Imperial College London

Department of Earth Science and Engineering

MSc in Applied Computational Science and Engineering

Independent Research Project
Final Report

Seismic Imaging on the Edge: Full Waveform Inversion on a Jetson Orin Nano for Autonomous Robotic Subsurface Exploration

by

Miraz Badrais

Email: miraz.badrais24@imperial.ac.uk
GitHub username: esemsc-mab124
Repository: https://github.com/ese-ada-lovelace-2024/irp-mab124.git

Supervisors:
Dr Ban-Sok Shin, Dr. Edward Caunt

29th August 2025

Contents

List of Tables

List of Figures

Abstract

This paper extends the ATC-FWI implementation on Nvidia Jetson Orin Nano Super with CUDA utilising DevitoPRO. A decentralised multi-agent network is designed such that each agent manages its local model, computes gradients from its own receivers, and exchanges information with neighbours in a two-phase Adapt-Then-Combine (ATC) structure. Four variations of ATC-FWI network are enforced by agents' distributing strategy and neighbouring strategy. These variations illustrate the effect of the seismic array and non-sequential wave propagation. To exploit parallelisation capabilities, Dask was integrated into the Python-level code in two different ways, per-shot and per-agent, enabling accelerated CPU runtime, in addition to HPC support. After stabilising the ATC network in terms of both result quality and runtime efficiency, the ATC-FWI has been ported to Jetson with CUDA utilising DevitoPRO's streamed CPU–GPU execution. Although 2D CUDA runs show longer wall-times than CPU due to data-movement overheads at small problem sizes, the results confirm correctness and pave the way for 3D, where higher arithmetic intensity should better utilise the Graphics Processing Unit (GPU). Overall, the proposed ATC-FWI pipeline enables in-situ seismic imaging on resource-constrained agents and supports scalable, cooperative subsurface exploration without a central node.

1 Introduction

The exploration of the Moon and Mars has long interested scientists due to their potential to be the future destinations for human missions beyond Earth's orbit [1]. One primary goal of exploration is target science, which determine a body's physical properties, history, origin, and suitability for future human bases. Several types of equipment with different configurations have been used in previous planetary missions addressing target science.

The Apollo 14, 16 and 17 missions deployed small arrays of geophones on the moon, which documented a globally uniform, highly porous, brecciated regolith with very low near-surface P-wave velocities [2]. Despite their large scientific value, these data were constrained by the sparse distribution of the deployed geophones. Alternatively, InSight mission deployed a single broadband seismometer placed on the surface, enabling interior mapping of Mars [3,4]. However, the single seismic station limits lateral resolution and source localisation.

The limitations of Apollo and InSight missions highlight the need for long-duration recording from more spatial coverage in future planetary landers. However, crewed missions imply high cost, risk, and communication delays, which are unsustainable for long-duration missions [5,6]. One solution is swarm based systems [7], a multi-agent robotic network with a decentralised wireless system that serves both communication and localisation via radio signal. In this network, the agents can share resources, process data, and make inferences over the network mimicking biological swarms behaviour in nature. The decentralised, multi-agent approach is an instance of distributed sensing, where a spatially dispersed network of sensors collaborates to achieve higher resolution, redundancy, and autonomy than a single station could provide [8]. Beyond planetary science, multi-agent networks are already deployed in search-and-rescue [9] and cooperative autonomous driving simulations and decision-making frameworks [10].

By distributing multiple agents across the planetary surface, the network can expand the local spatial coverage, improve lateral resolution, and reduce the risk of single-point failures [11]. Mainly since the decentralised scheme of the network eliminates dependence on a single central node: if one agent fails, the network can continue functioning. This fault tolerance is particularly important since planetary missions' environments might be uncertain and face the difficulty of on-site repair. Similar to crewed missions, communication delays influence the Earth-based control negatively with such a network; therefore, autonomous subsurface imaging techniques are important as onboard autonomy mitigates the impact of these delays, while allowing mission continuity without constant human oversight [12].

Seismic imaging methods are used to explore the subsurface [13]. One widely used technique is Full Waveform Inversion (FWI), an iterative computational data-fitting scheme that produces high-resolution models of physical properties using the full wavefield [14]. FWI was initially introduced in [15] using the acoustic wave approximation to solve the non-linear inverse problem of seismic imaging. FWI generates synthetic data with an initial guess model, then iteratively minimises the misfit between the synthetic and observed data with respect to specific model parameters m [14]. FWI uses the entire wavefield data rather than just the first arrival time as in the travel-time tomography [16]. This enable higher resolution and the recovery of fine-scale structures, showing potential targets such as caves or lava tubes, mineral reserves and hydrocarbon reservoirs. Thus, FWI is one of the most powerful and increasingly dominant imaging techniques. A typical choice of the misfit function in FWI is the least-squares L_2 norm.

$$\ell(m) = \frac{1}{2}||d_{obs} - d_{syn}(m)||^2 \tag{1}$$

Where d_{obs} is the observed data and d_{syn} is the generated synthetic data from the initial guess model. The gradient of the misfit function using the adjoint-state method is then:

$$\omega(x) = \nabla_m \ell = -\left(\frac{2}{m^3(x)}\right) \sum_{s=1}^{N_s} \int_0^T q_s(x, T-t) \, \frac{\partial^2 u_s(x,t)}{\partial t^2} \, dt. \tag{2}$$

Where $u_s(x,t)$ is the forward wavefield, at time t and position x, generated by injecting shot s that travels through a subsurface with using the velocity model m and $q_s(x,t)$ is the adjoint wavefield generated by injecting the residual at the receiver position x per shot s. The model update is then:

$$m^{k+1} = m^k + \alpha\omega \tag{3}$$

Where $\alpha > 0$ is the step size. This gradient-descent update with α is usually used to simplify the Newton method [14, 17].

The applications of FWI are not limited to geophysics [18–20]; it is also used in medical ultrasound imaging of the human brain [21] and breast [22]. Notably, FWI relies on a centralised scheme, which conflicts with the decentralised scheme of multi-agent robotic systems. Therefore, a decentralised and distributed subsurface imaging technique is required.

[23] proposed ATC-FWI to fulfil this requirement by combining the ATC approach from [24] with the FWI. Instead of relying on a central node to collect all agents' data, hold the subsurface model, and perform FWI, the ATC-FWI allows each agent to hold a local subsurface model, perform FWI, and then exchange information with its neighbours. The algorithm consists of two phases: a partial local model update performed by each agent, and a combination step with its neighbouring agents, in detail:

- **Adapt Phase**: This phase is a partial update of the agent's local model with the neighbouring agents' models. It is very similar to the FWI equation (3), except that a weighted average of the neighbouring gradients is incorporated, hence, the partial update. In detail, an intermediate model $\widetilde{m}_j^{k+1}$, per agent j and iteration k, holds the agent's local model m_j^k update with the weighted average of the neighbouring gradients multiplied by the step size $\alpha > 0$.

  $$\widetilde{m}_j^{k+1} = m_j^k + \alpha^k \sum_{i \in \mathcal{J}_j} c_{ij}\omega_i^k \tag{4}$$

  The misfit here considers the agent's receivers only. [23] has a single-receiver as an agent; therefore, the misfit function is:

  $$\ell(m)_j = \frac{1}{2}||d_{obs,j} - d_{syn,j}(m)||^2 \tag{5}$$

  Where $d_{obs,j}$ is the observed data per agent j, $d_{syn,j}$ is the generated synthetic data from the initial guess model per agent j.

Subsequently, [23] compute the gradient by the adjoint-state method, similar to the FWI equation (2) as:

$$\omega_j(x) = -\left(\frac{2}{m^3(x)}\right)\sum_{s=1}^{S}\int_0^T q_{j,s}(x, T-t)\,\frac{\partial^2 u_{j,s}(x,t)}{\partial t^2}\,dt. \tag{6}$$

Where $u_{j,s}(x,t)$ is the agent's j receiver-specific forward-solved wavefield, at time t and position x, generated by injecting shot s that travels through a subsurface with using the velocity model m and $q_{j,s}(x,t)$ is the agent's j receiver-specific adjoint wavefield generated by injecting the local agent receiver-specific residual at the receiver position x_j per shot s.

- **Combine Phase**: This phase is where the actual update of the agents takes place by a weighted average of the intermediate model $\widetilde{m}$.

$$m_j^{k+1} = \sum_{i\in\mathcal{J}_j} a_{ij}\,\tilde{m}_i^{k+1}. \tag{7}$$

The coefficients a and c determine how the agent neighbours are weighted in the adopt and combine phases. In [23] case, $a_{ij} = c_{ij} = \frac{1}{|J_j|} \iff i \in J_j$.

Moreover, [23] demonstrate that the gradient of FWI is theoretically similar to the sum of locally computed gradients at each agent in multi-agent networks. However, each agent has a unique subsurface model that simulate a specific region of the domain. This produce a similar inversion results as a centralised approach, yet localised to agents' receiver coordinates only. This In the context of planetary exploration, a decentralised multi-agent system can achieve high-resolution seismic imaging autonomously and at scale, without requiring centralisation or Earth-based communication.

