-3}$. The signal starts from a point force located at 5.5 cm from the center of the lens. As shown in Figure 1c, let us say that this point force is on the left side of the lens. Finally, we launch a signal at three different frequencies: 25, 30, and 35 kHz, and for each frequency, we keep the snapshot of the radiation pattern at the time $t=480\ \mu\text{s}$.

3.2 Time domain results

3.2.1 By using the morphing technique

Here we have worked on the Lens 2, still in a field of 50 cm by 50 cm, and once again with a force point at 250 kHz located at 12.5 cm from the top and at 25 cm from the left and from the right edge. In Figure 7a we show the snapshot time at $t=190\ \mu\text{s}$. Note that the reflected wave is just below the force point and starts to cross it. In Figure 7b we show the snapshot time at $t=200\ \mu\text{s}$. Note that the reflected wave is just above the force point. In Figure 7c we show the snapshot time at $t=195\ \mu\text{s}$. Note that the reflected wave is right in the middle of the force point. In Figure 7d we show the result of the morphing technique applied between Figures 7a and 7b. Note the strong similarity with Figure 7c. The reflected wave is also in the middle of the force point. Figures 7e is Figure 7c transformed into a grayscale image. Figures 7f is Figure 7d transformed into a grayscale image. Figure 7g is the absolute value of the difference between Figures 7e and 7f. We show here the small difference between Figures 7e and 7f, which is confirmed by the L^2 norm computation applied to Figure 7g, which is equal to 0.0392, that is, a difference of 1% between Figures 7e and 7f. However, we have to note that nothing happens at the bottom of the image while this part is counted for the L^2 norm computation. It explains such a good result.

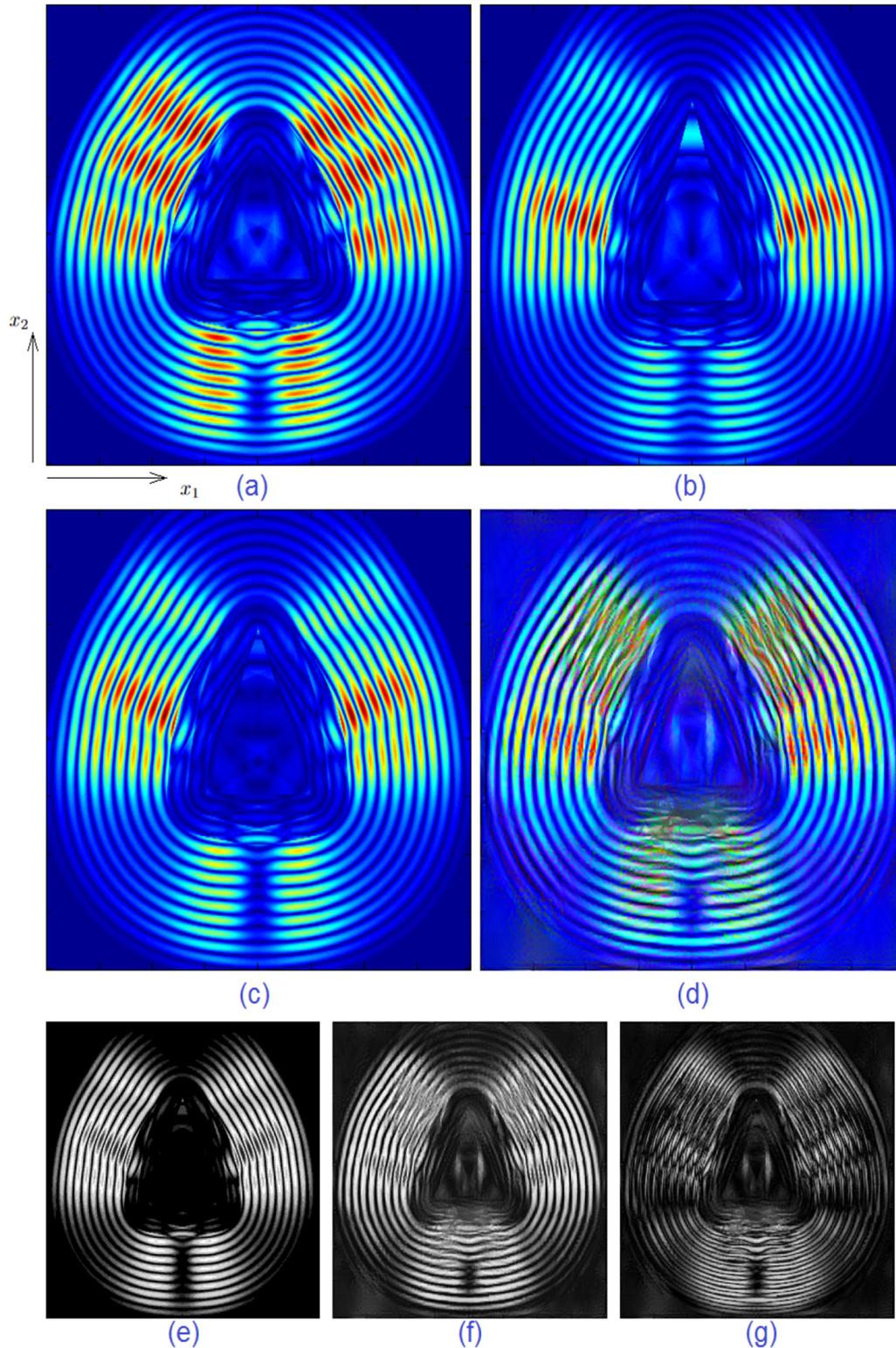


Fig. 5. Snapshot time $t = 710 \mu\text{s}$ for the radiation pattern of a point force (directed along x_2) vibrating at frequency 10 kHz and located at the gravity point of an isosceles triangle of height $\frac{4}{3} \times 10\sqrt{3}$ cm and base 20 cm (a), isosceles triangle of height $\frac{6}{3} \times 10\sqrt{3}$ cm and base 20 cm (b), isosceles triangle of height $\frac{5}{3} \times 10\sqrt{3}$ cm and base 20 cm (c); (d) show the neural style transfer result between (a) and (b). Note the weak similarities between (c) and (d). (e) is (c) transformed in a grayscale image, (f) is (d) transformed in a grayscale image and (g) is the absolute value of the difference between (e) and (f). Note that the size of the field is 80 cm by 80 cm.

