Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 399 (2022) 115280
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Finite element approximation of wave problems with correcting terms based on training artificial neural networks with fine solutions

Arnau Fabra[a], Joan Baiges[a,b], Ramon Codina[a,b,*]

[a] Universitat Politècnica de Catalunya, Barcelona Tech, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain
[b] Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE), Edifici C1, Campus Nord UPC, Gran Capità S/N, 08034 Barcelona, Spain

Received 23 November 2021; received in revised form 28 February 2022; accepted 21 June 2022
Available online 15 July 2022

Abstract

In this paper we present a general idea to correct coarse models by introducing a correcting term designed from fine solutions that can be then applied to situations in which a fine solution is not available. We apply this idea to the wave equation in both the time domain and the frequency domain. This correcting term is computed and trained by making use of learning algorithms, such as the least squares model or a model constructed from an artificial neural network. The performance of the method is tested through different acoustic numerical examples, where the fine solutions are characterized for having a finer discretization either in time or in space.

Keywords: Wave equation; Stabilized finite element methods; Least squares training; Artificial neural networks

1. Introduction

Numerical simulations have become one of the most important tools in a considerable amount of scientific fields. In spite of this, the computational cost is still the main limitation, as the growing of computational power leads to deal with more complex problems and to target more accurate simulations. In order to overcome this obstacle, one of the proposed solutions is the use of the so called Reduced Order Models (ROM) (see, e.g. [1–3]). Their main objective is to reproduce the features and accuracy of Full Order Models (FOM), at least for some outputs of interest to be defined for the problem at hand, with models that require less degrees of freedom and therefore less computational cost. Obviously, *classical* ROM is not the only method that makes use of the idea of reducing the number of degrees of the problem (see for example [4–7]). The simplest way to achieve this reduction is by *coarsening* the numerical approximation, either in space, in time or both, but having at our disposal samples of FOM solutions. These samples of FOM solutions can be used to design correcting terms for the coarse model, but with the objective of applying the coarse model to situations in which the FOM solution is not available. This can be due to several reason discussed in the paper, such as different physical parameters, different data (forcing terms, boundary conditions) or different time instants. In this sense, we can view the coarse model as a parametrized one, in the spirit of parametrized ROM models [2,3]. This is the idea explored in this paper. In this sense, this work is an

* Corresponding author at: Universitat Politècnica de Catalunya, Barcelona Tech, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain.
E-mail addresses: afabra@cimne.upc.edu (A. Fabra), joan.baiges@upc.edu (J. Baiges), ramon.codina@upc.edu (R. Codina).

https://doi.org/10.1016/j.cma.2022.115280
0045-7825/

extension of [8], where a ROM based on adaptive finite element (FE) meshes and correction terms combined with Artificial Neural Networks (ANNs) is presented. The aim of the present paper is threefold: first, to formalize the idea presented in [8], second, to broaden its scope, including time discretization and parameter dependent models, and third, to apply this to wave problems; so firstly, a little review of the wave problems is presented, secondly, how this specific method to reduce the number of degrees of freedom works is explained, and finally, the contributions of this paper are exposed.

Wave problems can be analyzed and solved in two separate frameworks, the time domain and the frequency domain, the change from one to the other being provided by the Fourier transform. Furthermore, the equations of each domain can be formulated in two different forms: the irreducible form, where just one variable needs to be solved, and the mixed form, where two or more variables have to be computed. In order to change from one form to the other, some analytical manipulation is needed. The variables of the equations depend on the problem being considered; for instance, in acoustics the irreducible form would only involve the acoustic pressure, whereas the mixed form would require solving for both the acoustic pressure and the acoustic velocity. For conciseness, we will refer to this case, although any other wave problem could be considered.

The irreducible form of the wave equation in the time domain is perhaps the most classical form of the wave equation. This is a very well-known second order linear partial differential equation which describes the temporal evolution of the velocity or the pressure of waves. The irreducible wave equation in the frequency domain is also an equation with vast literature, also known as the Helmholtz equation; it describes the velocity or the pressure of the wave for a certain frequency. The mixed form of the wave equation in the time domain is less common; it describes the temporal evolution of the pressure and the velocity of waves at the same time. One of its advantages compared to irreducible forms is that it can provide a better approximation of the acoustic velocity and can be applied to time dependent domains using an Arbitrary Lagrangian Eulerian (ALE) approach [9]. The last case is the least studied one, it is the mixed form of the wave equation in the frequency domain; it describes both the velocity and the pressure of the wave for a certain frequency [10]. This approach has the advantages of the mixed form of the wave equation and the computational savings associated to solving for a single frequency. For more information about the wave equations in the framework of acoustics, see [11,12].

One of the most popular methods to work in the field of acoustics is the boundary element method (see, e.g., [13–15]). However, in some situations, like small domains or aero-acoustic problems, the FE method is also a good option for the spatial discretization of the problem (see [16]). Since we are interested in working with small domains, the FE method is used in this paper. While for the irreducible form in the time domain the standard Galerkin method can be used for the spatial discretization, all other forms of the wave equation require some sort of stabilization. The Helmholtz equation needs to be stabilized when high frequencies are considered, leading to pollution errors (see [17] and references therein), whereas the mixed forms require stabilization if one wants to use equal interpolation for the acoustic pressure and the acoustic velocity (see [18]). The FE formulation we employ belongs to the computational framework of Variational Multi-Scale (VMS) methods [19,20], a group of techniques based on splitting the unknowns of the problem into two different scales, where one of those scales can be approximated by the FE mesh and the other one represents the sub-grid scale, which cannot be captured by the FE space. Thanks to these techniques, the instability problems of the Galerkin method can be solved.

To compute accurate solutions, the number of degrees of freedom of the problem can be huge, which causes an enormous computational cost and, many times, results in the need for using supercomputing facilities. This is where the aforementioned ROMs come in and, in particular, the mesh coarsening approach with correcting terms presented in [8], which departs from the work presented in [21]. The idea presented there consists of a training model based on two equal cases that have two different meshes; the case of the coarse mesh plays the role of the ROM and case of the fine mesh plays the role of the FOM. This second one is computationally expensive but necessary, since it provides the information that we need in order to create the correcting model. The required data of the fine mesh is sent to the coarse mesh and the results are compared and introduced to a training algorithm. This algorithm receives the results of the FOM and projects them to the coarse mesh. With this information, it is able to create a correction term that will be added to the coarse approximation in future simulations, thus yielding results closer to the ones that would be obtained with the fine model.

The main objective of this work is to construct and make use of this kind of training models in wave problems, and not only applying these correction models in the space discretization, but also adapting and using them for the time discretization. It is important to note that even though we use these models for this specific purpose,

they are actually a general method that allows one to transfer information from fine models (FOMs) to coarse models (e.g., classical ROMs), so they could be used in many other applications. In this sense, our approach can be considered a *hybrid* one, combining an underlying numerical approximation (a FE approximation, in our case) with a correction term designed by a training model. The most common hybrid models combine physical models with corrections, but the same idea applies by replacing physical models by numerical models. We will not put under question the accuracy of the fine models, which in fact could be not even numerical; we will give for granted that the coarse solution should be as close as possible to that of the fine model. Obviously, if the fine solution is not accurate, the correction term will not produce accurate results.

Regarding the correction model, we will explore two options. On the one hand, a classic Least Squares (LS) model will be used (see [22]); it turns out to work well in simple models but it has limitations in complex scenarios. On the other hand, ANNs will be used, its flexibility and capacity to adapt to almost every model being its main feature. The implementation used for this learning algorithm is the one provided by the open source library FANN (see [23]). The use of ANNs to train models that correct solutions of wave problems is one of the main novelties of this work. However, it is important to note that ANNs have been applied to FE problems in a lot of different ways and approaches (see [24,25]), although not as a correction tool as we propose here. ANNs have been also used to perform the numerical time integration itself [26,27], as a way to design ROMs (see [28]) or as a way to approximate the Helmholtz equation with an unsupervised learning technique (see [29]). Our objective *is not* to analyze in depth the use of ANNs and to study the best alternative, but only to show that we can obtain an improvement in the coarse solution from the knowledge of the fine one. Note that the lowest bound for the error in the coarse mesh is obviously the interpolation error, and this cannot be improved.

This paper is organized as follows. In Section 2, the equations of the wave problem, their variational forms and their discretization in time and space are described. In Section 3, the general concept of the correcting term is presented. In Section 4, the concept that has been explained in general in the previous section is applied to the wave equation. In Section 5, five acoustic numerical examples are shown. Finally, in Section 6 conclusions are drawn.

2. Problem statement and finite element approximation

2.1. The continuous wave equation

Wave phenomena can be described by the wave equation written in different forms. To fix ideas, in this paper we will use the terminology of acoustics, so that the unknowns to compute are the acoustic pressure (p) and the acoustic velocity ($\boldsymbol{u}$) in the case of mixed formulations, and just the acoustic pressure in the case of irreducible formulations.

The problems we consider are initial and boundary value problems posed in a spatial domain $\Omega \subset \mathbb{R}^d$, where $d = 1, 2, 3$ is the spatial dimension. Let $\partial\Omega$ be the boundary of the domain Ω. We split this boundary into two disjoint sets denoted as Γ_p and Γ_u, where the boundary conditions corresponding to the unknowns p and $\boldsymbol{n}\cdot\boldsymbol{u}$ will be enforced, respectively, $\boldsymbol{n}$ being the unit normal to $\partial\Omega$. The time domain is the interval $(0, T)$. Initial conditions need to be provided at time $t = 0$.

Even though we will only present numerical examples corresponding to irreducible formulations, the ideas to be proposed can be also applied to mixed formulations. Let us start writing the mixed formulation in the time domain, which consists of finding the acoustic pressure $p : \Omega \times (0, T) \to \mathbb{R}$ and the acoustic velocity $\boldsymbol{u} : \Omega \times (0, T) \to \mathbb{R}^d$ such that

$$\begin{cases} \mu_p \partial_t p + \nabla \cdot \boldsymbol{u} = f_p \\ \mu_u \partial_t \boldsymbol{u} + \nabla p = \boldsymbol{f}_u, \end{cases} \tag{1}$$

with the initial conditions

$$p(\boldsymbol{x}, 0) = p_0(\boldsymbol{x}), \quad \boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega, \tag{2}$$

$p_0(\boldsymbol{x})$ and $\boldsymbol{u}_0(\boldsymbol{x})$ being given functions, and with the boundary conditions

$$p = 0 \ \text{ on } \ \Gamma_p, \quad \boldsymbol{n}\cdot\boldsymbol{u} = 0 \ \text{ on } \ \Gamma_u, \quad t \in (0, T). \tag{3}$$

For simplicity, the boundary conditions have been considered homogeneous. Non-reflecting boundary conditions (NRBCs) could also be easily incorporated, and in fact we will use them in the numerical examples. The forcing terms f_p and $\boldsymbol{f}_u$ are assumed to be given and $\mu_p > 0$, $\mu_u > 0$ are physical properties.

The mixed problem in the frequency domain consists of finding $p : \Omega \to \mathbb{C}$ and $\boldsymbol{u} : \Omega \to \mathbb{C}^d$ such that

$$\begin{cases} -i\mu_p \omega\, p + \nabla \cdot \boldsymbol{u} = f_p \\ -i\mu_u \omega\, \boldsymbol{u} + \nabla p = \boldsymbol{f}_u, \end{cases} \tag{4}$$

with the same boundary conditions as in Eq. (3). In fact, the unknowns here are the amplitudes associated to the given frequency ω of the unknowns in Eqs. (1), although we have used the same symbol in both problems, and likewise for the forcing terms.

