**POLITECNICO**

MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

# Three-dimensional wave propagation analysis in SPEED through the Domain Reduction Method

Tesi di Laurea Magistrale in
Mathematical Engineering - Ingegneria Matematica

Author: **Micol Dalmonte**

Student ID: 968925
Advisor: Prof. Paola Francesca Antonietti
Co-advisors: Prof. Ilario Mazzieri
Academic Year: 2022-23

# Abstract

We present the Domain Reduction Method (DRM) in the context of three-dimensional numerical simulations of strong ground motion for the study of the effects due to the seismic source on geological and man-made structures. DRM is a sub-structuring technique that allows to divide the region under study in two subsequent models, in order to reduce the computational cost of the simulation and to guarantee greater accuracy in the results. We explain how the method is implemented, in the plane wave scenario, on code SPEED (*SPectral Elements in Elastodynamics with Discontinuous Galerkin*), and the way in which the input files are generated for a correct usage of the code.
The validity of the method and its advantages are tested on some preliminary geological models.

**Keywords:** DRM, seismic, SPEED, plane wave, earthquake

# Abstract in lingua italiana

In questa tesi viene presentato il Metodo del Dominio Ridotto (MDR) nel contesto di simulazioni numeriche tridimensionali del movimento del suolo, volte allo studio degli effetti provocati da una sorgente sismica su strutture di tipo geologico o artificiale. MDR si presenta come una tecnica di tipo sub-strutturale dal momento che comporta la divisione della regione di interesse in due modelli sequenziali, volti alla riduzione del costo computazionale della simulazione e al fine di garantire una migliore accuratezza dei risultati. Viene descritto il modo in cui il metodo è implementato, nel contesto di onde piane, all'interno del codice SPEED (*SPectral Elements in Elastodynamics with Discontinuos Galerkin*), e la procedura necessaria da seguire per la creazione dei file di input in modo da garantire un corretto uso del codice.

Il metodo viene poi validato facendo uso di alcuni modelli geologici introduttivi, grazie ai quali è possibile anche analizzare i vantaggi legati al MDR.

**Parole chiave:** MDR, sismico, SPEED, onde piane, terremoto

# Contents

# Introduction

Simulating earthquake ground motion in seismic regions for the study of its effects on geological or man-made structures is a topic of great interest, especially in applications involving civil and structural engineering.

In spite of that, three-dimensional simulations on seismic wave propagation analyses are still very hard to put into practice, because of the high computational cost and the complexity in properly defining the earthquake scenario.

Traditional numerical approaches typically consider a single model including in the domain the whole geological structure, from the seismic source to the localized structure under study. This strategy, however, has two major drawbacks: computationally, this necessarily leads to some approximations in the maximum frequency of propagation and in the lowest propagation velocity, moreover the construction of a mesh that takes into account features that can be modeled with a variation of about three/four orders of magnitude is very complicated. In addition, these methods appear particularly disadvantageous when the seismic source is very distant from the site of interest.

This thesis focuses on the Domain Reduction Method (DRM), that is a method representing an alternative approach to the one described before.

Referring to Bielak, Loukakis, Hisada, and Yoshimura [2003], the DRM is a sub-structuring technique in which the problem is subdivided into two sequential parts. The first one is an auxiliary problem where the structure under study is artificially substituted by the same materials of the surrounding soil. At this stage, it is computed the ground motion obtained in absence of the structure, the so-called *free-field motion*. After that, it is considered only a reduced part of the entire domain, referred to as reduced problem. The seismic source and most of the propagation path are not taken into consideration. Indeed, the reduced domain is only slightly larger than the structure of interest.

Seismic excitation are introduced to this model in the form of localized effective forces, calculated thanks to the free-field motion obtained with the auxiliary domain.

This procedure allows in particular to choose two different grid scales in the composition of the model, avoiding an high computational cost and leading to an accurate configuration.

The central approximation made for the development of this thesis is the assumption of either vertical or oblique plane waves. This hypothesis turns out to be reasonable especially when the source is located far away from the site under study. Indeed, the spherical wavefront propagating from the seismic source travels through several transmissions as it moves through the earth's crustal layers. Because of the distinct material that composes the different levels of the earth (hard rock material at depth, softer material near the surface) and accordingly to *Snell's law* (see Aki and Richards [2002]) waves are almost vertical when they reach the earth's surface.

The aim of this thesis is to implement the Domain Reduction Method in SPEED (*SPectral Elements in Elastodynamics with Discontinuous Galerkin*), a high performance open-source library, which primary use is to examine seismic wave propagation in viscoelastic heterogeneous three-dimensional media on both local and regional scale (see Mazzieri, Stupazzini, Guidotti, and Smerzini [2013], SPEED: `https://speed.mox.polimi.it`). This is done, in particular, by exploiting the implementation of the method already done by Smerzini [2010] on the code GeoELSE (*GeoELastodynamics by Spectral Elements*). The latter represents a sort of older version of SPEED.

Moreover, suitable Matlab's scripts have been developed for the phase before calculating the effective forces in SPEED. More precisely, they are involved in the computation of the free-field motion, necessary for the simulation of the reduced domain.

Some three-dimensional tests have been subsequently used to validate the code.

By summarizing the contents in each chapter, we describe the thesis' structure in detail.

**Chapter 1**. After a brief introduction to the Domain Reduction Method, we go trough its theoretical formulation. We start from the equations of motion formulated for the entire domain, until the expression of the effective forces computed in the reduced domain.

**Chapter 2**. We present here a brief overview of the code SPEED, that will be followed by the algorithmic aspects of the computation in SPEED both of the stiffness matrix and of the effective boundary forces. After that, we look more closely at the key aspects of the procedure for the analysis of plane wave propagation, the context in which the code was developed.

**Chapter 3**. We describe the steps necessary for the generation of the input files. We revise all the procedure required for the generation of the mate file with the functions FDRM (free-field motion). The latter is needed in order to execute the code SPEED coupled with DRM. The creation of such a file pass through the construction of the computational domain's mesh (software CUBIT) and all the Matlab scripts already used and implemented by Smerzini [2011].

**Chapter 4**. In this chapter we validate the code addressing three different test cases. The first one (homogeneous model) and the second one (heterogeneous model) are studied under the incidence of $SV$ and $P$ waves, both in the case of vertical wave (angle of incidence $\gamma = 0°$) and oblique wave (angle of incidence $\gamma = 10°$).
After that, a more physical test case is studied: croissant valley. The analysis concerns a valley surrounded by a bedrock subject to a vertical $SH$ wave.

**Chapter 5**. We provide an overview of the outcomes attained and discuss open problems for future research.

# 1 | The Domain Reduction Method (DRM)

The Domain Reduction Method, also referred to as DRM, represents an alternative method with respect to the traditional single-step approach used for simulating earthquake ground motion in seismic regions.

Indeed, as quoted by Bielak et al. [2003], in the 90's many models were developed for the three-dimensional modelling of the ground motion due to seismic events. For instance, for moderate-size problems with quite simple geometries and geological features, boundary element and discrete wavenumber approaches have become popular (e.g. Mossessian and Dravinski [1987], Kato, Aki, and Teng [1993], Kawase and Aki [1990]). Instead, for large-size problems, which includes realistic basin models with strongly heterogeneous materials, methods like finite elements (e.g. Lysmer and Drake [1971], Toshinawa and Ohmachi [1992], Bao [1998], Bao, Bielak, Ghattas, Kallivokas, O'Hallaron, Shewchuk, and Xu [1998] and Aagaard, Hall, and Heaton [2001]) and finite differences (e.g. Frankel and Vidale [1992], Frankel [1993], Graves [1993, 1996], Olsen, Pechmann, and Schuster [1995] and Pitarka [1999]) are preferred due to their perspicuity and flexibility. Another approach used in this framework is that of Spectral elements, see e.g. Faccioli, Maggio, Paolucci, and Quarteroni [1997] and Komatitsch, Tsuboi, Ji, and Tromp [2003], Komatitsch, Liu, Tromp, Suss, Stidham, and Shaw [2004].

However, traditional methods usually adopt uniform structured grids for large-size problems. This often forces to make prohibitive simplifications and approximations in three-dimensional simulations, including restricting the highest frequency or the slowest wave velocities that can be taken into account. Even though the softest soils only cover a small section of the computational domain, the grid size, which is proportional to the lowest shear wave velocity in the model and inversely proportionate to the highest frequency of interest, is held constant throughout it.

This explains why finite element and other methods with irregular grids are more preferable: thanks to their flexibility, they can better fit the mesh size to the local wavelength of the propagating waves.

Although the grid problem can be overcome in some way, another issue still remains: despite their differences, the traditional finite difference and finite element methods are characterised by the simultaneous computation of the ground motions near the causative fault, along the propagation path and within the region of interest. This is done using a single model that includes the whole geological structure, from the fault region to the site under study, namely adopting a *single-step* procedure. Despite this approach is good for a wide range of applications, it presents some problems when the seismic source is distant from the geological site under examination.

According to the work by Bielak et al. [2003], in the previous decades several research projects have been devoted to the development of **substructuring techniques** based on combinations of different computational methods aimed at overcoming this problem.
For example, some of these are: i) discrete wavenumber method coupled with finite difference methods (FDM) by Zahradník and Moczo [1996]; ii) modal summation method + FDM by Fäh, Suhadolc, and Panza [1994]; iii) Finite element method (FEM) coupled with boundary element method (BEM) by Bielak, MacCamy, McGhee, and Barry [1991] and Mita and Luco [1987]; iv) Finite element (FE) coupled with finite element by Bielak et al. [2003] and Yoshimura, Bielak, Hisada, and Fernández [2003]; v) Finite difference (FD) coupled with finite difference by Oprsal and Zahradnik [2002].

In this context, also the DRM represents an alternative approach to the single model, since it allows to divide the entire domain in order to avoid representing a single large region in which the seismic source is very far from the interested area or in which the scale of the elements varies significantly. In particular, DRM permits to reduce the computational cost of the analysis because is a two-step FE methodology in which an equivalent seismic excitation is applied at a fictitious boundary which ideally divides the region of interest (e.g., the Near field) from an external region (e.g., the Far field). For this reason DRM is a powerful substructuring technique. As a result, the seismic input is positioned closer to the region of interest and set in a way that ensures the same outcomes as the initial excitation modeling.
The approach has been originally proposed for soil-structure interaction (SSI) problems by Bielak and Christiano [1984] and subsequently applied to two-dimensional case-studies (e.g. Cremonini, Christiano, and Bielak [1988], Loukakis [1988], Loukakis [1994] and Loukakis and Bielak [1994]). Its extension to three-dimensional problems is provided by Yoshimura et al. [2003] and limited applications have been proposed in literature, such as in the works by Cremonini et al. [1988] and Oprsal and Zahradnik [2002]. The former applies to the finite element (FE) method and validates the formulation proposed by Bielak et al. [2003], while the second one is formulated for the finite-difference (FD) approach.

In the DRM the original problem (STEP 0) is divided into two numerical simpler *sub-problems* analysed sequentially, as depicted in Figure 1.1:

- STEP I: consider an *auxiliary problem*, that contains the seismic source and the propagation path, but from which the localized structure and the geological features of interest have been removed and replaced with a background structure having the same materials of the surrounding soil. In this step the *free-field ground motion* due to the seismic source is computed. The term *free-field* motion is used to denote the ground motion evaluated in absence of the structure or of any excavation.
  It is worth noticing that this computation only needs to be performed once for a certain earthquake source even when some features or system parameters of the localized structure are supposed to vary. Moreover, the grid or mesh only needs to be as fine as required by the background model's softest material.

- STEP II: consider subsequently a *reduced problem* where the initial domain is restricted in such a way as to delimit the presence of structures and geological features which are in turn examined. In this way, the new problem is modeled in a more accurate way, since a finer grid can be considered without compromising a lot the whole computational cost.
  Since now the fault of the earthquake ground motion has been excluded from the domain, we exploit the ground motion obtained in the previous step in order to determine the seismic input. This is introduced as a collection of equivalent effective nodal forces acting within a single layer of elements at an ideal interface between the region of interest and an external domain. The effective forces act as an equivalent dynamic excitation. This step needs to be repeated whenever changes in parameters of the localized structure of interest are introduced.

This methodology is capable of efficiently modeling three-dimensional wave fields for an arbitrary earthquake source in highly heterogeneous geological systems. Moreover, the main advantage of this modular two-step procedure is the possibility of computing the free-field displacements and, after that, the effective forces, by either numerical or analytical methods suitable for simplified geological settings (e.g. horizontally layered crustal model) and arbitrary seismic excitations (point/extended source or plane wave propagation).

STEP 0: ORIGINAL PROBLEM

STEP 1: AUXILIARY PROBLEM
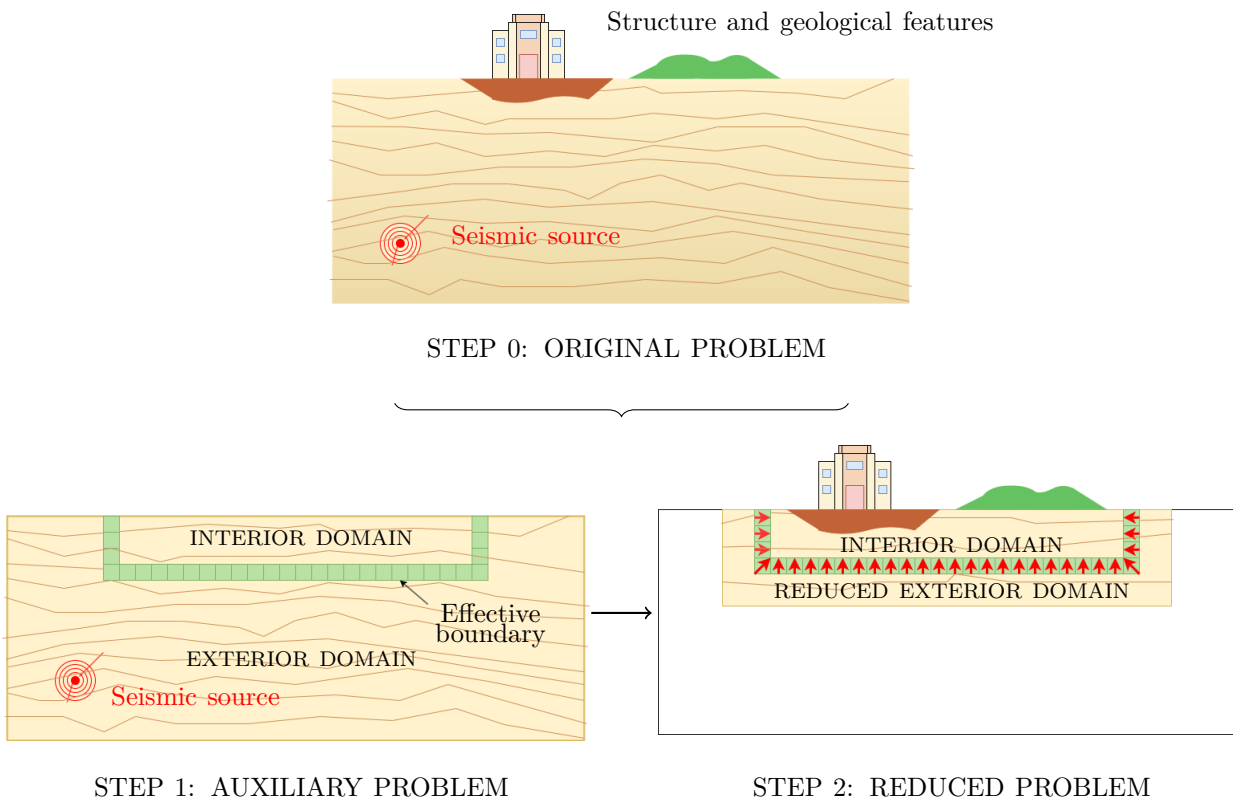
STEP 2: REDUCED PROBLEM

Figure 1.1: Domain Reduction Method (DRM) two-step procedure: the original problem is subdivided into two numerical submodels, namely the auxiliary problem and the reduced problem. **STEP 0** (top panel): original problem. **STEP 1** (left bottom panel): auxiliary problem. Seismic source and propagation path effects are included, while structure and geological features are removed and replaced with a background structure having the same materials of the surrounding soil. **STEP 2** (right bottom panel): reduced model including the localized geological structure of interest. The coupling of Step 1 and 2 is given by a set of effective forces, evaluated from the free-field solution (auxiliary problem) and applied within a strip of elements as an equivalent dynamic excitation. Adapted from Bielak et al. [2003] and Smerzini [2010].

## 1.1.   Theoretical formulation of the DRM

In this section we describe the three-dimensional theoretical formulation of the method as proposed by Bielak et al. [2003] with the contribution given by Smerzini [2010]. Then, in Chapter 2 we are going to outline its implementation in SPEED.

We consider the problem of a semi-infinite seismic region that contains localized geological features (e.g. basin, hill, cave) as well as man-made structures, henceforth simply called "structure", under earthquake excitation. The latter may be caused by a seismically active fault at depth, as depicted in Figure 1.2, or to plane waves propagating with arbitrary angle of incidence.
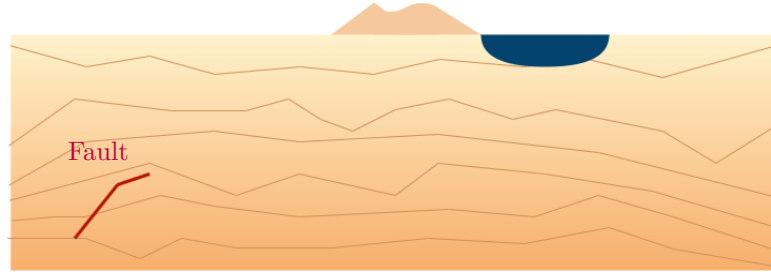


Figure 1.2: Simple representation of a semi-infinite seismic region characterised by the presence of a hill, a basin and an active fault.

Since the causative fault may be far from the structure, we want to define a new problem in which the seismic excitation is brought closer to the region of interest. Certainly this approach must occur in such a way as to ensure that the resulting ground motion within the region under study is identical to that caused by the original source.
To do so, consider $\Gamma_i$ (sketched by a dashed line in Figure 1.3a) as the fictitious surface dividing the original domain into two subdomains: the interior one, denoted as $\Omega_i$, including the structure, and the semi-infinite exterior one, denoted as $\Omega_e$, including the seismic fault. The new excitation has to be specified on this surface $\Gamma_i$. Note that the original semi-infinite area must be truncated for computational reasons; this is achieved including the absorbing boundary $\Gamma^+$ as indicated in Figure 1.3a.

Let the vector field of displacements in the exterior domain $\Omega_e$, in the interior domain $\Omega_i$, and in the boundary $\Gamma_i$ between them, be respectively expressed as $\boldsymbol{u_e}$, $\boldsymbol{u_i}$ and $\boldsymbol{u_b}$ as shown in Figure 1.3a. The earthquake excitation (extended fault or plane wavefront) can be expressed in terms of a set of equivalent body forces $\boldsymbol{P_e}$ operating close to the fault (e.g. Bao [1998]).

As explained before, we want to focus our study on the seismic response in a restricted region near the structure and not directly on the whole domain. For this purpose, we partition the total domain in two subdomains, as depicted in Figure 1.3b. It is worth underline that the displacements $\boldsymbol{u_b}$ are continuous across $\Gamma_i$ and $\boldsymbol{P_b}$ correspond to the forces transmitted from $\Omega_e$ onto $\Omega_i$.