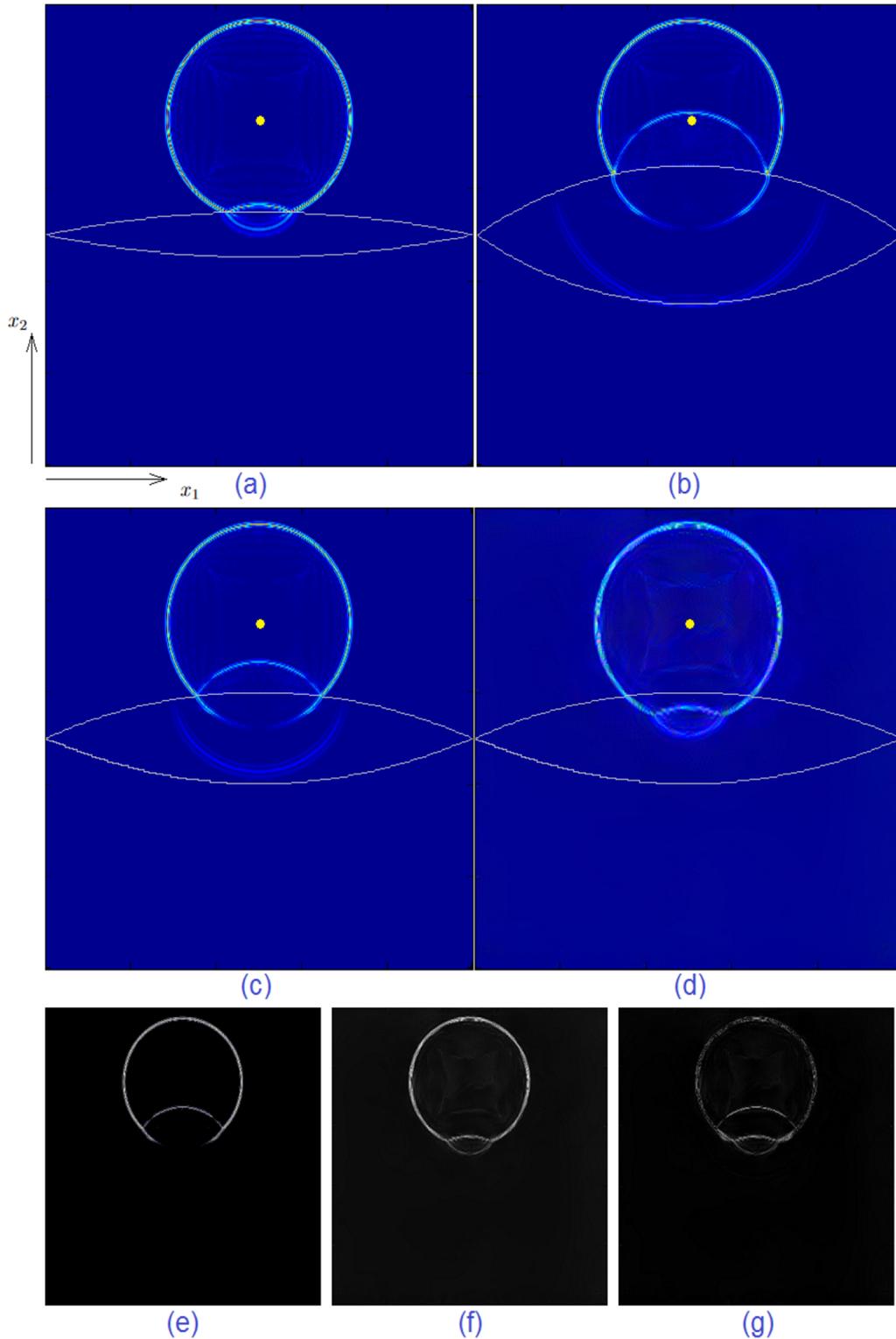


Fig. 6. Snapshot time $t = 140 \mu\text{s}$ for the radiation pattern of a point force (directed along x_2) generating a Ricker pulse with central frequency 250 kHz and spectral bandwidth 10 kHz, located at 12.5 cm from the top and 25 cm from the left and the right edge of a field of 50 cm by 50 cm, applied to three cases. (a) the “Lens 1”, (b) applied to the “Lens 3”, (c) applied to the “Lens 2” (c); (d) shows the neural style transfer result between (a) and (b), (e) is (c) transformed in a grayscale image, (f) is (d) transformed in a grayscale image and (g) is the absolute value of the difference between (e) and (f).

