

Solving the wave equation with physics-informed deep learning

B. Moseley* and A. Markham

*Department of Computer Science
University of Oxford*

T. Nissen-Meyer

*Department of Earth Sciences
University of Oxford*

(Dated: June 23, 2020)

We investigate the use of Physics-Informed Neural Networks (PINNs) for solving the wave equation. Whilst PINNs have been successfully applied across many physical systems, the wave equation presents unique challenges due to the multi-scale, propagating and oscillatory nature of its solutions, and it is unclear how well they perform in this setting. We use a deep neural network to learn solutions of the wave equation, using the wave equation and a boundary condition as direct constraints in the loss function when training the network. We test the approach by solving the 2D acoustic wave equation for spatially-varying velocity models of increasing complexity, including homogeneous, layered and Earth-realistic models, and find the network is able to accurately simulate the wavefield across these cases. By using the physics constraint in the loss function the network is able to solve for the wavefield far outside of its boundary training data, offering a way to reduce the generalisation issues of existing deep learning approaches. We extend the approach for the Earth-realistic case by conditioning the network on the source location and find that it is able to generalise over this initial condition, removing the need to retrain the network for each solution. In contrast to traditional numerical simulation this approach is very efficient when computing arbitrary space-time points in the wavefield, as once trained the network carries out inference in a single step without needing to compute the entire wavefield. We discuss the potential applications, limitations and further research directions of this work.

I. INTRODUCTION

Waves are the most effective means of information transfer over many distance scales, and they play a central role in natural as well as technological phenomena. Solving the wave equation is therefore a hugely important task for a wide range of physics problems. It can take many forms and arises in fields such as acoustics, electromagnetics, cosmology, fluid dynamics and not least in the field of geophysics, where it is needed to simulate earthquakes (Boore 2003, Cui et al. 2010), estimate subsurface structure (Fichtner 2010, Schuster 2017, Tarantola 1987, Virieux and Operto 2009), carry out non-destructive testing (Liu and Guo 2005), model tsunamis (Maeda and Furumura 2013) and characterise the interior of the Earth and other planets (Hosseini et al. 2019, Stähler et al. 2018), amongst many other applications.

The most popular methods for solving the wave equation are Finite Difference (FD) and spectral element methods, which are general methods for solving differential equations (Igel 2017, Komatitsch and Tromp 1999, Leng et al. 2019, Moczo et al. 2007). In these approaches the wave equation is discretised and iterative time-stepping schemes are used to solve it. These methods have benefited from decades of development, and highly sophisticated algorithms exist which are able to model a large range of physics; for example in geophysics,

the effects of solid-fluid interfaces, intrinsic attenuation and anisotropy can be readily incorporated into the solution (Colombi et al. 2012, Fernando et al. 2020, Tesoniero et al. 2020, van Driel and Nissen-Meyer 2014).

However, a key difficulty when using these methods is their computational cost. For large 3D solutions, billions of mesh points may be required needing large supercomputers to run (Bohlen 2002, Leng et al. 2019). Furthermore, they suffer from discretisation error and this typically requires the grid size to be made much smaller than the rate of change of the system for it to be acceptable (Courant et al. 1967), especially if the waves are propagating over thousands of wavelengths of distance (Nissen-Meyer et al. 2014). Unless local time stepping is used, temporal sampling of the dynamic evolution often needs to be extremely small to accommodate the constraints imposed by the meshing.

In recent years modern deep learning techniques have shown excellent promise in their ability to simulate physical phenomena. For example, Paganini et al. (2018) showed deep neural networks could simulate particle showers in particle colliders, Rasp et al. (2018) showed they could provide fast emulators for subgrid processes in climate simulations and Kasim et al. (2020) proposed a general neural architecture search which was able to accelerate simulations across a wide range of scientific tasks. Deep learning is also having a large impact in the field of fluid mechanics (Brunton et al. 2020). For solving the wave equation, Moseley et al. (2020) showed that deep neural networks can simulate waves in complex faulted media and Siahkoohi et al. (2019) showed

* bmooley@robots.ox.ac.uk

that FD simulation can be accelerated by using convolutional neural networks and larger time steps. However, a key issue with many of the existing approaches above is their inability to generalise outside of their training distribution (Moseley et al. 2020, Zhang and Lin 2018). These networks typically rely entirely on their training data and therefore perform poorly outside of it. Generalising outside of the training data is a challenge of deep neural networks in general and is important to overcome if these “naive” methods are to move on to more realistic applications (Moseley et al. 2020).

A recent development is the rise of physics-informed machine learning (Arridge et al. 2019, Karpatne et al. 2017, Raissi et al. 2019). Instead of replacing known physics with purely data-driven deep neural networks in a wholesale fashion, the idea is to blend physical constraints into the workflow in an attempt to combine the best of both worlds. In particular, a recent approach suggested by Raissi et al. (2019) proposes physics-informed neural networks (PINNs). They design a general scheme for solving the differential equations governing a physical system where a neural network represents a solution of the system. Importantly, the underlying equations are used to directly constrain the network whilst training. Raissi et al. showed that these networks could learn solutions which were far outside of the boundary data used to train them, which offers a potential way to address the generalisation issues of deep neural networks.

The introduction of PINNs has prompted a large amount of follow up work (Jagtap et al. 2020, Yang et al. 2020, Yang and Perdikaris 2019). For example, Sahli Costabal et al. (2020) used them to diagnose atrial fibrillation, Chen et al. (2020) used them to solve inverse scattering problems in nano-optics, and there has been much investigation in their use for fluid mechanics (Erichson et al. 2019, Raissi et al. 2020, Sun et al. 2020, Zhu et al. 2019). In geophysics Xu et al. (2020) and Costa Nogueira Junior et al. (2019) used PINNs to constrain seismic inversion, Shukla et al. (2020) used them for ultrasound inversion and Zhang et al. (2020) used them for structural response testing. Smith et al. (2020) presented a network to predict the travel times of waves in heterogeneous media by solving the Eikonal equation. However, little work has been carried out on their use with the wave equation.

Compared to the equations already studied, the wave equation presents unique challenges because of the multi-scale, propagating and oscillatory nature of its solutions, and it is unclear how well PINNs perform in this setting. In this work we investigate the effectiveness of physics-informed neural networks for solving the wave equation. Whilst we place emphasis towards its applications in geophysics, we believe our results are informative for solving the wave equation in general. Our contribution is as follows;

- We design a physics-informed neural network for solving the wave equation and show it is able to accurately solve the pressure response in complex 2D

acoustic media. The network is only given the first few timesteps of the solution as training data and is able to accurately predict the solution at much later times. It is able to model a wide range of physical phenomena induced by the wave equation, including the transmission and reflection of waves, the compression and expansion of waves through different velocity regions and the attenuation of waves due to spherical divergence.

- We extend the original PINN approach proposed by Raissi et al. (2019) by conditioning the network on the initial source location, showing that it is able to generalise over this initial condition without needing to retrain the network. This allows this network to be much more efficient at computing the solution for varying source locations compared to traditional FD modelling.
- We propose a curriculum-learning based strategy for training the PINN, rather than the standard approach used by Raissi et al. (2019), which improves convergence.
- We find and discuss specific challenges when using PINNs to solve the wave equation, such as dealing with discontinuities at the interfaces of media, and highlight ways to address them.

Compared to Smith et al. (2020) we solve the wave equation, rather than the Eikonal equation, and compared to Xu et al. (2020) and Costa Nogueira Junior et al. (2019) our focus is on simulation using the wave equation rather than inversion. Sitzmann et al. (2020) presented a physics-informed network for solving the wave equation, although they only considered a constant velocity model. We extend the original approach proposed by Raissi et al. (2019) by conditioning the network on its initial conditions and by proposing a curriculum learning strategy.

In Section II we describe PINNs and our workflow using them to solve the wave equation. In Section III we test our approach by simulating 2D waves in acoustic media of varying complexity and in Section IV we discuss the potential applications, limitations and further research directions of this work.