This finding is further supported by [25], which demonstrates how a Hardware-in-the-Loop (HIL) system conducts a 3D seismic exploration. In this system, a network of rovers, each of which carries a Raspberry Pi 4 (RPi4), collaboratively reconstructs the subsurface image using the 3D ATC-FWI. Such a distributed design demonstrates the potential of swarm-based geophysical exploration, where a number of low-cost instruments collaborate to perform a computationally demanding task. Compared to 2D imaging, this 3D approach enables broader data acquisition while supporting real 3D wave physics, such as out-of-plane scattering and diffraction by including the third spatial dimension; thus, resulting in a more detailed subsurface image [22]. The RPi4 is highly affordable and power-efficient, with 5 V / 3 A specifications, making it suitable for general embedded projects, prototyping, and low-power applications. However, the RPi4's quad-core Cortex-A72 CPU handles all calculations, resulting in slow performance when handling the computationally demanding 3D ATC-FWI.

One solution is the utilisation of a GPU to handle 3D ATC-FWI calculation. The GPU is a parallel hardware accelerator originally designed for rendering graphics, with early general-purpose computing capabilities introduced through CUDA [26, 27]. In seismic modelling methods such as FWI and ATC-FWI, the iterative wavefield propagation and gradient updates involve large-scale operations which are inherently data-parallel, considering the finite-difference kernels method in the propagation and gradient: the same stencil-based update is applied repeatedly across a spatial grid [28, 29]. Therefore, a GPU can accelerate the wavefield propagation and gradient updates by executing the identical tasks in parallel. This acceleration in gradient calculation is crucial for future planetary missions that utilise autonomous rovers or distributed sensor networks, as heavy computational tasks, primarily ATC-FWI, are performed locally under strict power and resource constraints in an edge environment. Additionally, the fast computation implies expedited decision-making, thus increasing data collection within the mission time window

The recent demand from machine learning and deep learning, which rely on parallelisable heavy computation as matrix and vector operations, has been leading the innovation of powerful GPU processors, making them equally valuable for scientific applications such as FWI. One of the recent GPU developments is the Nvidia Jetson Orin Nano Super [30], a heterogeneous system-on-chip designed with a wide range of computing capabilities to fulfil a variety of applications, such as modern robotics, autonomous vehicles, drones, and IoT systems [31]. These applications often demand cloud resources that are constrained by latency and limited bandwidth; thus, the requirement for fast local computation at the edge is increasingly common [32]. Unlike RPi4, Jetson Orin integrates an Ampere GPU with CUDA and Tensor Cores, resulting in higher performance according to [32]. The low power consumption of Jetson Orin Nano Super compared to Jetson NX and AGX [31], along with the fact that the same CUDA software ecosystem and TensorRT are utilised, makes it suitable for deployment in distributed, power-constrained field settings as rovers and multi-agent networks.

This paper's objective is to extend the ATC-FWI implementation on Nvidia Jetson Orin Nano Super processors within the Devito framework. The aim is to exploit Jetson's computation and parallelisation features so that agents could process seismic data in situ, supporting autonomous decision-making and broader data collection in future planetary missions that utilise multi-agent networks.

2 Methodology

To perform FWI computation for wave propagation and gradient, Devito is utilised. It is domain-specific language and compiler for finite difference computations. The compiler performs stencil-based computation on structured grids by converting the symbolic specification `SymPy` of the mathematical problem, e.g. isotropic-acoustic wave equation and gradient, into optimized C-level code [33, 34].

2.1 Experiment Description and Domain Setup

According to [23], FWI and ATC-FWI global convergence are expected to be comparable. Thereby, several experiments were defined to evaluate the ATC-FWI implementation under a controlled, yet geophysically meaningful synthetic setting. There are three main experiments:

- FWI and ATC-FWI network variations on CPU with Devito.
- CPU parallelisation via Dask for ATC-FWI.
- FWI and best CPU ATC-FWI variation on Jetson with DevitoPRO.

These experiments set up is motivated by mimicking realistic exploration surveys, while maintaining an reasonably inversionable ground-truth velocity as the primary focus of this paper is the implementation of ATC-FWI on Jetson rather than reproducing accurate geophysical properties.

The domain is defined as a $3\,\text{km}\times 1\,\text{km}$ subsurface area discretised with a uniform grid spacing of $10\,\text{m}$ in both dimensions and absorbing boundaries with 20 grid cells. This wide-aperture setup for the domain allows recovering of medium-to-long wavelength structure, while the highest frequency caps fine detail at about half a wavelength according to [14, 35]. Since [23] results form a base for the ATC-FWI, it was also considered during the domain setup to facilitate comparison.

The ground truth background velocity model consists of five horizontal layers, with P-wave velocities ranging $2\,\text{km/s}$ to $4\,\text{km/s}$ in increments of $0.5\,\text{km/s}$ per layer. This is a simple layer medium with increasing velocities, motivated by the generic planetary interior of the Earth, Moon, or Mars. To add some complexity to the model, two irregular anomalies are embedded at the centre of the model, each assigned a velocity 50% higher than the background mean. The initial velocity model smooths the ground truth model by applying a Gaussian filter with standard deviation $\sigma = 10$. The space order for discretisation is 16 and the time order for discretisation is 2.

For the exploration survey, the sources and receivers are evenly distributed on a line $10\,\text{m}$ below the surface. This setup prevents false reflection with the absorbing layers, while keeping both receivers and sources at the same depth. The sources are a Ricker wavelet with peak frequency $f = 15\,\text{Hz}$, and the total observation time is $T = 1\,\text{s}$. Additionally, most experiments used shots $S = 40$ and receivers $R = 80$ geometry. Since the higher number of sources/receivers implies better results. If the experiment involved fewer or extra sources/receivers, it is mentioned within its detailed subsection.

Figure 1 shows the ground truth and smooth models, with red dots denoting sources and green diamonds representing receivers.

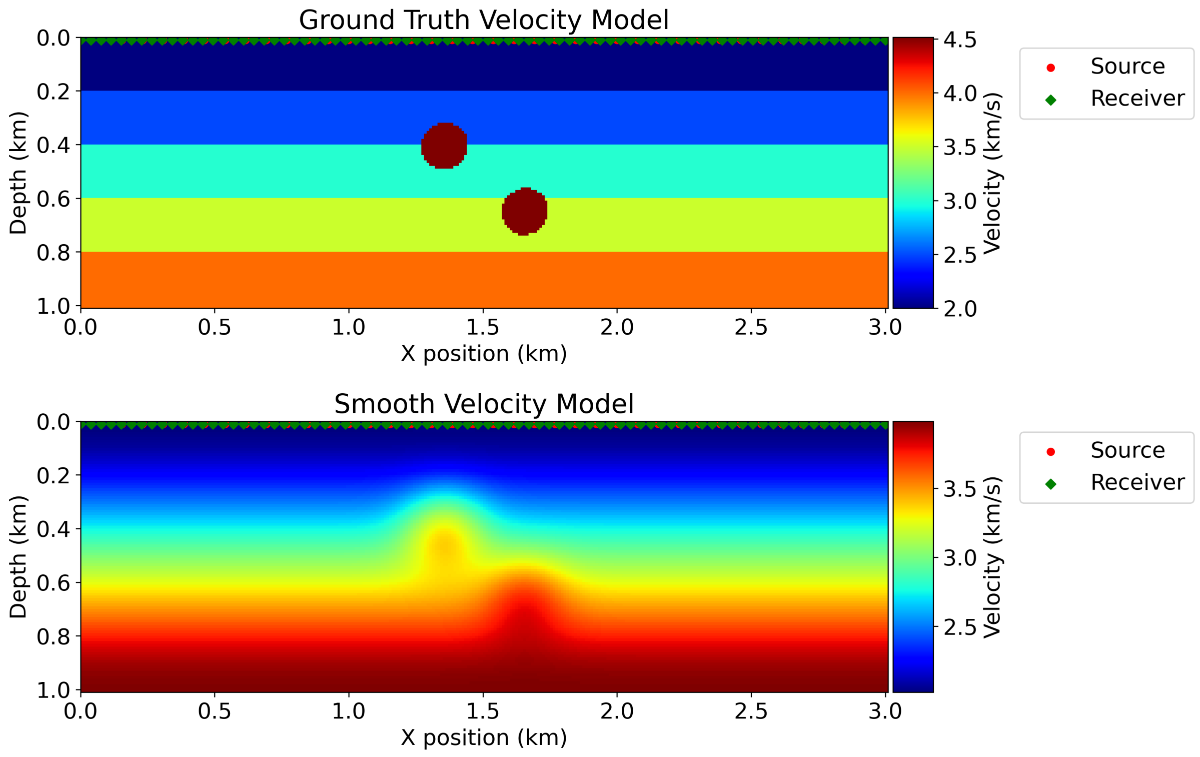


Figure 1: Ground Truth Velocity Model and Smooth Velocity Model

2.2 Generating Ground True Data

To ensure a fair comparison between FWI and ATC-FWI, a True Shots class was implemented, specifically to encapsulate the true data generation. This class consist of the following:

- A dynamic horizontal layered velocity function to mimic the planet's layered structure by generating the background model velocity. It constructs velocity layers based on the number of layers, an initial P-wave velocity (v_p), and a specified step size. These variables are editable, hence aiding in generating different ground truth models.
- Three functions were implemented to embed different irregular anomalies into the v_p model. The main function eventually chosen for this study places two relatively small anomalies at the model centre with higher velocity relative to the background mean (positive velocity contrast). The small anomalies minimise surface reflections, therefore allowing the wave to propagate and reflect from deeper layers. The anomalies' integration was initially motivated by the idea of lava tubes; however, the typical lava tubes are low-velocity voids or bounded structures which hinder FWI convergence. Therefore, it was replaced with this simple convention of high-velocity anomalies, as the primary focus of this paper is the implementation of ATC-FWI on Jetson rather than reproducing accurate geophysical properties.
- A function was implemented to uniformly distribute sources\shots $s = 40$ and receivers $J = 80$ just below the surface by $10\,\text{m}$ to minimise false reflection with the absorbing layers. These acquisition

geometries are used for both FWI and ATC-FWI experiments.