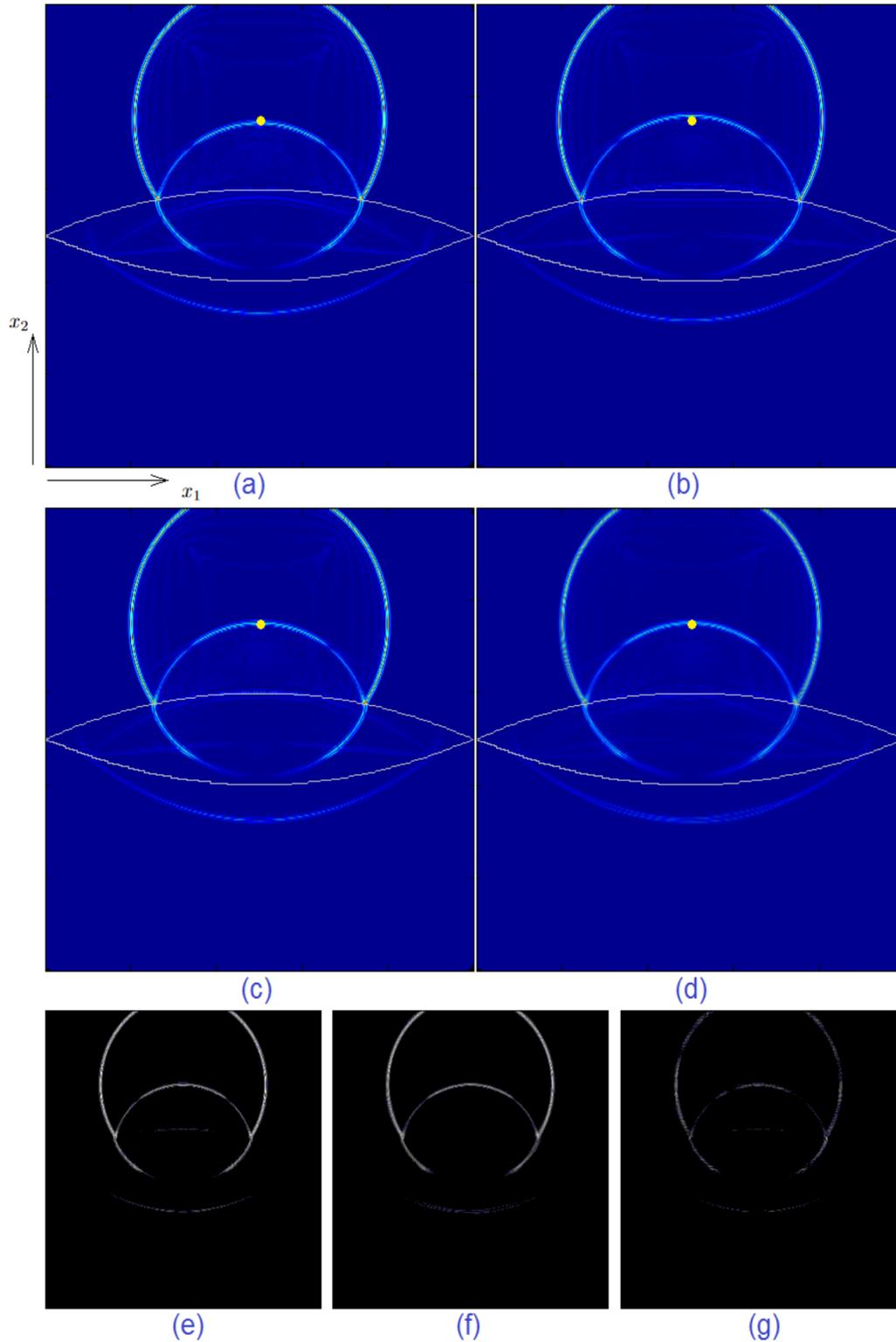


Fig. 7. Snapshot time from $t = 190 \mu\text{s}$ (a), $t = 200 \mu\text{s}$ (b), and $t = 195 \mu\text{s}$ (c), for the radiation pattern of a point force (directed along x_2) generating a Ricker pulse with central frequency 250 kHz and spectral bandwidth 10 kHz, located at 12.5 cm from the top and 25 cm from the left and the right edge of a field of 50 cm by 50 cm, applied to the “Lens 2”; morphing result (d) between (a) and (b), (e) is (c) transformed in a grayscale image, (f) is (d) transformed in a grayscale image and (g) is the absolute value of the difference between (e) and (f).

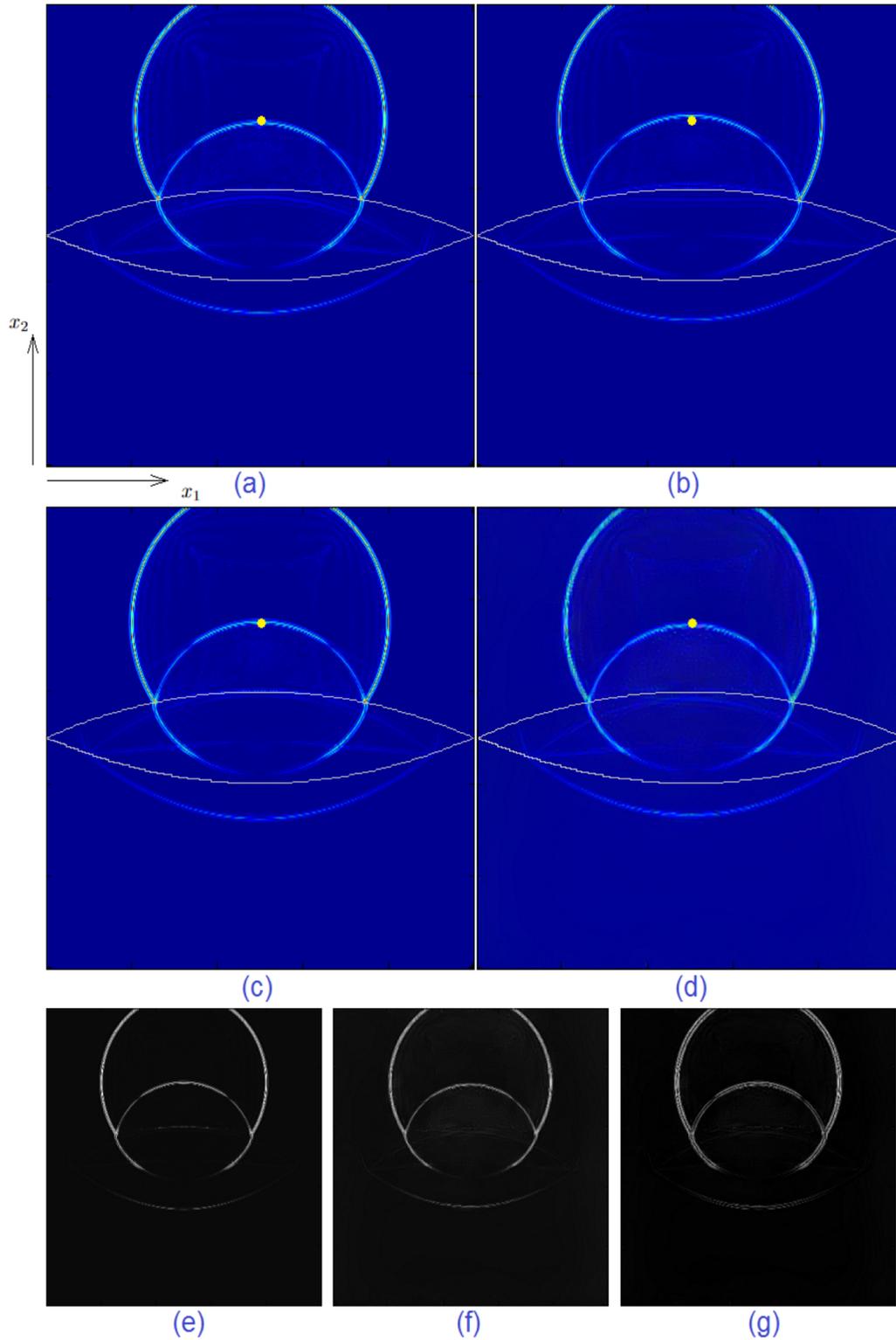


Fig. 8. Snapshot time from $t = 190 \mu\text{s}$ (a), $t = 200 \mu\text{s}$ (b), and $t = 195 \mu\text{s}$ (c), for the radiation pattern of a point force (directed along x_2) generating a Ricker pulse with central frequency 250 kHz and spectral bandwidth 10 kHz, located at 12.5 cm from the top and 25 cm from the left and the right edge of a field of 50 cm by 50 cm, applied to the “Lens 2”; style transfer result (d) between (a) and (b), (e) is (c) transformed in a grayscale image, (f) is (d) transformed in a grayscale image and (g) is the absolute value of the difference between (e) and (f).