Let $c = (\mu_u \mu_p)^{-1/2}$ be the speed of the wave. The irreducible wave equation in the time domain is obtained by taking the time derivative of the first equation in Eqs. (1) and making use of the second to eliminate the acoustic velocity as unknown. Still calling f_p the resulting right-hand-side (RHS), the equation found consists of finding $p : \Omega \times (0, T) \to \mathbb{R}$ such that

$$\frac{1}{c^2} \partial_{tt} p - \Delta p = f_p, \tag{5}$$

with the initial conditions

$$p(\boldsymbol{x}, 0) = p_0(\boldsymbol{x}), \qquad \partial_t p|_{t=0} = \dot{p}_0(\boldsymbol{x}), \qquad \boldsymbol{x} \in \Omega, \tag{6}$$

where $p_0(\boldsymbol{x})$ and $\dot{p}_0(\boldsymbol{x})$ are given functions, and boundary conditions

$$p = 0 \text{ on } \Gamma_p, \qquad \boldsymbol{n} \cdot \nabla p =: \frac{\partial p}{\partial n} = 0 \text{ on } \Gamma_u. \tag{7}$$

Let ω be a given frequency and $k = \frac{\omega}{c}$ the wave number associated to this frequency and the wave speed c. The irreducible form of the wave equation in the frequency domain consists of finding $p : \Omega \to \mathbb{C}$ solution of the Helmholtz equation

$$-\Delta p - k^2\, p = f_p, \tag{8}$$

with the same boundary conditions as in Eq. (7). The same comments regarding the notation as in the mixed form in the frequency domain apply now. This equation can be found considering that p is the amplitude associated to the frequency ω of the solution of Eq. (5) or directly multiplying by $i\omega$ the first equation in Eqs. (4) and making use of the second to eliminate the acoustic velocity as an unknown of the problem.

To write the weak form of the problem, we need to introduce some notation. Let w be a subdomain of Ω. $L^2(w)$ denotes the space of square integrable functions in w, with inner product $(\cdot, \cdot)_w$, $L^2(w)^d$ is the space of vector valued functions with components in $L^2(w)$, $H^1(w)$ is the space of functions in $L^2(w)$ with derivatives in $L^2(w)$ and $H(\mathrm{div}, w)$ is the space of vector functions with components and divergence in $L^2(w)$. We will use the simplification $(a, b)_\Omega \equiv (a, b)$.

Let $V_p := \{q \in H^1(\Omega) \mid q = 0 \text{ on } \Gamma_p\}$ and let us define the bilinear form $B(p, q) := (\nabla p, \nabla q)$ and the linear form $L(q) := (f_p, q)$ (the meaning of f_p depending on the problem). The weak form of the irreducible wave equation in the time domain consists of finding $p : (0, T) \to V_p$ such that

$$\begin{aligned} &\frac{1}{c^2}(\partial_{tt} p, q) + B(p, q) = L(q), \qquad \forall q \in V_p, \\ &(p(0), q) = (p_0, q), \qquad (\partial_t p(0), q) = (\dot{p}_0, q), \qquad \forall q \in L^2(\Omega), \end{aligned}$$

whereas the weak form in the frequency domain consists of finding $p \in V_p$ such that

$$-k^2 (p, q) + B(p, q) = L(q), \qquad \forall q \in V_p.$$

Note that in this case p may be complex valued, whereas for the problem in the time domain it is real valued.

For the irreducible problems (either in the time or in the frequency domains) p needs to be in $H^1(\Omega)$ and the associated acoustic velocity (which could be computed after knowing p) belongs to $L^2(\Omega)^d$. The weak form of the mixed formulation can be written considering two different functional settings, one which is the same as for the irreducible form and another one in which p belongs to $L^2(\Omega)$ in space and $\boldsymbol{u}$ to $H(\mathrm{div}, \Omega)$. However, since we will present numerical results for the irreducible formulation only, we shall omit details of the mixed formulation, for the sake of conciseness. See [10,18,30] for details.

2.2. Finite element approximation in space

Let us assume for simplicity that Ω is a polyhedral domain and let $\mathcal{T}_h = \{K\}$ be a FE partition of size h. The collection of interior edges will be denoted by $\{E\}$. The subscript h will be employed to refer to FE functions and FE spaces. The L^2 inner product in K (resp. E) will be represented as $(\cdot,\cdot)_K$ (resp. E). From $\mathcal{T}_h$ we may construct conforming FE spaces $V_{p,h} \subset V_p$ to approximate the pressure. In the case of mixed formulations, special care is needed to construct the velocity FE space to yield a stable numerical formulation; the approach we favor is to use arbitrary interpolations for velocity and pressure and modify the standard Galerkin method using a stabilized FE formulation, in our case based on the Variational Multi-Scale (VMS) approach [18–20,31].

For the irreducible formulation in the time domain, the standard Galerkin method is stable. The semi-discrete problem (discretized in space, continuous in time) reads: find $p_h : (0, T) \to V_{p,h}$ such that

$$\frac{1}{c^2}(\partial_{tt} p_h, q_h) + B(p_h, q_h) = L(q_h), \qquad \forall q_h \in V_{p,h}, \tag{9}$$

$$(p_h(0), q_h) = (p_0, q_h), \qquad (\partial_t p_h(0), q_h) = (\dot{p}_0, q_h), \qquad \forall q_h \in \tilde{V}_{p,h}, \tag{10}$$

where $\tilde{V}_{p,h}$ is constructed as $V_{p,h}$ but without the imposition of boundary conditions.

Let $\mathsf{P}(t)$ be the array of nodal unknowns of the pressure p_h. Denoting with a dot the temporal derivatives, the matrix version of Eq. (9) is

$$\frac{1}{c^2}\mathsf{M}\ddot{\mathsf{P}} + \mathsf{K}\mathsf{P} = \mathsf{F}_0,$$

where M is the standard mass matrix, K the matrix coming from the Laplacian and F_0 the resulting force vector. The initial conditions for this problem are obtained from Eq. (10).

The Galerkin formulation for the Helmholtz equation reads as follows: find $p_h \in V_{p,h}$ such that

$$-k^2(p_h, q_h) + B(p_h, q_h) = L(q_h), \qquad \forall q_h \in V_{p,h}. \tag{11}$$

It is well known that for high values of the wave number k this problem suffers from pollution errors. The general approach we propose to stabilize the problem and avoid the pollution effect is based on the VMS framework. Details can be found in [17], here we will just state the formulation. It consists of finding $p_h \in V_{p,h}$ such that

$$-k^2(p_h, q_h) + B(p_h, q_h) + \sum_K \tau(-k^2 q_h - \Delta q_h, \tilde{P}(-k^2 p_h - \Delta p_h - f_p))_K$$
$$+ \sum_E \delta \left(\left[\!\left[\frac{\partial q_h}{\partial n} \right]\!\right], \left[\!\left[\frac{\partial p_h}{\partial n} \right]\!\right] \right)_E = L(q_h), \qquad \forall q_h \in V_{p,h},$$

where $[\![\cdot]\!]$ stands for the jump over the edges and τ and δ are parameters of the formulation (see [17]). $\tilde{P}$ is a projection which can be taken as the identity (case in which we may set $\delta = 0$) or the projection orthogonal to the FE space [32,33]. In this last case, $\tilde{P}(-k^2 p_h - \Delta p_h - f_p) = \tilde{P}(-\Delta p_h - f_p) = 0$ for linear elements and f_p a FE function; it is thus clear that the stabilizing terms evaluated on the element boundaries are crucial.

Calling

$$B_s(p_h, q_h) := B(p_h, q_h) + \sum_K \tau(-k^2 q_h - \Delta q_h, \tilde{P}(-k^2 p_h - \Delta p_h))_K$$
$$+ \sum_E \delta \left(\left[\!\left[\frac{\partial q_h}{\partial n} \right]\!\right], \left[\!\left[\frac{\partial p_h}{\partial n} \right]\!\right] \right)_E$$
$$L_s(q_h) := L(q_h) + \sum_K \tau(-k^2 q_h - \Delta q_h, \tilde{P}(f_p))_K,$$

the stabilized version of Eq. (11) consists of finding $p_h \in V_{p,h}$ such that

$$-k^2(p_h, q_h) + B_s(p_h, q_h) = L_s(q_h), \qquad \forall q_h \in V_{p,h}. \tag{12}$$

The matrix structure of this problem is

$$-k^2\mathsf{M}\mathsf{P} + \mathsf{K}_s\mathsf{P} = \mathsf{F}_s, \tag{13}$$

where K_s is the matrix arising from the bilinear form $B_s(p_h, q_h)$ and F_s the forcing array arising from $L_s(q_h)$.

2.3. Time integration and fully discrete problem

In the case of the wave equation in the time domain, we need to discretize the problem in time. Our discussion is independent of the time integration scheme employed, but in the numerical examples we will use backward differences (BDF) of first order (BDF1).

Let $0 = t^0 < \cdots < t^n < \cdots < t^{n_\mathrm{f}} = T$ be a partition of the time interval $(0, T)$, for simplicity uniform and of size δt. Let $f(t)$ be a generic time dependent function and let f^n be an approximation to $f(t^n)$. The BDF1 scheme consists of approximating

$$\left.\frac{\mathrm{d}^2 f}{\mathrm{d}t^2}\right|_{t=t^n} \approx \frac{f^n - 2f^{n-1} + f^{n-2}}{\delta t^2}. \tag{14}$$

Thus, the fully discrete version of Eq. (9) is

$$\frac{1}{c^2\delta t^2}(p_h^n - 2p_h^{n-1} + p_h^{n-2}, q_h) + B(p_h^n, q_h) = L(q_h), \qquad \forall q_h \in V_{p,h}.$$

For $n = 1$, p_h^0 and p_h^{-1} can be determined from the initial conditions in Eq. (10).

The algebraic version of the problem can be written as

$$\frac{1}{c^2\delta t^2}\mathsf{M}\mathsf{P}^n + \mathsf{K}\mathsf{P}^n = \mathsf{F}_0 + \frac{2}{c^2\delta t^2}\mathsf{M}\mathsf{P}^{n-1} - \frac{1}{c^2\delta t^2}\mathsf{M}\mathsf{P}^{n-2}, \tag{15}$$

for $n = 2, \ldots, n_\mathrm{f}$.

3. Correction terms based on training: the general concept

In the previous section we have presented the numerical approximation we propose to solve the wave equation. In this section we present a general concept to correct fully discrete problems based on the knowledge of *fine* solutions, i.e., solutions corresponding to finer discretizations. The reason why we have at our disposal finer solutions is, in fact, irrelevant. It can be due to a finer time discretization, to a finer FE mesh approximation or to an interpolation of higher degree.

3.1. Coarse and fine problems

Suppose that we have a *coarse* and *linear* algebraic problem of the form:

$$\mathsf{A}\mathsf{U}_\mathrm{c} = \mathsf{F}, \tag{16}$$

with U_c, F arrays of m components, and A an $m \times m$ matrix, i.e., $\mathsf{U}_\mathrm{c}, \mathsf{F} \in \mathbb{R}^m$, $\mathsf{A} \in \mathbb{R}^{m\times m}$. Given F and matrix A, the unknown U_c could be computed by solving (16), but we intend to modify this system to enhance the accuracy of U_c. The example of interest in this work is that in which problem (16) is obtained from a coarse discretization of the wave equation, but the idea to be presented is general. In wave problems we would have:

$$\mathsf{A} = -k^2\mathsf{M} + \mathsf{K}_\mathrm{s}, \quad \mathsf{F} = \mathsf{F}_\mathrm{s}, \quad \mathsf{U}_\mathrm{c} = \mathsf{P},$$

in the case of the Helmholtz equation in Eq. (13) and

$$\mathsf{A} = \frac{2}{c^2\delta t^2}\mathsf{M} + \mathsf{K}, \quad \mathsf{F} = \mathsf{F}_0 + \frac{5}{c^2\delta t^2}\mathsf{M}\mathsf{P}^{n-1} - \frac{4}{c^2\delta t^2}\mathsf{M}\mathsf{P}^{n-2} + \frac{1}{c^2\delta t^2}\mathsf{M}\mathsf{P}^{n-3}, \quad \mathsf{U}_\mathrm{c} = \mathsf{P}^n,$$

in the case of the discrete wave equation in the time domain in Eq. (15).