- A true shot generator was implemented to generate and export the simulated wave fields into NumPy arrays for reuse in both FWI and ATC-FWI experiments.

2.3 FWI Implementation Specifications

The FWI serves as a benchmark for evaluating the performance of ATC-FWI since the global convergence of FWI and ATC-FWI are expected to be comparable [23]. To ensure a correct implementation, the Devito tutorials and examples were extensively consulted, resulting in the following setup:

- A smooth model creator that takes the TrueShots v_p, which include the horizontal layered background and two small fixed anomalies with higher velocity than the background mean, and smooths it using a Gaussian filter with standard deviation $\sigma = 10$.
- Take the TrueShots sources and receivers coordinate and initialise the Acquisition Geometry and Solver.
- A true-shot interpolator to align the true shots with the smooth model time step.
- A gradient computation function that loops over the shots to compute the forward wavefield, the residual (misfit), and the residual back-propagation to obtain the gradient $\omega(x)$.
- FWI function that update the model iteratively based on equation 3.
- The step size is adaptively scaled according to $\alpha = \min\left(\alpha,\ \frac{0.08}{\max(|\omega|)}\right)$, ensuring smaller gradients with each iteration, which implies more stable convergence.

2.4 AGENT Implementation Specifications

In [23], each agent is a single receiver; however, seismic arrays are advantageous since the summation of the individual array recordings improves the signal-to-noise ratio, enabling the study of seismic phases and small-scale structure of the interior, which is usually challenging to achieve with single receiver convention [36]. Also, the seismic array reduces overall computational cost for ATC-FWI on a given number of receivers. Therefore, in this paper, each agent consists of successive blocks of receivers based on the network split distribution (more detail are provided in Section 3.1).

The ATC-FWI network gives each agent a unique identifier (ID), an initial guess model, a receiver collection, and the corresponding true shots. Then, each agent performs the gradient computation $\omega(x)$, the adapt phase and the combine phase.

In [23], each agent exchanges its gradient with a set of 3-6 neighbours. Accordingly, in this paper each agent defines its neighbours using one of the following strategies:

- Non-Random neighbouring: each agent sets a successive block of neighbours based on its ID, clipped at the domain edges without wrap-around. Since the agents neighbours are assigned sequentially, the network will be connected.
- Random neighbouring: each agent sets a randomly sampled block of neighbours from the agent's pool, clipped at the domain edges with wrap-around. Since receivers are evenly distributed on a line, the previous neighbour to the agent is always included to ensure network connectivity. The motivation of this function is to investigate the randomness effect of ATC-FWI network.

2.5 ATC-FWI Network Implementation Specifications

The ATC-FWI network class defines the network of agents. Before initialising any agent, a graph is built by linking all receivers to ensure network connectivity. Then, the number of agents is chosen using one of the following strategies:

- Balanced distribution: distribute the receivers R evenly among agents J while minimising the difference between the number of agents and the number of receivers per agent $|J - (R/J)|$. The motivation is to expedite the FWI iteration computation by decreasing the number of agents, while examining the seismic array effect by increasing the number of receivers per agent.
- Maximum agents distribution: distribute the receivers R evenly among the maximum possible number of agents J such that R is divisible by J without remainder. The motivation is to minimise the computational effort per agent while maintaining a nearly single-receiver distribution.

The base ATC-FWI iteration consists of three loops over the agents; the first loop computes each agent's gradient $\omega(x)$, the second loop perform the adopt phase using the same step size of FWI as $\alpha = \min\left(\alpha, \frac{0.08}{\max(|\omega|)}\right)$, and a final loop to perform the combine phase.

`Dask`, a Python library for parallel and distributed computing, was intergraded to reduce the runtime at python codes level [37] `Dask` was utilised in two different ways, resulting in the following modifications:

- Instead of having a loop over the shots to compute the gradient, a distributed `submit- wait` workflow was implemented with `Dask`. The `submit` schedule a separate task for each shot, then these tasks compute the agent's gradient simultaneously for multiple shots based on half of the CPU cores. The `wait` ensure all the shots gradient are computed before moving forward with next agent.
- Instead of having a loop over the agents to compute their respective gradient, a distributed `scatter-map gather` workflow was implemented with `Dask` to evaluate the gradient. First the network is broadcast over the number of workers by `scatter`, then `map` compute the agent's gradient simultaneously for multiple agents based on half of the CPU cores, finally a `gather` to get back the gradient to the client.

2.6 Jetson Implementation Modifications

Data streaming means efficient data movement between computation nodes. Usually, a node consists of a host, a disk, and a device, which is the case with the Jetson. Data streaming is critical to utilise the node computation capabilities by overlapping data transfer with computation, therefore, keeping the GPU busy and enabling efficient, real-time processing without idle cycles [38] Until this point, the FWI and ATC-FWI utilised Devito utilities, although different compilers and architectures are supported by Devito package, the data streaming is not taking into consideration. Alternatively, DevitoPRO utilities was used to extend the FWI and ATC-FWI on the Jetson, which supports data streaming. The following modifications were necessary to enable DevitoPRO utilisation:

- The Acquisition Geometry and Acoustic Wave Solver of Devito were replaced with Acoustic Isotropic. This modification implied a sign change in residual computation.
- For better utilisation of DevitoPRO utilities, the Seismic Model of Devito changed to Wave Model, keeping the buoyancy $b = 1$ for constant density, since density is not the focus of this paper.

3 Results and Discussion

Experiments and domain setup described in Section **??** ran for $k = 20$ iterations each. The results are presented in the following subsections.

It is worth mentioning that the analyses in Subsections 3.1, **??**, and 3.2 are based on Devito implementation on CPU; however, this analysis is still valid even after porting to the Jetson with the DevitoPRO utilities due to the following:

- The neighbouring and agents' distribution strategies are based on the agents' IDs and the number of agents. None of these functions uses any utilities from Devito; therefore, the analysis remains valid when utilising DevitoPRO on the Jetson.
- Dask supports both CPU and GPU clusters; therefore, theoretically speaking, the analysis regarding which strategy is suitable for long-term runs and investigations is still valid, although the computation time might differ.

3.1 Agents Distribution and Neighbouring Strategy

As detailed in Section 2.4 and 2.5, four strategy were implemented that prompt four variations of ATC-FWI network:

- ATC-FWI with balanced agents distribution and non-random neighbouring.
- ATC-FWI with balanced distribution and random neighbouring.
- ATC-FWI with maximum agents distribution and non-random neighbouring.
- ATC-FWI with maximum agents distribution and random neighbouring.

The balanced distribution sets ATC-FWI network with $J = 8$ agents, each with $R = 10$ receivers. In comparison, the maximum agents distribution sets it to $J = 40$ agents, each with $R = 2$ receivers.

Figure 2 shows the convergence of FWI alongside the global convergence of ATC-FWI networks. The balanced distribution follows a similar global gradient trend to FWI, outperforming the maximum agents distribution. Across both neighbouring strategies, non-random neighbouring outperformed the random strategy quantitatively in terms of global convergence. Interestingly, the neighbouring strategies have a minimal effect on balanced distribution convergence. This behaviour is mostly attributed to the large number of receivers per agent, which considers a higher number of wave propagation corresponding to the agent's receivers, mimicking seismic array behaviour, where the signal-to-noise ratio enhances global convergence and stacks out the source-receiver imprint.

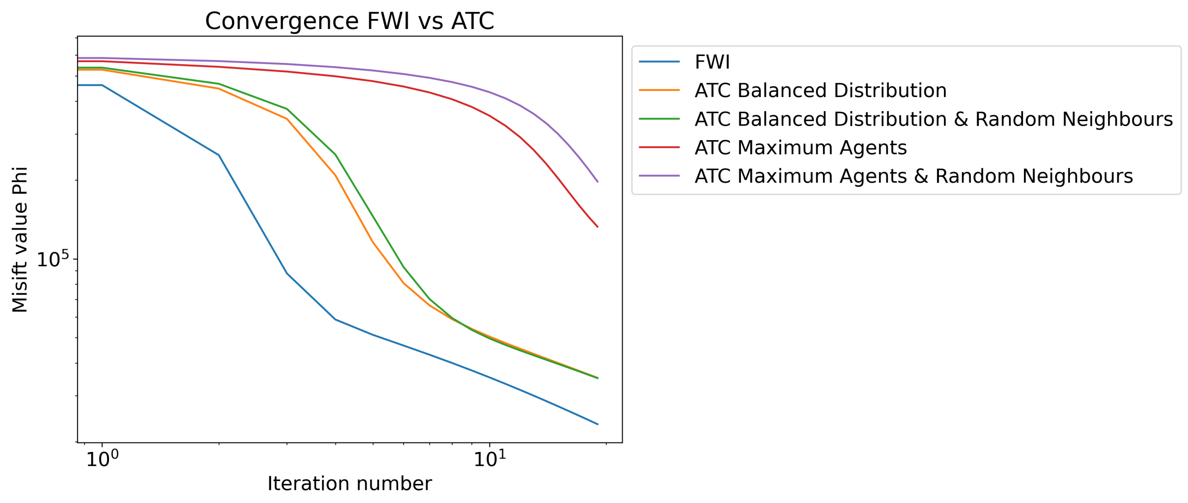


Figure 2: FWI vs. ATC-FWI network global converging

Figure 3 shows the convergence of each agent in different ATC-FWI networks. The balanced distribution starts from a relatively high objective value but converges rapidly in fewer than 10 iterations, whereas the maximum agents distribution starts from a lower objective value but converges more slowly. As it is the case with the global convergence, the neighbouring strategy has minimal effect on the local convergence trend on both distributions.