3.2.2 By using the style transfer technique

Now we keep exactly the same settings, but we replace the morphing technique with the style transfer technique. This means that Figure 8a is the same as Figures 7a, Figure 8b is the same as Figure 7b, and Figure 8c is the same as Figure 7c. In Figure 8d we show the result of the style transfer technique applied between Figures 8a and 8b. At first sight, we could think of a strong similarity with Figure 8c, but when we focus on the reflected wave, we see that it is just below the force point and that it starts to cross this force point, exactly like it does in Figure 8a. Actually, the reflected wave should have been right in the middle of the force point, like it does in Figure 8c. This shows the weakness of the style transfer technique in the time domain application. As made in the previous tests, we have in Figure 8e, Figure 8c transformed into a grayscale image, and in Figure 8f, Figure 8d transformed into a grayscale image. In Figure 8g we have the difference between Figures 8e and 8f. This shows much more white pixels than in Figure 7g, and when we compute the L^2 norm applied to Figure 8g, we get a result of 0.0745, that is, a difference of 2% between Figures 8e and 8f. It clearly confirms that in a time domain application, the morphing technique is more efficient than the style transfer technique. Moreover, there are not so many differences between Figures 8a, 8b, and 8c (or Figs. 7a, 7b and 7c), which explains why the style transfer technique seems to work in the time domain application. Once again, we have to note that nothing happens at the bottom of the image while this part is counted for the L^2 norm computation. It also explains such a good result.

3.3 Frequency domain results

3.3.1 By using the morphing technique

For our tests in the frequency domain, we have worked on a Luneburg lens, as described in Section 3.1. We have launched a signal at three different frequencies: 25 kHz, as shown in Figure 9a, 35 kHz, as shown in Figure 9b, and 30 kHz, as shown in Figure 9c. Then, we have applied the morphing technique in Figure 9d, between Figures 9a and 9b. At first sight, we see strong similarities between Figures 9c and 9d. In Figure 9e, we display Figure 9c transformed into a grayscale image, and in Figure 9f, Figure 9d transformed into a grayscale image. Figure 9g is the absolute value of the difference between Figure 9e and 9f. When we compute the L^2 norm on Figure 9g, we get a result of 0.1118, that is, a difference of 3% between Figure 9c and 9d. We have to note that nothing happens on the right part of the image (about a third of the image), while this part is taken into account for the L^2 norm computation. It explains such a good result. Whatever it is, in this test, the starting images in Figures 9a and 9b are quite different, or, let us say, much more different than in our test in the time domain, between Figures 7a and 7b,

which shows the efficiency of the morphing technique.

3.3.2 By using the style transfer technique

Now we keep exactly the same settings as previously, but we replace the morphing technique with the style transfer technique. This means that Figure 10a is the same as Figure 9a, Figure 10b is the same as Figure 9b, and Figure 10c is the same as Figure 9c. In Figure 10d, we show the result of the style transfer technique applied between Figure 10a and 10b. At first sight, we see that Figure 10d is very different from Figure 10c, but that it looks more like Figure 10a, particularly concerning the wavelength. The biggest problem is that the crest lines of the wave pattern are blurred. In other words, the colors are poorly defined, whereas they represent the signal amplitudes. This shows the weakness of the style transfer technique in the frequency domain application. As in the previous tests, we display in Figure 10e, Figure 10c transformed into a grayscale image, and in Figure 10f, Figure 10d transformed into a grayscale image. In Figure 10g, we have the absolute value of the difference between Figures 10e and 10f. This shows much more white pixels than in Figure 10g, particularly in a kind of internal crown, and when we compute the L^2 norm applied to Figure 10g, we get a result of 0.1285, that is, a difference of 4% between Figures 10e and 10f. It clearly confirms that in a frequency domain application, the morphing technique is more efficient than the style transfer technique. Once again, we should note that nothing happens on the right part of the image (about one-third of the image), while this part is taken into account in the L^2 norm computation. It explains such a good result.

4 Discussion, or strength and weakness of morphing as an acceleration tool for FDTD

4.1 Application field

Now that we know how morphing works, we can envisage many possible applications, and not only in an FDTD, or a TO context. Morphing can be applied to other wave phenomena providing we consider linear governing equations (what corresponds to linear transforms). In our case (an elastic wave application), as we have seen, morphing proves to be already an invaluable tool.

4.2 Interest

We have seen how well morphing performs to interpolate the results of simulations. Thus, we explained why it may work and how one can use this readily available tool to accelerate numerical simulations. The main interest of this renewed approach to FDTD is to considerably reduce computational time and resources.