Suppose that, by any means, we know a *finer* approximation to the same continuous problem, most likely obtained from a discrete problem like (16) but with dimension $M \geq m$. Different possibilities to obtain this finer solution for wave propagation problems will be discussed in the following section. Let $\mathsf{U}_\mathrm{f} \in \mathbb{R}^M$ be this finer solution (the system which it solves is in fact irrelevant), and suppose that we can construct a projection

$$\mathsf{P}_\mathrm{fc} : \mathbb{R}^M \longrightarrow \mathbb{R}^m,$$

i.e., a projection from the space where the fine solution belongs to the space where the coarse solution is defined. Again, the way to construct this projection is irrelevant, and will be discussed in the following section for the particular case of the wave equation.

The correction we will propose is based on the following fundamental assumption:

Assumption 1. The best coarse solution is $\mathsf{U}_\mathrm{c} = \mathsf{P}_\mathrm{fc}(\mathsf{U}_\mathrm{f})$.

In the case of a single linear problem, it is trivial to introduce a *correction* to (16), say $\mathsf{D}_\mathrm{ex} \in \mathbb{R}^m$, so that the solution is $\mathsf{P}_\mathrm{fc}(\mathsf{U}_\mathrm{f})$. Indeed, the solution to

$$\mathsf{A}\mathsf{U}_\mathrm{c} + \mathsf{D}_\mathrm{ex} = \mathsf{F} \quad \text{with} \quad \mathsf{D}_\mathrm{ex} = \mathsf{F} - \mathsf{A}\mathsf{P}_\mathrm{fc}(\mathsf{U}_\mathrm{f}), \tag{17}$$

is precisely $\mathsf{U}_\mathrm{c} = \mathsf{P}_\mathrm{fc}(\mathsf{U}_\mathrm{f})$, as it is trivially checked (A is assumed to be invertible, obviously).

For a single solve, such a correction is unnecessary, as we already have U_f. However, system (16) may be parameter dependent, and the fine solution may be known for some of the parameters, but not for all of them.

3.2. Configurations

Given a partial differential equation to be solved, the structure of the (coarse) discretization can be written as problem (16), but the particular expressions of matrix A and the RHS vector F depend on the data of the problem, including parameters. We will call *configuration* a particular realization of these data and parameters. Therefore, different configurations may correspond to changes in:

- Boundary conditions, which would affect both matrix A and the array F. Even though changes in the type of boundary conditions could change the dimension m of the problem, we will consider it fixed.
- Forcing terms, which would affect the array F.
- Physical properties, which again would affect both matrix A and the array F.
- Time instants, which would correspond to taking time t as a parameter of the problem. In a time evolution problem as the wave equation, problem (16) has to be understood as the problem to be solved at a certain time step, and thus different time steps can be considered different configurations.

Let us assume that we have N configurations for which we know the fine solution, $\mathsf{U}_{\mathrm{f},\alpha}$, $\alpha = 1, \dots, N$. We will use a subscript in matrix A, array F and the unknown of the coarse problem, U_c, to refer to each particular configuration. Thus, the *uncorrected* coarse problem for each configuration is

$$\mathsf{A}_\alpha \mathsf{U}_{\mathrm{c},\alpha} = \mathsf{F}_\alpha, \quad \alpha = 1, \dots, N.$$

Since these are the configurations in which we know the fine solution, $\mathsf{U}_{\mathrm{f},\alpha}$, we can introduce an *exact* correction $\mathsf{D}_{\mathrm{ex},\alpha}$, given by

$$\mathsf{D}_{\mathrm{ex},\alpha} = \mathsf{F}_\alpha - \mathsf{A}_\alpha \mathsf{P}_\mathrm{fc}(\mathsf{U}_{\mathrm{f},\alpha}), \quad \alpha = 1, \dots, N, \tag{18}$$

so that the solution to the *corrected* coarse problem

$$\mathsf{A}_\alpha \mathsf{U}_{\mathrm{c},\alpha} + \mathsf{D}_{\mathrm{ex},\alpha} = \mathsf{F}_\alpha, \tag{19}$$

is $\mathsf{U}_{\mathrm{c},\alpha} = \mathsf{P}_\mathrm{fc}(\mathsf{U}_{\mathrm{f},\alpha})$, considered optimal, for $\alpha = 1, \dots, N$.

3.3. The correcting term and main modeling assumption

As in the case of a single solve in which we know the fine solution, the corrected coarse problems (19) are unnecessary, since we know the fine solution $\mathsf{U}_{\mathrm{f},\alpha}$, $\alpha = 1, \dots, N$. We will call the set of these fine solutions *the training set*, and the associated configurations the *training configurations*.

The objective now is to introduce a correction $\mathsf{D} \in \mathbb{R}^m$ to the coarse problem (16) *for any* configuration, not only for the training configurations, so that the solution to

$$\mathsf{A}\mathsf{U}_\mathrm{c} + \mathsf{D} = \mathsf{F},$$

has improved accuracy with respect to that of problem (16). We now have to decide the structure of the correcting term D and the way it is computed. Our proposal is the following:

Assumption 2 (*Main Modeling Assumption*)**.** The correcting term D depends only on the unknown U_c, i.e., $\mathsf{D} = \mathsf{D}(\mathsf{U}_c)$, and it is designed under the condition that it is as close as possible to $\mathsf{D}_{ex,\alpha}$ in (18), $\alpha = 1, \ldots, N$, at the training configurations.

According to this assumption, the problem to be solved is of the form

$$\mathsf{A}\mathsf{U}_c + \mathsf{D}(\mathsf{U}_c) = \mathsf{F}. \tag{20}$$

What remains open is how to construct $\mathsf{D}(\mathsf{U}_c)$ to make it close to $\mathsf{D}_{ex,\alpha}$, $\alpha = 1, \ldots, N$, at the training configurations. In the following we describe two possibilities, namely, the case is which $\mathsf{D}(\mathsf{U}_c)$ is a linear function and that in which $\mathsf{D}(\mathsf{U}_c)$ is an ANN.

We do not intend to analyze quantitatively the improvement obtained by the introduction of $\mathsf{D}(\mathsf{U}_c)$, i.e., the reduction of the error with respect to the case $\mathsf{D}(\mathsf{U}_c) = \mathsf{0}$. In the FE context, we cannot expect a higher rate of convergence, as the FE space is not changed, but just a decrease in the constant that multiplies the power of the mesh size on which the error depends. We expect the improvement to be higher the richer the training set and the better $\mathsf{D}(\mathsf{U}_c)$ fits the training configurations.

3.4. Linear correction term: a least-squares approach

The simplest model for $\mathsf{D} = \mathsf{D}(\mathsf{U}_c)$ is a linear one:

$$\mathsf{D}(\mathsf{U}_c) = \mathsf{A}_{ls}\mathsf{U}_c + \mathsf{F}_{ls}, \tag{21}$$

where A_{ls} is an $m \times m$ matrix and F_{ls} an array of m components, both to be determined. The motivation for such a simple expression of the correction term can be found in stabilized FE methods. Indeed, *the matrix structure of the stabilizing terms is precisely* (21), derived from different heuristic arguments and with matrix A_{ls} and array F_{ls} possibly mesh-dependent.

Here we take as starting point expression (21) and design A_{ls} and F_{ls} according to Assumption 2. A possible way to fulfill it is *to impose that* $\mathsf{D}(\mathsf{P}_{fc}(\mathsf{U}_{f,\alpha}))$ *be as close as possible to* $\mathsf{D}_{ex,\alpha}$, $\alpha = 1, \ldots, N$, *in a least-square (LS) sense*. Thus, we obtain A_{ls} and F_{ls} by solving the optimization problem:

$$[\mathsf{A}_{ls}, \mathsf{F}_{ls}] = \arg \min_{\mathsf{A}_d, \mathsf{F}_d} \sum_{\alpha=1}^{N} \|\mathsf{D}_{ex,\alpha} - (\mathsf{A}_d \mathsf{P}_{fc}(\mathsf{U}_{f,\alpha}) + \mathsf{F}_d)\|^2, \tag{22}$$

where the norm involved in this expression can be taken as the standard Euclidian one, possibly scaled if the components of F have different units in the particular example of application.

As it will be shown in the numerical examples, the simple approach given by (21)–(22) is effective and the accuracy of the solution U_c to the resulting problem (20) is higher than that of problem (16). However, it suffers from two inconveniences: first, the limited scope of (21) and, secondly, the typical instabilities of the LS approximation (22). The latter are manifested for example by the fact that very different solutions $[\mathsf{A}_{ls}, \mathsf{F}_{ls}]$ can be obtained by introducing small changes in the training set, such as the introduction of a new element in this set.

3.5. Correcting term obtained from an artificial neural network

A much more general approach than using (21)–(22) is to use an ANN to model $\mathsf{D}(\mathsf{U}_c)$. In this work we consider ANNs simply as *best fitting functions* to a set of data. In particular, the ANN we wish to construct aims to fit the inputs $\{\mathsf{P}_{fc}(\mathsf{U}_{f,\alpha})\}_{\alpha=1}^N$ and the outputs $\{\mathsf{D}_{ex,\alpha}\}_{\alpha=1}^N$ given by Eq. (18). Thus, the ANN to be constructed will be a highly nonlinear mapping

$$\mathsf{D} : \mathbb{R}^m \longrightarrow \mathbb{R}^m,$$

trained with the pairs $\{\mathsf{P}_{fc}(\mathsf{U}_{f,\alpha}), \mathsf{D}_{ex,\alpha}\}_{\alpha=1}^N$.

The use of ANNs clearly solves the two deficiencies mentioned for the linear approach with a LS fitting. First, ANNs are highly flexible, allowing to construct in an efficient way complex nonlinear functions. And, second, new terms can be added to the training set with little computational effort and in a very stable manner. Moreover, the correction term constructed from ANNs can be introduced both to linear *and nonlinear* problems.

We will not analyze here the particular implementation of the ANN we use, and we defer its description to the numerical examples. Let us just say that we have found sufficient to use simple topologies (architectures) of the network, with only a few hidden layers (shallow networks) and a few neurons per layer. For the moment, we have only considered feedforward architectures with back-propagation algorithms to determine the weights and the biases, and with the sigmoid as activation function. However, the flexibility of ANNs would allow for a lot of possibilities in the context of correcting coarse models that we are describing.