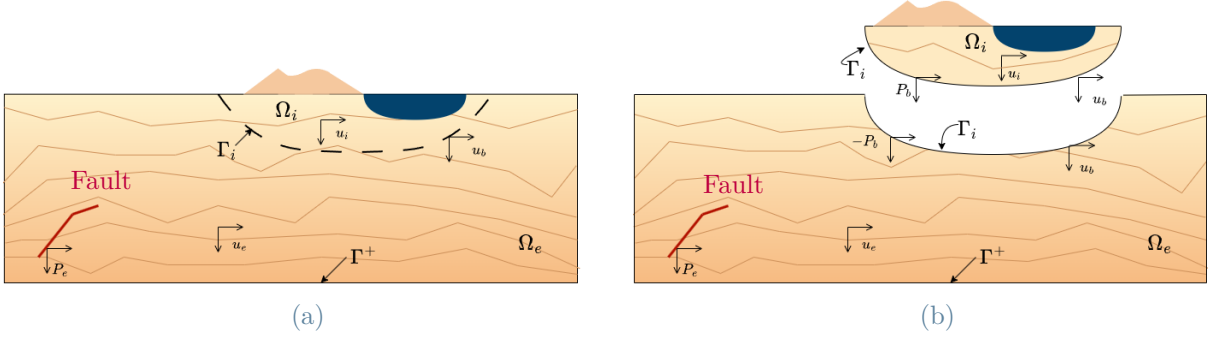(a)                                                               (b)

Figure 1.3: Limited seismic region: idealized seismic region confined by the boundary $\Gamma^+$ including a localized geological irregularity (alluvial basin and a hill) as well as a seismic source (active fault). a) The artificial interface $\Gamma_i$ subdivides the original model into two subdomains: $\Omega_e$, the exterior domain, which includes the seismic source, represented by a set of equivalent body forces $\boldsymbol{P_e}$, and the background geological model; $\Omega_i$, the interior domain, which contains only the localized geological feature. b) Explicit partition of the domain into two subdomains $\Omega_e$ and $\Omega_i$ across the fictitious interface $\Gamma_i$: displacements $\boldsymbol{u_b}$ are continuous across $\Gamma_i$, $\boldsymbol{P_b}$ are transmitted from $\Omega_e$ onto $\Omega_i$ and $-\boldsymbol{P_b}$ are the corresponding reactions equal in modulus but opposite in sign. Adapted from Bielak et al. [2003].

The equations of motion can be stated in $\Omega_i$ and $\Omega_e$ in the following partitioned form after spatial discretization (for example by the use of spectral elements or finite elements, without loss of generality):

Interior domain $\Omega_i$:

$$\begin{bmatrix} M_{ii}^{\Omega_i} & M_{ib}^{\Omega_i} \\ M_{bi}^{\Omega_i} & M_{bb}^{\Omega_i} \end{bmatrix} \begin{bmatrix} \ddot{u}_i \\ \ddot{u}_b \end{bmatrix} + \begin{bmatrix} C_{ii}^{\Omega_i} & C_{ib}^{\Omega_i} \\ C_{bi}^{\Omega_i} & C_{bb}^{\Omega_i} \end{bmatrix} \begin{bmatrix} \dot{u}_i \\ \dot{u}_b \end{bmatrix} + \begin{bmatrix} K_{ii}^{\Omega_i} & K_{ib}^{\Omega_i} \\ K_{bi}^{\Omega_i} & K_{bb}^{\Omega_i} \end{bmatrix} \begin{bmatrix} u_i \\ u_b \end{bmatrix} = \begin{bmatrix} 0 \\ P_b \end{bmatrix}. \qquad (1.1)$$

Exterior domain $\Omega_e$:

$$\begin{bmatrix} M_{bb}^{\Omega_e} & M_{be}^{\Omega_e} \\ M_{eb}^{\Omega_e} & M_{ee}^{\Omega_e} \end{bmatrix} \begin{bmatrix} \ddot{u}_b \\ \ddot{u}_e \end{bmatrix} + \begin{bmatrix} C_{bb}^{\Omega_e} & C_{be}^{\Omega_e} \\ C_{eb}^{\Omega_e} & C_{ee}^{\Omega_e} \end{bmatrix} \begin{bmatrix} \dot{u}_b \\ \dot{u}_e \end{bmatrix} + \begin{bmatrix} K_{bb}^{\Omega_e} & K_{be}^{\Omega_e} \\ K_{eb}^{\Omega_e} & K_{ee}^{\Omega_e} \end{bmatrix} \begin{bmatrix} u_b \\ u_e \end{bmatrix} = \begin{bmatrix} -P_b \\ P_e \end{bmatrix}. \qquad (1.2)$$

Where $\boldsymbol{M}$ denotes the mass matrix, $\boldsymbol{C}$ the damping matrix and $\boldsymbol{K}$ the stiffness matrix. Referring to Figure 1.3, the subscripts $i$, $e$ and $b$ stand for the nodes present in the interior domain, in the exterior domain or in the inner boundary, respectively. Indeed, the vectors $u$, $\dot{u}$ and $\ddot{u}$ represent the nodal displacements, velocities and accelerations for the corresponding subdomain in which they are defined, precisely identified by subscripts $i$, $e$ and $b$. Finally, the superscripts $\Omega_i$ and $\Omega_e$ represent the domain over which different matrices are defined.

In order to obtain the traditional form of the equation of motion, namely the one corresponding to the total domain, it is necessary to add the two equations (1.1) and (1.2) to obtain:

$$\begin{bmatrix} \boldsymbol{M}_{ii}^{\Omega_i} & \boldsymbol{M}_{ib}^{\Omega_i} & 0 \\ \boldsymbol{M}_{bi}^{\Omega_i} & \boldsymbol{M}_{bb}^{\Omega_i} + \boldsymbol{M}_{bb}^{\Omega_e} & \boldsymbol{M}_{be}^{\Omega_e} \\ 0 & \boldsymbol{M}_{eb}^{\Omega_e} & \boldsymbol{M}_{ee}^{\Omega_e} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}}_i \\ \ddot{\boldsymbol{u}}_b \\ \ddot{\boldsymbol{u}}_e \end{bmatrix} + \begin{bmatrix} \boldsymbol{C}_{ii}^{\Omega_i} & \boldsymbol{C}_{ib}^{\Omega_i} & 0 \\ \boldsymbol{C}_{bi}^{\Omega_i} & \boldsymbol{C}_{bb}^{\Omega_i} + \boldsymbol{C}_{bb}^{\Omega_e} & \boldsymbol{C}_{be}^{\Omega_e} \\ 0 & \boldsymbol{C}_{eb}^{\Omega_e} & \boldsymbol{C}_{ee}^{\Omega_e} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}}_i \\ \dot{\boldsymbol{u}}_b \\ \dot{\boldsymbol{u}}_e \end{bmatrix}$$

$$+ \begin{bmatrix} \boldsymbol{K}_{ii}^{\Omega_i} & \boldsymbol{K}_{ib}^{\Omega_i} & 0 \\ \boldsymbol{K}_{bi}^{\Omega_i} & \boldsymbol{K}_{bb}^{\Omega_i} + \boldsymbol{K}_{bb}^{\Omega_e} & \boldsymbol{K}_{be}^{\Omega_e} \\ 0 & \boldsymbol{K}_{eb}^{\Omega_e} & \boldsymbol{K}_{ee}^{\Omega_e} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_b \\ \boldsymbol{u}_e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \boldsymbol{P}_e \end{bmatrix}.$$

$$(1.3)$$

Now, we are going to consider an auxiliary domain in order to transmit the seismic excitation from the fault to the inner boundary $\Gamma_i$, and also to reduce the computational burden. As can be seen from Figure 1.4a, the geological features have been removed from the interior domain, now called $\Omega_i^0$, and substituted with the same material of the exterior domain $\Omega_e$, which instead remained unchanged, as well as the seismic source.

Clearly, the superscript 0 in $\boldsymbol{u}_e^{\boldsymbol{0}}$, $\boldsymbol{u}_b^{\boldsymbol{0}}$, $\boldsymbol{u}_i^{\boldsymbol{0}}$ and $\boldsymbol{P}_b^{\boldsymbol{0}}$ denotes that this time the nodal displacements and the forces are referred to the interior domain $\Omega_i^0$, while the subscripts $e$, $b$ and $i$ have the same meaning as before, as can be seen in Figure 1.4.

We know that the element size of the computational grid is determined by the lowest wave propagation velocity present in the model under study. The smaller propagation velocity is typically reach under the surface of the localized structures that, however, have been removed. For this reason the element size is now controlled by the lowest wave propagation velocity of the background geological structure and this leads to an optimization of the computational grid.
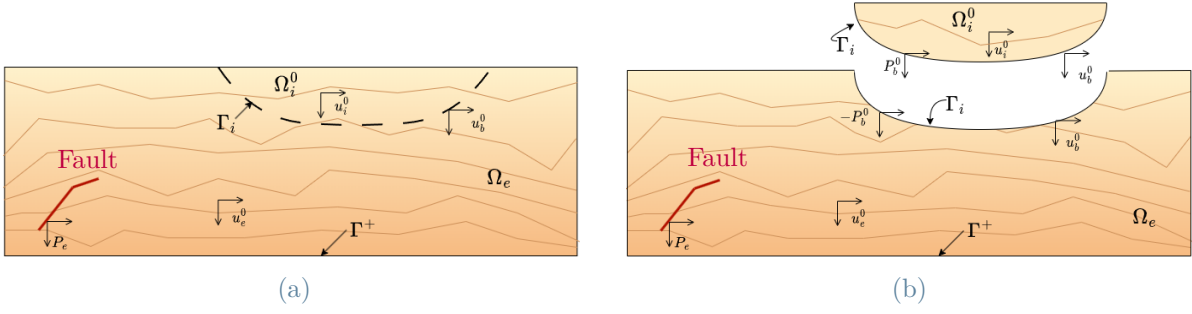
Figure 1.4: Auxiliary seismic region (STEP 1). The localized structure in $\Omega_i^0$ has been removed and substituted with a simpler background structure with the same material of $\Omega_e$. (a) Auxiliary domain in its entirety. (b) Auxiliary domain subdivided in exterior domain $\Omega_e$ and interior domain $\Omega_i^0$. $\Gamma_i$ represents the inner boundary between these two domains. This partition underline the condition of continuity of displacement $\boldsymbol{u_b^0}$ and forces $\boldsymbol{P_b^0}$ transferred through the interface $\Gamma_i$. Adapted from Bielak et al. [2003].

At this point, the equations of motion for the auxiliary problem can be stated in $\Omega_e$ in the following partitioned form after spatial discretization:

$$\begin{bmatrix} \boldsymbol{M}_{bb}^{\Omega_e} & \boldsymbol{M}_{be}^{\Omega_e} \\ \boldsymbol{M}_{eb}^{\Omega_e} & \boldsymbol{M}_{ee}^{\Omega_e} \end{bmatrix} \begin{bmatrix} \boldsymbol{\ddot{u}_b^0} \\ \boldsymbol{\ddot{u}_e^0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{C}_{bb}^{\Omega_e} & \boldsymbol{C}_{be}^{\Omega_e} \\ \boldsymbol{C}_{eb}^{\Omega_e} & \boldsymbol{C}_{ee}^{\Omega_e} \end{bmatrix} \begin{bmatrix} \boldsymbol{\dot{u}_b^0} \\ \boldsymbol{\dot{u}_e^0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{bb}^{\Omega_e} & \boldsymbol{K}_{be}^{\Omega_e} \\ \boldsymbol{K}_{eb}^{\Omega_e} & \boldsymbol{K}_{ee}^{\Omega_e} \end{bmatrix} \begin{bmatrix} \boldsymbol{u_b^0} \\ \boldsymbol{u_e^0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{P_b^0} \\ \boldsymbol{P_e} \end{bmatrix}. \qquad (1.4)$$

Note that equation (1.4) is the same as equation (1.2) except for the presence of the superscript 0 in the variables $\boldsymbol{u_i}$, $\boldsymbol{u_b}$, $\boldsymbol{u_e}$ and $\boldsymbol{P_b}$. In particular, the matrices $\boldsymbol{M}$, $\boldsymbol{C}$ and $\boldsymbol{K}$ are exactly the same as in equation (1.2) because nothing change unless the seismic source changes.

From the second equation in (1.4), we can now explicit the nodal body forces $\boldsymbol{P_e}$ as a function of the free-field motion:

$$\boldsymbol{P_e} = \boldsymbol{M}_{eb}^{\Omega_e} \boldsymbol{\ddot{u}_b^0} + \boldsymbol{M}_{ee}^{\Omega_e} \boldsymbol{\ddot{u}_e^0} + \boldsymbol{C}_{eb}^{\Omega_e} \boldsymbol{\dot{u}_b^0} + \boldsymbol{C}_{ee}^{\Omega_e} \boldsymbol{\dot{u}_e^0} + \boldsymbol{K}_{eb}^{\Omega_e} \boldsymbol{u_b^0} + \boldsymbol{K}_{ee}^{\Omega_e} \boldsymbol{u_e^0}. \qquad (1.5)$$

At this point, by substituting (1.5) into equation (1.3) it is possible to solve the latter equation for the entire domain in terms of the nodal displacements $\boldsymbol{u_i}$, $\boldsymbol{u_b}$, $\boldsymbol{u_e}$. However, this does not give any advantage with respect to the traditional formulation, since equation (1.5) contains the terms $\boldsymbol{M}_{ee}^{\Omega_e} \boldsymbol{\ddot{u}_e^0}$, $\boldsymbol{C}_{ee}^{\Omega_e} \boldsymbol{\dot{u}_e^0}$ and $\boldsymbol{K}_{ee}^{\Omega_e} \boldsymbol{u_e^0}$, that requires the free-field $\boldsymbol{u_e^0}$ to be stored through the domain $\Omega_e$. Of course this entails an excessive amount of calculation.

To overcome this problem, consider the transformation variable $\boldsymbol{w_e}$ defined as follows:

$$\boldsymbol{w_e} = \boldsymbol{u_e} - \boldsymbol{u_e^0}. \tag{1.6}$$

We can observe that the total displacement wavefield $\boldsymbol{u_e}$ can be expressed as the sum of the free-field $\boldsymbol{u_e^0}$ and the residual field $\boldsymbol{w_e}$, which in turn expresses the relative displacement with respect to the free-field solution $\boldsymbol{u_e^0}$.

Taking into consideration the just mentioned definition, we can substitute (1.6) into equation (1.3). After writing the terms containing the free-field on the right hand side, we obtain the following expression:

$$
\begin{bmatrix}
M_{ii}^{\Omega_i} & M_{ib}^{\Omega_i} & 0 \\
M_{bi}^{\Omega_i} & M_{bb}^{\Omega_i} + M_{bb}^{\Omega_e} & M_{be}^{\Omega_e} \\
0 & M_{eb}^{\Omega_e} & M_{ee}^{\Omega_e}
\end{bmatrix}
\begin{bmatrix}
\ddot{u}_i \\
\ddot{u}_b \\
\ddot{w}_e
\end{bmatrix}
+
\begin{bmatrix}
C_{ii}^{\Omega_i} & C_{ib}^{\Omega_i} & 0 \\
C_{bi}^{\Omega_i} & C_{bb}^{\Omega_i} + C_{bb}^{\Omega_e} & C_{be}^{\Omega_e} \\
0 & C_{eb}^{\Omega_e} & C_{ee}^{\Omega_e}
\end{bmatrix}
\begin{bmatrix}
\dot{u}_i \\
\dot{u}_b \\
\dot{w}_e
\end{bmatrix}
$$

$$
+
\begin{bmatrix}
K_{ii}^{\Omega_i} & K_{ib}^{\Omega_i} & 0 \\
K_{bi}^{\Omega_i} & K_{bb}^{\Omega_i} + K_{bb}^{\Omega_e} & K_{be}^{\Omega_e} \\
0 & K_{eb}^{\Omega_e} & K_{ee}^{\Omega_e}
\end{bmatrix}
\begin{bmatrix}
u_i \\
u_b \\
w_e
\end{bmatrix}
=
\begin{bmatrix}
0 \\
-M_{be}^{\Omega_e} \ddot{u}_e^0 - C_{be}^{\Omega_e} \dot{u}_e^0 - K_{be}^{\Omega_e} u_e^0 \\
P_e - M_{ee}^{\Omega_e} \ddot{u}_e^0 - C_{ee}^{\Omega_e} \dot{u}_e^0 - K_{ee}^{\Omega_e} u_e^0
\end{bmatrix}.
\tag{1.7}
$$

Finally, by substituting the expression of $\boldsymbol{P_e}$ determined in (1.5) into equation (1.7) we obtain the following desired equation:

$$
\begin{bmatrix}
M_{ii}^{\Omega_i} & M_{ib}^{\Omega_i} & 0 \\
M_{bi}^{\Omega_i} & M_{bb}^{\Omega_i} + M_{bb}^{\Omega_e} & M_{be}^{\Omega_e} \\
0 & M_{eb}^{\Omega_e} & M_{ee}^{\Omega_e}
\end{bmatrix}
\begin{bmatrix}
\ddot{u}_i \\
\ddot{u}_b \\
\ddot{w}_e
\end{bmatrix}
+
\begin{bmatrix}
C_{ii}^{\Omega_i} & C_{ib}^{\Omega_i} & 0 \\
C_{bi}^{\Omega_i} & C_{bb}^{\Omega_i} + C_{bb}^{\Omega_e} & C_{be}^{\Omega_e} \\
0 & C_{eb}^{\Omega_e} & C_{ee}^{\Omega_e}
\end{bmatrix}
\begin{bmatrix}
\dot{u}_i \\
\dot{u}_b \\
\dot{w}_e
\end{bmatrix}
$$

$$
+
\begin{bmatrix}
K_{ii}^{\Omega_i} & K_{ib}^{\Omega_i} & 0 \\
K_{bi}^{\Omega_i} & K_{bb}^{\Omega_i} + K_{bb}^{\Omega_e} & K_{be}^{\Omega_e} \\
0 & K_{eb}^{\Omega_e} & K_{ee}^{\Omega_e}
\end{bmatrix}
\begin{bmatrix}
u_i \\
u_b \\
w_e
\end{bmatrix}
=
\begin{bmatrix}
0 \\
-M_{be}^{\Omega_e} \ddot{u}_e^0 - C_{be}^{\Omega_e} \dot{u}_e^0 - K_{be}^{\Omega_e} u_e^0 \\
M_{eb}^{\Omega_e} \ddot{u}_b^0 + C_{eb}^{\Omega_e} \dot{u}_b^0 + K_{eb}^{\Omega_e} u_b^0
\end{bmatrix}.
\tag{1.8}
$$

On the one side, the mass matrix, the damping matrix and the stiffness matrix are still identical with those of equation (1.3). On the other side, the seismic forces $\boldsymbol{P_e}$ on the fault have been replaced by the effective nodal forces $\boldsymbol{P^{eff}}$, given by:

$$
\boldsymbol{P^{eff}} =
\begin{bmatrix}
P_i^{eff} \\
P_b^{eff} \\
P_e^{eff}
\end{bmatrix}
=
\begin{bmatrix}
0 \\
-M_{be}^{\Omega_e} \ddot{u}_e^0 - C_{be}^{\Omega_e} \dot{u}_e^0 - K_{be}^{\Omega_e} u_e^0 \\
M_{eb}^{\Omega_e} \ddot{u}_b^0 + C_{eb}^{\Omega_e} \dot{u}_b^0 + K_{eb}^{\Omega_e} u_b^0
\end{bmatrix}.
\tag{1.9}
$$

It is worth noticing that these effective forces $\boldsymbol{P_e}$ have an important property: in their computation are involved only the matrices $\boldsymbol{M_{be}}$, $\boldsymbol{M_{eb}}$, $\boldsymbol{C_{be}}$, $\boldsymbol{C_{eb}}$, $\boldsymbol{K_{be}}$ and $\boldsymbol{K_{eb}}$, that are defined on the exterior domain $\Omega_e$. The latter entails that these effective forces do not vanish only on a single layer of elements in $\Omega_e$ and adjacent to $\Gamma_i$. This small domain is situated between the surface $\Gamma_i$ and the surface $\Gamma_e$, adjacent to the previous one, as one can see in Figure 1.5. This entails that the nodes on $\Gamma_i$, the nodes on $\Gamma_e$ and the nodes between these two surfaces are the only points where the effective forces in (1.9) are computed from the auxiliary problem. The effective forces thus constitute the means by which seismic excitation is introduced into the reduced domain as an equivalent dynamic excitation. The principal advantage of formulation (1.6) therefore consists in the localization of the equivalent effective forces only in a strip of elements in the neighborhood of the geological structure.