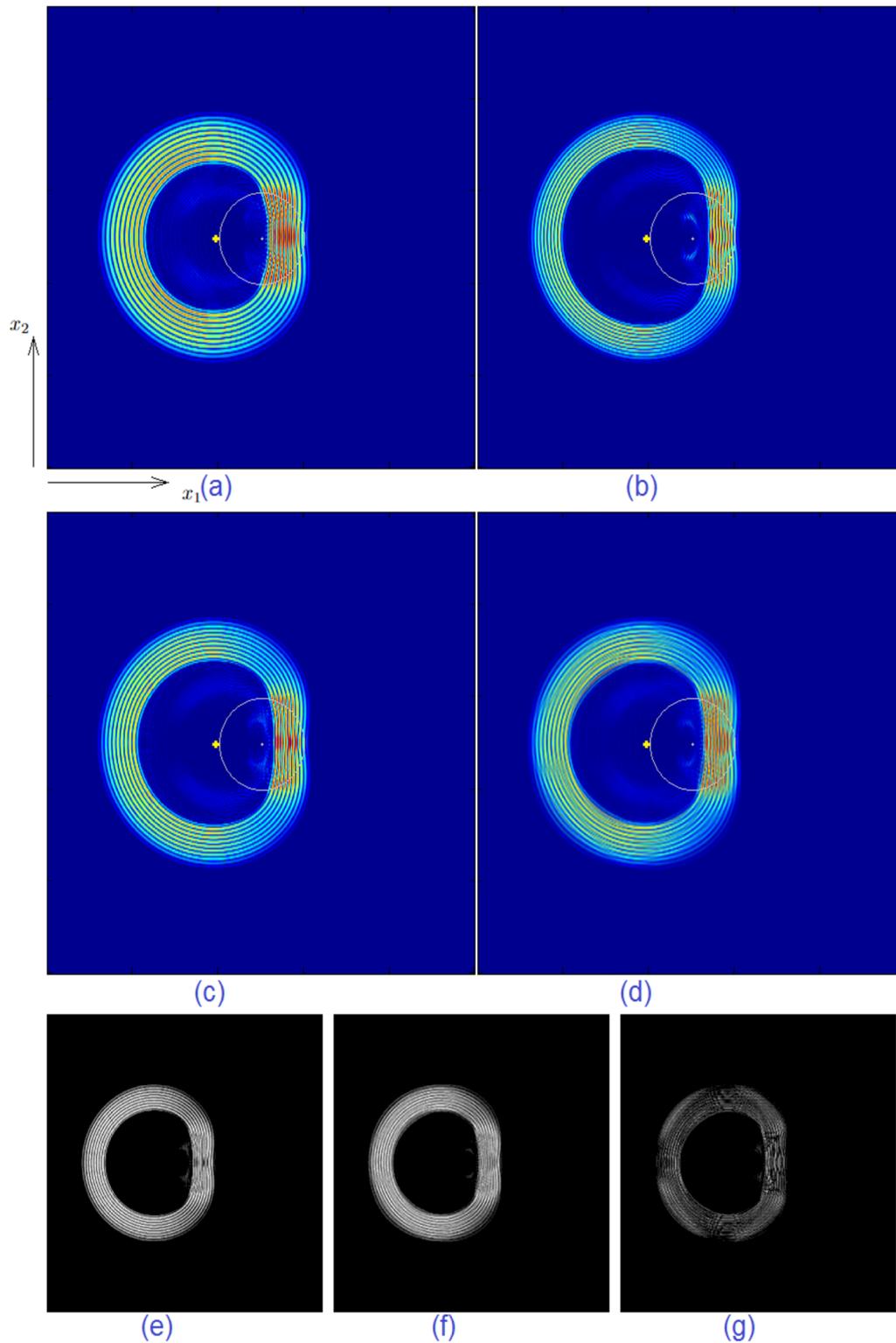


Fig. 9. Snapshot time $t = 480 \mu\text{s}$, for the radiation pattern of a point force directed along x_1 , and located at 5.5 cm from the center of a Luneburg lens with a radius of 5 cm based in the center of a field of 50 cm by 50 cm, with a frequency of 25 kHz (a), 35 kHz (b) and 30 kHz (c); morphing result (d) between (a) and (b), (e) is (c) transformed in a grayscale image, (f) is (d) transformed in a grayscale image and (g) is the absolute value of the difference between (e) and (f).

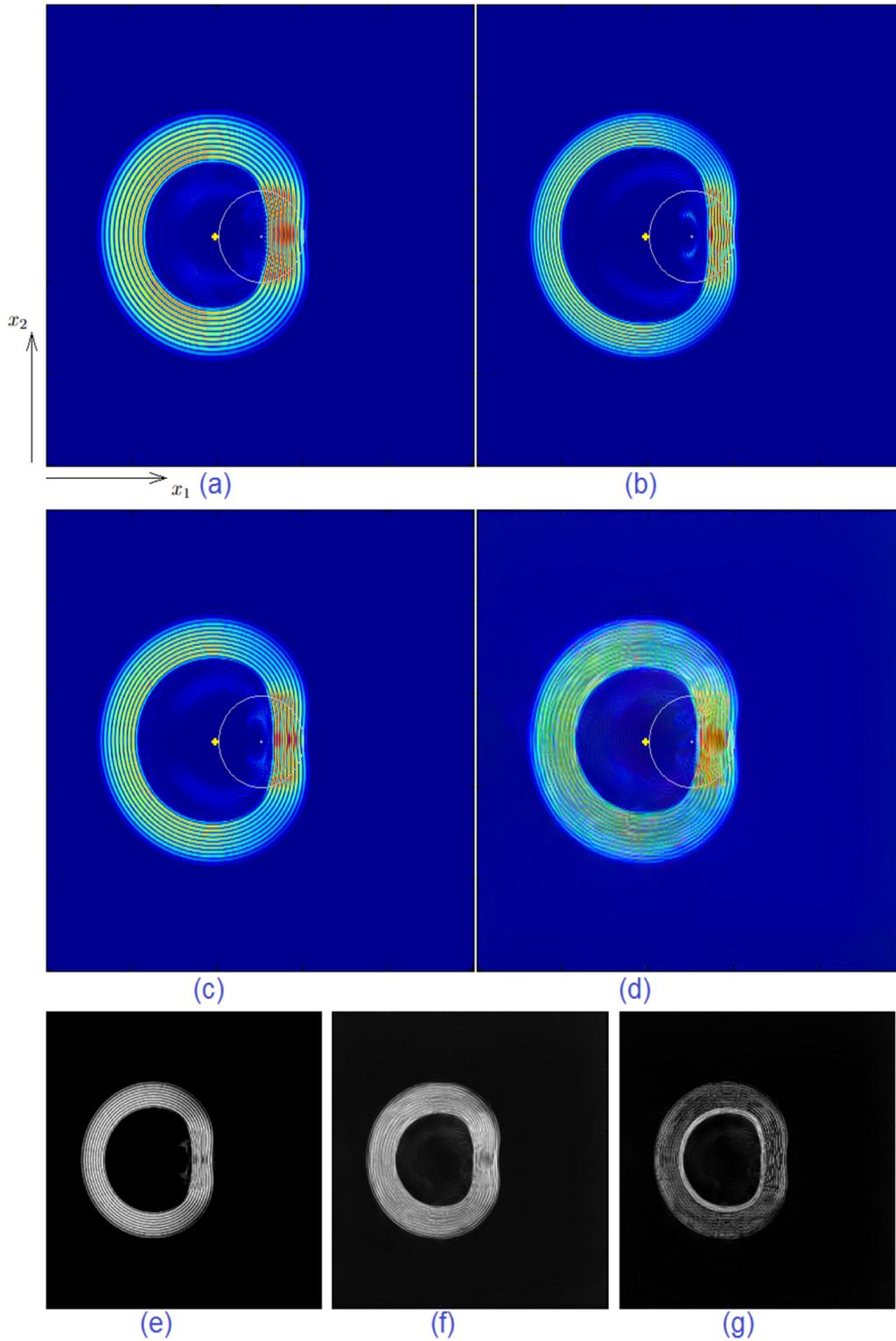


Fig. 10. Snapshot time $t = 480 \mu\text{s}$, for the radiation pattern of a point force located at 5.5 cm from the center of a Luneburg lens with a radius of 5 cm based in the center of a field of 50 cm by 50 cm, with a frequency of 25 kHz (a), 35 kHz (b) and 30 kHz (c); (d) show the style transfer result between (a) and (b), (e) is (c) transformed in a grayscale image, (f) is (d) transformed in a grayscale image and (g) is the absolute value of the difference between (e) and (f).

