

Deliverable

Deliverable D2.4: Using DAS and geophone chain for imaging

Report information	
Work package	WP2: Innovation in real-time monitoring and big-data analysis for EGS
Lead	DIAS
Authors	Nicolas Luca Celli, Christopher J. Bean
Reviewers	[Federica Lanza]
Approval	[Management Board]
Status	[final]
Dissemination level	[internal]
Will the data supporting this document be made open access?	Yes
If No Open Access, provide reasons	[...]
Delivery deadline	30.11.2022
Submission date	2.12.2022
Intranet path	[DOCUMENTS/DELIVERABLES/File Name]



Table of Contents

Summary	3
1. Publication	3
2. Codes	4

Summary

We developed a numerical solver to quantitatively simulate DAS records and used it to analyse what acquisition settings--e.g. ground-cable coupling, local site effects, presence of loose sediments around the cable---affect the response of a DAS cable. This tool allows the assessment of these effects and to separate them from the subtle signals targeted when imaging detailed structures both in synthetic models and real data.

1. Publication

We are in the advanced stages of the preparation of a publication on the topic, which we aim to submit at the end of January 2023. The delay until end of January is to allow us to work on a specific technical question (assessing the simulation results when increasing the resolution through finer model discretisation). The manuscript is largely written (see **Error! Reference source not found.**), and we submitted the following abstract for AGU 2022 where we will present the work that will go in the publication:

Full wavefield modelling of DAS cable and ground coupling response using Discrete Particle Schemes (Nicolas L. Celli, Nima Nooshiri, Christopher J. Bean, Gareth O'Brien)

Over the past several years, the use of fiber optic cables as ground motion sensors has become a central topic for seismologists, with successful applications of Distributed Acoustic Sensing (DAS) in various key fields such as seismic monitoring, structural imaging and source characterisation. DAS response is a combination of both instrument response (DAS interrogator) and cable-ground coupling, with the latter having a strong, spatially variable, but yet largely unquantified effect. This limits the application of a large number of staple seismological techniques (e.g., earthquake magnitude estimation, waveform tomography) that can require accurate knowledge of a signal's amplitude and frequency content. Here we present a method for accurately simulating a DAS cable and its ground coupling. The scheme is based on molecular dynamic-like particle-based numerical modelling, allowing the investigation of the effect of varying DAS-ground coupling scenarios. At first, we compute the full strain field directly, for each pair of neighbouring particles in the model. We then define a virtual DAS cable, embedded within the model and formed by a single string of interconnected particles. This allows us to control all aspects of the cable-ground coupling and their properties at an effective granular level through changing the bond strengths and bond types (e.g., non-linearity) for both the cable and the surrounding medium. Arbitrary cable geometries and heterogeneous materials can be accommodated at the desired scale of investigation. We observe that at the meter scale, realistic DAS materials, cable-ground coupling and the presence of unconsolidated materials around it dramatically affect wave propagation, each change affecting the synthetic DAS record, with differences reaching at times the same magnitude of the recorded signal. These differences show that cable coupling has to be considered both when designing a DAS deployment and analysing its data when either true or along-cable relative amplitudes are considered.