There is another important advantage in the formulation (1.9) that is worth pointing out. In the outer domain $\Omega_e$, all the waves are outgoing. As a result, the external domain's size can be significantly reduced to get a reduced exterior domain $\Omega_e^r$ for solving equations of motion (1.8), provided suitable absorbing boundaries are used to reduce the occurrence of spurious waves. These conditions were proposed for example by Clayton and Engquist [1977] and Stacey [1988].

It is exactly for this reason that the method is called *Domain Reduction Method*. Note that at least one strip of elements must be included in between the boundary $\Gamma_e$ and the boundary which delimits the reduced domain, i.e. $\Gamma^{+r}$ (see Figure 1.5).
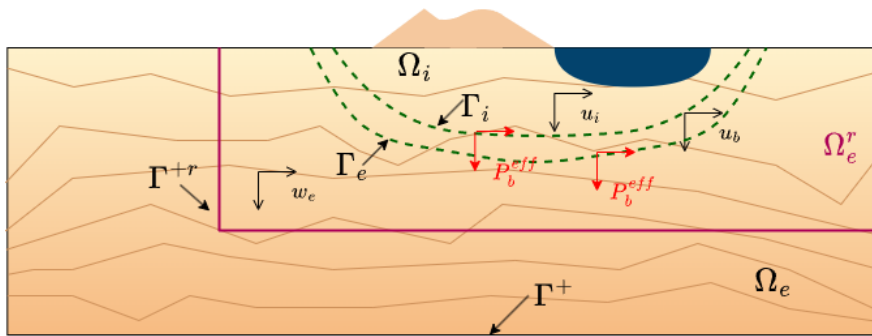


Figure 1.5: Reduced seismic region (STEP 2). The exterior domain $\Omega_e$ has been restricted to the reduced exterior domain $\Omega_e^r$, delimited by the surface $\Gamma^{+r}$. This domain must include at least one strip of elements around the boundary $\Gamma_e$. The effective forces $P_{eff}$, which correspond to a dynamic excitation equivalent to the seismic one, are computed only in the strip of elements in between the inner boundary $\Gamma_i$ and the outer boundary $\Gamma_e$.

Moreover, only the diffracted wavefield $\boldsymbol{w_e}$ has an impact on $\Omega_e^r$ and since the Absorbing Boundary Conditions (ABCs) only affect that wavefield, their effectiveness when applied to the exterior boundary $\Gamma^{+r}$ is greatly increased. This results in a notable reduction of spurious reflections inside the region of interest.

Compared to previous derivations like the one proposed by Bielak and Christiano [1984], Loukakis [1988], Loukakis and Bielak [1994], this turns out to be more rigorous and it considers explicitly the effect of an extended source on a finite fault. Moreover, a similar procedure was proposed by Loukakis [1988] and developed afterward by Aydinoğlu [1993], in the context of soil-structure interaction but without explicit analysis of the seismic source.

# 2 | Implementation of the Domain Reduction Method

In this chapter we discuss the implementation of the method described in Section 1.1 within the code SPEED (*SPectral Elements in Elastodynamics with Discontinuous Galerkin*).

The chapter is organized as follow: a brief overview of the code SPEED will be followed by the algorithmic aspects of the computation in SPEED of the stiffness matrix and of the effective boundary forces. Then, we will see in more detail the main steps employed for the analysis of plane wave propagation, the environment for which the Domain Reduction Method (DRM) was developed.

## 2.1. SPEED: a brief overview

SPEED, as explained by Mazzieri et al. [2013], is a high performance open-source library, which primary use is to approximately solve seismic wave propagation problems in viscoelastic heterogeneous three-dimensional media on both local and regional scale.

This code was developed in response to the growing requirement for creating models that simultaneously account for the seismic source, the propagation path through intricate geological features and localized superficial irregularities, like alluvial basins and man-made structures. Always in this framework, other three-dimensional numerical codes were developed mainly based on finite differences (see e.g. Graves [1996], Moczo, Lucká, Kristek, and Kristeková [1999], Pitarka [1999], Alford, Kelly, and Boore [1974] and Boore [1970]), finite elements (Bao [1998]), pseudo-spectral Fourier method (Furumura and Hayakawa [2007], Kosloff, Kessler, Filho, Tessmer, Behle, and Strahilevitz [1990]) and spectral elements (Faccioli et al. [1997], Komatitsch and Vilotte [1998], Wald and Graves [1998]).

The last approach, spectral element method (SEM), which is also the one implemented in SPEED, has proved to be one of the most effective procedures for solving three-dimensional seismic wave propagation problems in highly heterogeneous media. This is due to its native orientation toward high performance parallel computing and its ca-

pacity to provide quick and extremely accurate solutions. SEM was first introduced in fluid dynamics in the 80's (Maday and Patera [1989] and Patera [1984]) and was applied to elastodynamics in the mid 90's. Following the innovative work by Kosloff and Baysal [1982], which approximated the solution of the partial differential equations in elastic wave propagation using a Fourier decomposition of the displacement field, spectral methods for elastic wave propagation developed first with the introduction of Chebyshev polynomials (Kosloff et al. [1990], Priolo, Carcione, and Seriani [1994]), and then of Legendre polynomials combined with Legendre-Gauss-Lobatto quadrature formulas (Faccioli et al. [1997], Komatitsch and Tromp [1999]).

The SEM is a high order variational method that combines the finite element method (FEM) with the spectral method (SM). In particular, it inherits from the first one the geometrical flexibility, namely the capacity to take into account irregular interfaces and to adapt the mesh accordingly, while the spectral accuracy from the second one, i.e. few grid points per wavelength are necessary to maintain a minimal numerical dispersion due to the exponential convergence rate to the exact solution. In particular, the ability to provide an arbitrarily accurate numerical solution by simply increasing the polynomial approximation degree is one of the main feature of the SEM (Canuto, Hussaini, Quarteroni, and Zang [2007]).

The main aspect characterising the SEM is that the Lagrange polynomials, which are high-order (piecewise) interpolants, are samples of the finite-dimensional space taken at the Legendre-Gauss-Lobatto (LGL) quadrature points. To be more specific, this choice of polynomials allows to increase the numerical accuracy with respect to the low order polynomials shape functions usually used in finite element (FE) method. Indeed, the crucial aspect of the SEM is the capability of providing an arbitrary increase in accuracy simply enhancing the algebraic degree of these functions.
On the other hand, it is also possible to improve the accuracy of the numerical solution by refining the grid, as usually done for FE procedures.

Therefore, the SEM can be considered as a combination of the so-called $p$ (polynomial degree) and $h$ (grid size) versions of the FE method (see e.g. Babuska, Szabo, and Katz [1981], Babuška and Suri [1987]).

Consider now the Discontinuous Galerkin (DG) technique, which is a non-conforming discretization method (Arnold [1982], Wheeler [1978], Arnold, Brezzi, Cockburn, and Marini [2002], Hesthaven and Warburton [2007]). Referring to Mazzieri et al. [2013], there are several reasons to explain why it is coupled with SEM for the development of the code:

   &minus; complexity of the geometric constraints;

- seismic wave propagation issues due to a multiscale trait, which involves a large range of wavelengths;

- difficulty in keeping the computational cost low and, at the same time, dealing with complex three-dimensional heterogeneous setting.

The DG method, first introduced by Reed and Hill [1973] for neutron transport equation, was then developed for the treatment of parabolic end elliptic linear or non-linear partial differential equations (PDEs), for example by Arnold [1982] and Cockburn, Karniadakis, and Shu [2012]. After that, applications in parabolic equations by Falk and Richter [1999], Giraldo, Hesthaven, and Warburton [2002] and to acoustic or elastic wave propagation by Chung and Engquist [2006], Riviere, Shaw, Wheeler, and Whiteman [2003] followed.

Käser and Dumbser [2006] owe the first try of the high-order version of the DG approach for seismic wave propagation problems in heterogeneous media. However, this attempt was reviewed because of the high computational cost; in order to raise the computational efficiency and the performance of the method, a local time stepping scheme was introduced (Dumbser, Käser, and Toro [2007]). Finally, a similar DG method was exerted by Wilcox, Stadler, Burstedde, and Ghattas [2010] for the numerical solution of three-dimensional wave propagation problems in coupled elastic-acoustic media by using a velocity-strain formulation.

In order to cope the disadvantages of the above approaches and by virtue of the spectral element method, SPEED was implemented jointly between the Department of Structural Engineering and of Mathematics of Politecnico di Milano (SPEED: `https://speed.mox.polimi.it`).

To sum up, the code permit to perform simulation of seismic wave propagation in visco-elastic heterogeneous three-dimensional media, by exploiting non-conforming grids and/or variable approximation orders. One of the biggest advantages over the methods mentioned above, is the significant reduction of the computational cost per element due to its application only at a subdomain level (see discussion by De Basabe and Sen [2007] and De Basabe and Sen [2010]).

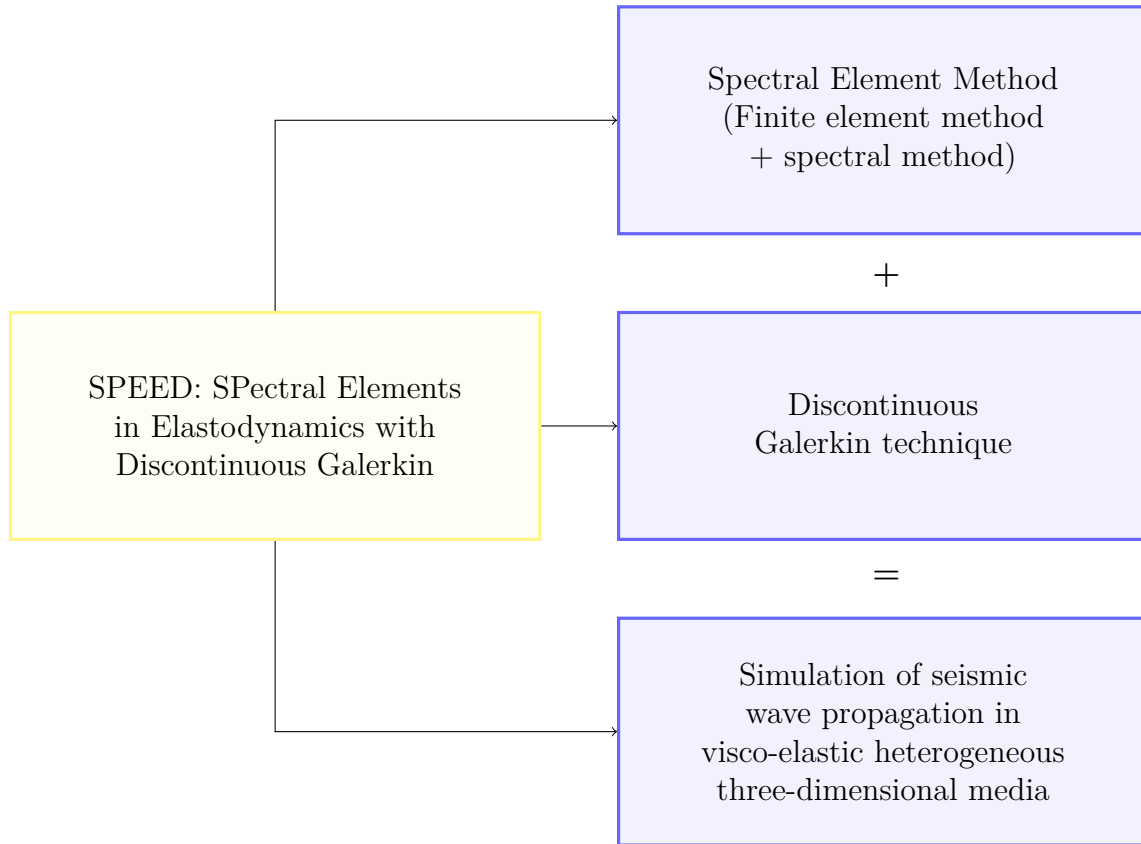In Figure 3.9 we report a scheme containing the main features of SPEED.

Figure 2.1: Graphical representation of the SPEED code's features.

## 2.2.   DRM and spectral elements

The formulation of DRM passed before from finite elements (Bielak et al. [2003], Loukakis [1988]) and then from spectral elements (Faccioli et al. [1997], Faccioli, Vanini, Paolucci, and Stupazzini [2005]). The implementation, instead, was first of type FE-SE within the code AHNSE (Advance Hybrid Numerical Solver for Elastodynamics) and, only after, was formulated using spectral elements by Scandella [2007] within the code GeoELSE, with particular attention to two-dimensional wave propagation analyses.

In this work, adapting the implementation done by Smerzini [2010] in the 3D parallel version of GeoELSE (*GeoELastodynamics by Spectral Elements*) (see Stupazzini, Zambelli, et al. [2005] and Scandella [2007]), the DRM was implemented in SPEED, for dealing with plane waves propagation problems.

The adoption of SEM leads to introduce some important features in order to clarify its formulation. Referring to Faccioli et al. [1997] and Smerzini [2010] for further details, we report here some of the basic aspects of the SEM taken into consideration.

Consider an arbitrary model with domain $\Omega \subset \mathbb{R}^d$, where $d$ is the dimension of the space, and boundary $\Gamma = \Gamma_N \cup \Gamma_D \cup \Gamma_{NR}$, where $\Gamma_D$ is the portion on which the displacement vector $\boldsymbol{u}$ takes prescribed values, $\Gamma_N$ the portion subjected to external forces (or tractions) and $\Gamma_{NR}$ is the fictitious non reflecting boundary.

Then, on this model, we can define the weak formulation of the differential wave equation through the principle of virtual work (see Zienkiewicz and Taylor [1989]). This can be written by multiplying the wave equation by an arbitrary displacement test function $\boldsymbol{v}(\boldsymbol{x})$ and, after that, integrating over the computational domain $\Omega$.

The problem can be formulated in the following way: find the displacement wavefield $\boldsymbol{u}(\boldsymbol{x}, t)$ such that $\forall t \in (0, T)$:

$$\frac{\partial^2}{\partial t^2} \int_\Omega \rho \boldsymbol{u} \cdot \boldsymbol{v} d\Omega + \int_\Omega \sigma_{ij}(\boldsymbol{u}) : \epsilon_{ij}(\boldsymbol{v}) d\Omega = \int_{\Gamma_N} \boldsymbol{t} \cdot \boldsymbol{v} d\Gamma + \int_{\Gamma_{NR}} \boldsymbol{t}^* \cdot \boldsymbol{v} d\Gamma + \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v} d\Omega,$$

$$\text{i, j} = 1,...,\text{d.} \tag{2.1}$$

In (2.1), $\rho(\boldsymbol{x})$ is the material density, $\sigma_{ij}$ is the stress tensor and $\epsilon_{ij}$ is the small strain tensor.

$\boldsymbol{v}(\boldsymbol{x})$ represents a generic function for admissible displacements, namely these are continuous through the domain. Moreover, the latter has been assumed to vanish over $\Gamma_D$, as well as making all the integrals in (2.1) finite.

$\boldsymbol{f}(\boldsymbol{x}, t)$ is the source term, given by a prescribed body force distribution. Finally, $\boldsymbol{t}(\boldsymbol{x}, t)$ is the vector of external tractions on $\Gamma_N$, while $\boldsymbol{t}^*(\boldsymbol{x}, t)$ is the traction vector on the absorbing boundary $\Gamma_{NR}$.

To describe this equation, additional details must be presented. First, suitable initial conditions must be given at time $t = 0$ for $\boldsymbol{u}$ and $\frac{\partial \boldsymbol{u}}{\partial t}$, $\forall \boldsymbol{x} \in \Omega$.

For what concerns the constitutive law, instead, we are under the hypothesis of linear elastic behaviour. Precisely, the *Hooke's law* describes the relation between stress and strain tensors with the displacement field:

$$\sigma_{ij} = \lambda \nabla \cdot \boldsymbol{u} \delta_{ij} + 2\mu \epsilon_{ij}(\boldsymbol{u}). \tag{2.2}$$

Where $\lambda$ and $\mu$ are the Lamé elastic coefficients, while $\delta_{ij}$ is the Kronecker Delta, such that:

$$\begin{cases} \delta_{ij} = 1, \text{ if } i = j, \\ \delta_{ij} = 0, \text{ otherwise.} \end{cases} \tag{2.3}$$

Instead, the strain tensor is expressed as follows:

$$\epsilon(\boldsymbol{u}) = \frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}). \tag{2.4}$$

To obtain an approximate numerical solution $\boldsymbol{u}_N$ at all grid points of the numerical model, we need to discretize equation (2.1) with respect to the space variables. In particular, the SEM is based on a generalized Galerkin approximation of the formulation (2.1).

$\boldsymbol{u}_N$, as well as the admissible displacements $\boldsymbol{v}_N$, are chosen in the space $V_N^0 := \{\boldsymbol{r} \in P^N$, set of continuous polynomials of degree $N$, such that $\boldsymbol{r}|_{\Gamma_D} = 0\}$.

Then, consider the semi-discrete formulation of equation (2.1): find $\boldsymbol{u}_N = \boldsymbol{u}_N(\boldsymbol{x}, t) \in V_N^0$ such that $\forall t \in (0, T)$:

$$\frac{\partial^2}{\partial t^2} \int_\Omega \rho \boldsymbol{u}_N \cdot \boldsymbol{v}_N d\Omega + \int_\Omega \sigma_{ij}(\boldsymbol{u}_N) : \epsilon_{ij}(\boldsymbol{v}_N) d\Omega =$$
$$\int_{\Gamma_N} \boldsymbol{t} \cdot \boldsymbol{v}_N d\Gamma + \int_{\Gamma_{NR}} \boldsymbol{t}^* \cdot \boldsymbol{v}_N d\Gamma + \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v}_N d\Omega, \tag{2.5}$$
$$\forall \boldsymbol{v}_N \in V_N^0.$$

It is now necessary to define the form of the approximate solution in a more precise way. However, since the use of global polynomials on the entire domain $\Omega$ brings many disadvantages (see Boyd [2001]), another approach is preferable.

We partition the computational domain $\Omega$ into disjointed quadrilaterals (in 2D) or hexahedra (in 3D), denoted by $\Omega_1, ..., \Omega_E$. Then, $\boldsymbol{u}_N$ and $\boldsymbol{v}_N$ are assumed to be piecewise continuous polynomials of degree $N$ in each of the domains $\Omega_e$, as well as being continuous at the interfaces.

Then, equation (2.5) must be computed for all the domains $\Omega_e$. In the latter, each volume integral is computed numerically by means of the Legendre-Gauss-Lobatto quadrature formula:

$$\int_{\Omega_e} f d\Omega \simeq \sum_j f(\boldsymbol{x}_j^{(e)}) \omega_j^{(e)}. \tag{2.6}$$

Where $\boldsymbol{x}_j^{(e)}$ is the vector of the coordinates of the LGL nodes in $\Omega_e$ and the $\omega_j^{(e)}$ are the corresponding weights, more precisely defined in Section 3.2 (see also Canuto, Hussaini, Quarteroni, Zang, Canuto, Hussaini, Quarteroni, and Zang [1988]).