II. METHODS

A. Physics-informed neural networks

PINNs attempt to solve physical systems which can be written as

$$\mathcal{N}[u(t, x); \lambda] = 0, \quad x \in \mathbb{R}^D, \quad t \in \mathbb{R}, \quad (1)$$

where $\mathcal{N}[u(t, x); \lambda]$ is an underlying differential operator describing the physical system, parameterised by λ , and $u(t, x)$ represents the solution of the system (Raissi et al. 2019). Many different physical phenomena can be

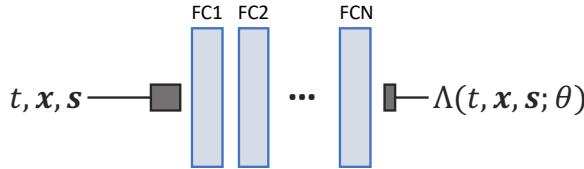


FIG. 1. Physics-informed neural network used to solve the wave equation. The input to the network is a single point in time and space (t, x) and its output is an approximation of the wavefield solution at this location. We extend the original PINN approach proposed by Raissi et al. (2019) by further conditioning the network on the initial source location s . We use a fully connected network architecture with 10 layers, 1024 hidden channels, softplus activation functions before all hidden layers and a linear activation function for the final layer.

described in this form, including conservation laws, equations of motion, fluid dynamics and many forms of the wave equation.

The basic idea of PINNs is to use a neural network $\Lambda(t, x; \theta)$, parameterised by θ , to approximate the solution of the physical system $u(t, x)$ and to use Equation 1 as a constraint when training the network. Importantly, the network is a direct functional approximation of $u(t, x)$. It is trained using a loss function which includes both a boundary condition and the underlying equation, given by

$$\mathcal{L} = \frac{1}{N_u} \sum_{i=1}^{N_u} \|u(t_i, x_i) - \Lambda(t_i, x_i)\|^2 + \frac{1}{N_\Lambda} \sum_{j=1}^{N_\Lambda} \|\mathcal{N}[\Lambda(t_j, x_j; \theta); \lambda]\|^2, \quad (2)$$

where $u(t_i, x_i)$ are known initial values of the solution at some boundary points (t_i, x_i) , and (t_j, x_j) are points sampled from the entire input space. By constructing the loss function in this way, the first term (denoted the “boundary loss”) attempts to ensure the network learns a unique solution and the second term (denoted the “physics loss”) tries to ensure that the network honours the underlying equation. A notable benefit of this approach is that it does not require discretisation, in contrast to traditional numerical methods. From a machine learning point of view, the second term in the loss function can be seen as an unsupervised regulariser and thus the network is likely to have better generalisation performance outside of the boundary data than one trained with just the boundary loss.

It is important to note that the gradients of neural networks with respect to their inputs are typically analytically available and therefore the physics loss in Equation 2 can be derived. Furthermore, a key practical enabler in this approach is the development of auto-differentiation in modern deep learning packages (Bay-

din et al. 2017, PyTorch 2018, TensorFlow 2015), which allows such gradients to be easily derived, and to be further differentiated with respect to the network weights θ in order to train the network.

B. Solving the wave equation

In this work we investigate the use of PINNs for solving the wave equation. To test our approach, we focus on the 2D acoustic wave equation, given by

$$\rho \nabla \cdot \left(\frac{1}{\rho} \nabla u \right) - \frac{1}{v^2} \frac{\partial^2 u}{\partial t^2} = -\rho \frac{\partial^2 f}{\partial t^2}, \quad (3)$$

where in this setting $u(t, x)$ represents the pressure response in an acoustic medium (known as the “wavefield”), $f(t, x)$ is a point source of volume injection, $v = \sqrt{\kappa/\rho}$ is the velocity of the medium, ρ is the density of the medium and κ is the adiabatic compression modulus (Long et al. 2013). In general both the density and velocity of the medium can vary spatially. When the density of the medium is constant and the source term is negligible Equation 3 reduces to the canonical form of the wave equation, given by

$$\nabla^2 u - \frac{1}{v^2} \frac{\partial^2 u}{\partial t^2} = 0. \quad (4)$$

Whilst we concentrate on the acoustic wave equation here, we note that our approach is general and is readily extendable to more complex forms.

Despite its linearity, the wave equation is notoriously difficult to solve in complex media, not least due to the broadband, oscillatory and dispersive nature of its solutions. The dynamics of the wavefield can be highly complex, and include reflected, transmitted and grazing waves at interfaces in the media (including prolonged “bouncing” around of the wavefield between interfaces), wavefront compression, expansion and healing through different velocity regions, spherical spreading, multiple types of waves interfering simultaneously and a large range of amplitudes and frequencies (Igel 2017). Compared to some of the nonlinear differential equations already investigated by PINNs, such as those in fluid mechanics, it is by no means easier to solve, and it is unclear how well PINNs perform in this setting.

From a machine learning perspective, a number of specific challenges arise when applying PINNs to the wave equation. Firstly, it is uncertain whether neural networks are expressive enough to represent the full range of dynamics stated above. Secondly, the wave equation is a second order partial differential equation and requires strict boundary conditions on both the initial wavefield and its derivatives for its solution to be unique; it is unclear if the boundary loss in Equation 2 will be sufficient in this setting. Thirdly, from a computational point of view the second order gradients of a network are typically much more expensive to compute than its first or-

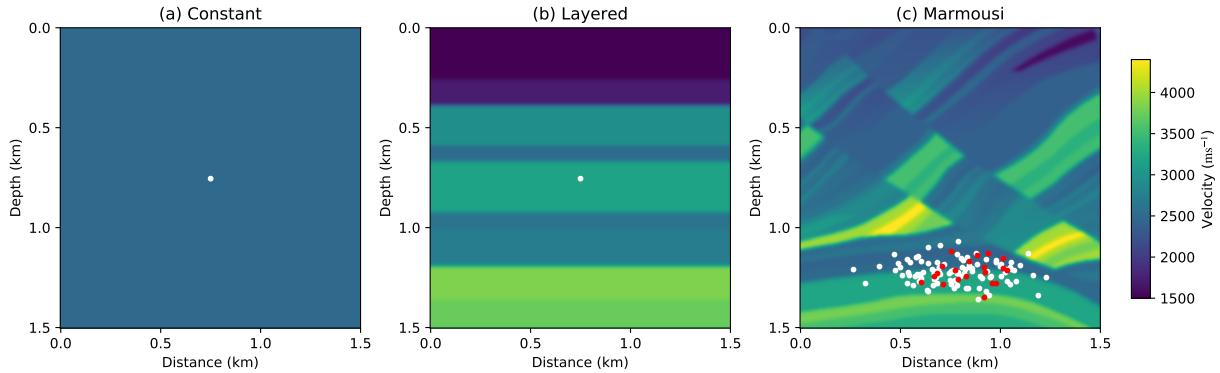


FIG. 2. Velocity models used to test our approach. We use a PINN to solve the wave equation in a variety of different media, starting from a simple homogenous velocity model (left), to a layered velocity model (middle) and finally a region of the Earth-realistic Marmousi model (Martin et al. 2006) (right). The velocity model is an implicit parameter in the training scheme and the network must be retrained to solve the wave equation for each case. White points show the locations of the sources used to train the network in each case. For the Marmousi model we train the network to generalise over different source locations and red points show the source locations used to test the network.

der derivatives and this potentially restricts the type and size of neural networks which can be used.

In this work we address the first point by testing the ability of PINNs to solve the wave equation in a variety of different conditions, starting from simple solutions in homogeneous media to complex dynamics in heterogeneous Earth-realistic media. We also introduce a curriculum learning-based strategy to help train the network. For the second point, we include multiple time steps in the boundary training data to help constrain both the temporal and spatial gradients of the initial solution and for the third point we restrict ourselves to a relatively small 10 layer fully connected network where its second order gradient computations with respect to its inputs are tractable. We also use a continuous softplus activation function, rather than the more standard piece-wise linear ReLU activation function, to ensure that the network has non-zero second order derivatives.