Table 1. Values of three metrics for morphing and style transfer experiments: L^2 norm, Mssim and ESER.

| Case | Figure | L^2 norm | Mssim | ESER |
|---------------------------------|--------|------------|--------|---------|
| Morphing spatial triangle | 3 | 0.0729 | 0.6725 | 11.2627 |
| Morphing spatial lens | 4 | 0.0473 | 0.9257 | 16.8309 |
| Style transfer spatial triangle | 5 | 0.2598 | 0.0990 | 5.6355 |
| Style transfer spatial lens | 6 | 0.0427 | 0.6927 | 15.5942 |
| Morphing time domain | 7 | 0.0392 | 0.9500 | 18.7929 |
| Style transfer time domain | 8 | 0.0745 | 0.5954 | 13.0028 |
| Morphing frequency domain | 9 | 0.1118 | 0.8858 | 10.7400 |
| Style transfer frequency domain | 10 | 0.1285 | 0.7060 | 9.0602 |

However, we still need someone to compute the starting and destination images and even more importantly someone who can judiciously place control points therein, before using any morphing algorithm.

4.3 Efficiency

Until now, morphing seems to be efficient, but this remains a qualitative feeling: by naked eyes, looking at the results in [Figures 3, 4, 7, or 9](#), we understand that the result obtained is good when a few pixels with elevated gray level values are present, each time, in Figure (g). We need, however, a quantitative metric. For this, we still consider the difference between the result (in gray level) of the computational method and either the morphing or the style transfer, and we compute the L^2 norm of this difference image, which has been scaled between 0 and 1. Note that this comparison function is not linear and behaves as a square root function. With this method, we obtained an average numerical value of 0.05 for the comparison between the two grayscale images, let us say with the difference image (g) of [Figures 3, 4, 7, and 9](#).

4.4 Limits of applicability of morphing

Even if the morphing technique has proved its efficiency with the monotonous functions, and, in particular, with the linear functions, on the other hand, this technique is no longer efficient as soon as we apply it to a non-monotonous function.

If in TO the function was the signal, in FDTD the function would correspond more to the homogeneity or heterogeneity of the medium, with a function all the more linear as the medium is homogeneous. This can be seen by comparing [Figures 9d to 9c](#): all the wave fronts that did not cross the Luneburg lens match very well between these two figures. On the other hand, the part of the wave front which crosses the Luneburg lens, and especially the part of the

wave fronts which is still on the right side of the Luneburg lens, gives a morphing result that does not match well the computed result.

If we focused our L^2 norm function between $-\frac{\pi}{6}$ and $\frac{\pi}{6}$, inside the Luneburg lens, of course we would get a quite bad result.

4.5 Comparison with other methods

To highlight the interest in morphing, it seemed interesting to us to compare it with other methods. Our first idea was to try with Machine Learning, or Deep Learning, but these kinds of methods require databases that, unfortunately, do not exist. This idea has therefore been abandoned. Then, we thought of making averages between the images of results obtained by the calculations. Here again, this idea could not work, because if the average coordinates between two points of the same image are half the distance between these two points, on the other hand, it is not the same between two points with different coordinates, each located on a different image from the other point. Indeed, the average between two images is done “point-to-point”, or, more precisely, with identical coordinates in the two images, which means that displacements are not taken into account. So the average between two images does not make sense at all. Finally, we thought about neural networks and more precisely the NST approach. At first glance, this idea might seem interesting, but the total lack of color sharpness is really catastrophic. Indeed, we must understand the importance of colors in numerical simulations: the maximum amplitude of the signal is in red, and the minimum amplitude of the signal is in blue. Each intermediate color between blue and red corresponds to a specific amplitude of the signal. If the colors are approximate, then the amplitudes are also approximate. In other words, the lack of precision of the colors leads to a false result. The tests that we have carried out with the NST give qualitative and quantitative results that are worse than those obtained using morphing.

We summarize the quantification of error between the fully numerical solutions and the morphing-based (or NST-based) solutions to the elastic wave equations in Table 1. We stress that the L^2 norm should be as close to 0 as possible. It is not a percentage of error between images because it is a square root-like curve (an L^2 norm value of 0.75 mean a difference of 50% between two images), whereas the Mssim (Mean Structural SIMilarity [40]) should be as close to 1 as possible. Finally, we also use what we call the “expected signal to error ratio” (ESER), which is a kind of “signal to noise ratio”. To calculate our ESER, we add the squares of the values of each component (red, green and blue) of each pixel of the reference image (obtained by the calculation) and we divide this result by the sum of the squares of each component of each pixel of the image to be compared (obtained by calculation result minus morphing or by style transfer result).

We calculate the decimal logarithm of the result of our division, and multiply the result by ten. If $x = \log(a)$ and $x + 1 = \log(b)$, there is only one unity between x and $x + 1$, but actually, it means a difference of ten times between a and b ($b = 10a$). However, we multiply by ten our logarithm, which means that from 10 onward, our result starts getting good.

In Figure 4, the L^2 norm is three times less than in Figure 9. One might think that the morphing has malfunctioned in Figure 9. However, the corresponding Mssim values are relatively close: 0.92 in Figure 4 and 0.88 in Figure 9.

In Figure 6 for the same application case as in Figure 4, the style transfer gives an L^2 norm which is very close to the result in Figure 4 obtained by morphing. On the other hand, the Mssim is much worse (0.69 for the style transfer against 0.92 for the morphing).

In general, morphing gives results that are slightly better in terms of L^2 norm and better in terms of Mssim.

The exception to the rule (or rather the case taken to the extreme) is Figure 5 compared to Figure 3: the result given by morphing is much better than that given by style transfer in terms of L^2 norm, and the result in terms of Mssim given by style transfer is catastrophic, which in fact makes the result obtained by morphing extremely satisfactory in comparison.

5 Concluding remarks

This paper deals with the acceleration of the computations required to study wave propagation phenomena. For this purpose, we adapt a morphing algorithm.

Firstly, we have extended a previous study to elastic phenomena, and to the frequency and time domains. In this framework, and although we mainly exemplify the abilities of morphing in an already explored application, a significant outcome of this paper, with respect to existing work, is the comparison of morphing with an up-to-date algorithm involving neural networks, namely style transfer.

Secondly, we have tested two new applications of morphing in FDTD, that is, in the temporal and frequency domains. Here again, we have proved the efficiency of the morphing technique. In the framework of these two new applications, we have further tested the above-mentioned NST technique, and here again, morphing has proved to be more efficient.