The resulting scheme can now be defined as a pseudo-spectral domain decomposition method. For its implementation, we need to select a set of independent polynomials

$\boldsymbol{r}_j \in V_N^0$. After that, we can express the numerical solution in the following way:

$$\boldsymbol{u}_N = \boldsymbol{r}_j \tilde{\boldsymbol{u}}_j. \tag{2.7}$$

Where $\tilde{\boldsymbol{u}}_j$ are the unknowns of the problem. The subscript $j$ represents the generic node of the grid. As a consequence, (2.5) needs to be satisfied only by such $\boldsymbol{r}_j$, i.e. taking $\boldsymbol{v}_N = \boldsymbol{r}_j$, for all j.

The choice made for $\boldsymbol{r}_j$ is the set of Lagrange polynomials of degree $N$, which is equal to 1 at the $j^{th}$ LGL node and vanishes at all the others nodes (see Canuto et al. [1988]). For further details see also Section 3.2.

This choice mainly leads to two advantages: on one side, it allows to preserve the exponential accuracy of the spectral method and, on the other side, to minimize the computational effort, since it yields diagonal matrices (for further details see Smerzini [2010], Section 3.4).

To be more precise, consider $(\zeta_i)_{i=0,...N}$ the $N+1$ LGL integration points. The Lagrange interpolants are the $N+1$ one-dimensional polynomials $\psi_q$ of degree $N$ that satisfy the following relationship:

$$\psi_q(\zeta_i) = \begin{cases} 1, & \text{if } q = i, \\ 0, & \text{otherwise.} \end{cases} \tag{2.8}$$

Precisely, in the one-dimensional case, the $q^{th}$ Lagrange polynomial of degree $N$ on the reference domain $[-1, 1]$ can be defined as follows:

$$\psi_q(\zeta) = \frac{(\zeta - \zeta_0)...(\zeta - \zeta_{q-1})(\zeta - \zeta_{q+1})...(\zeta - \zeta_N)}{(\zeta_q - \zeta_0)...(\zeta_q - \zeta_{q-1})(\zeta_q - \zeta_{q+1})...(\zeta_q - \zeta_N)}. \tag{2.9}$$

Since we deal with the three-dimensional case, the polynomial basis is obtained as the product of three one-dimensional polynomials of degree N, namely $\psi_{ijk}(\boldsymbol{\zeta}) = \psi_i(\zeta)\psi_j(\eta)\psi_k(\gamma)$, with $i, j, k = 0, ..., N$ and being $\boldsymbol{\zeta}$ the generic coordinate vector on the reference cube $[-1, 1]^3$.

Then, we can define the generic $i^{th}$ displacement component $u_{iN}$ as:

$$u_{iN}(\boldsymbol{x}) = u_{iq}\psi_q(\boldsymbol{x}). \tag{2.10}$$

We underline that $u_{iq} = u_{iN}(\boldsymbol{x}_q)$, where $\boldsymbol{x}_q$ denotes the global grid of LGL nodes. Precisely, these nodes are the images of the nodes $\boldsymbol{\zeta}$ on the reference element through the mapping $\boldsymbol{F}^e$.

Considering again equation (2.5), performing numerical integration by the LGL quadrature rule yields, for each domain $\Omega_e$ and for each scalar component $i$, the following expression:

$$\frac{\partial^2}{\partial t^2}(\rho u_{iN}, v_i)_{\Omega_e} + (\sigma_{lm}(\boldsymbol{u}_N), \epsilon_{lm}(\boldsymbol{v}_N))_{\Omega_e} = (t_i, v_i)_{\Gamma_N^e} + (t_i^*, v_i)_{\Gamma_{NR}^e} + (f_i, v_i)_{\Omega_e}. \quad (2.11)$$

In the equation above, $u_{iN}, v_i, t_i, t_i^*$ and $f_i$ are, respectively, the scalar components of the vectors $\boldsymbol{u}_N, \boldsymbol{v}_N, \boldsymbol{t}, \boldsymbol{t}^*$ and $\boldsymbol{f}$. Moreover, we have used the notation $(f, g)_{\Omega_e}$ to express the LGL numerical approximation of the integral $\int_{\Omega_e} f g d\Omega$. Similarly, $(f, g)_{\Gamma^{(e)}}$ indicates the approximation of the boundary integral $\int_{\Gamma^{(e)}} f g d\Gamma$.

Finally, by introducing equation (2.10) into equation (2.11) and summing for all subdomains, we obtain a system of algebraic equations in the unknowns $\boldsymbol{u}_N$, equal to the displacements at nodes $q$. The system is the following:

$$\sum_{e=1}^{E} \frac{\partial^2}{\partial t^2}(\rho u_{iq}\psi_q, \psi_k^{(i)})_{\Omega_e} + \sum_{e=1}^{E}(\sigma_{lm}(\boldsymbol{u}_q\boldsymbol{\psi}, \epsilon_{lm}(\boldsymbol{\psi}))_{\Omega_e} =$$
$$\sum_{e=1}^{E}(t_i, \psi_k^{(e)})_{\Gamma_N^e} + \sum_{e=1}^{E}(t_i^*, \psi_k^{(i)})_{\Gamma_{NR}^e} + \sum_{e=1}^{E}(f_i, \psi_k^{(i)})_{\Omega_e}. \quad (2.12)$$

As a result, for each component of displacement, there are as many equations as the nodal points.

Note that each equation is obtained by imposing $v_i = \psi_k^{(i)}$, namely equal to the Lagrange polynomial associated to the $k^{th}$ node for the $i^{th}$ displacement component. From this formulation, we obtain the following system of ordinary differential equations with respect to time (time dependence has so far been ignored for the sake of simplicity):

$$\boldsymbol{M}\ddot{\boldsymbol{U}}(t) + \boldsymbol{K}\boldsymbol{U}(t) = \boldsymbol{F}^{ext}(t) + \boldsymbol{T}(t). \quad (2.13)$$

Where $\boldsymbol{U}$ is the vector of nodal displacements, $\boldsymbol{M}$ is the mass matrix and $\boldsymbol{K}$ is the stiffness matrix. Moreover, vectors $\boldsymbol{F}^{ext}$ and $\boldsymbol{T}$ represent respectively the contributions of external forces and tractions.

Thanks to this formulation, one important simplification occurs in the computation of the effective forces, previously computed in (1.9):

$$\boldsymbol{P}^{eff} = \begin{bmatrix} \boldsymbol{P}_i^{eff} \\ \boldsymbol{P}_b^{eff} \\ \boldsymbol{P}_e^{eff} \end{bmatrix} = \begin{bmatrix} 0 \\ -\boldsymbol{K}_{be}^{\Omega_e}\boldsymbol{u}_e^0 \\ \boldsymbol{K}_{eb}^{\Omega_e}\boldsymbol{u}_b^0 \end{bmatrix}. \quad (2.14)$$

Indeed, the terms related to extra-diagonal components of the mass matrix $\boldsymbol{M}$ are 0 because, as mentioned before, the SE mass matrix is diagonal. Moreover, without loss of generality, we assume the viscosity terms $C_{ij}$ in (2.14) to be understood.

Finally, looking at (2.14), we can state that the effective forces only depend on the stiffness matrix related to the DRM elements, namely the elements at the interface between $\Gamma_i$ and $\Gamma_e$ (see section 1.1), and on the free-field displacements obtained from the auxiliary problem.

### 2.2.1.  Stiffness matrix for DRM

One of the necessary computation required for the implementation of DRM is the evaluation of the stiffness matrix within the strip made of DRM elements.

First, define as $\boldsymbol{N}$ the matrix of shape functions used for the approximation of the solution and $\boldsymbol{B}$ the matrix containing the derivatives of the same shape functions.

Consider the expression of the internal forces related to a generic spectral element $el$ belonging to the mesh::

$$\boldsymbol{F_i} = \int_{V_{el}} \boldsymbol{B^T}\boldsymbol{\sigma}dV. \tag{2.15}$$

where $\boldsymbol{\sigma}$ is the stress vector and $V_{el}$ indicates the volume of the element $el$.

The formulation will be done in the framework of linear elasticity; therefore we can express the stress vector in terms of the linear elastic law by introducing the elastic constitutive matrix $\boldsymbol{E}$:

$$\boldsymbol{\sigma} = \boldsymbol{E}\boldsymbol{\epsilon}, \tag{2.16}$$

where $\boldsymbol{\epsilon}$ is the elastic strain vector, which in turn can be defined as a function of $\boldsymbol{E}$, $\boldsymbol{B}$ and the vector of displacements $\boldsymbol{u}$:

$$\boldsymbol{\epsilon} = \boldsymbol{B}\boldsymbol{u}. \tag{2.17}$$

By substituting (2.17) into (2.16) we get:

$$\boldsymbol{\sigma} = \boldsymbol{E}\boldsymbol{B}\boldsymbol{u}. \tag{2.18}$$

That leads to:

$$\boldsymbol{F_i} = \int_{V_{el}} \boldsymbol{B^T}\boldsymbol{E}\boldsymbol{B}\boldsymbol{u}dV. \tag{2.19}$$

Being $\boldsymbol{F_i} = \boldsymbol{K_{el}}\boldsymbol{u}$, from (2.19) is immediate to verify that the element stiffness matrix becomes:

$$\boldsymbol{K_{el}} = \int_{V_{el}} \boldsymbol{B^T}\boldsymbol{E}\boldsymbol{B}dV. \tag{2.20}$$

Now, we want to explicitly define the element matrix for the three-dimensional case:

$$K_{ij} = \int_{V_{el}} [\lambda \boldsymbol{\Psi}_{,i}^T \boldsymbol{\Psi}_{,j} + \mu \boldsymbol{\Psi}_{,j}^T \boldsymbol{\Psi}_{,i} + \mu \delta_{ij} \boldsymbol{\Theta}] dV. \tag{2.21}$$

Where $\lambda$ and $\mu$ are the Lamé elastic coefficients and $\delta_{ij}$ is the Kroneker delta. The term $\Theta$, instead, is given by the following expression:

$$\boldsymbol{\Theta} = \boldsymbol{\Psi}_{,1}^T \boldsymbol{\Psi}_{,1} + \boldsymbol{\Psi}_{,2}^T \boldsymbol{\Psi}_{,2} + \boldsymbol{\Psi}_{,3}^T \boldsymbol{\Psi}_{,3} = \sum_{k=1}^{3} \boldsymbol{\Psi}_{,k}^T \boldsymbol{\Psi}_{,k}. \tag{2.22}$$

Note that $\boldsymbol{\Psi}_{,i}$ expresses the differentiation of the vector of shape functions with respect to the $i^{th}$ axis (for the three-dimensional case $i = 1, 2, 3$).

The integrals in equation (2.21) are computed by means of the LGL quadrature rule (see Canuto et al. [2007]).

Moreover, we note that the product $\boldsymbol{\Psi}_{,i}^T \boldsymbol{\Psi}_{,j}$ is a matrix $\bar{K}$ of dimensions $[N+1, N+1]$, being $N$ the spectral degree, with the following form:

$$\bar{K}_{mn}^{(ij)} = \Psi_{m,i} \Psi_{n,j}. \tag{2.23}$$

Therefore, $\boldsymbol{K_{el}}$ turns out to be a symmetric matrix of dimensions $[3(N+1)^3, 3(N+1)^3]$, always referring to the three-dimensional case. Since equation (2.23) refers only to one spectral element, the application of the DRM in three dimensions requires overall the allocation of a matrix $\boldsymbol{K_{DRM}}$ of dimensions $[3Nel_{DRM}(N+1)^3, 3(N+1)^3]$, where $Nel_{DRM}$ is the number of elements composing the effective boundary.

For further details on the implementation of the matrix $\boldsymbol{K_{DRM}}$ in SPEED see Appendix A.

### 2.2.2.    Computation of the free-field motion

Another fundamental computation required by the method is the free-field displacement. The latter can be calculated following several ways, as for example:

- analytical methods for plane wave propagation with arbitrary angle of incidence. The DRM is coupled with analytical solutions for *P-SV-SH* plane waves propagating in horizontally layered media with arbitrary angle of incidence. Such a method enables for the minimization of errors caused by spurious reflections emanating from the absorbing boundaries, making it particularly appropriate for three dimensional applications (see section 4.1.1).

- Numerical or semi-analytical methods. For what concerns two-dimensional applications, the use of the method developed by Hisada [1994, 1995], Hisada and Bielak [2003] for computing the near-fault strong ground motions due to an extended kinematic fault in a viscoelastic horizontally layered half-space can often lead to good results (see e.g. Faccioli, Vanini, Villani, Cauzzi, Smerzini, et al. [2010], Scandella [2007]).

The implementation of the free-field motion and more details on the used approach can be found in section 3.3.

### 2.2.3. Computation of the effective forces

At this point, if we look at equation (2.14), we are able to compute the effective forces by means of the stiffness matrix (see equation (2.20)) and the free-field motion at the interface nodes.

From a practical point of view, consider to isolate from the discretized model a generic spectral element from the strip that forms the effective boundary elements, as shown in Figure 2.2. Denoting 1 and 2 the vertices of the element belonging to the inner boundary $\Gamma_i$, while 3 and 4 are the one belonging to the outer boundary $\Gamma_e$. The algorithm for the computation of the effective nodal forces consists of two steps:

1. The degrees of freedom at the two nodes in $\Gamma_i$, denoted by 1 and 2, are fixed. Then, the free-field displacements are imposed at the other two nodes, namely 3 and 4, with opposite sign. After that, the reaction forces $P_1^{eff}$ and $P_2^{eff}$ are computed at the interface nodes 1 and 2 (see Figure 2.3a).

2. The degrees of freedom are now fixed at the two nodes in $\Gamma_e$, namely 3 and 4. Then, the free-field displacements are imposed at nodes 1 and 2, with opposite sign. Finally, the reaction forces $P_3^{eff}$ and $P_4^{eff}$ are computed at the fixed nodes (see Figure 2.3b).

The reaction forces calculated at steps 1 and 2 correspond to the effective nodal forces and can be calculated in the following way:

$$\boldsymbol{P}^{eff} = \begin{bmatrix} \boldsymbol{P_1^{eff}} \\ \boldsymbol{P_2^{eff}} \\ \boldsymbol{P_3^{eff}} \\ \boldsymbol{P_4^{eff}} \end{bmatrix} = \begin{bmatrix} -K_{13}\boldsymbol{u_3^0} - K_{14}\boldsymbol{u_4^0} \\ -K_{23}\boldsymbol{u_3^0} - K_{24}\boldsymbol{u_4^0} \\ K_{31}\boldsymbol{u_1^0} - K_{32}\boldsymbol{u_2^0} \\ K_{41}\boldsymbol{u_1^0} - K_{42}\boldsymbol{u_2^0} \end{bmatrix}. \tag{2.24}$$

From this derivation it is evident that the computation of effective forces should be performed only once unless the seismic source changes.

Moreover, the approach is suitable for non-linear analyses of structures because the effective forces disappear for nodes outside the strip of elements next to the boundary $\Gamma_i$. As a result, no forces act inside the interior region of the localized structure.
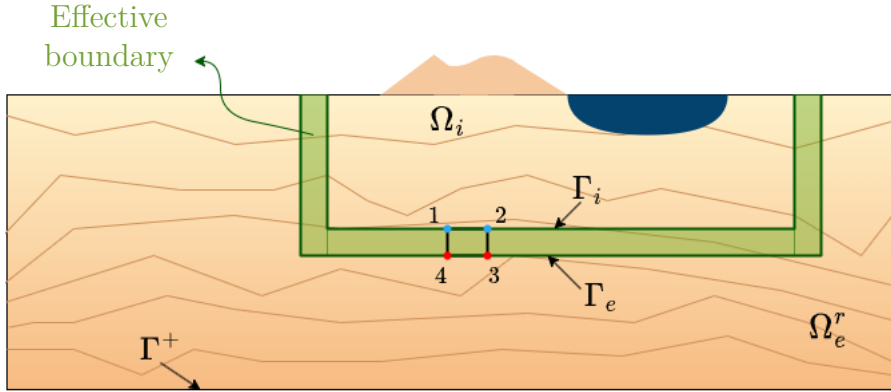


Figure 2.2: Computation of the effective nodal forces at the interface elements. Inside the reduced domain, the artificial green strip represents the spectral elements at the effective boundary. One of the spectral element, delimited by the four vertices 1 and 2, belonging to the internal boundary $\Gamma_i$, 3 and 4, belonging to the external boundary $\Gamma_e$, is idealistically isolated for the sake of explanation. Adapted from Smerzini [2010].
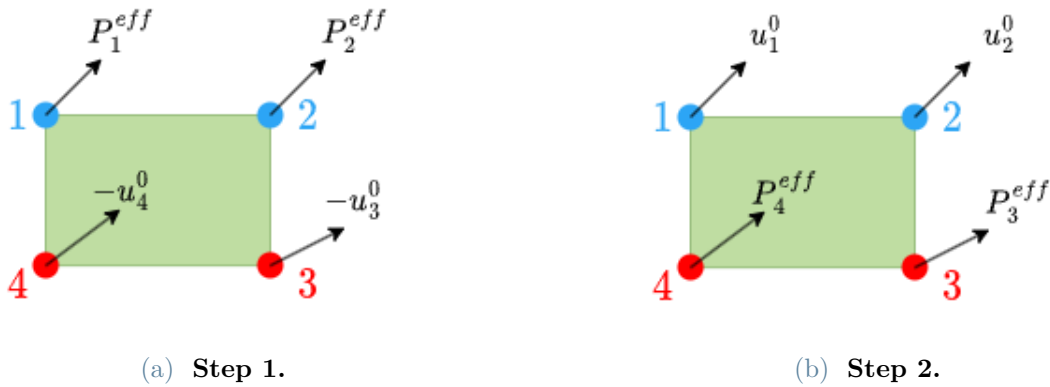


Figure 2.3: Two-steps procedure for the computation of the effective forces. **Step 1**: the degrees of freedom are fixed at nodes 1 and 2, while the free-field displacements $u_3^0$ and $u_4^0$ are imposed with opposite sign at nodes 3 and 4. The reaction forces $P_1^{eff}$ and $P_2^{eff}$ are computed. **Step 2**: the degrees of freedom are fixed at nodes 3 and 4, while the free-field displacements $u_1^0$ and $u_2^0$ are imposed at nodes 1 and 2. The reaction forces $P_3^{eff}$ and $P_4^{eff}$ are computed.

## 2.3. DRM for plane wave propagation

In this section we refer to the algorithm implemented by Smerzini [2011] in the framework of seismic wave propagation analyses. In particular, this approach is aimed at the study of arbitrarily complex geological structures under the action of plane waves with possibly any angle of incidence. The code, originally implemented in GeoELSE, is introduced here in SPEED.

Before exposing this procedure, that couples DRM with analytical methods for plane wave propagation, a brief overview of the previous version of the code will be given. In the latter, plane waves are instead implemented as a distribution of forces.

### 2.3.1. Plane waves as a distribution of body forces

The plane wave source was originally implemented as a force distribution along the wavefront that was computed in order to impose a specific displacement time history, $\bar{u}(t)$. It has been demonstrated that this is effectively obtained by including in the equilibrium equation an external force term that is proportional to the time derivative of the imposed displacement $\bar{u}(t)$.

Following Faccioli et al. [1997], let us consider the wave propagation equation in its one dimensional form:

$$\rho \frac{\partial^2 u}{\partial t^2} = (\lambda + 2\mu)\frac{\partial^2 u}{\partial x^2} + f(x,t). \tag{2.25}$$

Where $\lambda$ and $\mu$ are the Lamè coefficients and $f(x,t)$ is an equivalent uniform body force distribution corresponding to a vertically propagating displacement plane wavefront $\bar{u}(t)$.