A limitation of the original PINN approach is that the network must be retrained to solve every new initial condition of the physical system. This is practically undesirable, as it is likely to take as long to train the network as it takes to solve the underlying equations using traditional methods, or longer, at least for the wave equation. In this work, for our last case study (the Earth-realistic media), we condition the network on the initial source location and train over many different source locations, allowing it to generalise over this initial condition without needing to be retrained. We now discuss our network design and training scheme in detail.

C. PINN design and training strategy

Our PINN workflow used for solving the wave equation is shown in Figure 1. The input to the network is an arbitrary point in time and space (t, x) and its out-

put is an approximation of the wavefield solution at this location. A fully connected network architecture is used with 10 layers, 1024 hidden channels, softplus activation functions before all hidden layers and a linear activation function for the final layer. The network is designed to solve the wave equation in a fixed media (implicitly defined in the physics loss) and must be retrained for each new media.

We use data from FD simulation as a boundary condition to train the network. Given a fixed velocity model, density model, source location and source signature as initial conditions, we run FD simulation for a small number of starting time steps to obtain an initial (discretised) wavefield. Values from this initial wavefield $u_{\text{FD}}(t_i, x_i)$, $x_i \in [0, X_{\text{max}}]$, $t_i \in [0, T_1]$ are used to compute the boundary loss in Equation 2. Importantly, only a small number of initial timesteps ($t_i < T_1$) are used to train the network. For the physics loss in Equation 2, we randomly sample points (t_j, x_j) , $x_j \in [0, X_{\text{max}}]$, $t_j \in [0, T_{\text{max}}]$, $T_{\text{max}} \gg T_1$ over a much larger input space, which includes times much later than the initial wavefield, to attempt to learn the solution outside of the boundary training data. After training we use later time steps from FD simulation to test the generalisation ability of the network at these later times.

For the last case study, our network is extended by concatenating the source location, s , to its inputs (shown in Figure 1), and it is trained to simulate the wavefield over many different source locations. In order to sufficiently constrain the network solution in this case we run FD simulations with many different initial source locations in the boundary training set, and randomly sample over the source location when computing the physics loss.

To train all networks we modify the loss function given

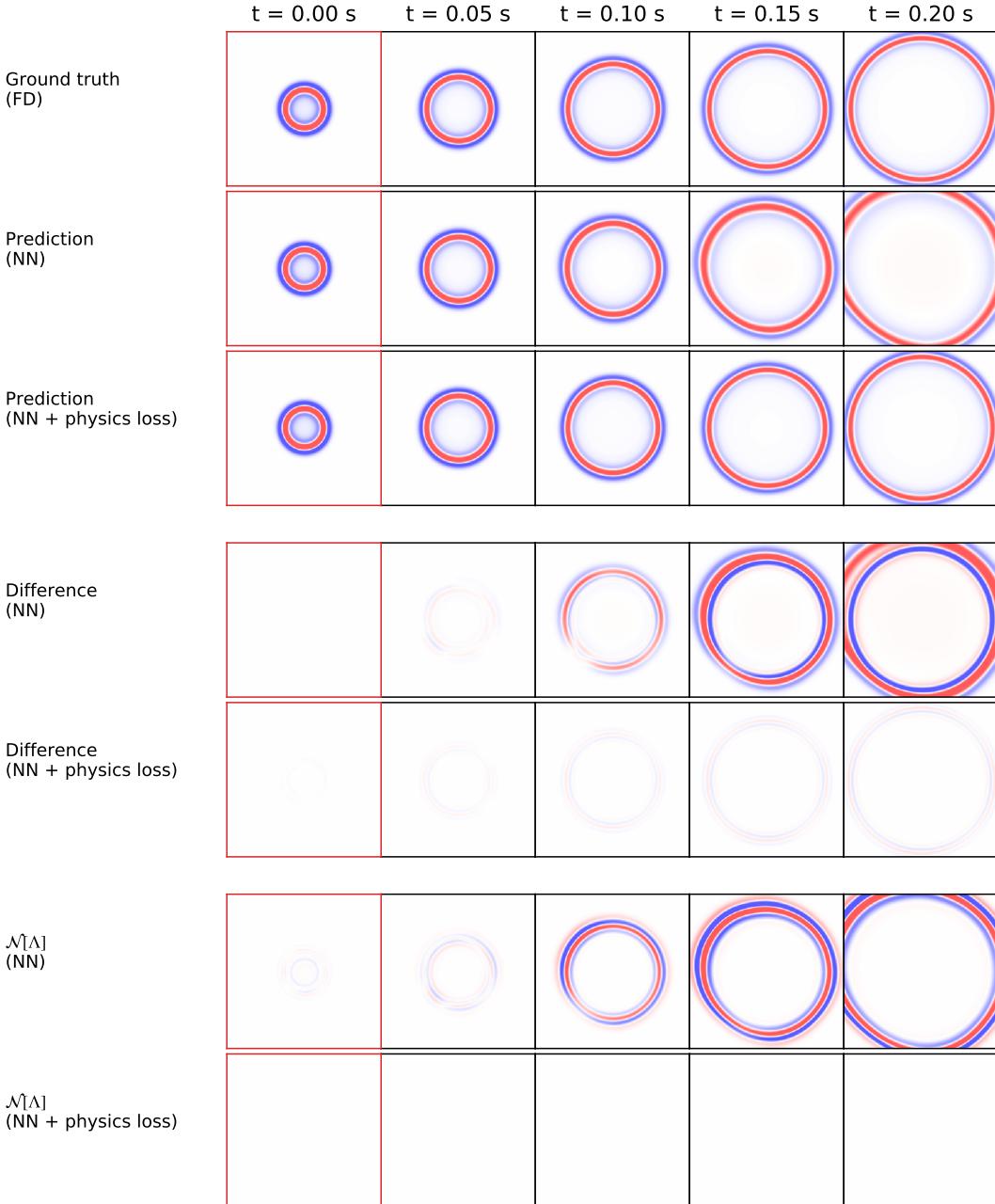


FIG. 3. Comparison of the PINN wavefield prediction to ground truth FD simulation, using a homogeneous velocity model. The PINN (“NN + physics loss”) is also compared to the same network trained using only the boundary loss (“NN”). Top three rows show the FD, PINN and uninformed network wavefield solutions through time. Middle two rows show the difference of the two network predictions to FD simulation. Bottom two rows show the value of the differential operator in the physics loss in Equation 5 (right hand term) for both networks, which is close to zero for an accurate solution to the wave equation. Plots bordered in black indicate wavefield times which are outside of the boundary training data. The boundary training data only covers the time range 0.00–0.02 s and the PINN is able to generalise to much later times (beyond 0.20 s, 10× the range of the boundary data). Colour bar ranges are the same between each type of plot, for fair comparison. The colour bar is shown in Figure 6.

by Equation 2 to

$$\begin{aligned} \mathcal{L} = & \frac{1}{N_u} \sum_{i=1}^{N_u} \|u_{\text{FD}}(t_i, x_i, s_i) - \Lambda(t_i, x_i, s_i)\|^2 + \\ & \frac{k}{N_\Lambda} \sum_{j=1}^{N_\Lambda} \|\mathcal{N}[\Lambda(t_j, x_j, s_j; \theta); v(x_j)]\|, \quad (5) \end{aligned}$$

where $v(x)$ is the fixed velocity model and $k \in \mathbb{R}^+$ is a hyperparameter added to allow more control over the balance between the two terms. In testing we find that a L1 norm on the physics term shows better convergence