Let us describe how to apply ANNs to our problem:

- We design $\mathsf{D}(\mathsf{U}_\mathrm{c})$ *component-wise*. This means that in fact we construct m fitting functions corresponding to the components of D, D_j, $j = 1, \dots, m$. These are the *outputs* of the ANN.
- The input of the ANN is in principle the set $\{\mathsf{P}_\mathrm{fc}(\mathsf{U}_{\mathrm{f},\alpha})\}_{\alpha=1}^{N}$. However, a particular implementation is possible in our context, since each $\mathsf{P}_\mathrm{fc}(\mathsf{U}_{\mathrm{f},\alpha})$ can be considered as the vector of nodal values of a certain function $u_{\mathrm{c},\alpha}(\boldsymbol{x})$ defined on the coarse mesh (scalar or vectorial). Thus, it makes perfect sense to use only a few components of $\mathsf{P}_\mathrm{fc}(\mathsf{U}_{\mathrm{f},\alpha})$ when trying to obtain the jth component of D, for example those associated to nodes linked to the node with the jth degree of freedom. Moreover, a way to introduce the effect of the neighboring nodes to that node can be through derivatives of $u_{\mathrm{c},\alpha}(\boldsymbol{x})$. Thus, considering for simplicity the case in which $u_{\mathrm{c},\alpha}(\boldsymbol{x})$ is a scalar field, a possible input to compute D_j can be $u_{\mathrm{c},\alpha}(\boldsymbol{x}_j)$ and $\nabla u_{\mathrm{c},\alpha}(\boldsymbol{x}_j)$, where $\boldsymbol{x}_j$ are the coordinates of the jth node, $j = 1, \dots, m$, $\alpha = 1, \dots, N$.
- The weights and biases of the ANN are determined by minimizing a certain loss function. According to Assumption 2, we take as loss function

$$\mathcal{C} = \sum_{\alpha=1}^{N} \|\mathsf{D}_{\mathrm{ex},\alpha} - \mathsf{D}(\mathsf{P}_\mathrm{fc}(\mathsf{U}_{\mathrm{f},\alpha}))\|^2. \tag{23}$$

  This means that *the ANN is trained with the configurations for which the exact correction is known*. As in the case of the LS fitting for linear corrections, the norm in this expression can be scaled when the components of D have different physical meaning.

Let us stress again that the construction of $\mathsf{D}(\mathsf{U}_\mathrm{c})$ only requires the fine solutions, but not the problem they solve. In fact, these could be obtained by means not necessarily numerical (such as experimental, analytical or semi-analytical). Our focus, however, is the case in which the training configurations are solved numerically with an approximation finer than the coarse one.

Once we have constructed the correction term $\mathsf{D}(\mathsf{U}_\mathrm{c})$, the problem to be solved is (20). While for the linear approximation with a LS fitting it reduces to the linear problem

$$(\mathsf{A} + \mathsf{A}_\mathrm{ls})\mathsf{U}_\mathrm{c} = \mathsf{F} - \mathsf{F}_\mathrm{ls},$$

for ANNs problem (20) is highly non-linear. Let D_ann be the correction term in this case. Standard linearization techniques can be used to linearize it. In our examples we have found efficient the simplest iterative procedure

$$\mathsf{A}\mathsf{U}_\mathrm{c}{}^{(k)} = \mathsf{F} - \mathsf{D}_\mathrm{ann}(\mathsf{U}_\mathrm{c}{}^{(k-1)}), \tag{24}$$

where the superscript denotes the iteration counter. In the case in which the problem corresponds to the system to be solved at a certain time step, the obvious initialization is to take $\mathsf{U}_\mathrm{c}{}^{(0)}$ as the value of U_c at the previous time step. This often suffices to improve the accuracy of the coarse solution and no iterations are needed, although it implies an explicit treatment of the correction term that could introduce instability problems in time.

The alternative to the simplest iteration (24) is a Newton–Raphson scheme:

$$(\mathsf{A} + \mathsf{J}(\mathsf{U}_\mathrm{c}{}^{(k-1)}))\mathsf{U}_\mathrm{c}{}^{(k)} = \mathsf{F} - \mathsf{D}_\mathrm{ann}(\mathsf{U}_\mathrm{c}{}^{(k-1)}) + \mathsf{J}(\mathsf{U}_\mathrm{c}{}^{(k-1)})\mathsf{U}_\mathrm{c}{}^{(k-1)}, \tag{25}$$

where

$$\mathsf{J}(\mathsf{U}_\mathrm{c}) = \frac{\partial \mathsf{D}_\mathrm{ann}(\mathsf{U}_\mathrm{c})}{\partial \mathsf{U}_\mathrm{c}},$$

is the Jacobian of the ANN. This implies computing derivatives of the ANN with respect to the inputs, a feasible alternative that we have not explored.

4. Discrete wave equation with correction terms

This section intends to be the combination of the two previous sections; the general concept of the correcting terms based on training explained in Section 3 will be applied to the wave problems that have been explained in Section 2.

4.1. Fine solutions and definition of the projection P_{fc}

As it has been explained in the previous section, a reference solution has to be provided in order to train the correction term for the coarse case. This solution is supposed to be more accurate than the coarse one, so it has to be supplied by a more accurate version of the same problem. In this case, this improvement on accuracy is produced by using a finer space discretization or a finer time discretization, in other words, with a smaller mesh size or a smaller time step. In the first case we would have $M > m$, whereas in the second $M = m$. The possibility of taking $M < m$ will not be considered; it could correspond to a solution obtained on a mesh coarser than that of U_{c} but with a smaller time step.

Once the coarse and the fine cases are created, the connection between them is crucial to perform the best possible training for the correction term. The two most important aspects at this point are what data we want to use from the fine case and how we project it to the coarse case.

On the one hand, three possible entry parameters for the correction term model could be the solution of the problem and its space derivatives, just the derivatives or just the solution. Physical parameters or other derivatives of the solution of the problem could also be used to improve the training. It would also be possible to just enter the solution and make the model calculate the space derivatives from it; however, we rather prefer to facilitate the construction of the model by calculating it separately.

On the other hand, how we project these data to the coarse case will depend on the type of the fine case. If the fine case corresponds to a finer time discretization, but the mesh used for the spatial discretization is the same, then P_{fc} is simply the identity, i.e., $\mathsf{P}_{\mathrm{fc}}(\mathsf{U}_{\mathrm{f}}) = \mathsf{U}_{\mathrm{f}}$. This projection needs to be performed at the time levels in which the fine solution is available and we wish to include in the training to solve the optimization problem in Eq. (22) in the LS approach or in Eq. (23) when ANNs are used. However, these problems require the system matrix A and the RHS vector F to be known at these time levels. For linear problems, A is constant, but F changes with the time level n. If the time step of the coarse model is not a multiple of the time step of the fine model, interpolation *in time* is required, for example interpolating the fine solution at the time levels in which the coarse solution is to be computed. In this case it has to be remarked that there are many options to perform the training, in particular it can be performed in certain time windows. For example, we can train the model with the fine solution during a certain number of time steps at the beginning of the simulation and then run the coarse model for longer time periods with the correction determined by this training. This is what we do in two of the numerical examples.

Let us consider now the situation in which the fine case corresponds to a finer space discretization. If the finer mesh is obtained from a hierarchical refinement of the coarse mesh, $\mathsf{P}_{\mathrm{fc}}(\mathsf{U}_{\mathrm{f}})$ can be taken simply as the restriction of the fine solution to the nodes of the coarse mesh that also belong to the fine mesh. In fact, this would be a particular case of projection on general meshes, namely, the *interpolation* of the fine solution to the coarse mesh. To do this, the position of each node of the coarse mesh on the fine one needs to be determined, and once this is done the value of the fine solution on the node of the coarse mesh can be calculated. This requires a standard search algorithm to obtain the element of the fine mesh to which a node of the coarse one belongs. Let us call I_{fc} the resulting interpolation.

The same search algorithm described above is required for another option to choose $\mathsf{P}_{\mathrm{fc}}(\mathsf{U}_{\mathrm{f}})$, namely, the L^2 projection from the fine mesh to the coarse one. Considering the unknown to be the acoustic pressure for conciseness, if p_{f} is the solution on the fine mesh and $P_h(p_{\mathrm{f}})$ its L^2 projection, it is obtained by solving

$$(q_h, P_h(p_{\mathrm{f}})) = (q_h, p_{\mathrm{f}}) \qquad \forall q_h \in \tilde{V}_{p,h}.$$

Let $\mathsf{P}_{2,\mathrm{fc}}$ be the L^2 projection at the algebraic level. The two options described are

$$\mathsf{P}_{\mathrm{fc}} = \begin{cases} \mathsf{I}_{\mathrm{fc}} & \text{in the case of standard interpolation} \\ \mathsf{P}_{2,\mathrm{fc}} & \text{in the case of the } L^2 \text{ projection} \end{cases}$$

Both options can be modified to incorporate restrictions of the projected solution [34].

The projection for the spatial derivatives of the solution of the problem can be done in the same way for both situations (finer solutions corresponding to finer time discretizations or to finer space discretizations). Physical parameters of the problem do not have to be projected, they can be directly sent from the fine case to the coarse one.

4.2. Fully discrete system

At this point, it is straightforward to obtain the fully discrete form of the problem. We just need to combine the matrix equations of the previous sections with the correction term of Eq. (21) in the case of the LS approach, or with the procedure of Eq. (24) in the case of ANNs.

Thus, the fully discrete form of the irreducible wave equation in the time domain for the LS training system is

$$\frac{2}{\delta t^2}\mathsf{M}\mathsf{U}^n + \mathsf{K}\mathsf{U}^n + \mathsf{A}_{\mathrm{ls}}\mathsf{U}^n = \mathsf{F} - \mathsf{F}_{\mathrm{ls}} + \frac{5}{\delta t^2}\mathsf{M}\mathsf{U}^{n-1} - \frac{4}{\delta t^2}\mathsf{M}\mathsf{U}^{n-2} + \frac{1}{\delta t^2}\mathsf{M}\mathsf{U}^{n-3}, \tag{26}$$

and the fully discrete form for the ANN training system, using an explicit treatment of the correction, is

$$\frac{2}{\delta t^2}\mathsf{M}\mathsf{U}^n + \mathsf{K}\mathsf{U}^n = \mathsf{F} + \frac{5}{\delta t^2}\mathsf{M}\mathsf{U}^{n-1} - \frac{4}{\delta t^2}\mathsf{M}\mathsf{U}^{n-2} + \frac{1}{\delta t^2}\mathsf{M}\mathsf{U}^{n-3} - \mathsf{D}_{\mathrm{ann}}(\mathsf{U}^{n-1}). \tag{27}$$

Two remarks are in order:

- In Eq. (26) the correction term depends only on U^n and in Eq. (27) only on U^{n-1}, in this case due to the explicit treatment of the correction. It could be argued that this correction could depend also on *previous* values of the unknown. The fact that we do not consider this dependence in our model can be understood as a sort of orthogonality of the correction term with respect to the unknown. In fact, this type of correction is what is encountered in stabilized FE methods based on the VMS concept using orthogonal sub-grid scales (see [31] for a review on the topic).
- Note that in Eq. (27) no iterations have been used in this approximation. This is possible because the equation belongs to the time domain and the previous time step can play the role that used to play the previous iteration. An implicit treatment of the correction term would require a linearization of $\mathsf{D}_{\mathrm{ann}}(\mathsf{U}^n)$ as explained in the previous section.

We can proceed similarly in the case of the equations in the frequency domain. The corrected version of Eq. (13) using the LS approach is

$$-k^2\mathsf{M}\mathsf{P} + (\mathsf{K}_{\mathrm{s}} + \mathsf{A}_{\mathrm{ls}})\mathsf{P} = \mathsf{F}_{\mathrm{s}} - \mathsf{F}_{\mathrm{ls}}, \tag{28}$$

whereas the corrected version using ANNs and a simple fix point iteration is

$$-k^2\mathsf{M}\mathsf{P}^{(i)} + \mathsf{K}_{\mathrm{s}}\mathsf{P}^{(i)} = \mathsf{F}_{\mathrm{s}} - \mathsf{D}_{\mathrm{ann}}(\mathsf{P}^{(i-1)}), \tag{29}$$

where i corresponds now to the iteration counter.