In a three dimensional infinite homogeneous medium, the displacement in the i-direction generated by a uniform distribution of body forces $\boldsymbol{f}(x,t) = \phi(t)\delta(x - x_0)\boldsymbol{e_i}$, being $\delta$ the spatial Dirac delta, acting on plane $x = x_0$, is given by:

$$u(x,t) = \frac{1}{2\rho c}H\left(t - \frac{|x - x_0|}{c}\right)\int_0^{t - \frac{|x - x_0|}{c}} \phi(\tau)d\tau. \tag{2.26}$$

Where $\rho$ is the soil mass density, $c$ is the propagation velocity and $H(\cdot)$ the step function.

Hence, in order to propagate an assigned displacement waveform $\bar{u}(t)$, we determine the time dependence of the force distribution by differentiating (2.26) with respect to time and evaluating the result at $x = x_0$ (see Graff [2012]). This leads to:

$$\phi(t) = 2\rho c\frac{\partial \bar{u}}{\partial t}. \tag{2.27}$$

These equivalent body forces are applied at the nodes of a horizontal grid line near the bottom boundary of the computational domain. Precisely, the nodal points must lie inside a horizontal layer of quadrilaterals (in 2D) or hexahedral (in 3D) elements with constant height $\bar{L}_x$. Indeed, the geometry must remain constant along the x axis since the numerical approximation seeks at resembling the 1D plane wave propagation of equation (2.25) This is also the reason why triangles and tetrahedra are discarded.

Such a strategy has two significant drawbacks:

- unless the free-surface is rotated, it does not apply to oblique plane waves propagation;

- the absorbing lateral boundaries, whose efficiency is minimal for lateral incidence, badly affect wave propagation.

## 2.3.2.   Plane waves as an effective seismic excitation

In the chosen approach, SPEED has been provided with a set of subroutines that can compute the effective nodal forces from the analytical free-field solution for *P-SV-SH* plane wave propagation across horizontally stratified soil media at any angle of incidence. More precisely, the analytical solutions are computed through the Haskell-Thomson (H-T) propagation matrix method (Haskell [1990], Thomson [1950]).

As depicted in Figure 2.4, the steps concerning plane wave propagation analyses performed in SPEED with DRM are the following:

1. definition of the input parameters:

    - time dependence of the imposed plane displacement wavefront. In this thesis, a Ricker wavelet is always used for validation, but any arbitrary displacement time history could be used without loss of generality;

    - mechanical properties of the background geological model that constitute the auxiliary problem. We assume the latter to be horizontally layered by $N_l$ layers. Each layer $l$, $l = 1, ..., N_l$, is identified by thickness $H_l$, soil mass density $\rho_l$, $S-$ velocity $V_{S_l}$, $P-$ wave velocity $V_{P_l}$ and quality factors $Q_{S_l}$ and $Q_{P_l}$;

    - plane wave source: type of polarization of seismic input (in-plane, $P$ or $SV$, or out-of-plane, $SH$ waves);

    - angle of incidence $\gamma$ ($\gamma = 0$ indicates vertical incidence, then $\gamma$ positive is measured in clockwise sense), in- or out- cropping input.

2. Computation of the effective boundary nodes in which the effective boundary forces will be evaluated, namely in the strip of spectral elements that constitutes the effective boundary. These nodes are Legendre-Gauss-Lobatto (LGL) nodes and their number depends on the spectral degree $N$;

3. Definition of a one-dimensional reference soil profile. Consider $N_f > N_l$ fictitious layers, each one described from its own mechanical properties $H_f$, $\rho_f$, $V_{S_f}$, $V_{P_f}$, $Q_{S_f}$ and $Q_{P_f}$. Every layer is delimited by two adjacent non-coinciding LGL nodes in the vertical ($z$) direction. Those nodes located inside the $l^{th}$ layer of the actual geological model (see point 1) have the same properties. Note that nodes located at the same height $z$ but different $x$ or $y$ coordinates are not duplicated within the soil profile, but only nodes at different depths are considered.

4. Resolution of the Haskell-Thomson (H-T) matrix method for the 1D soil profile determined at previous point. As a result, the free-field displacement time histories at the $(N_f + 1)$ interfaces are computed.
   It is worth underlying that H-T matrix method provides results in the frequency domain. Therefore, the code must take into account a specific convolution process for the analytical transfer function with the imposed displacement time history.

5. Computation of the free-field displacements at the entire set of effective boundary nodes of the reduced problem. In order to "distribute" the free-field displacement solution to all spectral nodes where the equivalent seismic excitation must be applied, allowing for any direction of propagation, a proper translation in the time domain is achieved. To be more precise, let us recall the solution of the scalar Helmholtz wave equation in three-dimensional space: $u_o(t - \boldsymbol{n} \cdot \boldsymbol{r}/V)$. $n = \begin{bmatrix} n_x & n_y & n_z \end{bmatrix}$ is the vector normal to the plane wave front, $r = \begin{bmatrix} x & y & z \end{bmatrix}$ is the position vector and $V$ is the propagation velocity. Now, calling $x_0$ and $y_0$ the reference coordinates with respect to which the temporal time shift $\tilde{t}$ is calculated, the free-field displacement of the nodes located at the same depth $\bar{z}$ can be computed as follow:

$$u_0(x_k, y_k, \bar{z}; t) = u_0(x_0, y_0, \bar{z}; t + \tilde{t}), \quad \text{with } \tilde{t} = \frac{n_x \cdot (x_k - x_0) + n_y \cdot (y_k - y_0)}{V}.$$

Of course, the choice of the reference coordinate depends on the prescribed direction of the incident plane wave. Moreover, $n_x$ and $n_y$ can be expressed as a function of the angle of incidence $\gamma$ and the azimuth $\phi$ of the plane wavefront, namely:

$$n_x(\gamma, \phi) = \sin \gamma \cdot \cos \phi, \quad n_y(\gamma, \phi) = -\sin \gamma \cdot \sin \phi.$$

6. Computation of the effective nodal forces $\boldsymbol{P}^{eff}$ exploiting the free-field displacements obtained at the previous step. Then the resolution of the wave propagation analysis takes place (reduced model).
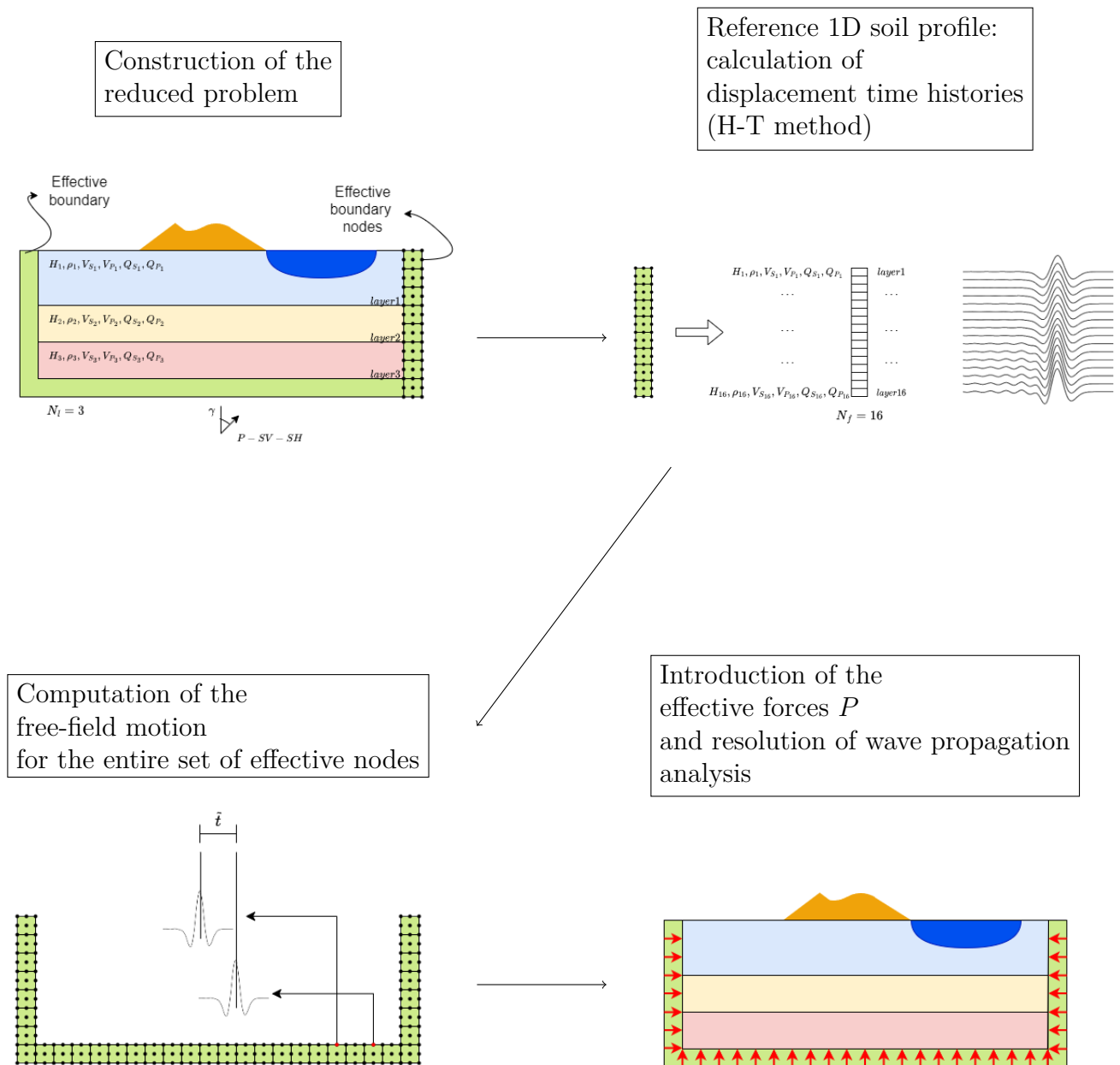


Figure 2.4: Plane wave propagation analyses in arbitrary complex media: sketch of the implementation. Adapted from Smerzini [2010].

In particular, the first five steps, implemented in Matlab, are described in more details in Chapter 3, while the last step, the only one directly implemented in SPEED, can be found within the SPEED subroutine *MAKE_LOAD_MATRIX.f90*.

As a final remark, it should be noted that this method is particularly appropriate for the applications that are frequently used in engineering practice. Indeed, a similar procedure is implemented in the finite difference commercial code FLAC (Fast Lagrangian Analysis of Continua, `https://www.itascacg.com/software/FLAC3D`, Board [1989]) for advanced continuum modeling of geotechnical analysis of rock, soil and structural support in 2D and 3D, even though it is only applicable to vertical plane waves.

The use of DRM for modeling plane wave propagation leads to several advantages with respect to the conventional approach, such as:

- The spurious reflections due to the absorbing boundary conditions (ABCs) applied on the external boundary of the computational domain are minimized. Beyond their efficacy, the ABCs are not exempt from numerical errors: non physical reflections from the external boundary towards the localized structure of interest may affect seismic wave propagation in the region under study. In general, absorbing conditions perform well for vertical wave incidence ($\gamma = 0°$), while spurious effects tend to increase when $\gamma$ increases. Precisely, in the standard approach, namely without DRM, (see Section 4.1.1), non reflecting conditions act on the total wavefield, while in the sub-structuring approach boundary conditions on $\Gamma^+$ act only on the diffracted wavefield. As a result, the ABCs perform better, and it is no longer essential to locate the exterior boundary distant from the region of interest.

- Coupling SPEED with DRM allows the treatment of non orthogonal plane wave incidence to the free surface. The latter can be done by simply changing the auxiliary problem according to the assumed seismic excitation.
  Thanks to this, such an approach is much more versatile in doing parametric studies with respect to the seismic input.

# 3 | Input files generation

In this section, we go over the steps necessary to apply the plane wave propagation analyses. Precisely, we focus on the generation of the input files necessary for the use of the code SPEED coupled with the Domain Reduction Method.

This process can be summarized as follows:

1. Create the spectral element (SE) reduced model using the software CUBIT. In order to apply the effective forces to the irregular structure of interest, the computational grid must include an appropriate strip of spectral elements, i.e. the effective DRM boundary.

2. Exploit the Matlab script *read_grid_test.m* that in turn involves the Matlab function *lgl_block.m* to be able to obtain the coordinates of the LGL (Legendre - Gauss - Lobatto) nodes in the effective DRM boundary. These are listed in the file LGL_coords.txt.

3. Run the *input_4else_3D.m* interface Matlab program. This file allows to create the PDRM nodes (effective DRM nodes) and FDRM (free-field displacements computed at PDRM) functions which are subsequently inserted in the file *.mate.

4. 3D SE simulation by SPEED: the displacement time histories at selected sites of the model under study are provided as output in the folder `MONITORS`: file MONI-TORXXXXXX.D.

The algorithm is schematised in Figure 3.1.

Filename.cub → Filename.mesh



Generation of the SE reduced model using CUBIT

Matlab script
*read_grid_test.m* + Matlab function *lgl_block.m*

LGL_coords.txt

Interface Matlab program
*input_4else_3D.m*

Filename.mate
with PDRM nodes
FDRM functions

3D simulation on SPEED

MONITORXXXXXX.D
at selected sites
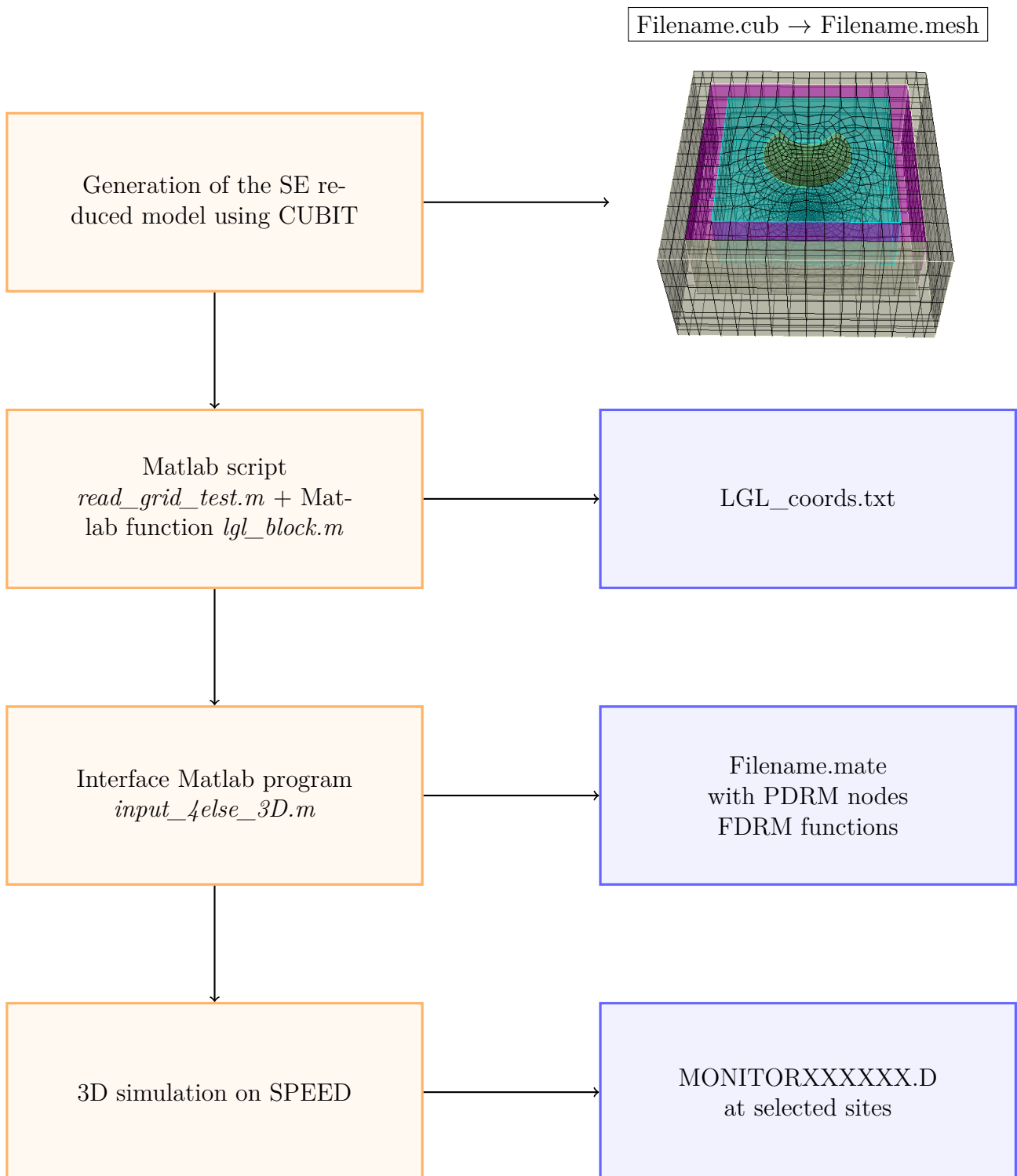
Figure 3.1: Main steps for the use of SPEED coupled with the Domain Reduction Method. Starting from the generation of the mesh file using the software CUBIT, up to the generation of the monitor at the selected sites.

# 3.1.  Spectral Elements mesh generation using CUBIT

It is necessary to create an unstructured mesh of spectral elements for the spatial discretization of an arbitrarily complex model that honors the topography of the earth and includes, in addition to an extended seismic fault, localized irregular structures (like alluvial basins) or civil engineering elements of various types (viaducts, bridges, underground structures). This discretization takes place by means of quadrilaterals elements (two-dimensional case) and hexahedral elements (three-dimensional case).

For the simulations in SPEED, the computational grid is generated making use of the software CUBIT (`https://cubit.sandia.gov/`). It includes a collection of strong and sophisticated meshing techniques created to address challenging unstructured meshing issues.

Once the geometry is set on CUBIT, we have to define the mesh on it. In Section 3.1.1 we report some general criteria for spatial discretization.

After that, one has to define the different blocks that characterise the domain under study. These blocks can be of different type: soil layers, DRM boundary elements, absorbing boundary conditions etc. The geological features of each block are then declared inside the .mate file. In Figure 3.2 we present an example of a homogeneous model divided in blocks.