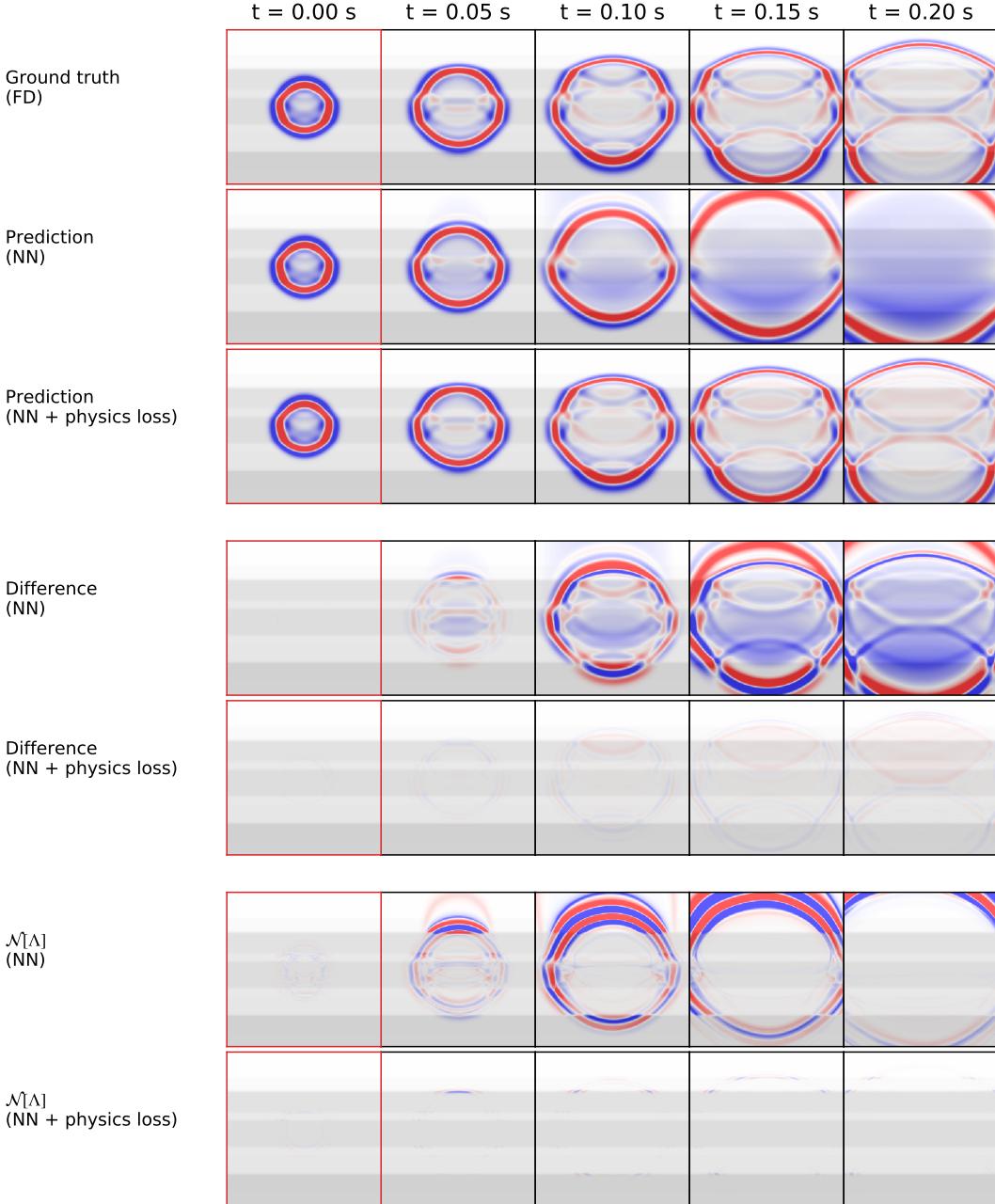


FIG. 4. Comparison of the PINN wavefield prediction to ground truth FD simulation, using a layered velocity model. The same layout as Figure 3 is used; top three rows show the FD, PINN and uninformed network wavefield solutions through time; middle two rows show the difference of the two network predictions to FD simulation; bottom two rows show the value of the differential operator in the physics loss in Equation 5 (right hand term) for both networks, which is close to zero for an accurate solution to the wave equation; plots bordered in black indicate wavefield times which are outside of the boundary training data. In this case the plots are overlaid on the velocity model, shown in grey. Similar to Figure 3 the boundary training data only covers the time range 0.00-0.02 s and the PINN is able to generalise to much later times (beyond 0.20 s, 10× the range of the boundary data).

than an L2 norm for this task (discussed further in Section III C). For simplicity we only use starting time steps after the source term becomes negligible in the boundary data, and use a constant density model, which allows the differential operator $\mathcal{N} = \nabla^2 - (1/v^2)\partial_{tt}$ to be used in

the physics loss in Equation 5. We write our own code to analytically compute the second order derivatives of the network in order to compute the physics loss, although auto-differentiation could also be used.

Curriculum learning is used to help the network con-

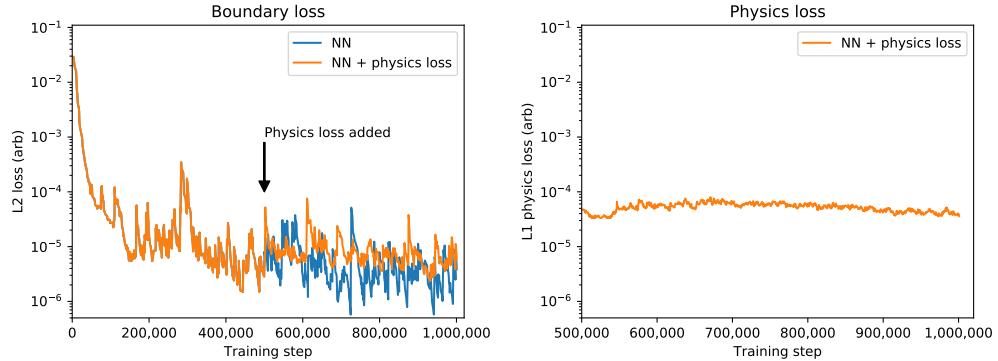


FIG. 5. Training loss for the layered velocity model against training step. Left shows the boundary loss and right shows the physics loss in Equation 5. The physics loss is added halfway through training and includes an expanding time horizon to help the network converge. The PINN (“NN + physics loss”) is compared to a network trained only using the boundary loss (“NN”).

verge to a solution. We start by training the network to reconstruct the boundary data only by using the boundary loss, and then “switch on” the physics loss halfway through training. This allows the network to learn the initial wavefield before learning to solve the wave equation through time. Furthermore a linearly growing time horizon is used when feeding points to the physics loss, analogous to the way a numerical solver iteratively solves the wave equation through time. Both schemes are intended to allow the network to learn incrementally rather than all at once and in testing they are found to improve convergence (see Appendix A).

The network is trained using the Adam stochastic gradient descent algorithm (Kingma and Ba 2014). For each update step a random set of discretised points are sampled from the initial wavefield to compute the boundary loss and a random set of continuous points over the full input space up to the current horizon time are sampled to compute the physics loss. A batch size of 500 samples in both sets is used for each update step with a learning rate of 1×10^{-5} .

D. Boundary data generation

We solve the wave equation using three different velocity models to test our approach, shown in Figure 2. The PINN is retrained in each setting and the models are chosen in increasing complexity to test the limits of the approach. They consist of a homogeneous velocity model with a fixed value of 2500 ms^{-1} , a horizontally layered velocity model and a region of the Earth-realistic Marmousi P-wave velocity model (Martin et al. 2006). 2D Gaussian smoothing with a standard deviation of 2 grid points is applied to the each model before it is used to alleviate discontinuity issues (discussed in more detail in Section III and Appendix A).