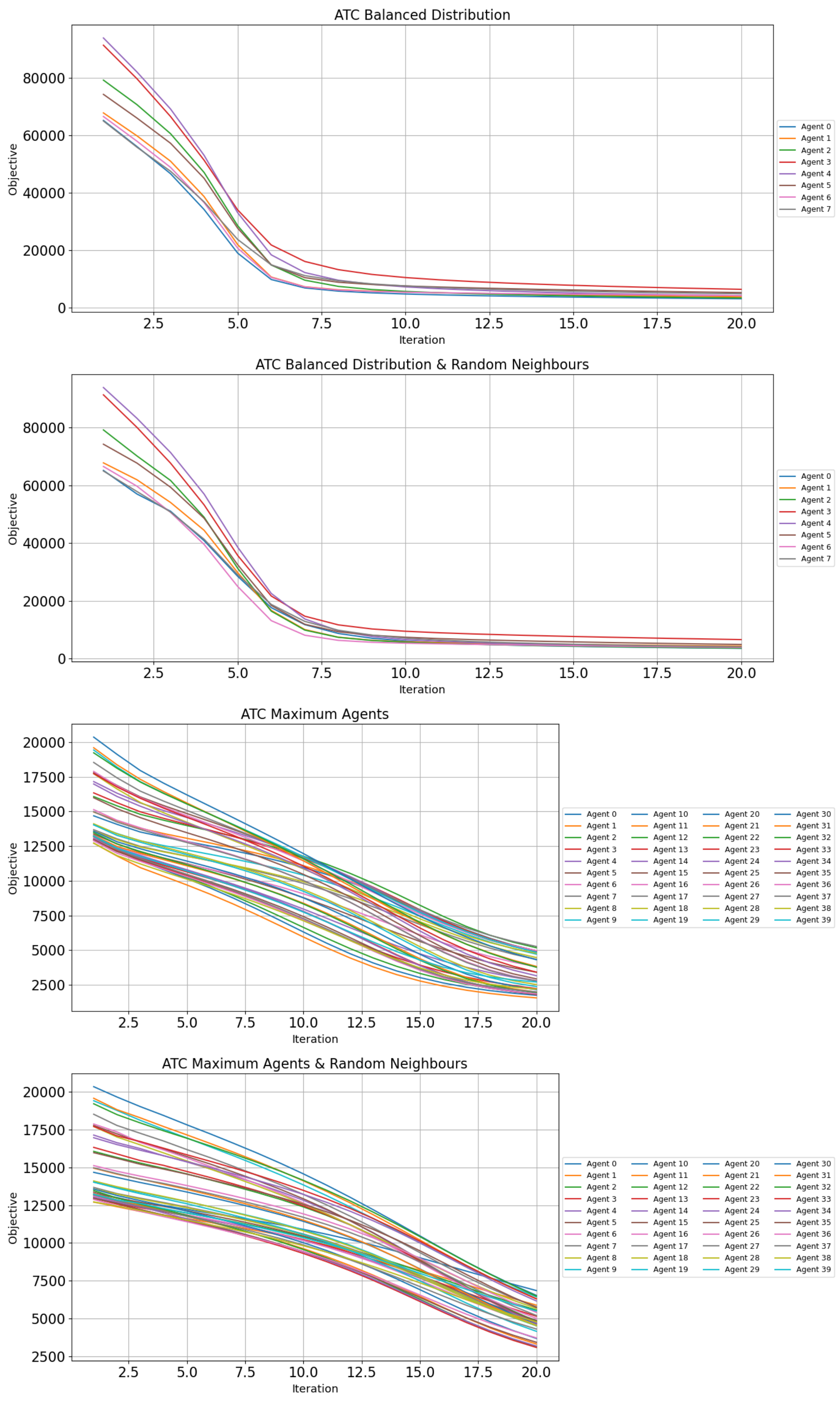


Figure 3: ATC-FWI Agent's converging

Figure 4 shows that all ATC-FWI variation resulted in sharpened layered, particularly, the first layer. The balanced distribution variation, generally produced a more comprehensive final models despite the neighbouring strategy. The upper anomaly is more framed while few to no artefacts are produced around the edge agents as annotated by red circles on the same figure. The improved models quality in the variation can again be attributed to the higher receivers number per agent, which mimics the seismic array effect and enhances robustness.

However, the maximum agent distribution shows different final models based on the neighbouring strategy as the green boxes shows in Figure 4. The non-random neighbouring resulted in more detailed, yet skewed, final models compared to random neighbours, which led to some artefacts between the agents. This behaviour can be attributed to the allowance of sequential data propagation between the neighbours with the non-random strategy. Particularly noticeable with edge agents that have fewer neighbours, typically three, resulting in more localised skewed area-specific reflections $\iff$ the number of agents is high and the number of receivers per agent is low. Subsequently, this requires more iteration to achieve global convergence, similar to the FWI, which explains the high trend in the maximum agents' distribution.

The runtime comparison in Figure 5 shows that the fewer number of agents in balanced distribution required generally shorter runtime, despite neighbouring strategy, in comparison to maximum agents distribution; therefore, theoretically speaking, less computational effort. This is an expected behaviour, as a fewer number of agents imply less gradients loop; therefore, shorter runtime. However, maximum agent distribution with the non-random neighbouring strategy was slower than the random. One reason for such unexpected behaviour might the sequential data propagation over the network.

After evaluating the strategies effects, the balanced distribution produced high-resolution final models with minimal artefacts in relatively short runtime with a global convergence similar to FWI,therefore it is selected as the default option when initialising ATC-FWI network.

As for the neighbouring, the non-random strategy outperformed the random method in global convergence, especially with maximum agent distribution. However, when the number of receivers per agent is large, as in the balanced distribution, the neighbouring strategy effect is minimal due to seismic array behaviour. Therefore, to serve both distributions, the non-random strategy is adopted as the default option since it resulted in more detailed final models.

Therefore, the best variation of ATC-FWI is balanced distribution with non-random neighbouring. Figure 6 shows the FWI vs. agent 1, 4, and 8 of ATC-FWI network best variation. As expected, the results are very similar even within the different agents of ATC-FWI network. Table 1 quantifies this similarity using the L2 norm and SSIM between the ATC-FWI agents and FWI. Notably, the L2 norm difference between FWI and agents of ATC-FWI is very small L2 norms range from 2.64–3.30 and SSIM values consistently exceed 0.992 across agents 1, 4, and 8. This also support the observation of seismic array effect.

Comparison	**L2**	**SSIM**
ATC-FWI Balanced Agent 1 vs Ground Truth	537.657737	0.695587
ATC-FWI Balanced Agent 1 vs FWI	3.299054	0.992639
ATC-FWI Balanced Agent 4 vs Ground Truth	537.286409	0.695977
ATC-FWI Balanced Agent 4 vs FWI	3.047676	0.993259
ATC-FWI Balanced Agent 8 vs Ground Truth	536.813918	0.696669
ATC-FWI Balanced Agent 8 vs FWI	2.639143	0.994325
FWI CPU vs Ground Truth	501.308569	0.719331
FWI CUDA vs Ground Truth	552.393185	0.690589
FWI CPU vs FWI CUDA	4.350088	0.991572

Table 1: L2 error and SSIM comparison between CPU, CUDA, and ATC-FWI Balanced Agents against the ground truth and baseline FWI.