At this point it could be argued that the correction term plays a role similar to that of the stabilizing terms arising from the VMS formulation, and the interaction between both needs being explained. This is particularly obvious in the case of the LS approach, in which the stabilizing and the correction terms have the same structure. However, let us point out that the stabilization terms are designed mainly for *stability reasons*, without any information about finer solutions, whereas the new correcting terms are precisely introduced *to improve accuracy* from the knowledge of accurate solutions. Thus, both terms pay different roles. Furthermore, the VMS-based stabilizing terms have no significant computational cost, whereas the correcting terms have the cost of training.

4.3. Practical implementation

Let us explain now how our algorithm specifically works. First of all, it is important to differentiate the two main phases of the algorithm: the training phase and the execution phase. It is obvious that before starting the execution phase, the training one has to be completed at least for one case, so let us start with the training phase.

Once the coarse and fine training cases are ready, the training model is selected and the data that we want to use from the fine problem is defined; we then launch the coarse version of the problem with one processor and

the fine version with another parallel processor. At this point, depending on which kind of training we want, the communication between the processors is different. On the one hand, in the situation of the finer mesh discretization and in the case of nested meshes, the processor that is launching the fine case only sends the desired information on the coincident nodes but at every single time step; in the case of non-nested meshes, non-trivial projections are required, as explained earlier. On the other hand, in the situation of the finer time discretization, the processor that is launching the fine case only sends the desired information at the coincident times, in other words, at the times when the coarse case solves the problem, so the coarse case has to wait for the fine case to do its extra time steps; again, in the case in which time levels never coincide, interpolation in time is required. Obviously, in this second case the data is sent to every node, assuming the mesh of the two cases to be exactly the same. When the processor of the coarse case receives the data, it is entered to the model. Depending on whether it is the LS or the ANN model, the information is treated in a different way; however, the objective of the model is the same in both situations. While this process is underway, the model creates a new file and writes on it its specific correction parameters. Once the two cases are finished this new file is saved, this is where the information that will be used in the execution phase is stored.

Now that the training phase is finished, it is time to prepare the execution case. The possible changes in the execution cases are called configurations and have been explained in Section 3. When this new case is launched, the algorithm acts in a completely different way. Firstly, the data that has been written in the previous phase is read, so the model has all the information that needs to improve the solution of the problem. Secondly, the first time step or iteration of the problem is solved without any correction; nevertheless, this solution will be used to make the correction that is used to improve the solution of the second time step or iteration. Finally, each solution is used to calculate the correction of the next one until the problem ends. An important point is that if we are dealing with a problem that is unstable in its first time steps, we can add a delay so that the correction starts at the time we choose.

Additionally, there is something that can be significant in the performance of the correction algorithm, which is to use the data of the previous time step or iteration in the training phase instead of the data of the same time step or iteration. This is pretty obvious if we take into account that in the execution phase we are forced to use the data of the previous time step or iteration, so doing this we would be training the model in the same way as we use it.

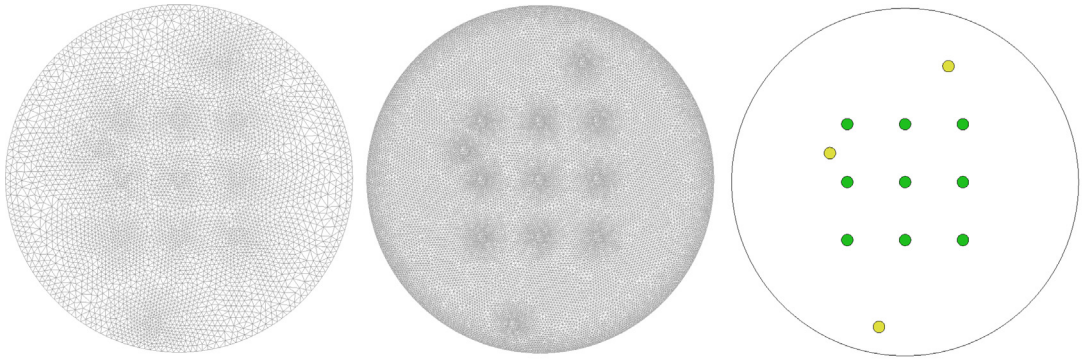
5. Numerical examples

In this section, some numerical examples that illustrate the improvement of wave problem results with correcting terms based on training with ANN and LS models are shown. These specific examples have been solved with the irreducible form of the wave equation, the first two in the frequency domain and the other two in the time domain. Moreover, the two corresponding to the time domain have been improved using a finer time discretization, and the two corresponding to the frequency domain have been improved using a finer space discretization. However, other combinations as using mixed forms or improving time domain problems with finer space discretization could be considered. In all the examples we have used linear triangular elements for the spatial discretization, and the BDF1 scheme in the case of transient problems. When using ANNs, the sigmoid has been employed as activation function.

One of the typical problems when using ANNs is the under and over-fitting. This has to be avoided by choosing appropriately the parameters of the ANN in each case. Taking into account that our models and equations are not very complex, we have to be consequent with the chosen number of neurons, as if it were too large over-fitting could be encountered. On the contrary, if the training set were too small (for example, less than a period in a wave problem) the model could not give us good results because of under-fitting. Another possible problem in using ANNs in the coarse time training cases is the fine time step that can be used. Depending on the complexity of the problem, the difference in the time steps of the fine and coarse models cannot be too large; this is explained in the last numerical example.

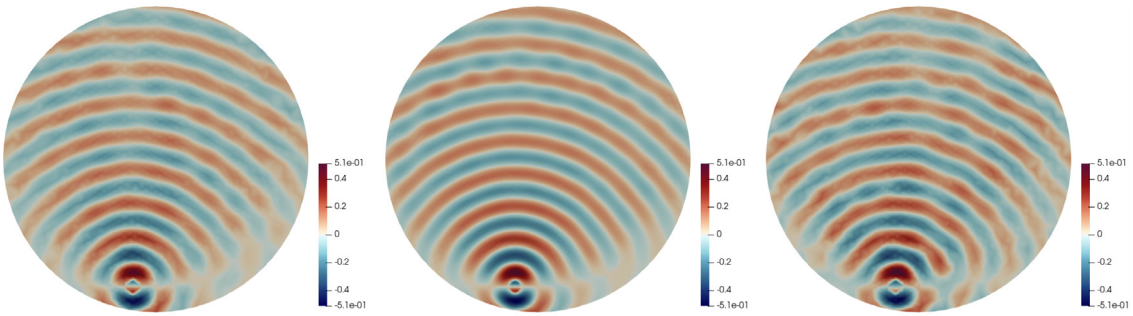
5.1. Dipole localization coarse mesh training

This first example consists in training a model to give a better solution for whatever is the position of the source in a circular domain that has a radius of $R = 300$ m. More precisely, the source is a circular dipole of radius $r = 10$ m. In this case, we use a simple ANN model of 2 layers and 3 neurons each, with a learning momentum of 0.1, the number of maximum epochs is set to 1000 and the maximum tolerated error as 1. This problem has



(a) Mesh with $h = 10$ m. (b) Mesh with $h = 5$ m. (c) Trained dipoles (green) and used dipoles (yellow).

Fig. 1. Spatial discretization and source distribution of the numerical example 5.1.. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Coarse. (b) Fine. (c) Corrected with ANN.

Fig. 2. Imaginary part of the acoustic pressure for the first of the non-trained dipoles of the numerical example 5.1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

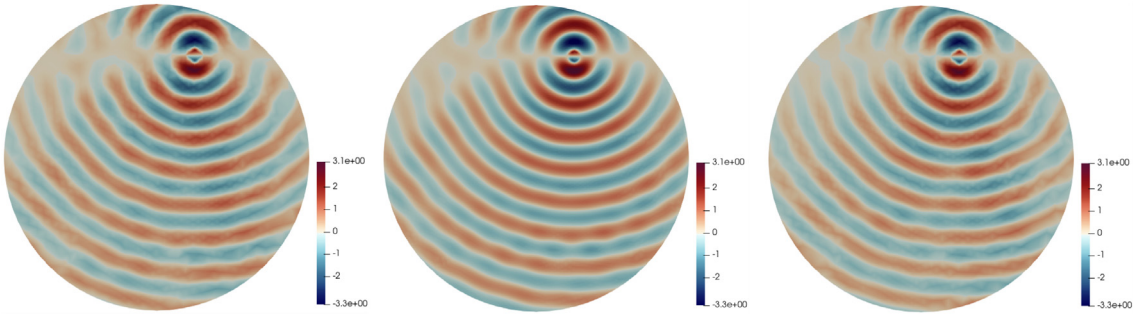
been solved with the irreducible wave equation in the frequency domain. NRBCs of Sommerfeld's type, the same as in the following examples, have been used in order to avoid reflected waves. The aim of this example is to show how our model approaches a situation where the geometry of the problem is changed between the training and the execution phases.

The training set is a grid of dipoles launched one by one in the training phase. The coarse and the fine meshes and their element sizes can be seen in Fig. 1(a) and (b), respectively. The element size is smaller at the border of the dipoles in both meshes so as to have better results. The specific position of the training set dipoles and the external ones that will be used to test the model can be seen in Fig. 1(c).

It is easy to see that the results of the trained cases are more similar to the results of the finer cases than to the coarse cases. For example, if we look at the top of the circular domain of Fig. 2, we can see that the fine and the trained cases have a clear last red stripe, while in the coarse case this red stripe is almost imperceptible. In Figs. 3 and 4, similar qualitative differences can be found.

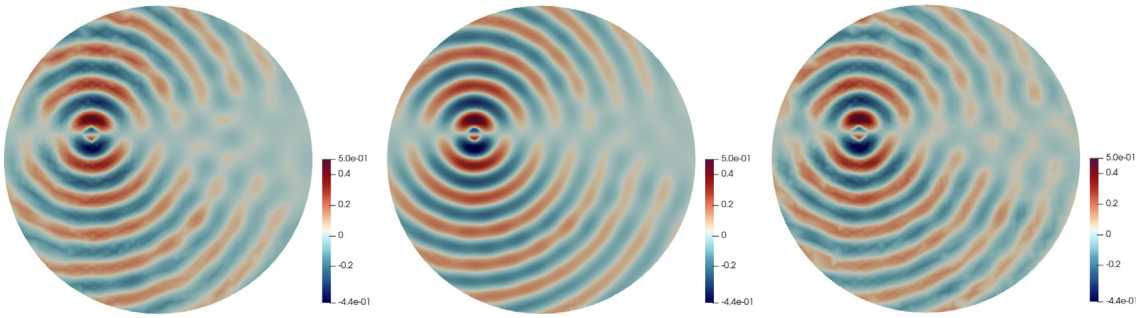
5.2. Plane wave propagation coarse mesh training

The second example consists in training a model to give a better solution for a determined range of frequencies. We will consider a freely propagating two-dimensional plane wave, the same example that has been used in [17]. The domain is a square with side $s = 1$ m. The element size is $h = 0.05$ m for the coarse mesh and $h = 0.025$ m for the fine mesh. This problem has been solved with the irreducible wave equation in the frequency domain and



(a) Coarse. (b) Fine. (c) Corrected with ANN.