We use the SEISMIC_CPMML FD modelling code (Komatitsch and Martin 2007) to generate the initial wave-

field training data for each case. All simulations use a 300×300 grid with a spacing of 5 m in each direction (i.e. $X_{\max} = 1500$ m in both directions), a 0.002 s time interval, a constant density model of 2200 kgm^{-2} and a 20 Hz Ricker source time function. For the first two velocity models a single simulation is used. The point source is placed in the centre of the velocity model and the first 10 time steps (i.e. $T_1 = 0.02$ s or 0.14 source cycles) after the source term becomes negligible are taken as training data. For the Marmousi velocity model 100 simulations are used each with a random source location drawn from a 2D Gaussian distribution located towards the bottom of the model (shown in Figure 2), and 20 time steps (i.e. $T_1 = 0.04$ s or 0.28 source cycles) are taken. For all cases the points used for computing the physics loss are sampled up to a maximum time of $T_{\max} = 0.4$ s.

III. RESULTS

A. Homogeneous velocity model, single source

Our first case study tests the ability of the PINN in the simplest possible case: a homogeneous velocity model. In this media the initial wavefield propagates outwards with a fixed velocity and its amplitude reduces due to spherical divergence. We train the network and find that both the boundary loss and physics loss converge, and predictions of the wavefield after training are shown in Figure 3. The network agrees with the FD solution very accurately, capturing both its kinematics and its amplitude attenuation. We also train the same network with only the boundary loss, and show its wavefield predictions in Figure 3. In contrast to this network, the physics-informed network is able to accurately predict the wavefield at times much later than the initial wavefield, showing much better generalisation capability outside of the boundary training data. Despite its poor accuracy, the uniformed network is still able to capture the notion of the wavefront prop-

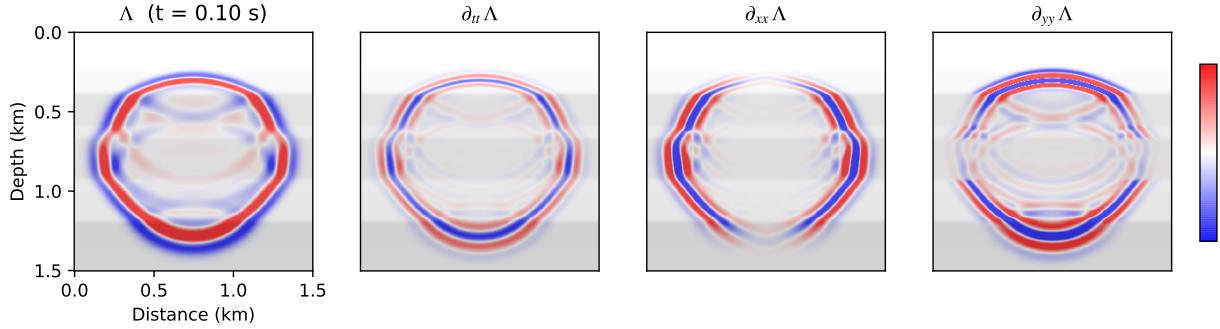


FIG. 6. Second order derivatives of the PINN trained using the layered velocity model. The derivatives are shown at a single snapshot in time and the plots are overlaid on the velocity model. From left to right: network wavefield prediction, its second order time, second order x and second order y derivatives. Sharp changes occur in the second order y derivative at velocity interfaces. Arbitrary scale used for colour bar.

agating outwards outside of the training data. This case study confirms the validity of the approach and gives us confidence to move onto the more complex case studies below.

B. Layered velocity model, single source

Our second case study uses the horizontally layered velocity model, which induces much more complex wavefield dynamics; reflected and transmitted waves form at each velocity interface and the wavefield is compressed and expanded when travelling through different regions of the velocity model. The boundary and physics loss converge and the predictions of the PINN after training are shown in Figure 4. The network is able to accurately capture the full range of dynamics: it is able to simulate transmitted and reflected waves which are not present in the initial wavefield, in contrast to the same network just trained using the boundary loss. This suggests that it is able to capture the specific physics phenomena occurring at the interfaces of the media. It is also able to accurately capture the compression, expansion and spherical spreading attenuation of the wavefield.

For this case we plot the boundary and physics loss during training in Figure 5. The boundary loss slightly increases when the physics loss is added and then remains stable. The physics loss is relatively constant throughout training, even as later times are added in the expanding time horizon, suggesting on average it is kept at the same low level as the initial wavefield. However, looking in more detail at Figure 4 we observe that the loss is higher along the interfaces in the velocity model. This is perhaps due to the discontinuities in the velocity model which translate into discontinuities in the second order gradients of the wavefield along the y -direction, which the network may struggle to represent. We plot the second order gradients of the network in Figure 6 and find that the network has learnt sharp, although not fully discontinuous, contrasts at the interfaces.

C. Marmousi model, multiple sources

Our final case study uses the Marmousi velocity model. This is a complex, Earth-realistic model which includes faults, velocity inversions and a large range of dips (angles of interfaces). For this case study we further increase the complexity of the problem by conditioning the network on the source location, using the source locations shown in Figure 2 to train the network. To test the network we generate a separate test set of 20 input source locations which are not used during training. We find that the boundary and physics loss converge, albeit at a slower rate than for the homogeneous and layered velocity models, and the predictions of the PINN for three of the source locations in this test set are compared to FD simulation in Figure 7.

In this case the network is able to accurately simulate the kinematics of the initial wavefront throughout the media to times much later than the initial wavefield. The network is also able to generalise across the source position, accurately modelling the initial wavefield and subsequent dynamics without needing to be retrained. However, the network struggles to learn the secondary reflected waves of the initial wavefront as it propagates through the model. The reason for this is unclear, and one explanation could be that because of their small relative amplitudes these reflections do not make a large enough contribution to the loss function in Equation 5 to be modelled accurately. In testing we found that an L1 norm on the physics loss improved the accuracy of the network compared to an L2 loss, perhaps by increasing the contribution of lower wavefield amplitudes, and it is possible that further amplitude balancing is required. Another explanation is that there are also inaccuracies in the FD simulation. We discuss this issue in more detail in Section IV.

For this case study the PINN takes approximately 1 day to train using a single Nvidia Titan V GPU, although we make little effort to optimise its training time. The majority of this time is spent computing the second order