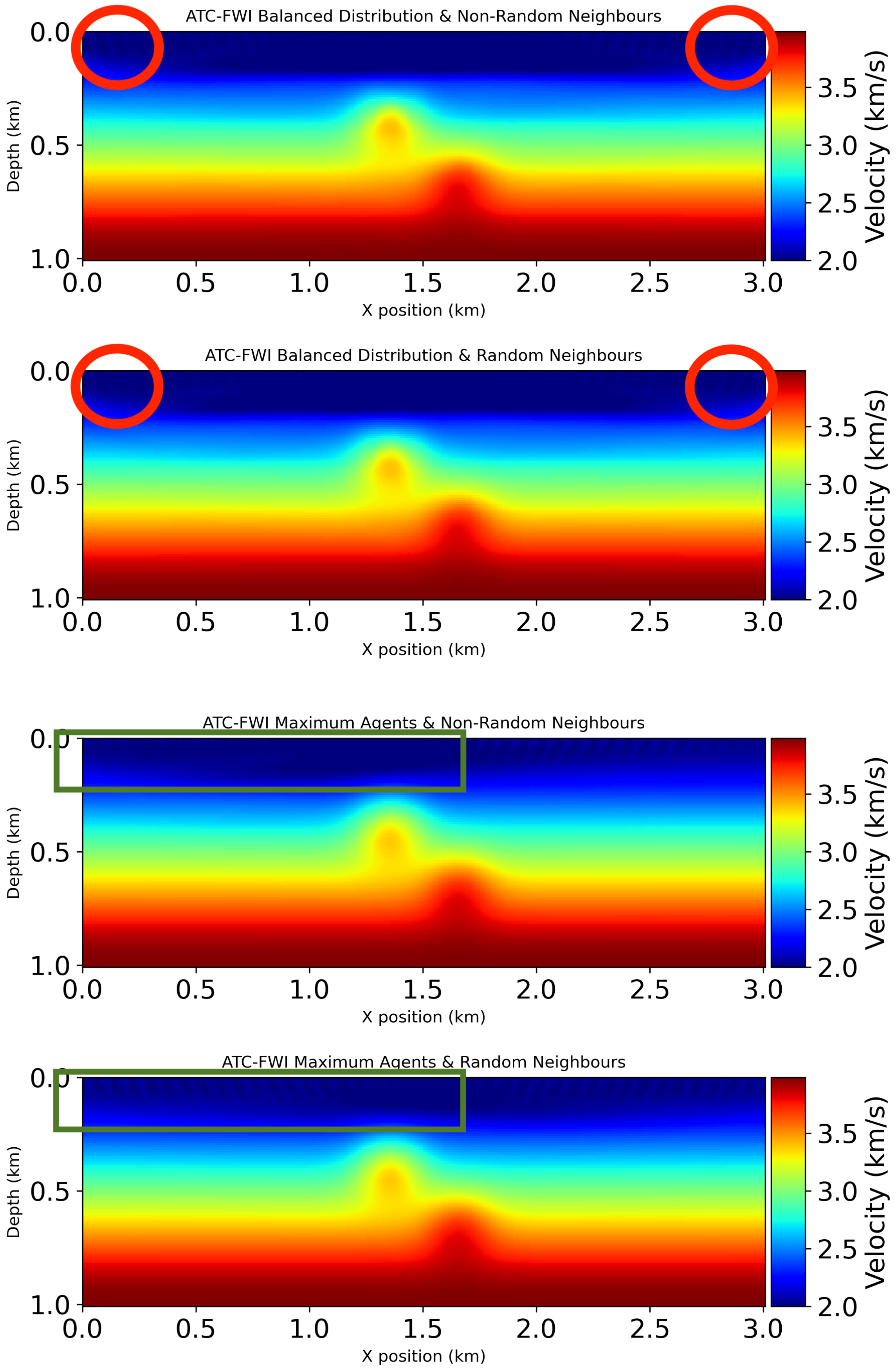


Figure 4: ATC-FWI final models of Agent ID 6. Annotated by red circles on balanced distribution edges vs. green boxes on maximum agents distribution skewed reflection

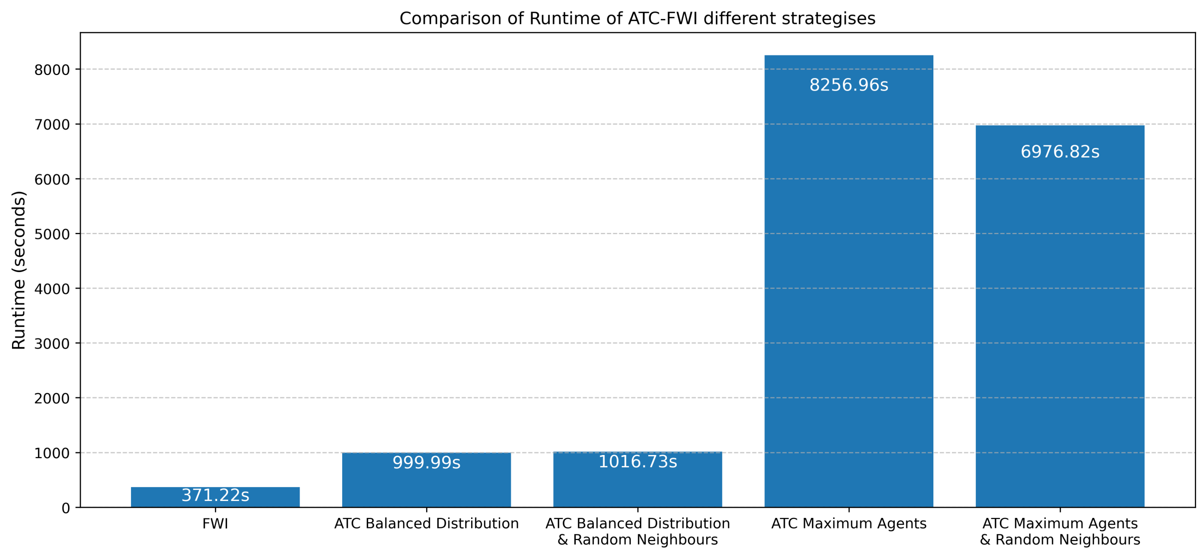


Figure 5: FWI and ATC-FWI Runtime comparison

3.2 Dask Integration

Before discussing Dask integration, it worth mentioning that OpenMP was utilised to expedite Devito's under-the-hood low-level C kernels. OpenMP is an API supporting C/C++ and Fortran multi-platform shared-memory parallel programming. Additionally, Dask was integrated to expedited gradient computation at python level as described in Section 2.5.

All computations were performed on a MacBook Air (M1, 2020) equipped with an Apple M1 chip (8-core CPU: 4 performance and 4 efficiency cores, 7-core integrated GPU) and 16 GB unified memory, running macOS Sequoia 15.6.1.

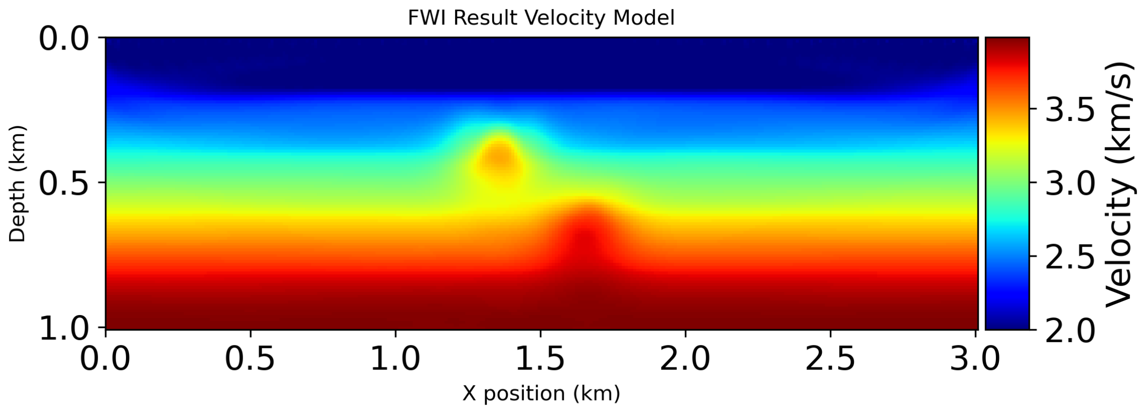
The default type of ATC-FWI network was chosen to test Dask effect on the runtime, which was ATC-FWI with balanced distribution and non-random neighbouring. Figure 7 shows a 1 iteration runtime comparison between the dask integration strategises, OpenMP, and regular FWI with OpenMp. As the figure show, the per-shot Dask integration was found to be 2.05 slower than the base ATC-FWI. This indicates that the gradient loop over the shots is most likely not the reason for the long runtime. In fact, the Dask produce unnecessary overhead when distributing the per-shot misfit and gradient computation.

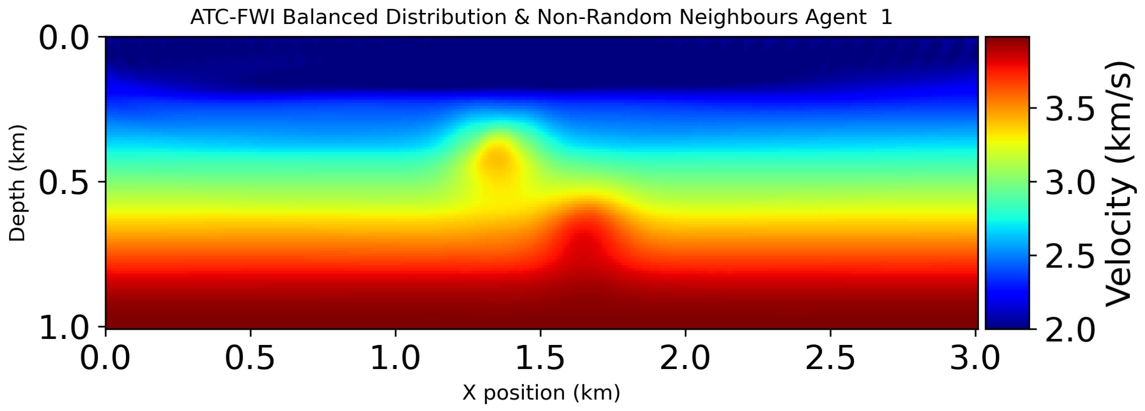
In contrast, the per-agent Dask integration is 1.7 times faster than the base ATC-FWI, and of course 3.0 times faster than the per-shot interrogation due to parallelising gradient computation over the agents network. This resulted in improved CPU utilisation and reduced Python scheduling overhead. Therfore, the integration of Dask over agents was utilised in CPU implementation of ATC-FWI.