Fig. 3. Imaginary part of the acoustic pressure for the second of the non-trained dipoles of the numerical example 5.1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



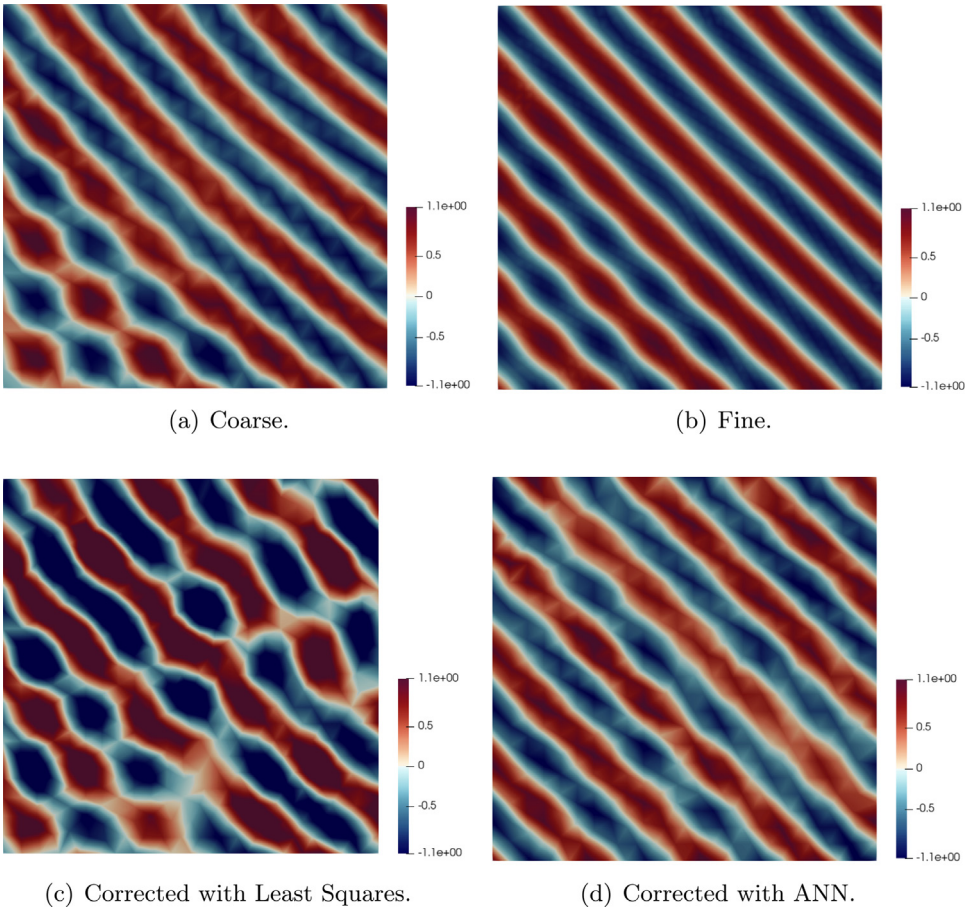
(a) Coarse. (b) Fine. (c) Corrected with ANN.

Fig. 4. Imaginary part of the acoustic pressure for the third of the non-trained dipoles of the numerical example 5.1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

without stabilization, i.e., using the standard Galerkin method, to explore the possibility of the correction to alleviate the pollution error. The most important physical parameter of the equation that we solve in this example is the dominant frequency, which is the one we want to train and change. The aim of this example is to show how our model approaches a situation where a physical parameter of the problem is changed between the training and the execution phases.

The training set is a range of frequencies that goes from $\omega = 10$ s^{-1} to $\omega = 42$ s^{-1} with intervals of 4 s^{-1}. This time, the LS method and an ANN model of 3 layers and 3 neurons are compared with the same training conditions. The latter also has a learning momentum of 0.5, its maximum number of epochs is 1000 and the maximum tolerated error is 1. For the purpose of testing this model, the execution phase is launched with a frequency of $\omega = 32$ s^{-1}, which does not appear in the training set. If the frequency chosen in the execution phase were outside the interval of frequencies of the training set, the results would not be as good as they are.

In Fig. 5(a), it can be seen that the coarse case has stability problems, the stripes are not perfectly straight and there is some pollution down left the domain. On the contrary, in the fine case these problems do not appear due to the smaller element size, as it can be seen in Fig. 5(b). On the one hand, as we can see in Fig. 5(c), the solution of the model corrected using LS is not improved but worsened; this is because the lack of adaptability that this method has; it is clear that the correction term here is not working. On the other hand, regarding the solution corrected with the ANN model, that we can see in Fig. 5(d), the result is the opposite. It is true that it can seem that the solution is better in the lower-left region but worse in the rest of the domain; however, we have to take into account that this training corrects more than just the pollution error. The curvature of the stripes of the coarse case is clearly higher than the curvature of the stripes of the corrected case, that is almost imperceptible. Moreover, there is one extra stripe in the trained case, that can also be seen in the fine one. So we can state that the solution in the corrected case is better than the coarse one in all the domain. Finally, it is important to note one of the limitations of the



(a) Coarse. (b) Fine.

(c) Corrected with Least Squares. (d) Corrected with ANN.

Fig. 5. Real part of the acoustic pressure of the numerical example 5.2 for $\omega = 32$ s^{-1}. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

ANN model that we have observed in this case, which is the choice of the dominant frequency of the corrected case. Even though we have proved that we can launch frequencies that do not belong directly to the training set, we cannot launch frequencies that are outside the interval between the smallest and the highest frequencies in the training set. This behavior is also found in ROMs without correcting terms run with physical parameters different from those of the FOM.

5.3. Dipole coarse time training in the time domain

This third example consists in training a model to give a better solution with any time step inside a time interval. The domain here is exactly the same that we have used in the first numerical example, the source is also the same and has the same size but this time it is fixed and positioned at the center of the circle. A triangular mesh of element size $h = 5$ m around the source and progressively increased up to $h = 20$ m at the external borders has been used. NRBCs have been used in order to avoid reflected waves. The training set is composed by the solution in all the time steps between $t = 400$ s and $t = 700$ s, the finer time step for the whole training being fixed at $\delta t = 0.1$ s. On the contrary, the time step of the coarse case is not fixed, so in the same run we can use different time steps for various time intervals. Specifically, a coarse time step of $\delta t = 1$ s from $t = 400$ s to $t = 500$ s has been used, $\delta t = 2$ s from $t = 500$ s to $t = 600$ s and $\delta t = 4$ s from $t = 600$ s to $t = 700$ s have been employed. This problem has been solved with the irreducible wave equation in the time domain.

To show that not only ANNs can correct and improve the solutions of wave problems, in this example we have also used a LS model. The parameters of the ANN are 3 layers, 7 neurons per layer, 0.1 of learning momentum,

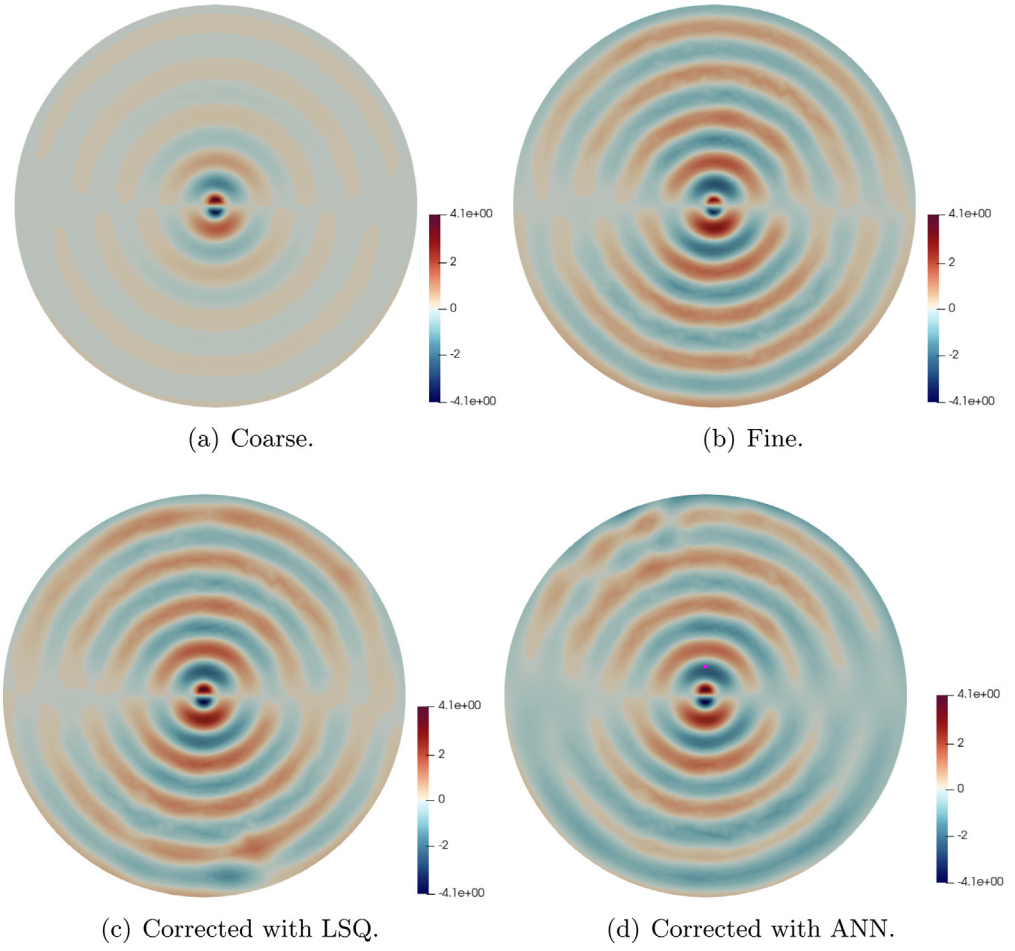


Fig. 6. Acoustic pressures of the numerical example 5.3 at $t = 891$ s. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

1000 maximum epochs and the tolerated error has been set to 1. For the purpose of testing this model, the execution phase is launched with a fixed time step of $\delta t = 3$ s, a time step that has not been used in the training phase.

On the one hand, a big qualitative improvement of the solution can be seen in Fig. 6, which shows the solution of the problem for a concrete time step. Even though the stripes are perfectly located in the four situations, the amplitudes of the waves are clearly better in the fine and corrected cases. On the other hand, in order to make a quantitative analysis, in Fig. 7 the time evolution of the solution of the problem at one specific point is plotted for the four cases. It can be seen that from $t = 0$ s to $t = 400$ s, the corrected cases and the coarse case are exactly equal; this is because the correcting term is activated at $t = 400$ s. At this point, the corrected cases starts to change, approaching the behavior of the fine case almost perfectly. In spite of the fact that the training set ends at $t = 700$ s, the periodicity of the solution allows us to extend the time of the simulation as far as we want and get the same improvement.

This example allows us to compare the two different types of corrections in Fig. 6(c) and (d). Regarding the qualitative analysis, it can be seen that even though both solutions give a clear improvement with respect to the coarse case, the LS is a bit more stable. This is because when the LS method is able to produce a corrected solution, this is very stable; however, it is not something that always happens, as it has been observed in the previous numerical Section 5.2. Regarding the quantitative analysis of the selected point, the same happens. Both corrections give almost the same behavior as the fine case, but the ANN oscillates a bit more. We consider that this slightly smaller

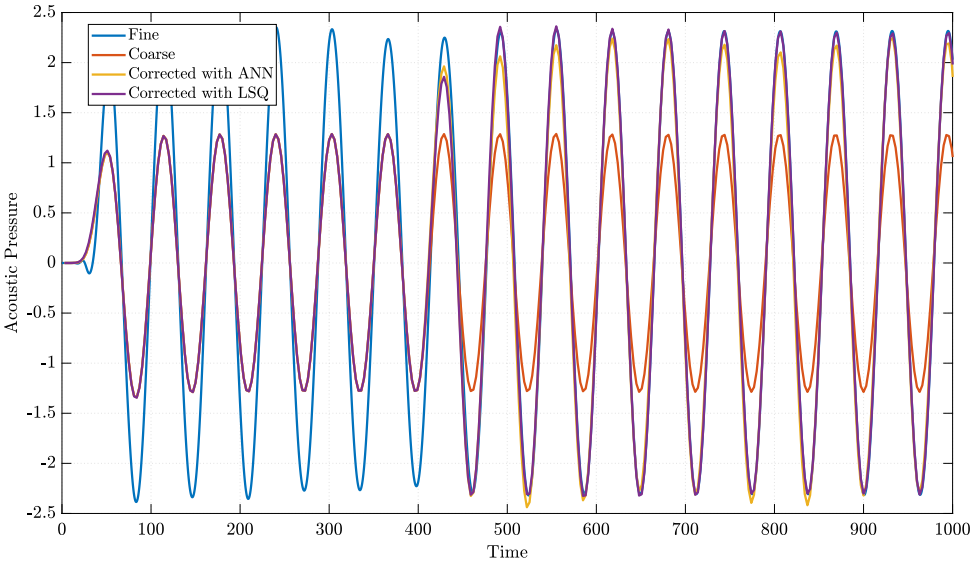


Fig. 7. Acoustic pressures evolution on time of the four cases of the numerical example 5.3 at point marked in Fig. 6(d). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

accuracy of the ANN approach is compensated by its adaptability, as the LS is not always capable of adapting to the given problem.