After the generation of the mesh, in order to obtain the mesh file necessary to run SPEED for the desired test, it is necessary to follow the following steps:

- Export the file *test.cub* containing the geometry of the test in the Exodus format in order to obtain *test.e*;

- Convert *test.e* into an ASCII file *test.txt* exploiting the filter ncdump.exe;

- Run the Matlab file *MAKE_MESHinput.m* in order to rewrite the file *test.txt* in the format *test.mesh*. The mesh file of the test contains all the information about the geometry and grid in which the test has been subdivided, see Figure 3.3.
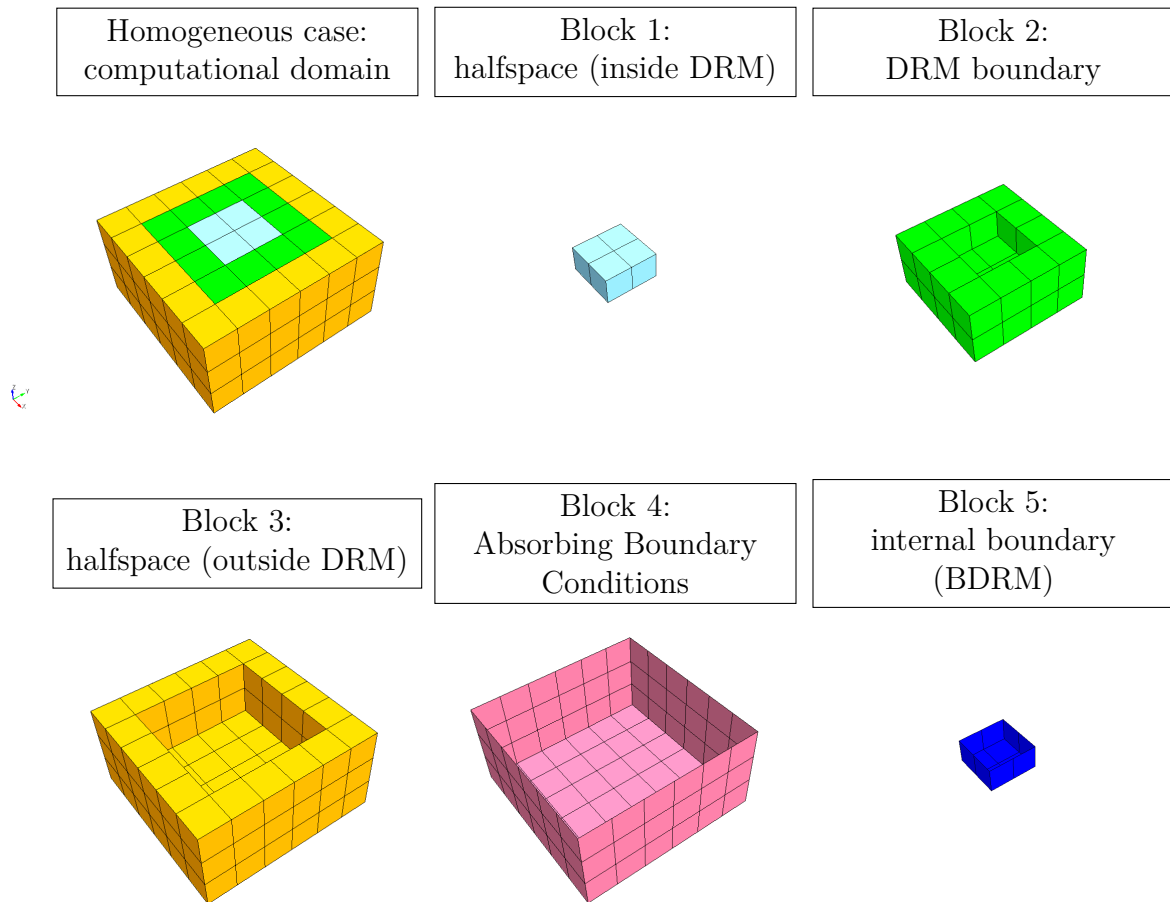
Figure 3.2: Division into blocks of a simple computational case. Here, there is no difference in the materials of the blocks. This division allows the simulation with SPEED coupled with DRM method, in particular the presence of the green strip of DRM elements (block2). Note that there is a block also for absorbing boundary conditions applied to the external boundary of the computational domain (block 4) and for the DRM boundary elements (block 5).

```
# number of total nodes \number of total elements
196 228 0 0 0

# number of the node \x, y, z coordinates
1 +2.5000000e+02 -2.5000000e+02 +0.0000000e+00
2 +0.0000000e+00 -2.5000000e+02 +0.0000000e+00
3 +0.0000000e+00 +0.0000000e+00 +0.0000000e+00
...

# number of the element \block it belongs to \type of element \vertices
1 4 quad 18 12 10 17
2 4 quad 12 11 9 10
3 4 quad 16 15 13 14
...
1 1 hex 1 2 3 4 5 6 7 8
2 1 hex 2 9 10 3 6 11 12 7
3 1 hex 4 3 13 14 8 7 15 16
...
```

Figure 3.3: Example of mesh file. In this file are listed all the nodes delimiting the elements together with their coordinates. Then, for each block, all the elements are listed specifying their type (namely quadrilateral / hexahedral) and the four / eight vertices that delimit them.

### 3.1.1.   Spatial discretization criteria

It is important to take into consideration some general criteria for spatial discretization. In particular, Nyquist theorem (see Por, Van Kooten, and Sarkovic [2019]) states that if we have to examine a signal or a waveform, we would need samples with a frequency larger than twice the maximum frequency contained in the signal, namely $f_{sample} \geq 2f_{max}$. This is necessary in order to avoid aliasing phenomena, due to an undersampled signal.

From this, it is possible to derive the maximum spatial sampling step for a suitable discretization of the computational grid. Always to avoid aliasing phenomena, it follows from the Nyquist theorem that the minimum number of points to represent the minimum wavelength, called $G_\lambda$, must be equal to 2, and this translates into terms of maximum spatial sampling step as:

$$\Delta x \leq \frac{\lambda_{min}}{2}, \quad \text{with} \quad \lambda_{min} = \frac{V_{min}}{f_{max}}, \tag{3.1}$$

where $V_{min}$ is the minimum wave propagation velocity and $f_{max}$ is the maximum frequency to propagate.

Despite this request parameter, Faccioli et al. [1997] provide another criteria in order to obtain better results: the mesh should be designed in order to ensure more than 2.5 points per minimum wavelength, namely $G_\lambda \geq 2.5$, for a homogeneous environment, while $G_\lambda \geq 3.5 - 4$ for a strongly heterogeneous one. To give an idea of these values, consider that low order finite elements (FE) schemes require in general $G_\lambda \approx 10 - 12$.

Note that $\Delta x$ in equation (3.1) is equal to the spatial step of the mesh for finite differences (FD) method and it is constant. However, for spectral elements (SE) method $\Delta x$ is not kept constant, but corresponds to an average inter-node distance between two adjacent LGL nodes. As a general rule, we can assume characteristic dimension $\Delta l$ of the spectral element given by the following expression:

$$\Delta l \leq \frac{\lambda_{min}}{G_\lambda} N. \tag{3.2}$$

Here $G_\lambda$ is equal to 4 and $N$ is the degree of the interpolant polynomial. Note that this rule is valid for a relative low spectral degree, like $N \leq 5$.

Therefore, the dependence of the discretization on the spectral degree constitutes a great advantage in using this method: the accuracy of the method can be enhanced not only by decreasing $\Delta l$, namely by refining the grid, but also by increasing $N$. More precisely, for the user it is sufficient to impose the spectral degree without changing the mesh. Then, the code will calculate the new LGL nodes consequently.

## 3.2.  LGL nodes generation

The first step consists in creating the list of the Legendre-Gauss-Lobatto (LGL) nodes. These nodes are necessary for the computation of the effective forces.

Before moving on to the calculation of LGL nodes, we make some theoretical remarks on their construction.

First of all, let us consider the expression of the Legendre polynomials in their recursive expression (Quarteroni, Sacco, Saleri, and Gervasio [2014]):

$$\begin{cases} L_0(x) = 1, \\ L_1(x) = x, \\ L_{k+1}(x) = \frac{2k+1}{k+1} x L_k(x) - \frac{k}{k+1} L_{k-1}(x), \quad k = 1, 2, \dots . \end{cases} \tag{3.3}$$

These polynomials are defined in the domain $[-1, 1]$ and it is easy to demonstrate that $L_k \in P_k$ for every $k = 1, 2, \dots$, where $P_k$ is the space of polynomials with order $k$.

In one dimension, the LGL nodes $\zeta_i$ are defined as the roots of the first space derivatives of the Legendre polynomial of degree $N$, $L_N$ (see Abramovitz and Stegun [1966]), i.e.:

$$\begin{cases} i = 0 : \zeta_0 = -1; \\ i = 1, ..., N - 1 : \zeta_i = L'_N(\zeta_i) = 0; \\ i = N : \zeta_i = 1. \end{cases} \tag{3.4}$$

Therefore, for a Legendre polynomial of degree $N$, there are $N + 1$ LGL nodes in each space dimension: $N - 1$ are the zeros of $L'_N$, in addition to the two extrema.

In order to obtain the LGL nodes in higher dimensional spaces, it is necessary a tensorization process. The latter is one of the reason for which in spectral element method (SEM) quadrilateral and hexahedral elements are preferred to triangular and tetrahedral elements.

For computing the LGL nodes for the entire strip of DRM elements, which are three-dimensional physical hexahedral elements, the user has to run the Matlab script *read_grid_test.m*.

First of all, one has to set the spectral degree $N \geq 1$. Then, the script receives as input two matrices:

1. `coords`: contains in the first column the number of all the nodes in the entire domain in ascending order, and in the last three columns the three respective coordinates. This matrix has dimensions $[nn \times 4]$, where $nn$ indicates the number of total nodes.

2. `drm_blocks`: contains in the first column the numbers of all hexahedral elements that make up the DRM boundary, always in ascending order. The other eight columns are instead characterised by the eight nodes that correspond to the vertices of the respective hexahedron. This matrix has dimensions $[Nel_{DRM} \times 9]$, being $Nel_{DRM}$ the number of DRM elements.

These two matrices are used in order to extract, for each DRM element, the coordinates of its eight vertices.
These coordinates are necessary to generate the parameters needed for mapping the LGL nodes into the desired DRM element. Indeed, exploiting the Matlab function *lgl_block.m* we obtain the LGL nodes calculated for the reference hexahedron element that in fact must then be able to be mapped on the generic hexahedron that makes up the DRM boundary (see Casadei and Gabellini [1998] for more details). The reference hexahedron element in the three-dimensional case is nothing but the cube having dimensions $[-1, 1]^3$.

In Figure 3.4 we report, as an example, the computation of LGL nodes for the reference element having $N = 3$. This leads to the computation of $N + 1 = 4$ nodes for each interval span $[-1, 1]$ and, consequently, to the creation of $4^3 = 64$ LGL nodes.

Note that $\xi$, $\eta$ and $\zeta$ are used to express the reference axis in the reference domain, while $x$, $y$ and $z$ correspond to the axis in the physical domain.
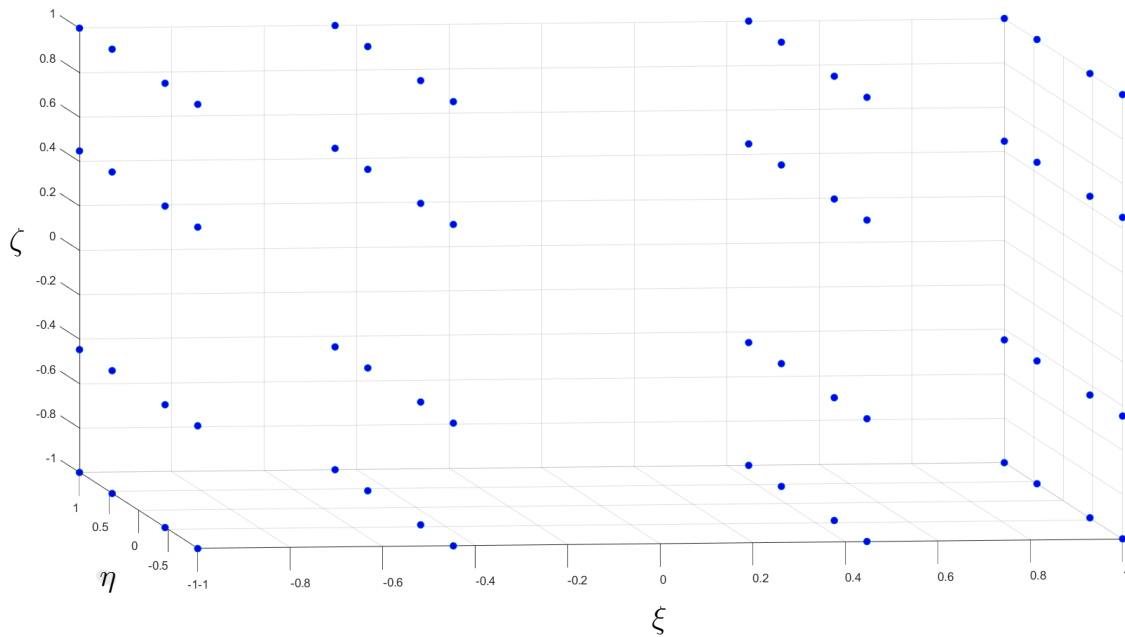


Figure 3.4: Computation of the LGL nodes for the reference hexahedron, that corresponds to the cube having dimensions $[-1, 1]^3$. In this example, the spectral degree $N$ is equal to 3. This leads to the computation of $4^3 = 64$ LGL nodes.

In algorithm 3.1 we report the Matlab code described util the call of the Matlab function *lgl_block.m* at line 22.

**Listing 3.1:** read_grid_test.m

```
1 N = 4; % spectral degree
2 coords = [1   +2.5000e+02   -2.5000e+02   +0.0000e+00 .. ];
3 drm_blocks = [1205   729   736   8   5   733   738   4   1 .. ];
4
5 % extract the coordinates of each vertex of every DRM element
6 for i = 1:length(drm_blocks)
7     x1 = coords(drm_blocks(i,2),2);
```

```matlab
 8      y1 = coords(drm_blocks(i,2),3);
 9      z1 = coords(drm_blocks(i,2),4);
10      ...
11      z8 = coords(drm_blocks(i,9),4);
12 end
13
14 % for every block I need the coefficients a1, a2,..., h3
15 % in order to pass from the reference to the physical domain
16 a1 = - x1 + x2 - x3 + x4 + x5 - x6 + x7 - x8;
17 a2 = - y1 + y2 - y3 + y4 + y5 - y6 + y7 - y8;
18 a3 = - z1 + z2 - z3 + z4 + z5 - z6 + z7 - z8;
19 ...
20 h3 = z1 + z2 + z3 + z4 + z5 + z6 + z7 + z8;
21
22 % call lgl_block.m
23 [coor_ref] = lgl_block(N+1);
24 coor_phys_block = zeros(length(coor_ref),3);
25
26 for j = 1:length(coor_ref)
27      xi = coor_ref(j,1);
28      eta = coor_ref(j,2);
29      zeta = coor_ref(j,3);
30      x = (1/8)*(a1*xi*eta*zeta + b1*xi*eta + c1*eta*zeta + d1*
          xi*zeta + e1*xi + f1*eta + g1*zeta + h1);
31      y = (1/8)*(a2*xi*eta*zeta + b2*xi*eta + c2*eta*zeta + d2*
          xi*zeta + e2*xi + f2*eta + g2*zeta + h2);
32      z = (1/8)*(a3*xi*eta*zeta + b3*xi*eta + c3*eta*zeta + d3*
          xi*zeta + e3*xi + f3*eta + g3*zeta + h3);
33      coor_phys_block(j,1) = x;
34      coor_phys_block(j,2) = y;
35      coor_phys_block(j,3) = z;
36 end
37      coor_phys_tot = [coor_phys_tot ; coor_phys_block];
38 end
39 C = uniquetol(coor_phys_tot,'ByRows',true); % returns the
      unique rows of coor_phys_tot
```

Then, as said before, the function *lgl_block.m* returns the LGL nodes relative to the reference hexahedron element. This is done by exploiting the Matlab function *xwlgl.m* created by Gervasio [2007]. This function returns the $np = N + 1$ weigths and nodes of the corresponding Legendre Gauss-Lobatto quadrature formula in the reference interval $[-1, 1]$. After that, *lgl_block.m* function allows to generate all LGL nodes composing the hexahedron having as dimensions $[-1, 1] \times [-1, 1] \times [-1, 1]$. This is achieved as reported in 3.2.

**Listing 3.2:** lgl_block.m

```matlab
function [coor] = lgl_block(np)
% Input: np = N + 1 number of nodes
[x,~] = xwlgl(np);
[y,~] = xwlgl(np);
[z,~] = xwlgl(np);
coor = zeros(np^3,3); % coordinates matrix of LGL nodes
cont1 = 1;
cont2 = 1;

% assign x:
for k = 1:np*np:np^3
    coor(k:k+np^2-1,1) = x(cont1);
    cont1 = cont1 + 1;
end

% assign y:
for k = 1:np:np^3
    if (rem(cont2,np) == 0)
        coor(k:k+np-1,2) =  y(np);
    else
        coor(k:k+np-1,2) =  y(rem(cont2,np));
    end
    cont2 = cont2 + 1;
end

% assign z
for k = 1:np^3
    if (rem(k,np) == 0)
```

```
29        coor(k,3)  =   z(np);
30     else
31        coor(k,3)  =   z(rem(k,np));
32     end
33 end
34 end
```

After the call of the function *lgl_block.m*, inside *read_grid_test.m*, the LGL reference coordinates are mapped into the physical DRM element. This last computation is reported in Algorithm 3.1 starting from line 23.

Finally, the script returns as output the file LGL_coords.txt containing the coordinates of the LGL nodes in the interested strip of DRM elements.

## 3.3. Computation of PDRM and FDRM

In this section we explain how PDRM (effective DRM nodes) nodes and FDRM (free-field displacements computed at PDRM) functions are calculated.
To be more specific, pdrm.dat is the file containing the nodal (LGL) coordinates of DRM strip of elements, while fdrm.dat is the file containing the displacement time histories computed at the DRM boundary nodes.

The algorithm implemented in Matlab by Smerzini [2010] and requires the following steps:

1. Matlab script *input_4PSVQ_3D.m*.
   **Input files**:

   - LGL_coords.txt: contains the list of coordinates $(x, y, z)$ of all LGL nodes belonging to the strip of DRM elements as calculated in the previous section;

   - layers_plax.txt: It is characterised by the mechanical properties of the horizontal layers that constitute the background geological model that forms the auxiliary problem. These properties are: layer number, vertical upper bound $z_{sup}(m)$, vertical lower bound $z_{inf}(m)$, density $\rho(\frac{t}{m^3})$, S- velocity $V_s(\frac{m}{s})$, quality factor $Q_s$, P- velocity $V_p(\frac{m}{s})$, quality factor $Q_p$.

   **Output files**:

   - lista_ord_all_coor_plax.txt: list of LGL node coordinates in descending order;

- PSVQ_layers_plax.txt: corresponds to the reference soil profile used for 1D linear visco-elastic analyses. Every layer is delimited by two adjacent LGL nodes in $z$ direction.

2. *PSVQ_drm.m*: takes as **input** the file test_3d.ini, that contains the input parameters for 1D linear visco-elastic seismic wave propagation. In Figure 3.5 is depicted an example of this file and its features. It is always referred to the homogeneous case of Figure 3.2. Note that the construction of this file takes advantage from the file PSVQ_layers_plax.txt, previously computed, to introduce the reference soil profile.

The **output** of this script is the file test_3d.out that contains the free-field displace-

```
# type of wave (0=SH,1=P,2=SV) - angle of incidence
0.0 0.0
# Number of layers (excluding the halfspace) - plot layers (1=sup.;0=all) IQ=2 Q=q*f
4 0 2
# depth (m) - ρ - V_S - Q_S - V_P - Q_P
125.000 2.000 300.000 1000.000 520.000 2000.000
125.000 2.000 300.000 1000.000 520.000 2000.000
125.000 2.000 300.000 1000.000 520.000 2000.000
125.000 2.000 300.000 1000.000 520.000 2000.000
250.000 2.000 300.000 1000.000 520.000 2000.000
1 # 1 = read accelerogram
ricker.1
# Number of lines to skip - number of columns
0 1
# dt - npoints - iconv(2 on surf.) - idec (=2 deconvolution) - itap (=1 tapering)
0.02 501 1 1 0
```

Figure 3.5: Example of file test_3d.ini. In this file are listed all the input parameters for one dimensional linear visco-elastic seismic wave propagation analyses.

ments at all interfaces of the 1D soil profile defined at the previous point. These displacements are calculated from the analytical free-field solution for $P$-$SV$-$SH$ plane wave propagation with arbitrary angle of incidence through horizontally stratified soil media. The analytical solution, as explained in Section 2.3.2, is obtained through the Haskell-Thomson (H-T) matrix method (Thomson [1950], Haskell [1990]).