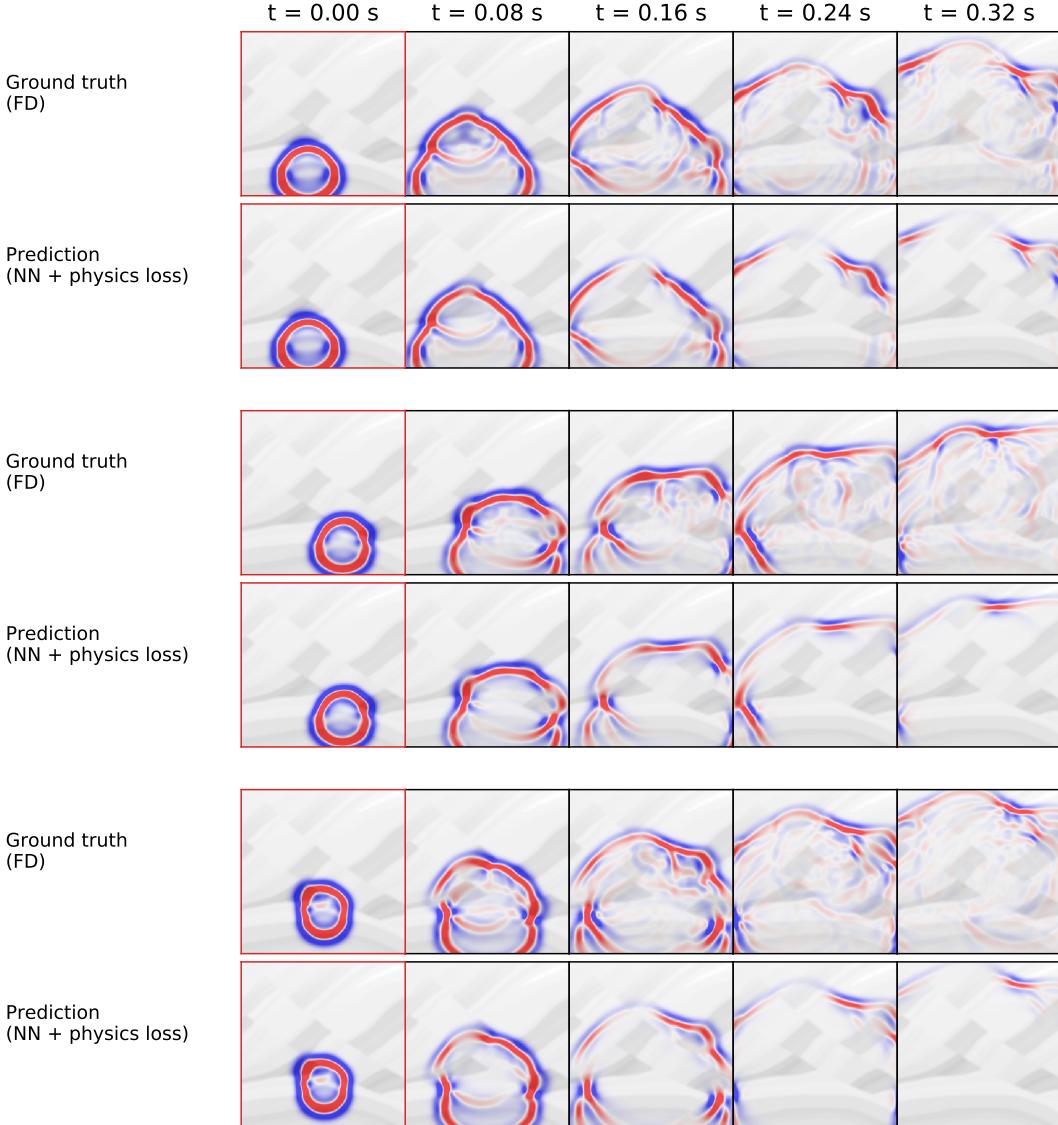


FIG. 7. Comparison of the PINN wavefield prediction to ground truth FD simulation, using the Marmousi velocity model. Each pair of rows shows the PINN prediction and FD simulation for a randomly selected source location in the test set. The plots are overlaid on the velocity model, shown in grey. For this case the boundary training data only covers the time range 0.00–0.04 s and the PINN is able to generalise to much later times (beyond 0.32 s, 8× the range of the boundary data).

derivatives of the network, which are approximately 10× as slow to calculate as the forward pass of the network. After training, the network is very efficient when computing the wavefield for arbitrary source positions at arbitrary points in time and space (of the order of 0.001 s), whilst 2D FD modelling of the full media is much slower for each source simulation (of the order of 1 s, or 1000× slower).

IV. DISCUSSION

We have shown that physics-informed neural networks are able to accurately solve the wave equation in increasingly complex media. They are able to capture a large range of physics, including the transmission and reflection of waves at interfaces, wavefield compression and expansion through different velocity regions and spherical spreading attenuation. Impressively they are able to accurately model the wavefield dynamics at much later times than the boundary data used to train them, and are therefore a promising approach for addressing the generality issues in existing deep learning approaches.

We extended the original approach proposed by Raissi et al. (2019) by conditioning our PINN on source location, which is practically desirable as it allows multiple simulations to be carried out without needing to retrain the network. A potential application of this work could be for seismic hazards assessment, where the quantification of hazards in a given media across different source locations is desired (Jena et al. 2020, Reiter 1990). Our method could provide a much more efficient approach than using multiple FD simulations for this task, both because the network does not need to be retrained and because the network can query the wavefield at specific points in time and space (rather than needing to solve for the entire wavefield). Our network could also be used for source inversion, as its gradients with respect to its inputs are readily accessible, and it is fully flexible to where the observed wavefield measurements are located. Another benefit of this approach is that the physics loss can be used to quantify the accuracy of the network’s solution, which could be useful to understand its reliability.

A challenge identified by this work is the modelling of waves at interfaces, where they introduce discrete changes in the second order derivatives of the wavefield. We note that modelling these discontinuities is also a challenge for numerical methods, where oversampling is typically required to ensure the solution remains accurate (Igel 2017). The performance of our network using the layered velocity model degraded at these interfaces and we somewhat avoided the problem by smoothing the velocity model before use, although errors were still observed along the interfaces. We show the network prediction when the layered velocity model without smoothing was used in Figure A2. The accuracy of this solution was significantly worse than in Figure 4. Another related issue was that the Marmousi test struggled to model reflected waves with low relative amplitudes. Future work could investigate these issues, for example by forcing the network to be discontinuous (e.g. by using separate networks for each velocity section), by changing the activation function (Sitzmann et al. 2020), by using the specific physics constraints (such as wavefield continuity (Aki and Richards 1980)) that arise at interfaces in the loss function, or by using a more intelligent sampling scheme, such as adaptive co-location (Anitescu et al. 2019), instead of random sampling, to sample interfaces more often in the loss function.

Another extension of this work is to condition the network on further initial conditions, such as the velocity model, as well as the source location. This would allow it to become even more general and potentially be useful for a wider range of tasks. Our last case study showed that our network has a high representational capacity

and is able to model the initial wavefield and subsequent dynamics for many different sources, and it may therefore be able to generalise further. One approach may be to use an “encoder” network to reduce the velocity model into a set of salient features before feeding it to the fully connected network to help the network converge.

Finally, our approach is readily extendable to 3D and more complex wave equations. 3D extension is trivial in the network design, as the dimensionality of the input tensor just needs to be increased by 1. Furthermore it may not be too practically challenging to generate 3D training data, as our approach only requires the first few time steps of FD simulation to train the network. One speed up of our approach could be to simulate multiple sources in each FD simulation and separate their wavefields before using them as training data, to reduce the number of FD simulations required. For more complex wave equations, such as the elastic wave equation, the underlying physics loss in Equation 5 just needs to be modified. The performance of our PINN on the wave equation suggests that PINNs may perform well for other hyperbolic systems too. Another exciting possibility is to investigate how efficiently the network solves high-frequency simulations, which take considerably more computation using numerical methods. Our future work will investigate these directions.

V. CONCLUSION

We have shown that physics-informed deep neural networks are able to solve the wave equation and generalise outside of their training set by adding physics constraints directly to their loss function. This offers a powerful new approach for simulating the wavefield which is efficient once trained and does not rely on discretisation. We further condition our network on source location, allowing it to predict the wavefield across many different source locations without needing to be retrained. Future work includes understanding the practical applications of this approach and extending it to solve more complex, 3D wave equations.

ACKNOWLEDGMENTS

The authors would like to thank the Computational Infrastructure for Geodynamics (www.geodynamics.org) for releasing the open-source SEISMIC_CPM1 FD modelling libraries.