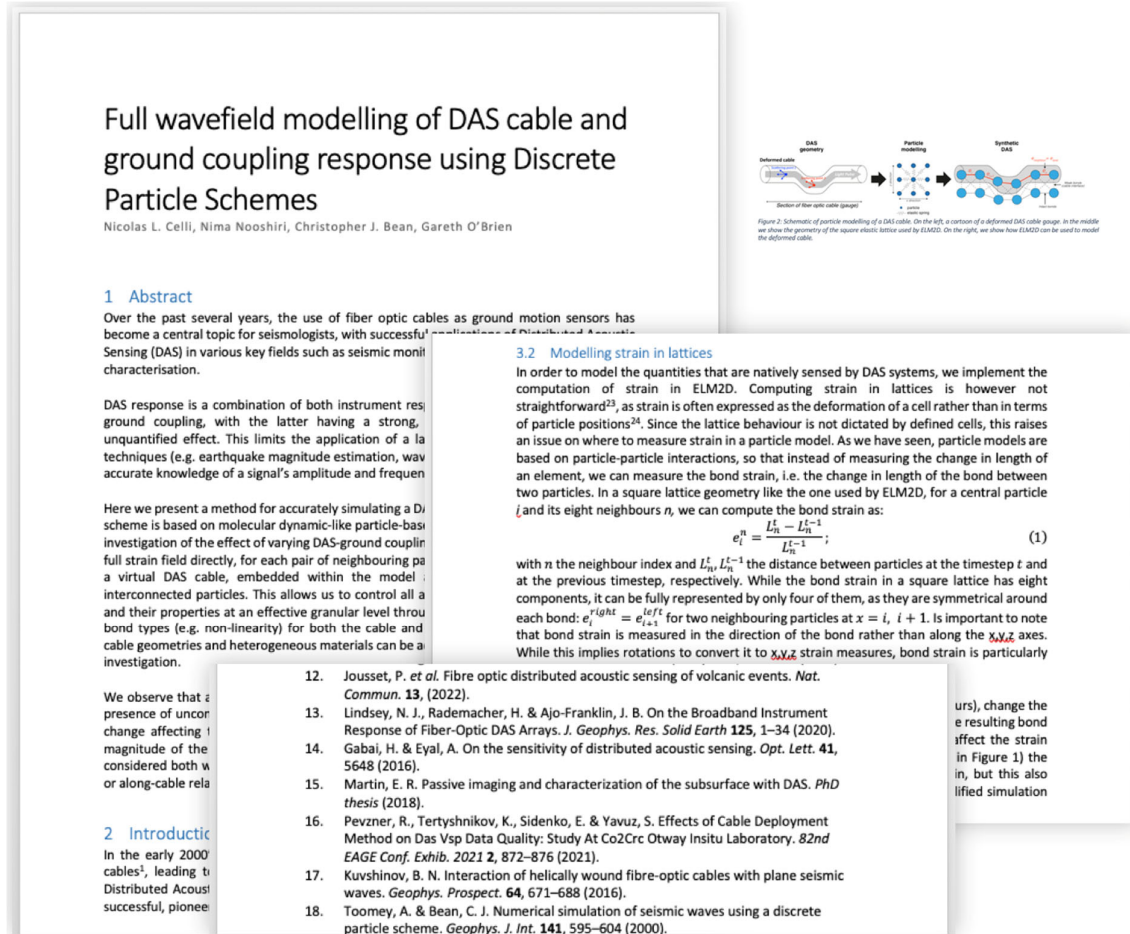
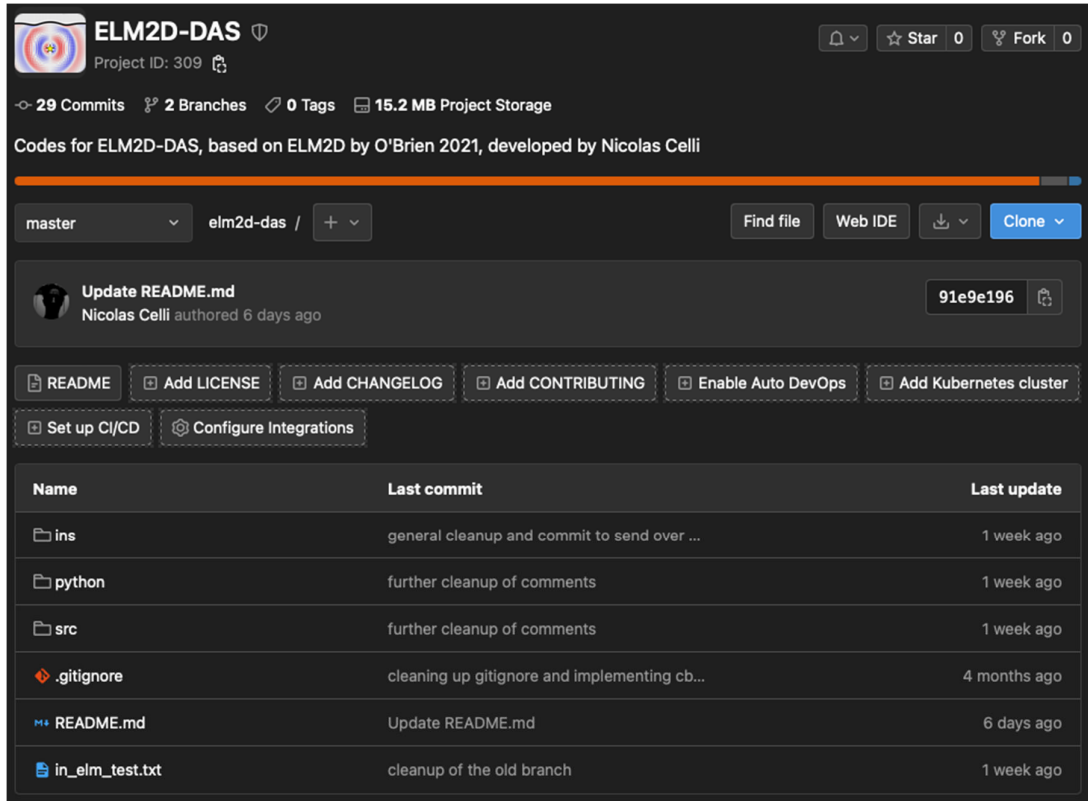