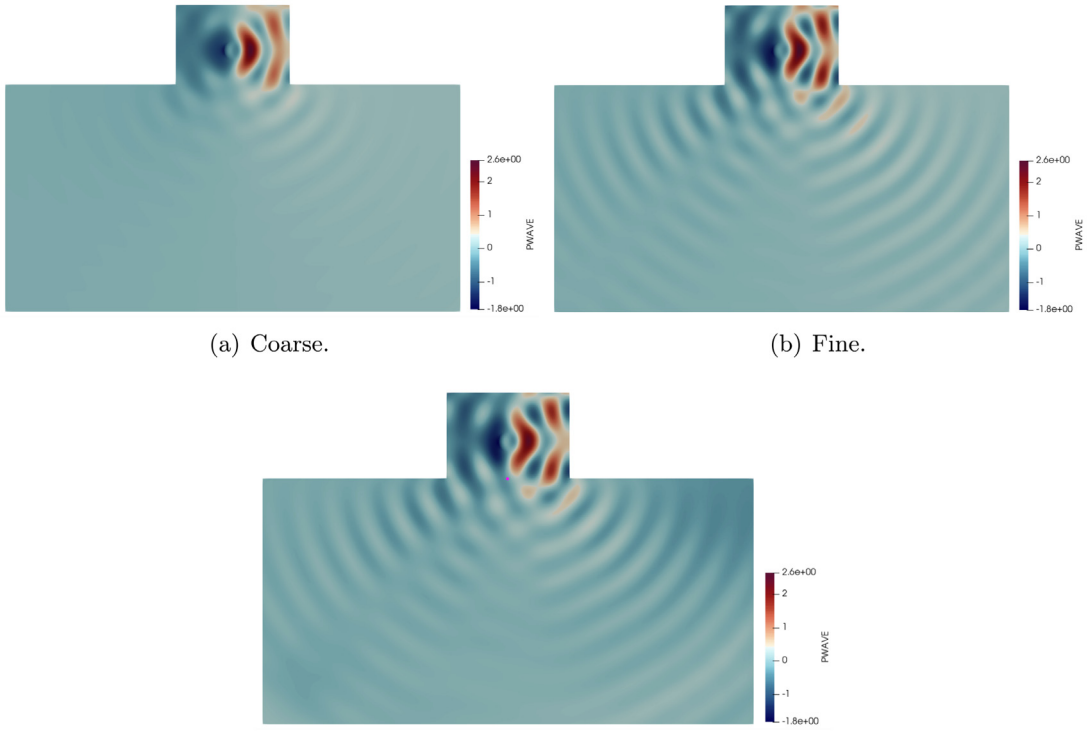
We have used this example to explain and compare the computational cost of each phase of the training in both methods, LS and ANN. Considering that the fine case corresponds to the 100% of the computational time, the coarse case corresponds to the 3.3%. Then, the cost of the training phases are 105.5% and 123.4% for the LS and the ANN approaches, respectively. Finally, the execution phases are 3.5% and 3.6% for the LS and the ANN methods, respectively. In the fist place, we can say that the ANN training is more demanding than the LS one, what makes sense knowing the mechanics of both. In the second place, we can state that there is almost no extra cost in execution cases with respect to the coarse one, so once the model is trained, it gives a significant improvement with a really low cost. In the last place, we consider that the extra cost added to the training phase is worth if the created model is strong enough to adapt to changes in the simulations, which is our case.

5.4. Dipole coarse time step training in a cavity

This example is similar to the previous one, the kind of model that we want to create and the used equation are the same, so the training is performed in the same way. The source is also a circular dipole with radius $r = 4$ m. A triangular mesh of element length $h = 1$ m has been used. However, this time the domain is different and we will create the model using an ANN with 3 layers, 7 neurons per layer, 0.7 of learning momentum, 1000 maximum epochs and tolerated error 1. The domain in this case is a cavity composed by a big rectangle of length $L = 200$ m and width $W = 100$ m and a small rectangle of length $l = 50$ m and width $w = 35$ m centered at the top of the big one (see Fig. 8). NRBCs have only been used on the borders of the big rectangle, and therefore there are reflecting waves inside the cavity, adding complexity to the problem.

The training set is composed by the solution in all the time steps between times $t = 200$ s and $t = 400$ s; the finer time step for the whole training is fixed and it is $\delta t = 0.1$ s. As it has been done in the previous example, we use different time steps in the coarse case. Concretely, a coarse time step of $\delta t = 0.2$ s has been used from $t = 200$ s to $t = 240$ s, and $\delta t = 0.4$ s from $t = 240$ s to $t = 300$ s. For the purpose of testing the model, the execution phase is launched with a fixed time step of $\delta t = 0.3$ s.

The results of this example are similar to the results of the previous one. On the one hand, a big qualitative improvement of the solution of the trained coarse model can be seen in Fig. 8(c); the waves that are far from the source are almost imperceptible in the coarse (a) case but perfectly defined in the fine (b) and trained (b) cases.



(a) Coarse.

(b) Fine.

(c) Corrected with ANN.

Fig. 8. Acoustic pressures of the numerical example 5.4 at $t = 400$ s. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

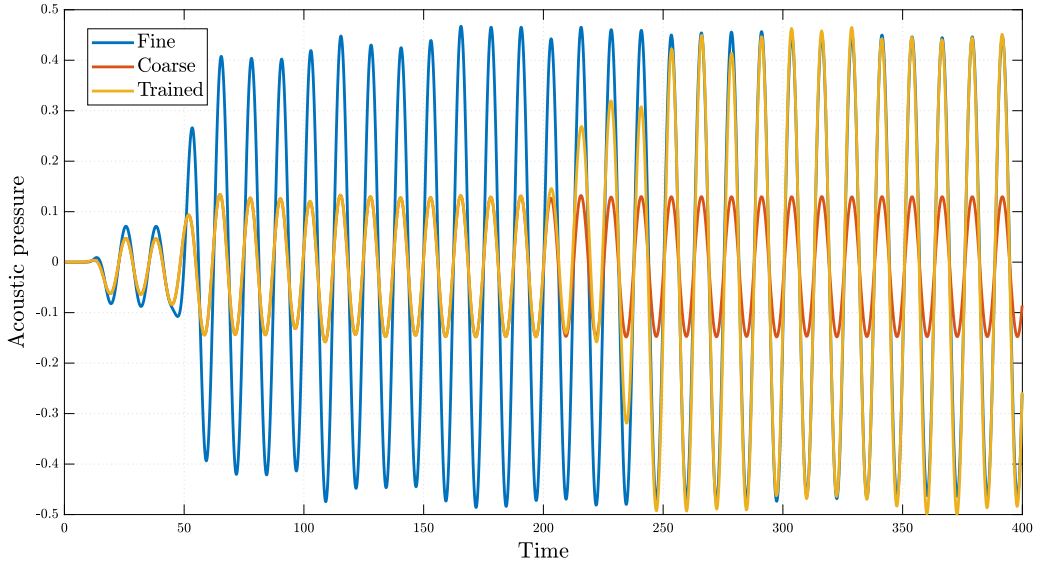
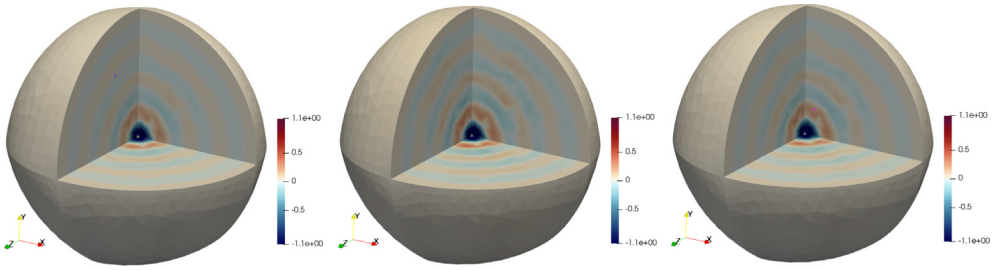


Fig. 9. Acoustic pressures evolution on time of the three cases of the numerical example 5.4 at point (0,50) m, marked in Fig. 8 c. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

On the other hand, in order to make a quantitative evaluation of the problem, the time evolution of the acoustic pressure at a specific point can be seen in Fig. 9. Although the point is really close to the source, it can be seen



(a) Coarse. (b) Fine. (c) Corrected with ANN.

Fig. 10. Acoustic pressures of the numerical example 5.5 at $t = 1000$ s. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

that even there the improvement in the trained case is enormous. As in the previous example, the correcting term is activated at time $t = 200$ s; at this point the amplitude of the oscillations starts to increase until it reaches the amplitude of the fine case, making the correction almost perfect.

5.5. 3D sphere coarse time training of a centered dipole

This fifth example is actually the 3D adaptation of the numerical Section 5.3. The domain consists of two spheres centered at the origin of coordinates, with radiuses of $R = 300$ m and $r = 10$ m for the big and the small spheres, respectively. The small sphere is the source of a dipole. NRBCs have been used in order to avoid reflected waves in all the exterior surface. The mesh is composed by tetrahedral elements of size $h = 5$ m around the small sphere that are progressively increased to $h = 30$ m at the external borders of the big sphere. The training set is composed by the solution in all the time steps between $t = 400$ s and $t = 800$ s, the finer time step for the whole training being fixed at $\delta t = 0.5$ s. On the contrary, the time step of the coarse case is not fixed, so in the same run we can use different time steps for various time intervals. Specifically, we have used a coarse time step of $\delta t = 1$ s from $t = 400$ s to $t = 500$, $\delta t = 2$ s from $t = 500$ s to $t = 650$ s and $\delta t = 3$ s from $t = 650$ s to $t = 800$ s. This problem has also been solved with the irreducible wave equation in the time domain. The parameters of the ANN are 3 layers, 4 neurons per layer, 0.1 of learning momentum, 1000 maximum epochs and the tolerated error has been set to 1. For the purpose of testing this model, the execution phase is launched with a fixed time step of $\delta t = 1.5$ s, a time step that has not been used in the training phase.

As we have done in the previous sections, in Fig. 10 the solutions for the coarse, fine and corrected cases are displayed. It is important to note that in this example we have used a simpler ANN than in the two previous ones, so the improvement of the corrected case is smaller. On the one hand, regarding the qualitative analysis, it is easy to see that the corrected case of Fig. 10(c) is more similar the fine case (b) than the coarse one (c), and the stripes are more marked. On the other hand, regarding the quantitative analysis, in Fig. 11 we can see the time evolution of a random point $(24, 87, 0)$ for the three different cases. As before, the correction term is activated at time $t = 400$ s, where the amplitude of the oscillations is incremented, approaching the behavior of the fine case. The time execution phase has been incremented again with respect to the training phase. It is clear that there is improvement but smaller than in the previous examples, as the amplitude of the waves of the corrected case does not reach the one of the fine case; this is explained due to the complexity to simulate a 3D case and the simpler ANN employed.