- ers, *Materials and Continua*, 59(1):345–359, jan 2019. ISSN 15462226.
- Simon Arridge, Peter Maass, Ozan Öktem, and Carola-Bibiane Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019. ISSN 0962-4929.
- Atilim Günes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *The Journal of Machine Learning Research*, 18(1):5595–5637, 2017.
- Thomas Bohlen. Parallel 3-D viscoelastic finite difference seismic modelling. *Computers & Geosciences*, 28(8):887–899, oct 2002. ISSN 00983004.
- David M. Boore. Simulation of ground motion using the stochastic method. *Pure and Applied Geophysics*, 160(3-4):635–676, 2003. ISSN 00334553.
- Steven L. Brunton, Bernd R. Noack, and Petros Koumoutsakos. Machine Learning for Fluid Mechanics. *Annual Review of Fluid Mechanics*, 52(1):477–508, 2020. ISSN 0066-4189.
- Yuyao Chen, Lu Lu, George Em Karniadakis, and Luca Dal Negro. Physics-informed neural networks for inverse problems in nano-optics and metamaterials. *Optics Express*, 28(8):11618, apr 2020. ISSN 1094-4087.
- Andrea Colombi, Tarje Nissen-Meyer, Lapo Boschi, and Domenico Giardini. Seismic waveform sensitivity to global boundary topography. *Geophysical Journal International*, 191(2):832–848, 2012. ISSN 0956-540X.
- A. Costa Nogueira Junior, J. L. De Sousa Almeida, M. Paredes Quiñones, and L. Silveira De Albuquerque Martins. Physics-based machine learning inversion of subsurface elastic properties. In *81st EAGE Conference and Exhibition 2019*, volume 2019, pages 1–5. EAGE Publishing BV, jun 2019. ISBN 9789462822894.
- R. Courant, K. Friedrichs, and H. Lewy. On the Partial Difference Equations of Mathematical Physics. *IBM Journal of Research and Development*, 11(2):215–234, apr 1967. ISSN 0018-8646.
- Yifeng Cui, Kim B Olsen, Thomas H Jordan, Kwang-yoon Lee, Jun Zhou, Patrick Small, Daniel Roten, Geoffrey Ely, Dhabaleswar K. Panda, Amit Chourasia, John Levesque, Steven M. Day, and Philip Maechling. Scalable Earthquake Simulation on Petascale Supercomputers. In *2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, number November, pages 1–20. IEEE, nov 2010. ISBN 978-1-4244-7557-5.
- N. Benjamin Erichson, Michael Muehlebach, and Michael W. Mahoney. Physics-informed Autoencoders for Lyapunov-stable Fluid Flow Prediction. *ArXiv e-prints*, may 2019.
- Benjamin Fernando, Kuangdai Leng, and Tarje Nissen-Meyer. Oceanic high-frequency global seismic wave propagation with realistic bathymetry. *Geophysical Journal International*, 222(2):1178–1194, 2020. ISSN 0956-540X.
- Andreas Fichtner. *Full Seismic Waveform Modelling and Inversion*. Springer, 2010. ISBN 978-3-642-15806-3.
- Kasra Hosseini, Karin Sigloch, Maria Tsekhnistrenko, Afshaneh Zaheri, Tarje Nissen-Meyer, and Heiner Igel. Global mantle structure from multifrequency tomography using P, PP and P-diffracted waves. *Geophysical Journal International*, 220(1):96–141, 2019. ISSN 0956-540X.
- Heiner Igel. *Computational seismology: a practical introduction*. Oxford University Press, 2017. ISBN 9780198717409.
- Ameya D. Jagtap, Kenji Kawaguchi, and George Em Karniadakis. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *Journal of Computational Physics*, 404, jun 2020. ISSN 10902716.
- Ratiranjan Jena, Biswajeet Pradhan, Ghassan Beydoun, Abdullah Al-Amri, and Hizir Sofyan. Seismic hazard and risk assessment: a review of state-of-the-art traditional and GIS models, jan 2020. ISSN 18667538.
- Anuj Karpatne, Gowtham Atluri, James H. Faghmous, Michael Steinbach, Arindam Banerjee, Auroop Ganguly, Shashi Shekhar, Nagiza Samatova, and Vipin Kumar. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2318–2331, oct 2017. ISSN 10414347.
- M. F. Kasim, D. Watson-Parris, L. Deaconu, S. Oliver, P. Hatfield, D. H. Froula, G. Gregori, M. Jarvis, S. Khatiwala, J. Korenaga, J. Topp-Muggleton, E. Viezzzer, and S. M. Vinko. Up to two billion times acceleration of scientific simulations with deep neural architecture search. *ArXiv e-prints*, jan 2020.
- Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *ArXiv e-prints*, dec 2014.
- Dimitri Komatsitsch and Roland Martin. An unsplit convolutional perfectly matched layer improved at grazing incidence for the seismic wave equation. *Geophysics*, 72(5):SM155–SM167, 2007.
- Dimitri Komatsitsch and Jeroen Tromp. Introduction to the spectral element method for three-dimensional seismic wave propagation. *Geophysical Journal International*, 139(3):806–822, dec 1999. ISSN 0956540X.
- Kuangdai Leng, Tarje Nissen-Meyer, Martin vanDriel, Kasra Hosseini, and David Al-Attar. AxiSEM3D: broad-band seismic wavefields in 3-D global earth models with undulating discontinuities. *Geophysical Journal International*, 217(3):2125–2146, feb 2019. ISSN 0956-540X.
- Lanbo Liu and Tieshuan Guo. Seismic non-destructive testing on a reinforced concrete bridge column using tomographic imaging techniques. *Journal of Geophysics and Engineering*, 2(1):23–31, mar 2005. ISSN 17422132.
- Guihua Long, Yubo Zhao, and Jun Zou. A temporal fourth-order scheme for the first-order acoustic wave equations. *Geophysical Journal International*, 194(3):1473–1485, sep 2013. ISSN 1365-246X.
- Takuto Maeda and Takashi Furumura. FDM Simulation of Seismic Waves, Ocean Acoustic Waves, and Tsunamis Based on Tsunami-Coupled Equations of Motion. *Pure and Applied Geophysics*, 170(1-2):109–127, nov 2013. ISSN 00334553.
- Gary S. Martin, Robert Wiley, and Kurt J. Marfurt. Marmousi2: An elastic upgrade for Marmousi. *Leading Edge (Tulsa, OK)*, 25(2):156–166, apr 2006. ISSN 1070485X.
- Peter Moczo, Johan O.A. Robertsson, and Leo Eisner. The Finite-Difference Time-Domain Method for Modeling of Seismic Wave Propagation. *Advances in Geophysics*, 48:421–516, 2007. ISSN 00652687.
- Benjamin Moseley, T Nissen-Meyer, and A Markham. Deep learning for fast simulation of seismic waves in complex media. *Solid Earth Discussions, In Review*, pages 1–23, 2020.
- T Nissen-Meyer, M. Van Driel, S C Stähler, K Hosseini, S Hempel, L Auer, A Colombi, and A Fournier. AxiSEM: Broadband 3-D seismic wavefields in axisymmetric media. *Solid Earth*, 5(1):425–445, jun 2014. ISSN 18699529.
- Michela Paganini, Luke De Oliveira, and Benjamin Nachman. Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multilayer Calorimeters. *Physical Review Letters*, 120(4):1–6, 2018. ISSN 10797114.