Figure 1: samples of various sections of the manuscript in preparation.

2. Codes

The codes for our numerical solver are hosted on the git repository at DIAS (<https://git.dias.ie/seismology/elm2d.das>). The repository is currently available for internal access only to control dissemination of code before the peer-review process, and we will make the code publicly available once the manuscript is accepted for publication. The repository contains all source codes for the solver, codes to plot the simulation results as well as a full online manual documenting its usage (see Figure 2, Figure 3).



ELM2D-DAS Project ID: 309

29 Commits 2 Branches 0 Tags 15.2 MB Project Storage

Codes for ELM2D-DAS, based on ELM2D by O'Brien 2021, developed by Nicolas Celli

master elm2d-das / Find file Web IDE Clone

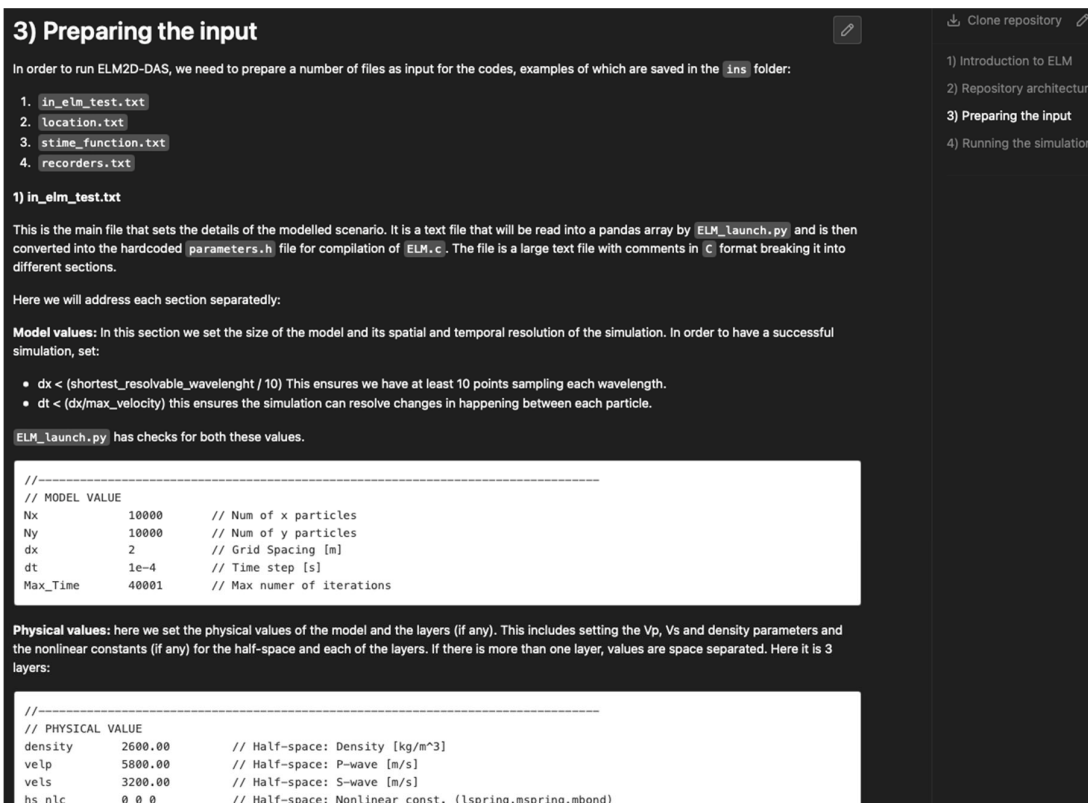
Update README.md
Nicolas Celli authored 6 days ago 91e9e196