The elastic phenomena which are studied are linear, and morphing is also based on linear interpolations. So, valuable property of morphing in this case, is that the Courant–Friedrichs–Lewy conditions are inherently respected.

Referring to three different numerical criteria, namely the L^2 norm, Mssim, and ESER, morphing provides results that are slightly, or even significantly better. Referring to the visual aspect of the images obtained through morphing, and especially for the experiments in the frequency domain, the shape of the waves in the morphing result is much better preserved than in the case where style transfer is applied.

Morphing has thus proved to be an invaluable tool for the exploration of transformation-based elastic metamaterials in the time and frequency domains, as an accelerator of an FDTD algorithm. The usefulness of morphing for analysis of transformed-based electromagnetic non-dispersive metamaterials in the frequency domain has already been established in [22] for a finite element algorithm, and we are confident this should carry through for FDTD algorithms provided this remains within the framework of non-dispersive media. Extension of our study to dispersive electromagnetic and elastic metamaterials could be envisaged making use of the formalism of auxiliary fields [41,42], as this would circumvent the challenge of a non-linear setting for morphing. Finally, we point out that morphing could also be used to accelerate designs of metamaterials in combination with topological optimization techniques such as used in [43].

R. Aznavourian acknowledges funding from the National Research Technology Association for the CIFRE grant #2020/0078 of his PhD thesis.

Appendix A

Test of L^2 norm

Note that this comparison function is not linear and behaves as a square root function if we choose the L^2 norm, cf. Figure A1, wherein we also show for comparison what a norm looks like (clearly, the latter norm cannot be used to make fine estimates for small discrepancies between images, which is why we opt for the L^2 norm).

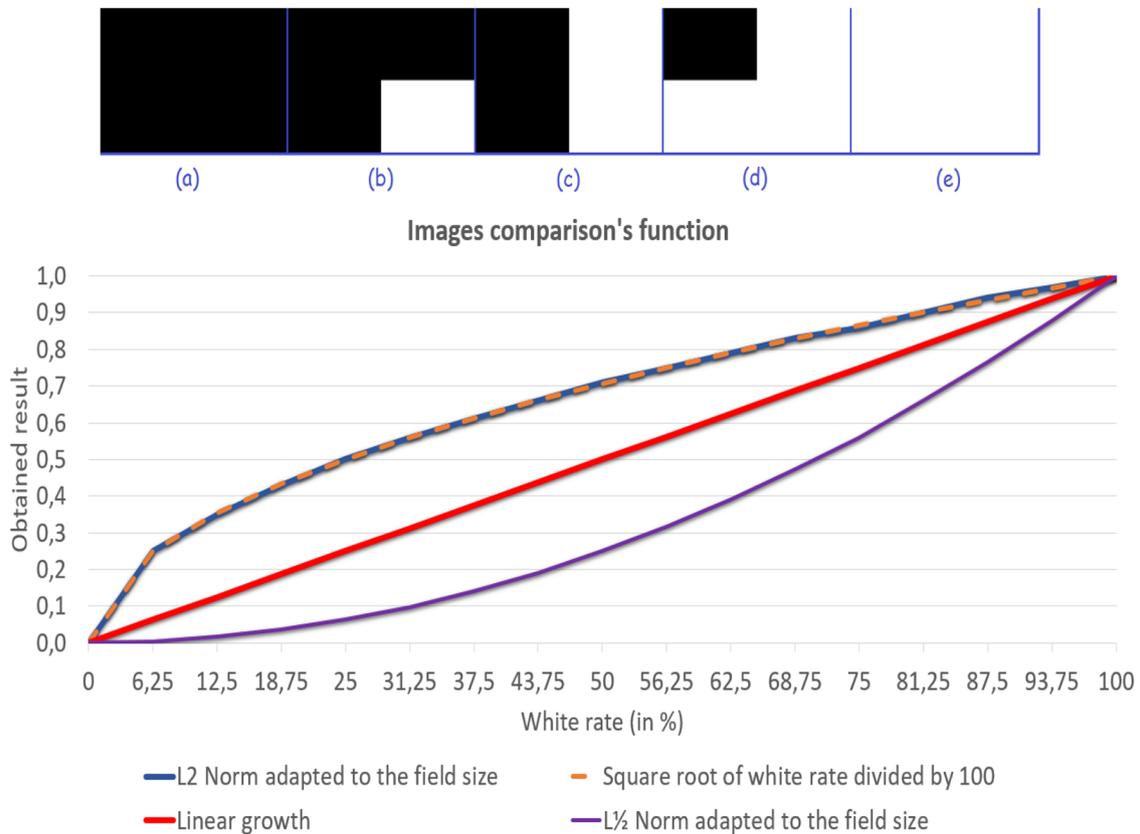


Fig. A1. Test of L^2 norm for image comparison's function (f) with black and white images with respectively 0% (a), 25% (b), 50% (c), 75% (d), and 100% (e) of white. [Reprinted/Adapted] with permission from [22] © The Optical Society.