- PyTorch, 2018. URL <https://www.pytorch.org>.
- M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. ISSN 10902716.
- Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481):1026–1030, feb 2020. ISSN 10959203.
- Stephan Rasp, Michael S. Pritchard, and Pierre Gentine. Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences of the United States of America*, 115(39):9684–9689, sep 2018. ISSN 10916490.
- L Reiter. *Earthquake Hazard Analysis: Issues and insights—Springer Seismic Hazard Assessment: Issues and Alternatives*. Columbia University Press, 1990. ISBN 0231065345.
- Francisco Sahli Costabal, Yibo Yang, Paris Perdikaris, Daniel E. Hurtado, and Ellen Kuhl. Physics-Informed Neural Networks for Cardiac Activation Mapping. *Frontiers in Physics*, 8:42, feb 2020. ISSN 2296424X.
- Gerard T. Schuster. *Seismic Inversion*. Society of Exploration Geophysicists, jan 2017. ISBN 9781560803416.
- Khemraj Shukla, Patricio Clark Di Leoni, James Blackshire, Daniel Sparkman, and George Em Karniadakis. Physics-informed neural network for ultrasound nondestructive quantification of surface breaking cracks. *ArXiv e-prints*, may 2020.
- Ali Siahkoohi, Mathias Louboutin, and Felix J. Herrmann. Neural network augmented wave-equation simulation. *ArXiv e-prints*, sep 2019.
- Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit Neural Representations with Periodic Activation Functions. *ArXiv e-prints*, jun 2020.
- Jonathan D. Smith, Kamyr Azizzadenesheli, and Zachary E. Ross. EikoNet: Solving the Eikonal equation with Deep Neural Networks. *ArXiv e-prints*, mar 2020.
- Simon C. Stähler, Mark P. Panning, Steven D. Vance, Ralph D. Lorenz, Martin van Driel, Tarje Nissen-Meyer, and Sharon Kedar. Seismic Wave Propagation in Icy Ocean Worlds. *Journal of Geophysical Research: Planets*, 123(1):206–232, jan 2018. ISSN 21699100.
- Luning Sun, Han Gao, Shaowu Pan, and Jian Xun Wang. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Computer Methods in Applied Mechanics and Engineering*, 361, jun 2020. ISSN 00457825.
- Albert Tarantola. *Inverse problem theory: methods for data fitting and model parameter estimation*. Elsevier, 1987. ISBN 0444427651.
- TensorFlow, 2015. URL <https://www.tensorflow.org>.
- Andrea Tesoniero, Kuangdai Leng, Maureen D. Long, and Tarje Nissen-Meyer. Full wave sensitivity of SK(K)S phases to arbitrary anisotropy in the upper and lower mantle. *Geophysical Journal International*, 222(1):412–435, 2020. ISSN 0956-540X.
- Martin van Driel and Tarje Nissen-Meyer. Optimized viscoelastic wave propagation for weakly dissipative media. *Geophysical Journal International*, 199(2):1078–1093, sep 2014. ISSN 0956-540X.
- J. Virieux and S. Operto. An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74(6), dec 2009. ISSN 00168033.
- Yiran Xu, Jingye Li, and Xiaohong Chen. Physics informed neural networks for velocity inversion. In *SEG International Exposition and Annual Meeting 2019*, pages 2584–2588. Society of Exploration Geophysicists, 2020.
- Liu Yang, Xuhui Meng, and George Em Karniadakis. B-PINNs: Bayesian Physics-Informed Neural Networks for Forward and Inverse PDE Problems with Noisy Data. *ArXiv e-prints*, mar 2020.
- Yibo Yang and Paris Perdikaris. Adversarial uncertainty quantification in physics-informed neural networks. *Journal of Computational Physics*, 394:136–152, nov 2019. ISSN 10902716.
- Ruiyang Zhang, Yang Liu, and Hao Sun. Physics-guided convolutional neural network (PhyCNN) for data-driven seismic response modeling. *Engineering Structures*, 215, jul 2020. ISSN 18737323.
- Zhongping Zhang and Youzuo Lin. Data-driven Seismic Waveform Inversion: A Study on the Robustness and Generalization. *ArXiv e-prints*, pages 1–13, 2018.
- Yinhao Zhu, Nicholas Zabaras, Phaedon Stelios Koutsourelakis, and Paris Perdikaris. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *Journal of Computational Physics*, 394:56–81, jan 2019. ISSN 10902716.

Appendix A: Supplementary figures

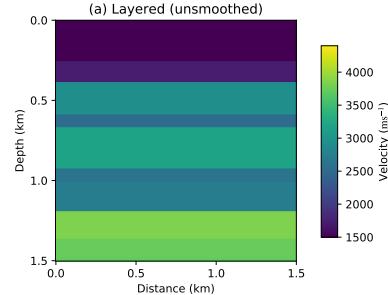


FIG. A1. Layered velocity model without smoothing, used in Figure A2.

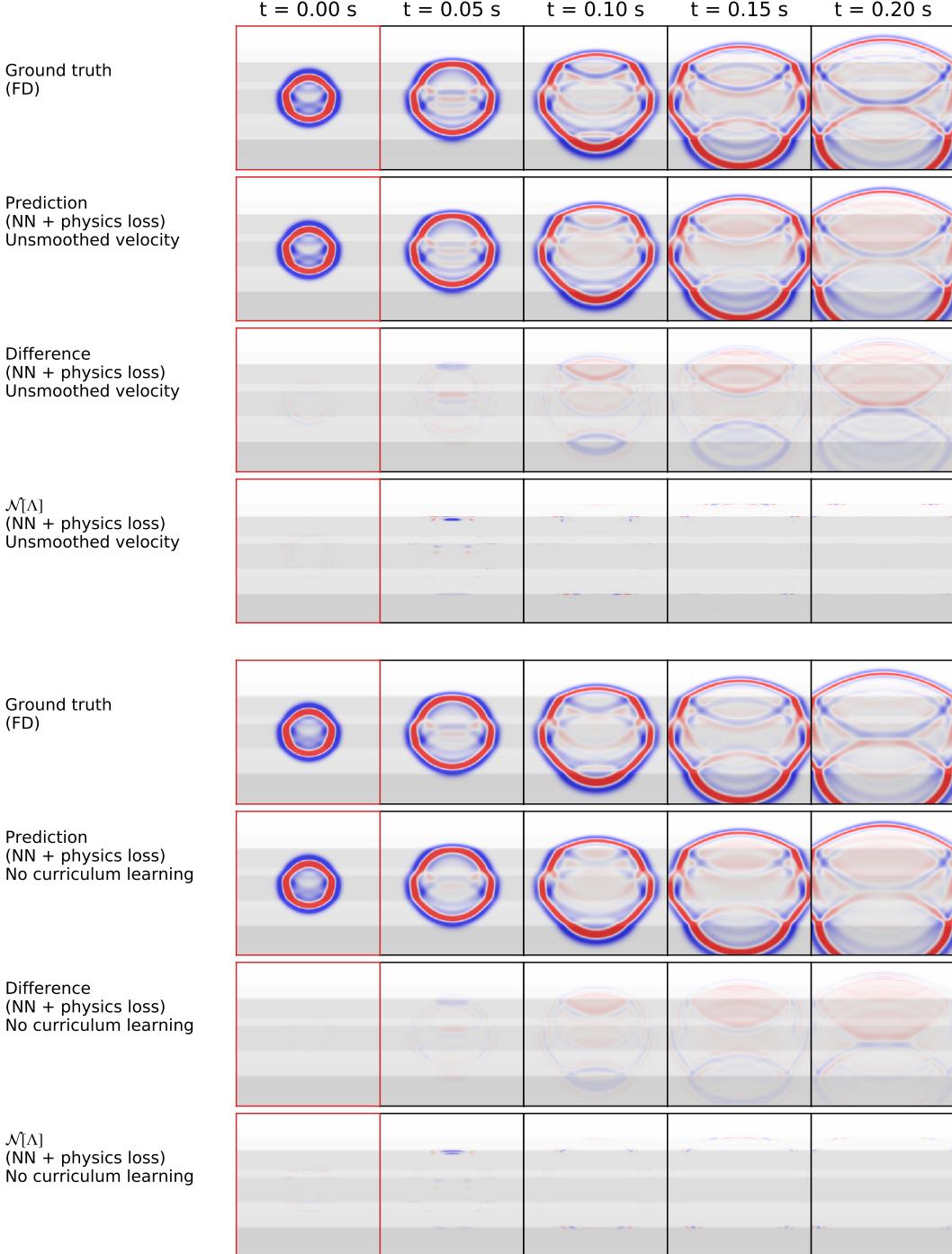


FIG. A2. Comparison of the PINN wavefield prediction to ground truth FD simulation for two additional PINN training setups. In the first test the layered velocity model is used without smoothing (“Unsmoothed velocity”), shown in Figure A1. In the second test curriculum learning is not used; that is, the physics loss is used from the start of training and an expanding time horizon is not used (i.e. t_j values for computing the physics loss in Equation 5 are sampled from their full input range $[0, T_{\max}]$ from the start of training). Top four rows show the results from the first test and bottom four rows show the results from the second test. For each test, top two rows show the FD and PINN wavefield solutions through time. Third row shows the difference of the network prediction to FD simulation. Fourth row shows the value of the differential operator in the physics loss in Equation 5 (right hand term), which is close to zero for an accurate solution to the wave equation. Plots bordered in black indicate wavefield times which are outside of the boundary training data. Plots are overlaid on the velocity model, shown in grey. Larger errors can be seen in both cases compared to the PINN in Figure 4.