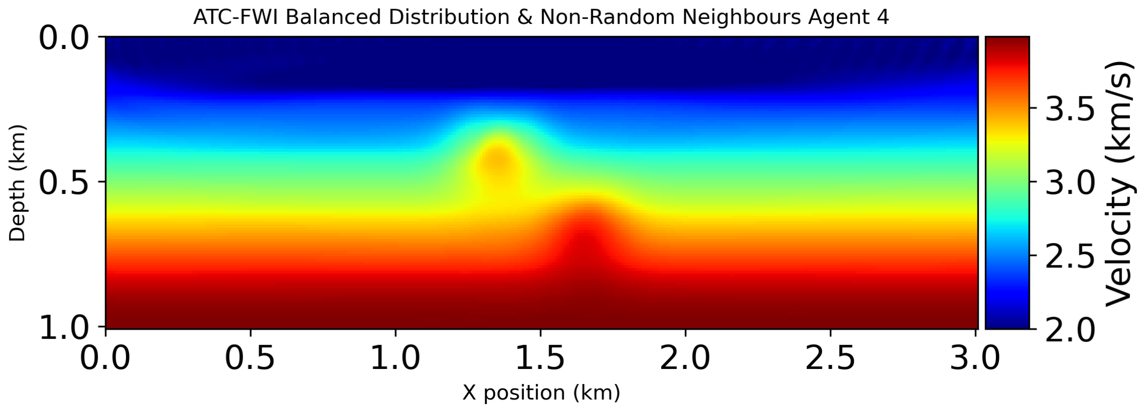
It worth mentioning that since Jetson has a single GPU, Dask parallelisation might conflict with DevitoPRO data streaming between the CPU and GPU, therefore, it was deduced from ATC-FWI implementation on Jetson.

3.3 Porting to Jetson with CUDA

After stabilising the ATC-FWI network in terms of both result quality and runtime efficiency, the implementation was further extended to run on the Jetson platform with CUDA backend, enabling GPU parallelisation. The Jetson device integrates a 6-core Arm Cortex-A78AE CPU, an Ampere-architecture GPU with 1024 CUDA cores and 32 Tensor Cores, and 8 GB of LPDDR5 memory with a bandwidth of 102 GB/sec. With NVMe storage and a power envelope of only 25 W, the system is capable of delivering up to 67 INT8 TOPS of AI performance, making it suitable for rovers limited power and high demanding







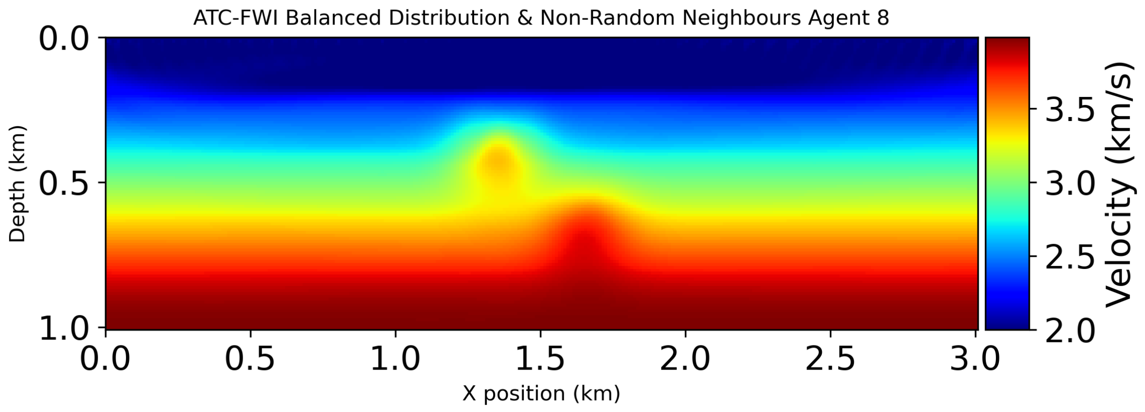


Figure 6: FWI vs. agent 1, 4, and 8 of ATC-FWI network with balanced distribution and non-random neighbouring

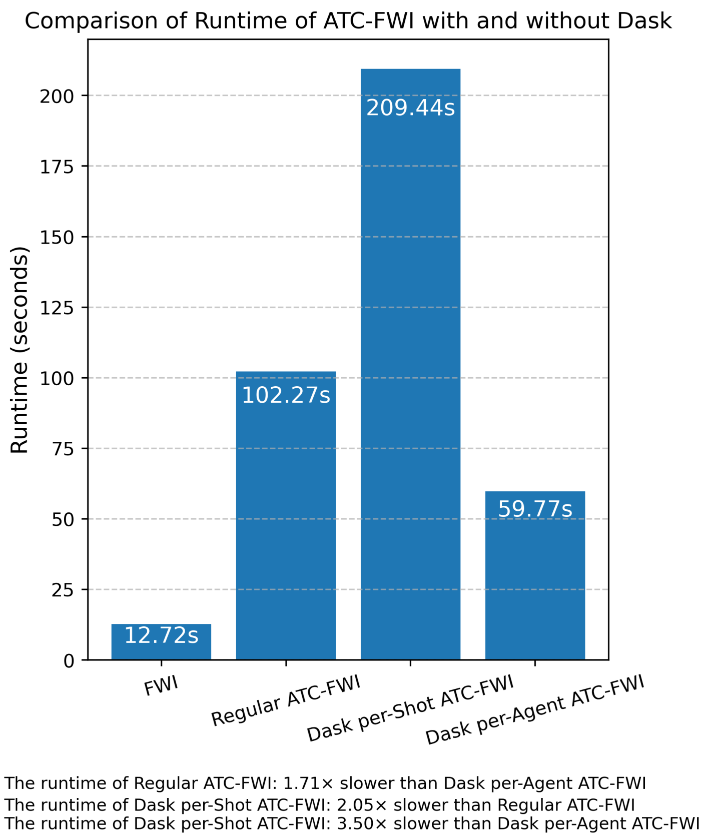


Figure 7: With vs. Without Dask Integration

computation with ATC-FWI. As described in Section 2.6, DevitoPRO was employed to streamline data movement between CPU and GPU, aiming to improve GPU utilisation.

Figure 8 and Figure 9 represent the CPU vs GPU results for FWI and ATC-FWI respectively. It worth mentioning that the slow runtime of ATC-FWI force the decrease of iteration to $k = 12$, specificallyfor this experiment. Although results convey less comprehensive final models with CUDA implementation since the upper anomaly is less framed (see also Table 1). It prove that ATC-FWI can run successfully on Jetson with CUDA utilising streamline data movement by DevtioPRO. The 2D domain structure runtime was much longer compared to the CPU implementation, however, it allows debugging and comparison to the CPU result. The slow runtime may be attributed to data movement overhead between CPU and GPU, as 2D problems are relatively small in compare to 3D.

It is worth noting that a 3D domain was implemented for generating ground-truth data and initial FWI experiments. However, due to time constraints, the ATC-FWI extension to the 3D case was not fully realised.

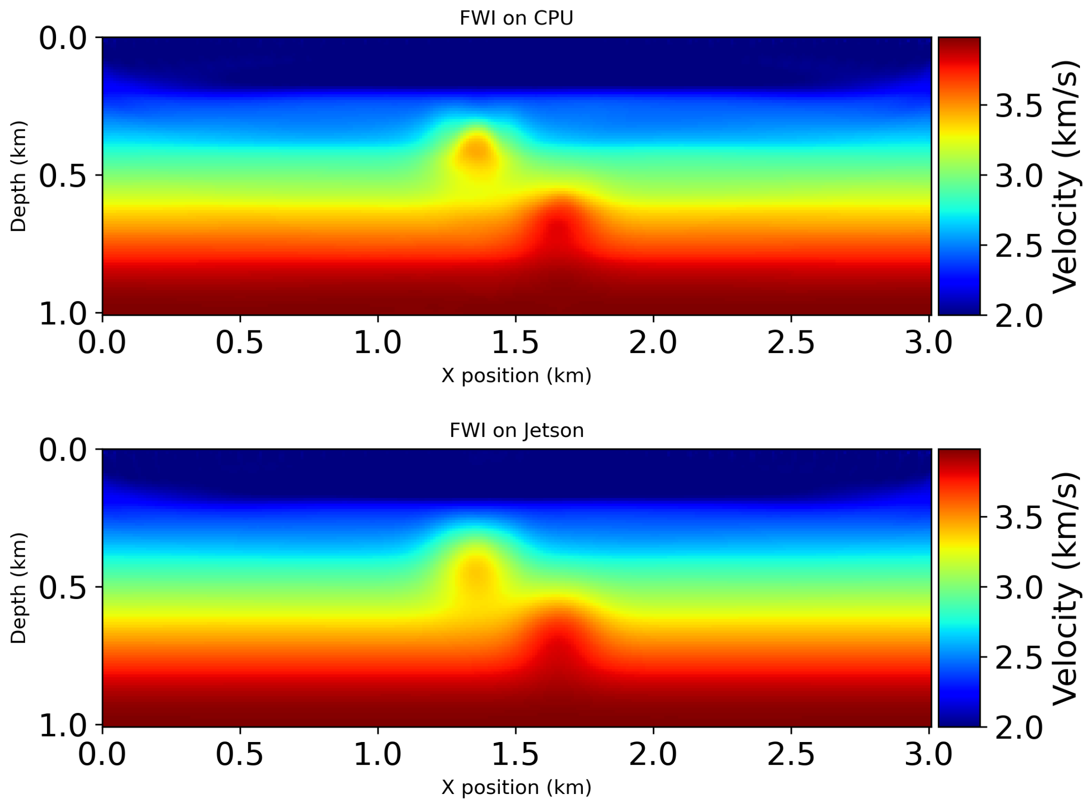


Figure 8: FWI CPU vs. GPU (Jetson)