README Add LICENSE Add CHANGELOG Add CONTRIBUTING Enable Auto DevOps Add Kubernetes cluster

Set up CI/CD Configure Integrations

Name	Last commit	Last update
ins	general cleanup and commit to send over ...	1 week ago
python	further cleanup of comments	1 week ago
src	further cleanup of comments	1 week ago
.gitignore	cleaning up gitignore and implementing cb...	4 months ago
README.md	Update README.md	6 days ago
in_elm_test.txt	cleanup of the old branch	1 week ago

Figure 2: Main page of the git repository hosting the codes for the numerical solver.



3) Preparing the input

In order to run ELM2D-DAS, we need to prepare a number of files as input for the codes, examples of which are saved in the `ins` folder:

- `in_elm_test.txt`
- `location.txt`
- `stime_function.txt`
- `recorders.txt`

1) in_elm_test.txt

This is the main file that sets the details of the modelled scenario. It is a text file that will be read into a pandas array by `ELM_launch.py` and is then converted into the hardcoded `parameters.h` file for compilation of `ELM.c`. The file is a large text file with comments in `C` format breaking it into different sections.

Here we will address each section separately:

Model values: In this section we set the size of the model and its spatial and temporal resolution of the simulation. In order to have a successful simulation, set:

- $dx < (\text{shortest_resolvable_wavelength} / 10)$ This ensures we have at least 10 points sampling each wavelength.
- $dt < (dx / \text{max_velocity})$ this ensures the simulation can resolve changes in happening between each particle.

`ELM_launch.py` has checks for both these values.

```

//-----
// MODEL VALUE
Nx      10000    // Num of x particles
Ny      10000    // Num of y particles
dx       2      // Grid Spacing [m]
dt      1e-4    // Time step [s]
Max_Time 40001  // Max numer of iterations

```

Physical values: here we set the physical values of the model and the layers (if any). This includes setting the V_p , V_s and density parameters and the nonlinear constants (if any) for the half-space and each of the layers. If there is more than one layer, values are space separated. Here it is 3 layers:

```

//-----
// PHYSICAL VALUE
density  2600.00 // Half-space: Density [kg/m^3]
velp    5800.00 // Half-space: P-wave [m/s]
vels    3200.00 // Half-space: S-wave [m/s]
hs_nlc  0 0 0   // Half-space: Nonlinear const. (lspring,mspring,mbond)

```

Figure 3: snapshot capturing one of the manual pages detailing the usage of the numerical solver.

Liability Claim

Do we need something like this?