Furthermore, regarding the computational times of this example, and considering that the fine case corresponds to the 100% of the computational time, the coarse case corresponds to the 6.8%. In this case, the cost of the training phase is the 104.5% and the cost of the execution phase is the 7.0%. These results match with the computational costs that we got in Section 5.3, with the main difference of the reduced cost of the training phase. This is directly related to the difference between the coarse time steps and the chosen fine time step; in Section 5.3 there was a factor of 10 between the smallest of the coarse time steps and the fine one, whereas here this factor is 2. It can be again observed that there is almost no extra cost in the execution phase if we compare it to the coarse case, so the conclusion is that the correction is really worthy.

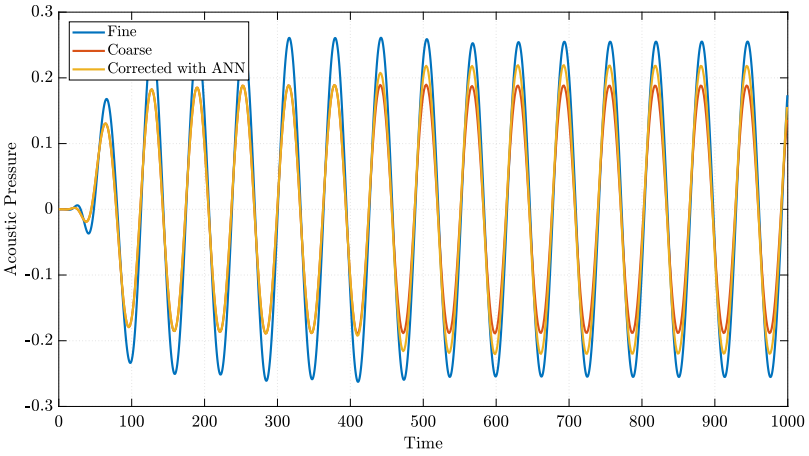


Fig. 11. Acoustic pressure evolution in time of the three cases of the numerical example 5.5 at the point marked in red in Fig. 10(c). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

As it has been explained at the beginning of this section, in some examples we have limitations to choose the fine time step. If in this example we choose a time factor of 10 between the smallest coarse time step and the fine one, we get an error on the mean value of the waves, is spite of the fact that the amplitudes are still partially corrected. This is the reason why the time ratio between coarse and fine time steps in this example and in the previous one has been chosen to be just 2.

6. Conclusions

In this paper, a general idea to enhance the solutions of coarse simulations through learning algorithms from fine solutions has been presented. This improvement is possible thanks to the training of correcting terms, that can be performed by LS or ANN models. This idea has been applied in this paper to the wave equation, both in the time and in the frequency domain.

The general concept of correction terms has been explained and some of its different possibilities have been described. Moreover, the specific correcting terms of the two different learning algorithms (LS and ANN) have been set and compared. The need for using iterative schemes in the case of using an ANN model has been explained, as well as the algorithmic issues of the application of this model in the time domain version of the wave equation.

Finally, the performance of the correcting terms based on training has been tested in different numerical examples. The first one corresponds to the Helmholtz equation trained with an ANN. The fine solutions have been chosen to be computed from a set of source positions with a fine mesh. It has been checked that the algorithm improves the coarse solution no matter which is the position of the source for a given frequency, so the model is capable to adapt to a change of geometry. The second example corresponds to the Helmholtz equation; this has been trained with an ANN and also solved with the LS method. The fine solutions have been chosen to be those obtained from a set of different frequencies and computed with a fine mesh. Again, it has been checked that the algorithm improves the coarse solution no matter what is the frequency in a certain range, so the model is capable to adapt to a change of a physical parameter. However, it has been checked that the LS method does not improve the solution in this case, confirming its lack of generality. The third, the fourth and the fifth examples are similar, all of them have been done with the irreducible form of the wave equation in the time domain, perhaps trained with more than one coarse time step, and launched with a time step that did not belong to the training set. The first of them has a simple domain and has been corrected using an ANN and with the LS method, due to the simplicity of the problem, this time the LS method has worked well and even better than the ANN; however, both have provided a significant improvement of the solution. The second of them has a more complex domain, but it has been successfully corrected by an ANN. The last one is the 3D adaption of the first one; it has also been solved with an ANN, but a simple one. The correction is not as good as that obtained in the previous examples, but nevertheless it is quite significant.

Declaration of competing interest

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

Acknowledgments

A. Fabra acknowledges the support obtained from the Spanish Government through the FPI grant RTI2018-098276-B-I00. J. Baiges gratefully acknowledges the support of the Spanish Government through the Ramón y Cajal grant RYC-2015-17367. R. Codina gratefully acknowledges the support received through the ICREA Acadèmia Research Program of the Catalan Government. This work was partially funded through the TOP-FSI: RTI2018-098276-B-I00 project of the Spanish Government. CIMNE is a recipient of a "Severo Ochoa Programme for Centers of Excellence in R&D" grant (CEX2018-000797-S) by the Spanish Ministry of Economy and Competitiveness.

References

[1] D.J. Lucia, P.S. Beran, W.A. Silva, Reduced-order modeling: new approaches for computational physics, Prog. Aerosp. Sci. 40 (1–2) (2004) 51–117.
[2] J.S. Hesthaven, G. Rozza, B. Stamm, et al., Certified Reduced Basis Methods for Parametrized Partial Differential Equations, Vol. 590, Springer, 2016.
[3] A. Quarteroni, A. Manzoni, F. Negri, Reduced Basis Methods for Partial Differential Equations: An Introduction, Vol. 92, Springer, 2015.
[4] F. Chinesta, P. Ladeveze, E. Cueto, A short review on model order reduction based on proper generalized decomposition, Arch. Comput. Methods Eng. 18 (4) (2011) 395–404.
[5] P. Chen, C. Schwab, Adaptive sparse grid model order reduction for fast Bayesian estimation and inversion, in: Sparse Grids and Applications-Stuttgart 2014, Springer, 2016, pp. 1–27.
[6] L. Feng, Review of model order reduction methods for numerical simulation of nonlinear circuits, Appl. Math. Comput. 167 (1) (2005) 576–591.
[7] J. Burkardt, M. Gunzburger, H.-C. Lee, POD and CVT-based reduced-order modeling of Navier–Stokes flows, Comput. Methods Appl. Mech. Engrg. 196 (1–3) (2006) 337–355.
[8] J. Baiges, R. Codina, I. Castanar, E. Castillo, A finite element reduced-order model based on adaptive mesh refinement and artificial neural networks, Internat. J. Numer. Methods Engrg. 121 (4) (2020) 588–601.
[9] O. Guasch, M. Arnela, R. Codina, H. Espinoza, A stabilized finite element method for the mixed wave equation in an ALE framework with application to diphthong production, Acta Acust. United Acust. 102 (1) (2016) 94–106.
[10] P. Monk, J. Schöberl, A. Sinwel, Hybridizing Raviart-Thomas elements for the Helmholtz equation, Electromagnetics 30 (1–2) (2010) 149–176.
[11] L. Pierce, Acoustics, Springer, 2019.
[12] F. Ihlenburg, Finite Element Analysis of Acoustic Scattering, Springer, 1998.
[13] L.C. Wrobel, The Boundary Element Method, Volume 1: Applications in Thermo-Fluids and Acoustics, Vol. 1, John Wiley & Sons, 2002.
[14] S.M. Kirkup, The Boundary Element Method in Acoustics, Integrated sound software, 2007.
[15] M.R. Bai, Application of BEM (boundary element method)-based acoustic holography to radiation analysis of sound sources with arbitrarily shaped geometries, J. Acoust. Soc. Am. 92 (1) (1992) 533–549.
[16] I. Harari, F. Magoulès, Numerical investigations of stabilized finite element computations for acoustics, Wave Motion 39 (4) (2004) 339–349.
[17] J. Baiges, R. Codina, A variational multiscale method with subscales on the element boundaries for the Helmholtz equation, Internat. J. Numer. Methods Engrg. 93 (6) (2013) 664–684.
[18] H. Espinoza, R. Codina, S. Badia, A sommerfeld non-reflecting boundary condition for the wave equation in mixed form, Comput. Methods Appl. Mech. Engrg. 276 (2014) 122–148.
[19] T.J. Hughes, Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods, Comput. Methods Appl. Mech. Engrg. 127 (1–4) (1995) 387–401.
[20] T.J. Hughes, G.R. Feijóo, L. Mazzei, J.-B. Quincy, The variational multiscale method—A paradigm for computational mechanics, Comput. Methods Appl. Mech. Engrg. 166 (1–2) (1998) 3–24.
[21] J. Baiges, R. Codina, S. Idelsohn, Reduced-order subscales for POD models, Comput. Methods Appl. Mech. Engrg. 291 (2015) 173–196.
[22] S.J. Miller, The method of least squares, in: The Probability Lifesaver, Princeton University Press, 2017, pp. 625–635.
[23] S. Nissen, et al., Implementation of a fast artificial neural network library (FANN), Rep., Dep. Comput. Sci. Univ. Copenhagen (DIKU) 31 (2003) 29, URL https://github.com/libfann/fann.
[24] P. Ramuhalli, L. Udpa, S.S. Udpa, Finite-element neural networks for solving differential equations, IEEE Trans. Neural Netw. 16 (6) (2005) 1381–1392.
[25] H. Qu, X. Liu, Z. She, Neural network method for fractional-order partial differential equations, Neurocomputing 414 (2020) 225–237.

[26] A.A. Anastassi, Constructing Runge–Kutta methods with the use of artificial neural networks, Neural Comput. Appl. 25 (1) (2014) 229–236.
[27] Z. Zhe-Zhao, W. Yao-Nan, W. Hui, Numerical integration based on a neural network algorithm, Comput. Sci. Eng. 8 (4) (2006) 42–48.
[28] J.S. Hesthaven, S. Ubbiali, Non-intrusive reduced order modeling of nonlinear problems using neural networks, J. Comput. Phys. 363 (2018) 55–78.
[29] A. Stanziola, S.R. Arridge, B.T. Cox, B.E. Treeby, A Helmholtz equation solver using unsupervised learning: Application to transcranial ultrasound, J. Comput. Phys. 441 (2021) 110430.
[30] R. Codina, Finite element approximation of the hyperbolic wave equation in mixed form, Comput. Methods Appl. Mech. Engrg. 197 (13–16) (2008) 1305–1322.
[31] R. Codina, S. Badia, J. Baiges, J. Principe, Variational multiscale methods in computational fluid dynamics, in: E. Stein, R. Borst, T.J.R. Hughes (Eds.), Encyclopedia of Computational Mechanics, John Wiley & Sons Ltd., 2017, pp. 1–28, http://dx.doi.org/10.1002/9781119176817.ecm2117.
[32] R. Codina, Stabilized finite element approximation of transient incompressible flows using orthogonal subscales, Comput. Methods Appl. Mech. Engrg. 191 (39–40) (2002) 4295–4321.
[33] R. Codina, Analysis of a stabilized finite element approximation of the oseen equations using orthogonal subscales, Appl. Numer. Math. 58 (3) (2008) 264–283.
[34] A. Pont, R. Codina, J. Baiges, Interpolation with restrictions between finite element meshes for flow problems in an ALE setting, Internat. J. Numer. Methods Engrg. 110 (2017) 1203–1226.