4 Conclusion and Future Work

In conclusion, this paper began by generating ground truth data, motivated by the layered structure of the planet's interiors, with added complexity due to anomaly integrations. A novel implementation of FWI and ATC-FWI with Devito on the CPU was achieved, utilising various strategies for ATC-FWI to investigate the seismic array effect and convergence speed. Additionally, Dask was integrated on two different levels to expedite the runtime and explore parallelisation capabilities. Finally, ATC-FWI has been successfully extended on NVIDIA Jetson with CUDA utilising DevitoPRO. Although the implementation is limited to a 2D domain, the successful run implies the possibility of a 3D extension. The 3D domain will primarily result in better GPU utilisation and less data movement overhead, which is contributed to the fact that the computation of a 3D wave is more demanding than that of a 2D wave. Additionally, Dask integration over multiple Jetson will imply more realistic set up simulating multiple agents carrying their own hardware.

AI Acknowledgment Statement: ChatGPT-4o and 5 by OpenAI (https://chatgpt.com) were consulted with concepts clarifications, code debugging and plotting. All submitted work is my own interpretation

References

[1] N. B. Douglas, P. Jeffrey, M. Cintala, J. Levine, P. Lowman, R. Mancinelli, W. Mendell, C. Stoker, and S. Suess, "Science Exploration Opportunities for Manned Missions to the Moon, Mars, Phobos, and an Asteroid," NASA, Washington, D.C., Tech. Rep. NASA-CR-186295; JPL-PUBL-89-29, Jun. 1989. [Online]. Available: https://ntrs.nasa.gov/api/citations/19900007460/downloads/19900007460.pdf

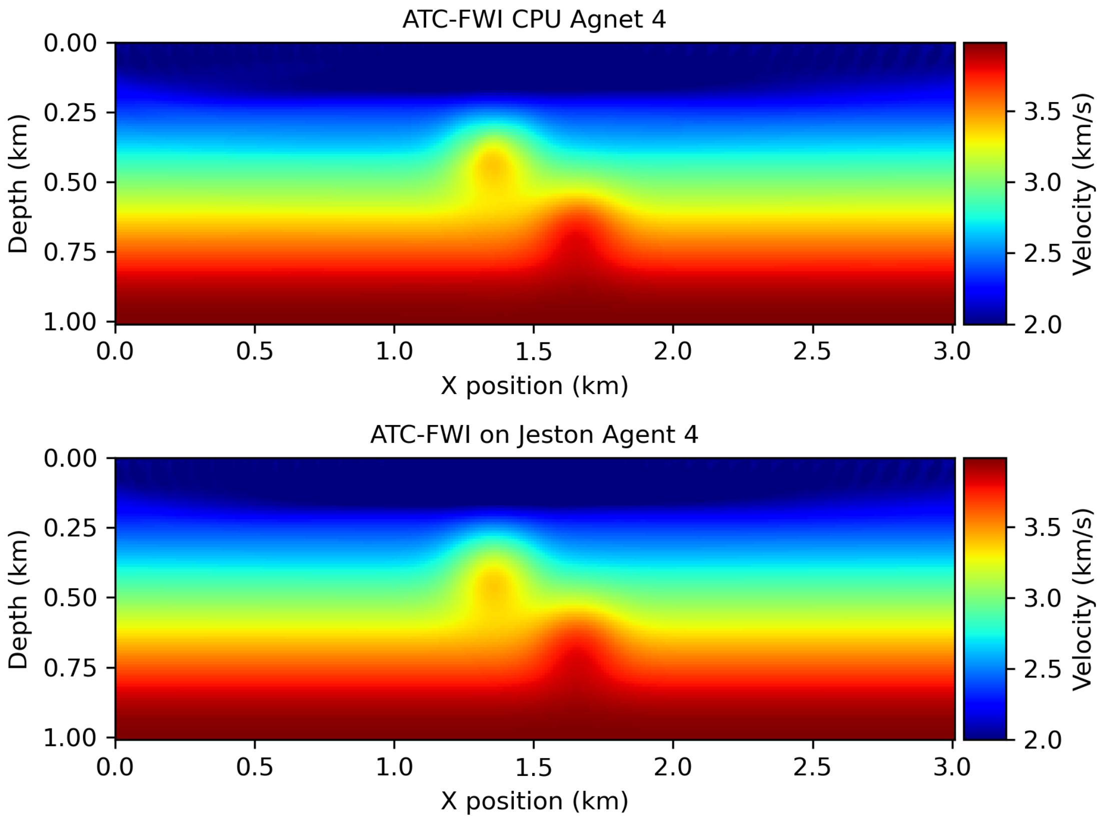


Figure 9: ATC-FWI Agent 4 on CPU vs. GPU (Jetson)

[2] R. L. Kovach and J. S. Watkins, "Apollo 14 and 16 Active Seismic Experiments & Apollo 17 Lunar Seismic Profiling," NASA, Washington, D.C., Tech. Rep. N76-25140, Apr. 1976. [Online]. Available: https://ntrs.nasa.gov/api/citations/19760018052/downloads/19760018052_V1.pdf

[3] B. Knapmeyer-Endrun and T. Kawamura, "NASA's InSight mission on Mars—first glimpses of the planet's interior from seismology," *Nature Communications*, vol. 11, no. 1, p. 1451, Mar. 2020, publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41467-020-15251-7

[4] B. Knapmeyer-Endrun, M. P. Panning, F. Bissig, R. Joshi, A. Khan, D. Kim, V. Lekić, B. Tauzin, S. Tharimena, M. Plasman, N. Compaire, R. F. Garcia, L. Margerin, M. Schimmel, Stutzmann, N. Schmerr, E. Bozdağ, A.-C. Plesa, M. A. Wieczorek, A. Broquet, D. Antonangeli, S. M. McLennan, H. Samuel, C. Michaut, L. Pan, S. E. Smrekar, C. L. Johnson, N. Brinkman, A. Mittelholz, A. Rivoldini, P. M. Davis, P. Lognonné, B. Pinot, J.-R. Scholz, S. Stähler, M. Knapmeyer, M. Van Driel, D. Giardini, and W. B. Banerdt, "Thickness and structure of the martian crust from InSight seismic data," *Science*, vol. 373, no. 6553, pp. 438–443, Jul. 2021. [Online]. Available: https://www.science.org/doi/10.1126/science.abf8966

[5] National Aeronautics and Space Administration, "NASA Strategic Plan 2018," NASA, Washington, D.C., Government agency strategic plan document, Feb. 2018.

[6] A. Ellery, *Planetary Rovers: Robotic Exploration of the Solar System.* Berlin, Heidelberg: Springer, 2016. [Online]. Available: https://link.springer.com/10.1007/978-3-642-03259-2

[7] S. Zhang, R. Pohlmann, E. Staudinger, and A. Dammann, "Assembling a Swarm Navigation System: Communication, Localization, Sensing and Control," in *2021 IEEE 18th Annual Consumer*

Communications & Networking Conference (CCNC). Las Vegas, NV, USA: IEEE, Jan. 2021, pp. 1–9. [Online]. Available: https://ieeexplore.ieee.org/document/9369547/

[8] R. Pöhlmann, E. Staudinger, S. Zhang, F. Broghammer, A. Dammann, and P. A. Hoeher, "Cooperative Radio Navigation for Robotic Exploration: Evaluation of a Space-Analogue Mission," in *2023 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, Apr. 2023, pp. 372–380, iSSN: 2153-3598. [Online]. Available: https://ieeexplore.ieee.org/document/10140026

[9] D. S. Drew, "Multi-Agent Systems for Search and Rescue Applications," *Current Robotics Reports*, vol. 2, no. 2, pp. 189–200, Jun. 2021. [Online]. Available: https://doi.org/10.1007/s43154-021-00048-3

[10] W. Jin, H. Du, B. Zhao, X. Tian, B. Shi, and G. Yang, "A Comprehensive Survey on Multi-Agent Cooperative Decision-Making: Scenarios, Approaches, Challenges and Perspectives," Mar. 2025, arXiv:2503.13415 [cs]. [Online]. Available: http://arxiv.org/abs/2503.13415