The program, created by F.J. Sanchez-Sesma and R. Paolucci and modified by C. Smerzini, makes use of the following subroutines:

- *HASSH.m*: compute the transfer function under SH wave propagation;

- *HASSVP.m*: compute the transfer function under either P or SV wave propagation;

- *convol.m*: compute the convolution of the provided input accelerogram with the calculated transfer function;

- *taper_fun.m*: taper the input signal if required;

- *read_acc.m*: reads an external file containing the input time history.

3. *input_4else_3D.m*: calculate the free-field solution at the entire set of effective boundary nodes. Starting from the displacement time histories computed at all interfaces of the 1D reference soil model, the solution at all effective nodes is obtained by applying a suitable time shift depending on the assumed direction of propagation. Note that for $\gamma = 0$, time shift is equal to 0 for all nodes.

**Input files:**

- lista_ord_all_coor_plax.txt;

- test_3d.out;

- param_4else_3d.txt: contains the input parameters of the model under study. An example of this file, always referred to the homogeneous test, is reported in Figure 3.6.

**Output files:**

- pdrm.dat;

- fdrm.dat.

```
# iwave (0=sh,1=p,2=sv) - dip == angle inc. (deg) - strike (deg) - dt convolution
# R hypo (m) - Vs halfspace (m/s) - Ttot (s)
0 - 0 - 0 - 0.02
0 - 300 - 10
```

Figure 3.6: Example of file param_4else_3d.txt. It contains the parameters describing the model under study.

In Figure 3.7 we report a scheme of the procedure just described.
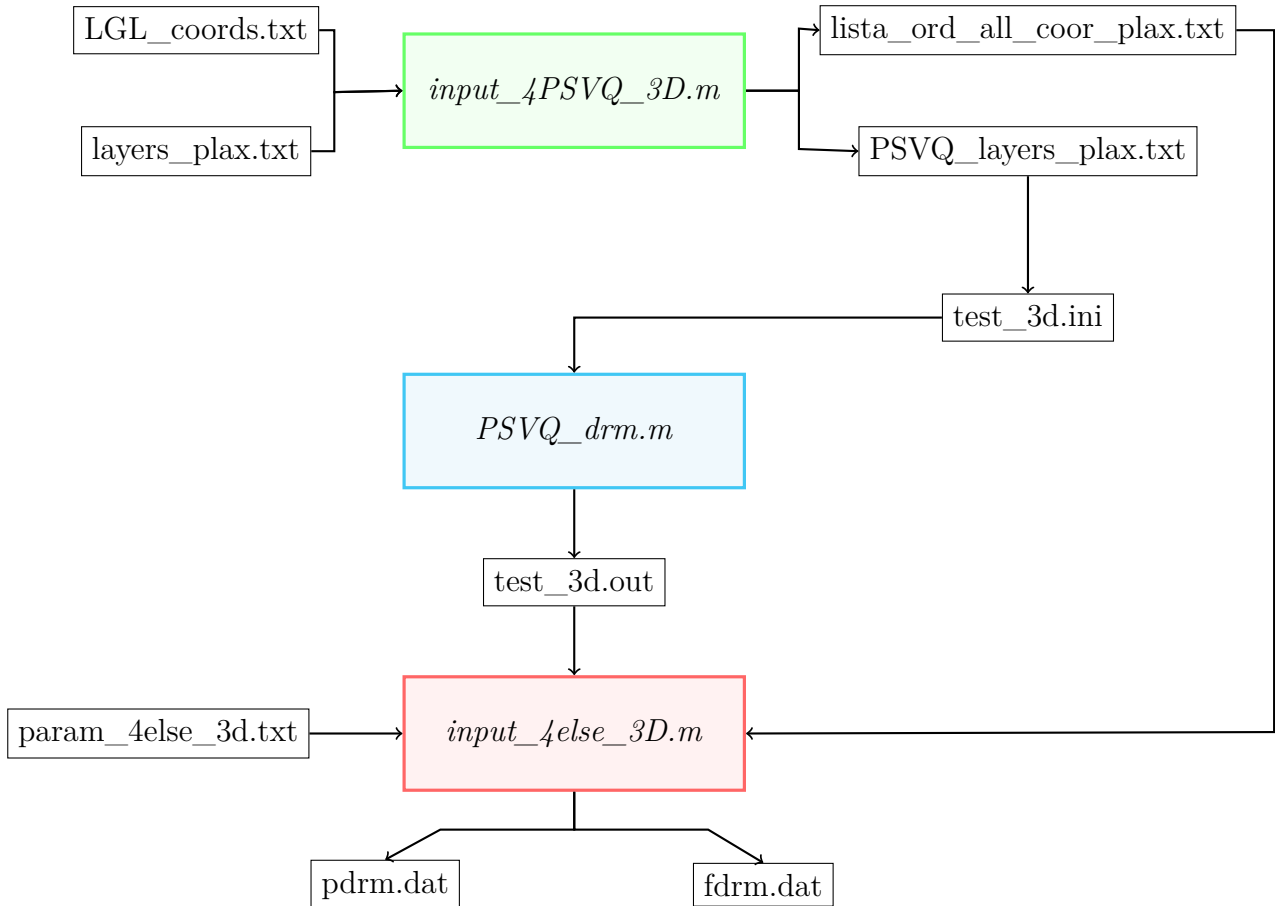


Figure 3.7: Steps for the computation of the PDRM nodes and FDRM free-field displacements. Starting from the LGL nodes computed my means of the Matlab scripts *read_grid_test.m* and *lgl_block.m*, it necessary to pass through all the Matlab scripts depicted above. Finally, one gets as outputs the files pdrm.dat and fdrm.dat that will be inserted in the file mate in order to run the simulation with SPEED coupled with DRM.

## 3.4. Three-dimensional simulation of reduced problem by SPEED coupled with DRM

Given the files generated in the previous sections, we are now able to perform in SPEED, coupled with the Domain Reduction Method, the simulation of seismic wave propagation model under the action of a plane wave.

Precisely, in order to start the simulation, the following files are required:

- File mesh: it contains all the information about the mesh of the model, as explained in Section 3.1.

- File mate: the effective DRM nodes (PDRM) and the free-field displacements (FDRM) are added to the initial form of the file mate, namely the one in which are present only the features of the blocks that compose the reduced domain. The presence of PDRM and FDRM allows to evaluate at run time the effective nodal forces necessary to propagate the target plane wavefront. An example of this file can be found in Figure 3.8.

- SPEED.input. It is the header file in which are fixed the fundamental parameters of the analysis, and the files and directories that are used for the simulation. We report in Figure 3.9 an example of this file.

- LS.input. It is the file containing all the coordinates of the monitored points.

It is worth making some considerations on the choice of the time step $\Delta t$ used for the simulation in SPEED (see also Figure 3.9). As explained by Mazzieri et al. [2013], the procedure adopted for time discretization is the standard leap-frog scheme. The latter, being an explicit scheme, is not unconditionally stable, but it must satisfy the Courant-Friedrichs-Levy (CFL) condition, that reads as follows:

$$\Delta t \leq C_{CFL} \frac{\Delta x_{min}}{V_{max}}. \tag{3.5}$$

Here $C_{CFL}$ is a constant depending on the dimension, the order of the scheme, the mesh geometry and the polynomial degree. It takes value in between 0 and 1. $\Delta x_{min}$ represents the minimum distance between any couple of adjacent LGL nodes, while $V_{max}$ is the maximum propagation velocity.

Since $\Delta x_{min}$ is associated to the nodes close to the element edges, where the grid size scales as $N^{-2}$, the stability condition becomes restrictive for large values of spectral degree, for which an implicit time scheme would be recommended.

| MATE label | degree | $\rho \left(\frac{kg}{m^3}\right)$ | $V_S \left(\frac{m}{s}\right)$ | $V_P \left(\frac{m}{s}\right)$ | $Q_S$ | $Q_P$ |
|---|---|---|---|---|---|---|
| MATE   1 | 2 | 2000 | 300 | 519.6 | 30 | 0 |
| MATE   2 | 2 | 2000 | 300 | 519.6 | 30 | 0 |
| MATE   3 | 2 | 2000 | 300 | 519.6 | 30 | 0 |

Absorbing boundary conditions
ABSO 5

DRM boundary
MDRM 2

Internal boundary
BDRM 4

Tag step of the problem: reduced model
SDRM 2

List of effective DRM nodes with their coordinates
PDRM 1 -375.000000 -500.000000 0.000000 1
...

Maximum frequency
FMAX 1

Free-field displacements corresponding to effective PDRM nodes.
The time instant is followed by the $x$, $y$ and $z$ components of the displacements.
FDRM 1 50 501 0 0 0 0
0.02 0 -2.697460e-05 0
0.04 0 -2.835880e-05 0
...

Figure 3.8: Example of mate file. In addition to the geological properties of each block, PDRM and FDRM are added to the file in order to perform the simulation on SPEED coupled with DRM.

Name of the mesh file
GRIDFILE test3d
Name of the mate file
MATFILE test3d
Name of the directory for MPI files
MPIFILE FILESMPI
Name of the directory for MONITOR files
MONFILE MONITOR

DAMPING 1: frequency proportional damping
DAMPING 1

Option for the output: displacement
OPTIOUT 1 0 0 0 0 0 0 0

$\Delta t$: time integration step adopted (sec)
TIMESTEP 1.0e-2
Write output results at times t* multiple of TMONITOR*TIMESTEP.
TMONITOR 2
Final instant of the simulation (sec)
STOPTIME 10

Starting depth for monitored point search
MLST -450 0

Figure 3.9: Example of file SPEED.input. It contains the names of all the files used for the simulation, as well as folder names to which the output files are placed inside. Moreover, it is specified the type of damping, the type of output and the time step used for the simulation.

# 4 | Tests and results

The implementation of the code SPEED coupled with the Domain Reduction Method (DRM) for plane wave propagation has been validated on simple geological models to check the accuracy of the numerical results against semi-analytical solutions.

Precisely, these models are studied under the action of two different types of seismic body waves. The first type are the so-called Primary waves ($P$ waves): they are compressional waves such that the direction of the ground motion they cause is parallel to the wave propagation direction. They are called "Primary" because they travel faster than other waves through the earth and, because of this, they arrive at seismograph stations first. The second type of waves are the Second waves ($S$ waves), which name is due to the fact that, in a seismic event, they arrive later than the $P$ waves. They are shear waves for which, differently from before, the ground motion is perpendicular to the wave propagation direction. For the latter case, we further distinguish between vertically polarized $SV$ waves, when the polarization of an $S$ wave lies in a vertical plane, and horizontally polarized $SH$ waves, when, instead, the polarization is in a horizontal plane.

In Figure 4.1 we depict a simple representation of the two different waves.
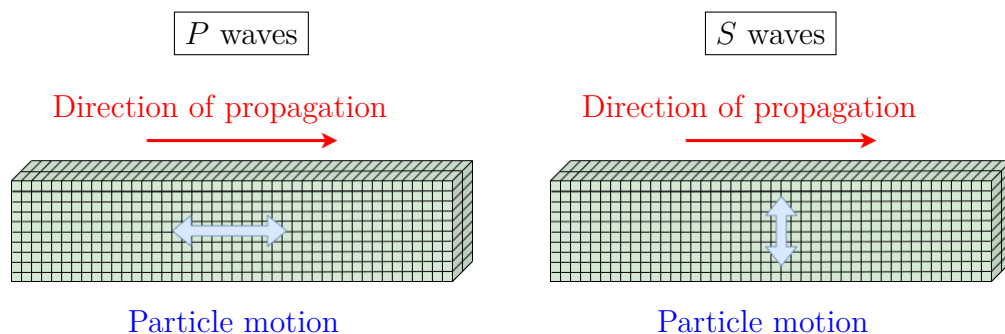


Figure 4.1: $P$ and $S$ waves. For the $P$ waves (on the left), the direction of propagation (red arrow) is parallel to the ground motion caused by the wave (blue arrow). For the $S$ waves (on the right), instead, the direction of propagation (red arrow) is perpendicular to the ground motion caused by the wave (blue arrow).

As a final remark, we denote as $V_P$ the $P$ wave velocity and $V_S$ the $S$ wave velocity. Respectively, $V_P = \sqrt{\frac{\lambda+2\mu}{\rho}}$ and $V_S = \sqrt{\frac{\mu}{\rho}}$, where $\lambda$ and $\mu$ are the Lamé elastic coefficients, while $\rho$ is the material density.

In this section we analyse three different test cases:

- TEST CASE 1: **Homogeneous model**. The test, analysed both under the action of $SV$ and $P$ waves, is performed for the cases of vertical plane wave (angle of incidence $\gamma = 0°$) and oblique plane wave (angle of incidence $\gamma = 10°$);

- TEST CASE 2: **Heterogeneous model**. Like TEST CASE 1, it is analysed both under the action of $SV$ and $P$ waves and the simulation is performed for the cases of vertical plane wave ($\gamma = 0°$) and oblique plane wave ($\gamma = 10°$).

- TEST CASE 3: **"Croissant" valley**. The last one, a more physical test case, it is analysed under the action of vertical ($\gamma = 0°$) $SH$ plane wave.

For all these test cases, as said before, we compare the numerical solution obtained with SPEED coupled with DRM with the semi-analytical solution. The way in which the latter is obtained is explained in the following and it depends on the test case.

Moreover, only in the cases with vertical plane waves, we are going to compare the numerical solution obtained using SPEED coupled with DRM also with the numerical solution obtained using SPEED in the traditional way, namely without DRM. This is because SPEED without DRM is not able to deal with plane waves with an angle of incidence different from zero, while this is possible for DRM since it takes into account the angle of incidence in the computation of the *free-field ground motion*, as explained in the previous chapter.

Before proceeding with the results, a final remark on the way in which the solution are obtained. Recall that, as explained in Chapter 3, after the execution of the code we obtain the file MONITORXXXXXX.D. After that:

1. Run the MATLAB script *REWRITE_MONITOR_FORMAT.m* to rewrite the output files from MONITORXXXXXX.D to the $N_{LS}$ files **monitorxxxxxx.d**, where $N_{LS}$ is the number of the monitored points. These files contain, for the corresponding monitored point, the temporal history of the displacements computed for each instant of time involved in the simulation.

2. Run the MATLAB script *PLOT_MONITOR.m* to plot the output results.

## 4.1.  TEST CASE 1: HOMOGENEOUS MODEL

As a first validation of the three-dimensional numerical code, we consider an idealized halfspace subject to the incidence of both vertical ($\gamma = 0°$) and oblique ($\gamma = 10°$) plane waves from below. The temporal dependence of the incident plane wavefront is given by a Ricker wavelet with maximum frequency equal to $3Hz$.

Figure 4.2 represents the mesh of the reduced domain used for the analyses. Moreover, in the same figure, the blocks that form the computational domain are isolated one by one in order to simplify the comprehension of the domain's division. Note that block 4 is introduced in order to make the simulation with code SPEED without DRM. Indeed, this is the block in which the plane wave is applied for that specific simulation.

In Table 4.1 we report the mechanical properties of the blocks. Obviously, all the four volumes blocks have the same properties since the model is homogeneous.

Note that the spectral degree $N$ is chosen equal to $N = 4$ for all the blocks. Because of this, the number of degrees of freedom for each spectral element of the simulation is equal to $3(N + 1)^3 = 375$.

| | label | degree | $\rho$ $(kg/m^3)$ | $V_S$ (m/s) | $V_P$ (m/s) | $Q_S$ | $Q_P$ |
|---|---|---|---|---|---|---|---|
| **MATE** | 1 | 4 | 2000 | 300 | 519.6 | 0 | 0 |
| **MATE** | 2 | 4 | 2000 | 300 | 519.6 | 0 | 0 |
| **MATE** | 3 | 4 | 2000 | 300 | 519.6 | 0 | 0 |
| **MATE** | 4 | 4 | 2000 | 300 | 519.6 | 0 | 0 |

Table 4.1: **Test case 1**. Mechanical properties related to the four blocks composing the computational domain. The spectral degree is equal to $N = 4$. All the properties are the same for every block since the model is homogeneous.

In Figure 4.3 we depict the monitored points of the model during the simulation. These are the points in which we are going to analyse the numerical displacement solutions in the following sections. They are nine points taken along the $z$ axis, starting from the top $(z = 0)$.

Computational domain

Block 1: halfspace (inside DRM)

Block 2: DRM boundary (MDRM)

Block 3: halfspace (outside DRM)

Block 4: plane wave

Block 5: Absorbing BCs

Block 6: internal boundary (BDRM)
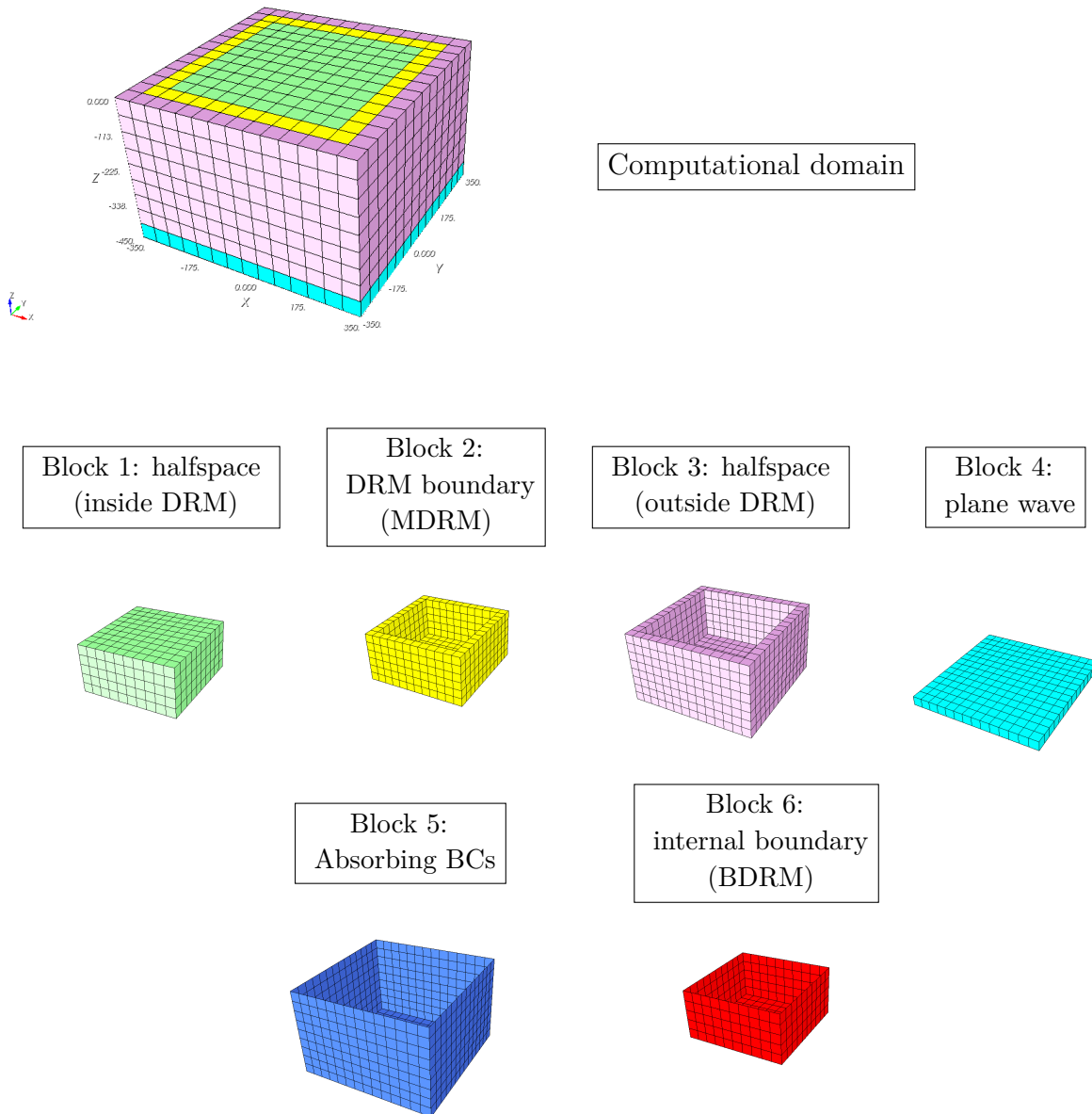
Figure 4.2: **Test case 1**. Blocks representation of the homogeneous model. The figure in the top panels represents the mesh of the entire computational model. Then, all the blocks are isolated one by one, included those regarding the boundary conditions of the model.
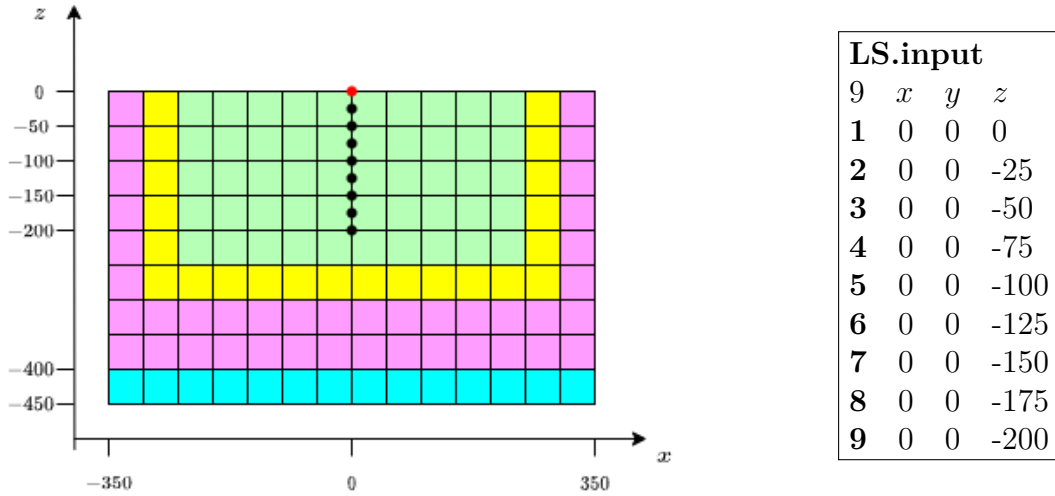
Figure 4.3: **Test case 1**. Monitored points of the simulation. The picture on the left indicates where the points are located inside the computational domain. Note that the sketch represents the $y$ section of the entire domain. On the right, instead, it is provided an entire list of the monitored points with their respective coordinates, namely the file LS.input. Note that these points are all taken inside the DRM block of elements.