References

- E. Macé, I. Cohen, G. Montaldo, R. Miles, M. Fink, M. Tanter, In vivo mapping of brain elasticity in small animals using shear wave imaging, *IEEE Trans. Med. Imag.* **30**, 550–558 (2010)
- P. Guéguen, P. Yves Bard, F.J. Chávez-García, Site-city seismic interaction in mexico city -like environments: an analytical study, *Bull. Seismolog. Soc. Am.* **92**, 794–811 (2002)
- S. Brûlé, E.H. Javelaud, S. Enoch, S. Guenneau, Experiments on seismic metamaterials: molding surface waves, *Phys. Rev. Lett.* **112**, 133901 (2014)
- Y. Achaoui, T. Antonakakis, S. Brûlé, R.V. Craster, S. Enoch, S. Guenneau, Clamped seismic metamaterials: ultra-low frequency stop bands, *New J. Phys.* **19**, 063022 (2017)
- M. Miniaci, A. Krushynska, F. Bosia, N.M. Pugno, Large scale mechanical metamaterials as seismic shields, *New J. Phys.* **18**, 083041 (2016)
- A. Palermo, S. Krödel, K.H. Matlack, R. Zaccherini, V.K. Dertimanis, E.N. Chatzi, A. Marzani, C. Daraio, Hybridization of guided surface acoustic modes in unconsolidated granular media by a resonant metasurface, *Phys. Rev. Appl.* **9**, 054026 (2018)
- D. Komatitsch, J.-P. Vilotte, The spectral element method: an efficient tool to simulate the seismic response of 2d and 3d geological structures, *Bull. Seismolog. Soc. Am.* **88**, 368–392 (1998)
- B. Lombard, J. Piraux, Numerical treatment of two-dimensional interfaces for acoustic and elastic waves, *J. Comput. Phys.* **195**, 90–116 (2004)
- A. Colombi, S. Guenneau, P. Roux, R.V. Craster, Transformation seismology: composite soil lenses for steering surface elastic rayleigh waves, *Sci. Rep.* **6**, 1–9 (2016)
- R. Aznavourian, T.M. Puvirajesinghe, S. Brûlé, S. Enoch, S. Guenneau, Spanning the scales of mechanical metamaterials using time domain simulations in transformed crystals, graphene flakes and structured soils, *J. Phys.: Condens. Matter* **29**, 433004 (2017)
- M. Kadic, T. Bückmann, R. Schittny, M. Wegener, Metamaterials beyond electromagnetism, *Rep. Prog. Phys.* **76**, 126501 (2013)
- A. Nicolet, J.-F. Remacle, B. Meys, A. Genon, W. Legros, Transformation methods in computational electromagnetism, *J. Appl. Phys.* **75**, 6036–6038 (1994)
- J.-P. Berenger, A perfectly matched layer for the absorption of electromagnetic waves, *J. Comput. Phys.* **114**, 185–200 (1994)
- A. Diatta, M. Kadic, M. Wegener, S. Guenneau, Scattering problems in elastodynamics, *Phys. Rev. B* **94**, 100105 (2016)
- Z.A. Kudyshev, V.M. Shalaev, A. Boltasseva, Machine learning for integrated quantum photonics, *ACS Photonics* **8**, 34–36 (2020)
- W.W. Ahmed, M. Farhat, X. Zhang, Y. Wu, Deterministic and probabilistic deep learning models for inverse design of broadband acoustic cloak, *Phys. Rev. Res.* **3**, 013142 (2021)

17. S. Chu, A. Vallecchi, C.J. Stevens, E. Shamonina, Fields and coupling between coils embedded in conductive environments, *EPJ Appl. Metamat.* **5**, 2 (2018)
18. R. Palmeri, T. Isernia, Inverse design of ebg waveguides through scattering matrices, *EPJ Appl. Metamat.* **7**, 10 (2020)
19. L. Pomot, C. Payan, M. Remillieux, S. Guenneau, Acoustic cloaking: Geometric transform, homogenization and a genetic algorithm, *Wave Motion* **92**, 102413 (2020)
20. L.A. Gatys, A.S. Ecker, M. Bethge, Image style transfer using convolutional neural networks, in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 2414–2423
21. Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, M. Song, Neural style transfer: a review, *IEEE Trans. Visual. Comput. Graph.* **26**, 3365–3385 (2019)
22. R. Aznavourian, S. Guenneau, Morphing for faster computations in transformation optics, *Opt. Express* **22**, 28301–28315 (2014)
23. D. Terzopoulos, J. Platt, A. Barr, K. Fleischer, Elastically deformable models, in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (1987), pp. 205–214
24. U. Leonhardt, T. Philbin, *Geometry and light: the science of invisibility* (Courier Corporation, 2010)
25. K. Yee, Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media, *IEEE Trans. Antennas Propag.* **14**, 302–307 (1966)
26. A. Hrennikoff, Solution of problems of elasticity by the framework method, *J. Appl. Mech.* **8**, A169–A175 (1941)
27. R. Courant et al., Variational methods for the solution of problems of equilibrium and vibrations, *Bull. Amer. Math. Soc.* **49**, 1–23 (1943)
28. A. Bossavit, Solving maxwell equations in a closed cavity, and the question of spurious modes', *IEEE Trans. Magn.* **26**, 702–705 (1990)
29. R. Courant, K. Friedrichs, H. Lewy, Über die partiellen differenzengleichungen der mathematischen physik, *Math. Annalen* **100**, 32–74 (1928)
30. S.-Y. Lee, K.-Y. Chwa, J. Hahn, S.Y. Shin, Image morphing using deformation techniques, *J. Visual. Comput Anim.* **7**, 3–23 (1996)
31. xiberpix, Sqirlzmorph. <http://www.xiberpix.net/SqirlzMorph.html>, 2009
32. E. Bossy, Simsonic. <http://www.simsonic.fr>
33. E. Bossy, M. Talmant, P. Laugier, Effect of bone cortical thickness on velocity measurements using ultrasonic axial transmission: a 2d simulation study, *J. Acoust. Soc. Am.* **112**, 297–307 (2002)
34. Y. Kartynnik, A. Ablavatski, I. Grishchenko, M. Grundmann, Real-time facial surface geometry from monocular video on mobile gpus. [arXiv:1907.06724](https://arxiv.org/abs/1907.06724) (2019)
35. A. Zadeh, Y.C. Lim, T. Baltrusaitis, L.-P. Morency, Convolutional experts constrained local model for 3d facial landmark detection, in *Proceedings of the IEEE International Conference on Computer Vision Workshops* (2017), pp. 2519–2528
36. F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, M. Grundmann, Mediapipe hands: On-device real-time hand tracking. [arXiv:2006.10214](https://arxiv.org/abs/2006.10214) (2020)
37. L. Ge, Z. Ren, Y. Li, Z. Xue, Y. Wang, J. Cai, J. Yuan, 3d hand shape and pose estimation from a single rgb image, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019)
38. G. Ghiasi, H. Lee, M. Kudlur, V. Dumoulin, J. Shlens, Exploring the structure of a real-time, arbitrary neural artistic stylization network. [arXiv:1705.06830](https://arxiv.org/abs/1705.06830) (2017)
39. A. Mokhtari, A. Ribeiro, Global convergence of online limited memory bfgs, *J. Mach. Learn. Res.* **16**, 3151–3181 (2015)
40. Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE Trans. Image Process.* **13**, 600–612 (2004)
41. B. Gralak, A. Tip, Macroscopic maxwell's equations and negative index materials, *J. Math. Phys.* **51**, 052902 (2010)
42. C. Bellis, B. Lombard, Simulating transient wave phenomena in acoustic metamaterials using auxiliary fields, *Wave Motion* **86**, 175–194 (2019)
43. B. Vial, Y. Hao, Topology optimized all-dielectric cloak: design, performances and modal picture of the invisibility effect, *Opt. Express* **23**, 23551–23560 (2015)

Cite this article as: Ronald Aznavourian, Sébastien Guenneau, Bogdan Ungureanu, Julien Marot, Morphing for faster computations with finite difference time domain algorithms, *EPJ Appl. Metamat.* **9**, 2 (2022)