[11] M. J. Schuster, M. G. Muller, S. G. Brunner, H. Lehner, P. Lehner, R. Sakagami, A. Domel, L. Meyer, B. Vodermayer, R. Giubilato, M. Vayugundla, J. Reill, F. Steidle, I. Von Bargen, K. Bussmann, R. Belder, P. Lutz, W. Sturzl, M. Smisek, M. Maier, S. Stoneman, A. F. Prince, B. Rebele, M. Durner, E. Staudinger, S. Zhang, R. Pohlmann, E. Bischoff, C. Braun, S. Schroder, E. Dietz, S. Frohmann, A. Borner, H.-W. Hubers, B. Foing, R. Triebel, A. O. Albu-Schaffer, and A. Wedler, "The ARCHES Space-Analogue Demonstration Mission: Towards Heterogeneous Teams of Autonomous Robots for Collaborative Scientific Sampling in Planetary Exploration," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5315–5322, Oct. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9134730/

[12] I. A. Nesnas, L. M. Fesq, and R. A. Volpe, "Autonomy for Space Robots: Past, Present, and Future," *Current Robotics Reports*, vol. 2, no. 3, pp. 251–263, Sep. 2021. [Online]. Available: https://link.springer.com/10.1007/s43154-021-00057-2

[13] J.-L. Mari, *Seismic imaging: a practical approach*. EDP Sciences, Nov. 2020. [Online]. Available: https://www.degruyter.com/document/doi/10.1051/978-2-7598-2351-2/html

[14] J. Virieux and S. Operto, "An overview of full-waveform inversion in exploration geophysics," *GEOPHYSICS*, vol. 74, no. 6, pp. WCC1–WCC26, Nov. 2009, publisher: Society of Exploration Geophysicists. [Online]. Available: https://library.seg.org/doi/10.1190/1.3238367

[15] A. Tarantola, "Inversion of seismic reflection data in the acoustic approximation," *GEOPHYSICS*, vol. 49, no. 8, pp. 1259–1266, Aug. 1984. [Online]. Available: https://library.seg.org/doi/10.1190/1.1441754

[16] G. Nolet, "Seismic wave propagation and seismic tomography," in *Seismic Tomography*, G. Nolet, Ed. Dordrecht: Springer Netherlands, 1987, pp. 1–23. [Online]. Available: http://link.springer.com/10.1007/978-94-009-3899-1_1

[17] J. Nocedal and S. J. Wright, *Numerical optimization*, 2nd ed., ser. Springer series in operations research. New York: Springer, 2006.

[18] J. Virieux, S. Operto, H. Ben-Hadj-Ali, R. Brossier, V. Etienne, F. Sourbier, L. Giraud, and A. Haidar, "Seismic wave modeling for seismic imaging," *The Leading Edge*, vol. 28, no. 5, pp. 538–544, May 2009. [Online]. Available: http://library.seg.org/doi/10.1190/1.3124928

[19] R. Brossier, S. Operto, and J. Virieux, "Seismic imaging of complex onshore structures by 2D elastic frequency-domain full-waveform inversion," *GEOPHYSICS*, vol. 74, no. 6, pp. WCC105–WCC118, Nov. 2009, publisher: Society of Exploration Geophysicists. [Online]. Available: https://library.seg.org/doi/10.1190/1.3215771

[20] M. Warner, A. Ratcliffe, T. Nangoo, J. Morgan, A. Umpleby, N. Shah, V. Vinje, I. Štekl, L. Guasch, C. Win, G. Conroy, and A. Bertrand, "Anisotropic 3D full-waveform inversion," *GEOPHYSICS*, vol. 78, no. 2, pp. R59–R80, Mar. 2013, publisher: Society of Exploration Geophysicists. [Online]. Available: https://library.seg.org/doi/full/10.1190/geo2012-0338.1

[21] F. Wu, Q. He, Y. Li, B. Han, and Y. Wang, "Truncated Newton full waveform inversion method for the human brain imaging," *Journal of Physics: Conference Series*, vol. 2822, no. 1, p. 012013, Sep. 2024, publisher: IOP Publishing. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/2822/1/012013

[22] R. Ali, G. Jin, M. Singh, T. Mitcham, and N. Duric, "3D Frequency-Domain Full Waveform Inversion for Whole-Breast Imaging With a Multi-Row Ring Array," *IEEE Open Journal of Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 5, pp. 77–81, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/11003981

[23] B.-S. Shin and D. Shutin, "ADAPT-Then-Combine Full Waveform Inversion for Distributed Subsurface Imaging In Seismic Networks," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Toronto, ON, Canada: IEEE, Jun. 2021, pp. 4700–4704. [Online]. Available: https://ieeexplore.ieee.org/document/9414072/

[24] J. Chen and A. H. Sayed, "Diffusion Adaptation Strategies for Distributed Optimization and Learning over Networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4289–4305, Aug. 2012, arXiv:1111.0034 [math]. [Online]. Available: http://arxiv.org/abs/1111.0034

[25] B.-S. Shin, D. Patel, L. Wientgens, D. Shutin, and A. Dekorsy, "Multi-Agent 3D Seismic Exploration Using Adapt-then-Combine Full Waveform Inversion in a hardware-in-the-loop System," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Seoul, Korea, Republic of: IEEE, Apr. 2024, pp. 12 911–12 915. [Online]. Available: https://ieeexplore.ieee.org/document/10447703/

[26] J. D. Owens, M. Houston, D. Luebke, S. Green, J. E. Stone, and J. C. Phillips, "GPU Computing," *Proceedings of the IEEE*, vol. 96, no. 5, pp. 879–899, May 2008. [Online]. Available: https://ieeexplore.ieee.org/document/4490127

[27] J. Nickolls, I. Buck, M. Garland, and K. Skadron, "Scalable Parallel Programming with CUDA: Is CUDA the parallel programming model that application developers have been waiting for?" *Queue*, vol. 6, no. 2, pp. 40–53, Mar. 2008. [Online]. Available: https://dl.acm.org/doi/10.1145/1365490.1365500

[28] P. Micikevicius, "3D finite difference computation on GPUs using CUDA," in *Proceedings of 2nd Workshop on General Purpose Processing on Graphics Processing Units*, ser. GPGPU-2. New York, NY, USA: Association for Computing Machinery, Mar. 2009, pp. 79–84. [Online]. Available: https://doi.org/10.1145/1513895.1513905

[29] T. Okamoto, H. Takenaka, T. Nakamura, and T. Aoki, "Accelerating large-scale simulation of seismic wave propagation by multi-GPUs and three-dimensional domain decomposition," *Earth, Planets and Space*, vol. 62, no. 12, pp. 939–942, Dec. 2010, publisher: SpringerOpen. [Online]. Available: https://earth-planets-space.springeropen.com/articles/10.5047/eps.2010.11.009

[30] NVIDIA, "Jetson Orin Nano Module Datasheet," https://developer.nvidia.com/embedded/jetson-orin, 2023, accessed: 2025-06-12.

[31] A. Archet, N. Gac, F. Orieux, and N. Ventroux, "Embedded AI performances of Nvidia's Jetson Orin SoC series," in *17ème Colloque National du GDR SOC2*, Lyon, France, Jun. 2023. [Online]. Available: https://hal.science/hal-04186977

[32] N. Surantha and N. Sutisna, “Key Considerations for Real-Time Object Recognition on Edge Computing Devices,” *Applied Sciences*, vol. 15, no. 13, p. 7533, Jan. 2025, publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2076-3417/15/13/7533

[33] M. Louboutin, M. Lange, F. Luporini, N. Kukreja, P. A. Witte, F. J. Herrmann, P. Velesko, and G. J. Gorman, “Devito (v3.1.0): an embedded domain-specific language for finite differences and geophysical exploration,” *Geoscientific Model Development*, vol. 12, no. 3, pp. 1165–1187, 2019. [Online]. Available: https://www.geosci-model-dev.net/12/1165/2019/

[34] F. Luporini, M. Louboutin, M. Lange, N. Kukreja, P. Witte, J. Hückelheim, C. Yount, P. H. J. Kelly, F. J. Herrmann, and G. J. Gorman, “Architecture and performance of devito, a system for automated stencil computation,” *ACM Trans. Math. Softw.*, vol. 46, no. 1, apr 2020. [Online]. Available: https://doi.org/10.1145/3374916

[35] R. G. Pratt, “Seismic waveform inversion in the frequency domain, Part 1: Theory and verification in a physical scale model,” *GEOPHYSICS*, vol. 64, no. 3, pp. 888–901, May 1999. [Online]. Available: https://library.seg.org/doi/10.1190/1.1444597

[36] S. Rost and C. Thomas, “Array Seismology: Methods and Applications,” *Reviews of Geophysics*, vol. 40, no. 3, pp. 2–1–2–27, 2002, _eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2000RG000100. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1029/2000RG000100

[37] Dask Developers, “Dask documentation.” [Online]. Available: https://docs.dask.org/en/stable/

[38] R. Li, B. Hanindhito, S. Yadav, Q. Wu, K. Kavi, G. Mehta, N. J. Yadwadkar, and L. K. John, “Performance Implications of Pipelining the Data Transfer in CPU-GPU Heterogeneous Systems,” *ACM Trans. Archit. Code Optim.*, Jun. 2025, just Accepted. [Online]. Available: https://dl.acm.org/doi/10.1145/3746231