### 4.1.1.   Vertical plane wave: $\gamma = 0°$

In this section we analyse the results relative to the vertical ($\gamma = 0°$) plane wave, both in the case of $P$ and $SV$ waves.

The resulting displacements are plotted in two different figures. Figure 4.4 shows the horizontal $u_x$ and vertical $u_z$ displacement time histories obtained due to the incidence of plane $SV$ wave. Precisely, the top panel shows the displacements corresponding to all the monitored point except the one on the free surface (see red point in Figure 4.3), that, instead, is plotted in the bottom panel.

In the same way, in Figure 4.5 we report the computed displacements obtained in the case of a vertical incident $P$ plane wave.

In both representations we make a comparison between the semi-analytical solution (computed with the Haskell-Thomson (HT) method and depicted as a blue continuous line), the numerical solution obtained with SPEED coupled with Domain Reduction Method (dashed green line), and the numerical solution obtained with SPEED without the implementation of DRM (dashed red line).

For each case, the comparison between the three solutions turns out to be very good since the solutions are basically superimposed. Moreover, only for the $P$ wave case, we report

in Table 4.2 the errors in norm $L^\infty$ of the difference between the semi-analytical solution (denoted as $u$) and the numerical solution obtained with SPEED coupled with DRM (denoted as $u_N$), namely $|u - u_N|_{L^\infty} = max(|u - u_N|)$. Note that both the solutions refer to the vertical displacement $u_z$ (see Figure 4.5, top right panel). The solutions taken into consideration are the one computed in the eight monitor points below the free-surface of the model and the error turns out to be good in each of the eight cases.

We underline that the spectral degree used for both the numerical simulations is equal to $N = 4$, while the time discretization $\Delta t$ is chosen equal to $2 \cdot 10^{-4}$, again for both the simulations involving SPEED.

| Depth $z(m)$ | $-25$ | $-50$ | $-75$ | $-100$ | $-125$ | $-150$ | $-175$ | $-200$ |
|---|---|---|---|---|---|---|---|---|
| $|u - u_N|_{L^\infty}$ | 0.0275 | 0.0204 | 0.0126 | 0.0111 | 0.0126 | 0.0172 | 0.0192 | 0.0186 |

Table 4.2: **Test case 1**. Errors in $L^\infty$ norm of the numerical solution $u_N$ (obtained with SPEED coupled with DRM) with respect to the semi-analytical solution $u$, both referred to the vertical displacement $u_z$. The measure of the error is given for the case of $P$ vertical incident plane wave. The solutions taken into considerations are the one computed in the eight monitors point below the free-surface of the model.

SV WAVES

Horizontal displacement $u_x(m)$       Vertical displacement $u_z(m)$
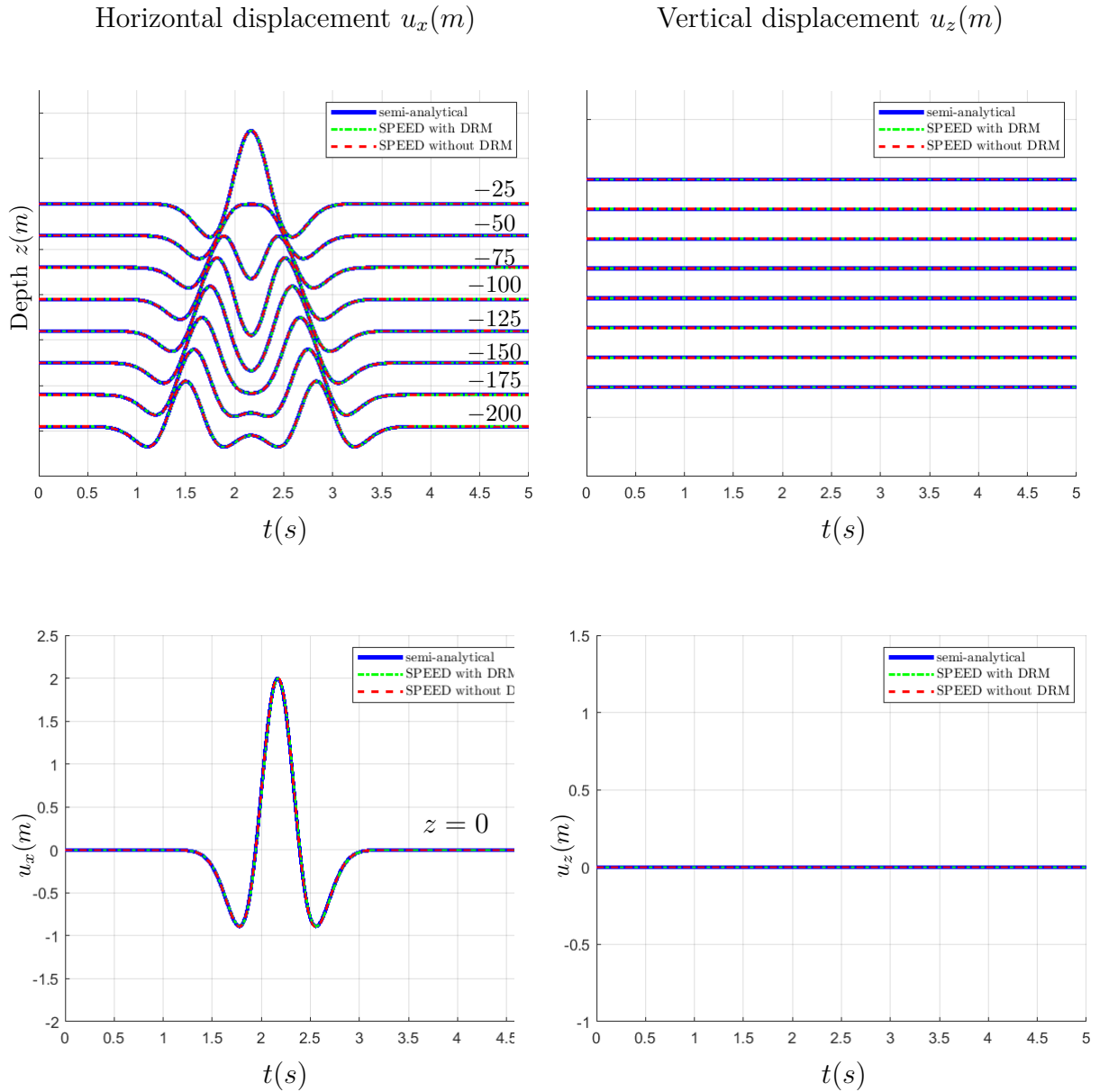


Figure 4.4: **Test case 1**: homogeneous model. Plot of horizontal $u_x$ and vertical $u_z$ displacements corresponding to the internal monitored points (top panel) and to the point located at the free surface (bottom panel) due to the incidence of a vertical ($\gamma = 0°$) plane $SV$ wave. Three solutions are compared: the blue line indicates the semi-analytical solution obtained with the H-T method, the dashed green line represents the solution obtained with SPEED coupled with DRM, while the red dashed line the one obtained with SPEED without DRM.

P WAVES

Horizontal displacement $u_x(m)$        Vertical displacement $u_z(m)$
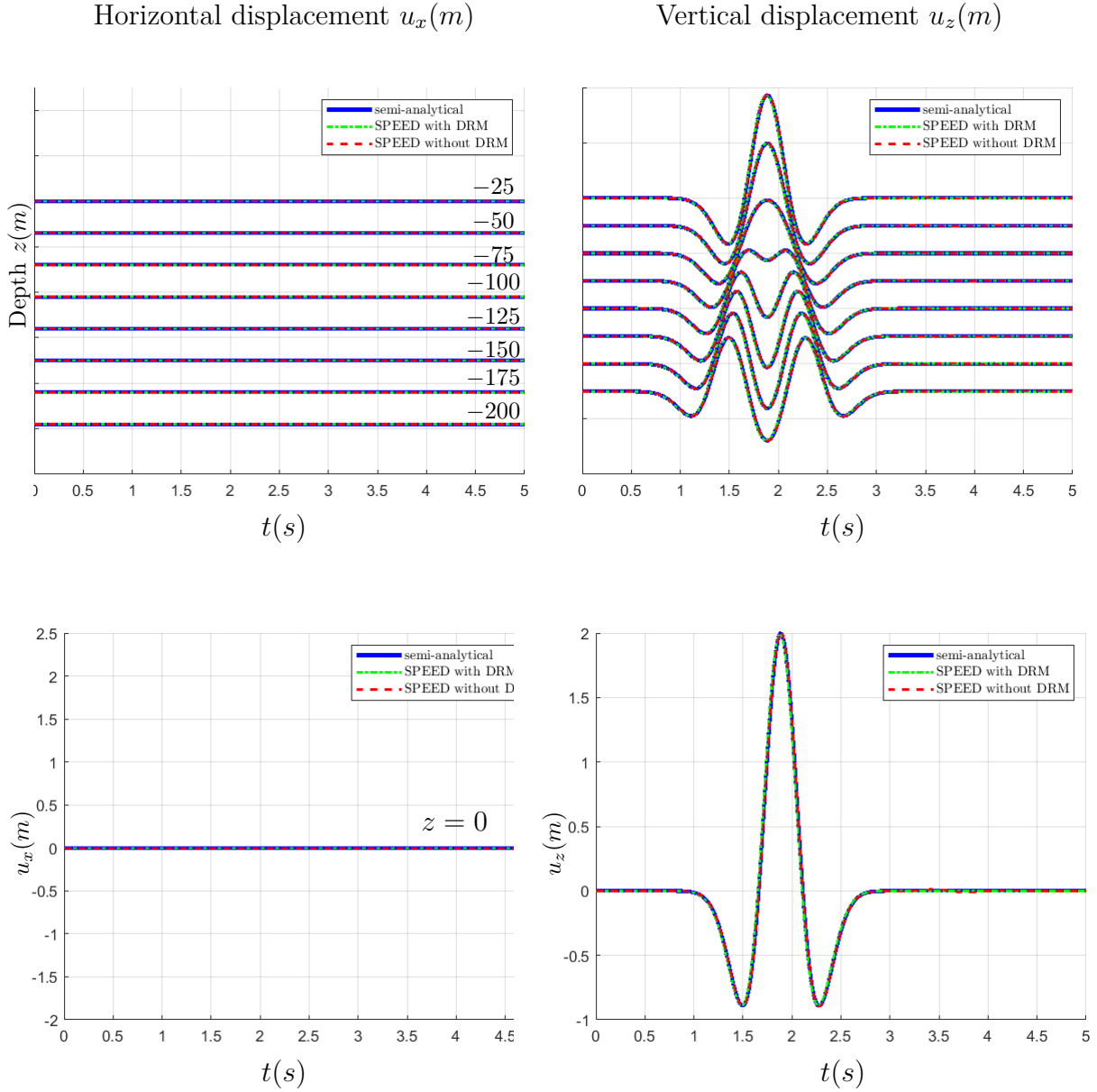


Figure 4.5: **Test case 1**: homogeneous model. Plot of horizontal $u_x$ and vertical $u_z$ displacements corresponding to the internal monitored points (top panel) and to the point located at the free surface (bottom panel) due to the incidence of a vertical ($\gamma = 0°$) plane $P$ wave. Three solutions are compared: the blue line indicates the semi-analytical solution obtained with the H-T method, the dashed green line represents the solution obtained with SPEED coupled with DRM, while the red dashed line the one obtained with SPEED without DRM.

## 4.1.2. Inclined plane wave: $\gamma = 10°$

In the same way, in this section we analyse what happens in the case of oblique incidence plane wave. Here $\gamma$, representing the angle of incidence with respect to the vertical direction, is taken equal to 10°. It is positive since it is obtained in a clockwise sense with respect to the vertical direction.

As done before, we show in Figure 4.6 the horizontal $u_x$ and vertical $u_z$ displacement time histories obtained due to the incidence of plane $SV$ wave, and, in Figure 4.7, the same results but due to the incidence of a plane $P$ wave.

Again, the top panel shows the displacements corresponding to all the monitored points except the one on the free surface (see red point in Figure 4.3), that, instead, is plotted in the bottom panel.

Differently from before, the comparison occurs only between the semi-analytical solution (computed with H-T method and depicted as a blue continuous line) and the numerical solution obtained with SPEED coupled with Domain Reduction Method (dashed green line). Indeed, in the traditional use of SPEED, namely without DRM, it is not possible to perform simulations in which the plane wave has an angle of incidence different from 0.

This time, the results present some minimal spurious effects probably due to the presence of Absorbing Boundary Conditions (ABCs), that are applied to the bottom and the lateral surfaces of the reduced model. This is especially visible in the vertical displacement $u_z$ for $SV$ plane wave case, and in the horizontal displacement $u_x$ for $P$ plane wave case. These displacements are the one arising because of the inclined plane wave. Indeed, as we can see from Figures 4.4 and 4.5, they are both equal to 0 in the case of vertical plane wave.

However, looking at the solutions, the comparison is still very good.

$$\boxed{\text{SV WAVES}}$$

Horizontal displacement $u_x(m)$        Vertical displacement $u_z(m)$
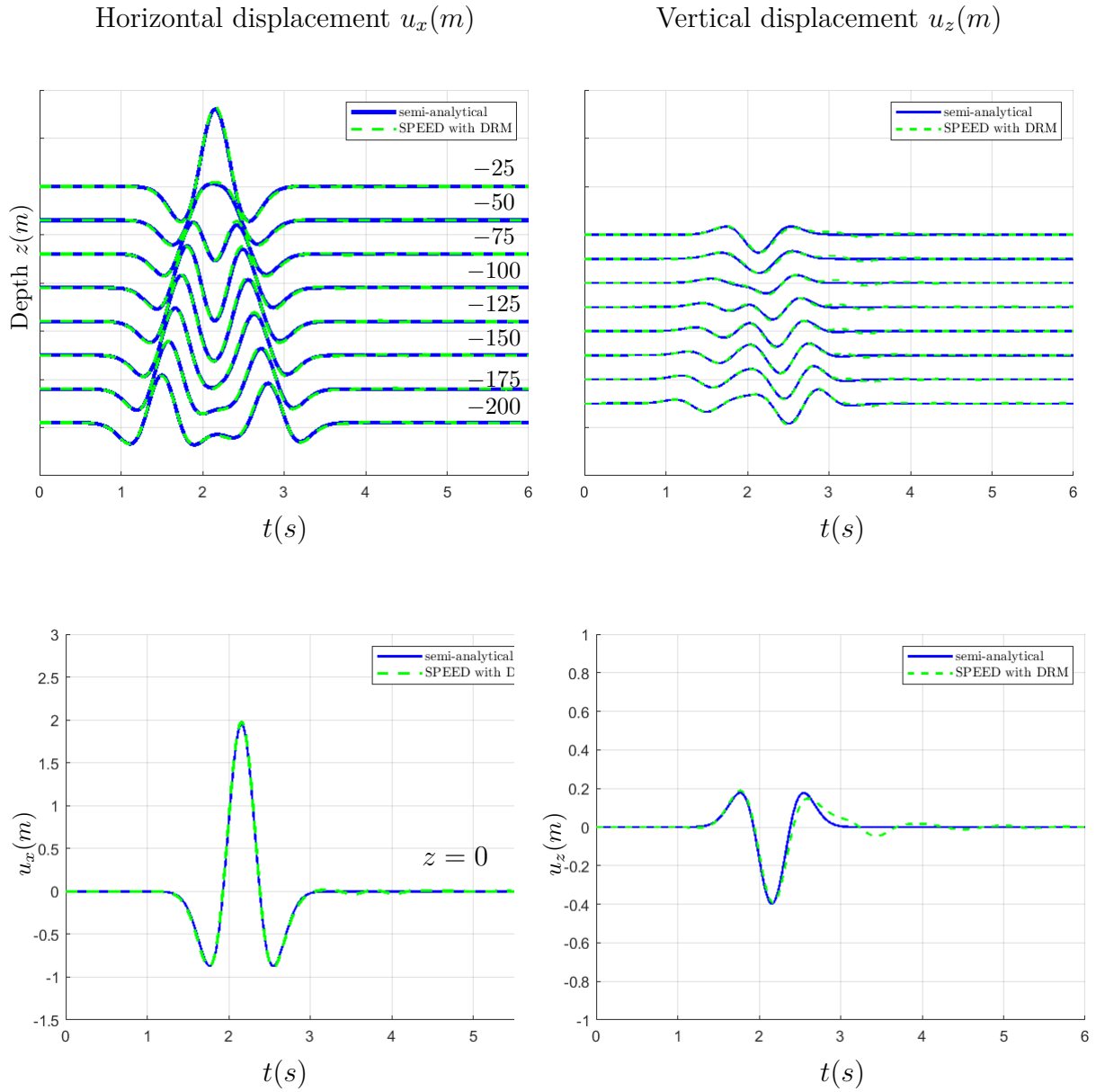


Figure 4.6: **Test case 1:** homogeneous model. Plot of horizontal $u_x$ and vertical $u_z$ displacements corresponding to the internal monitored points (top panel) and to the point located at the free surface (bottom panel) due to the incidence of an oblique ($\gamma = 10°$) plane $SV$ wave. Two solutions are compared: the blue line indicates the semi-analytical solution obtained with the H-T method, while the dashed green line represents the solution obtained with SPEED coupled with DRM.

P WAVES

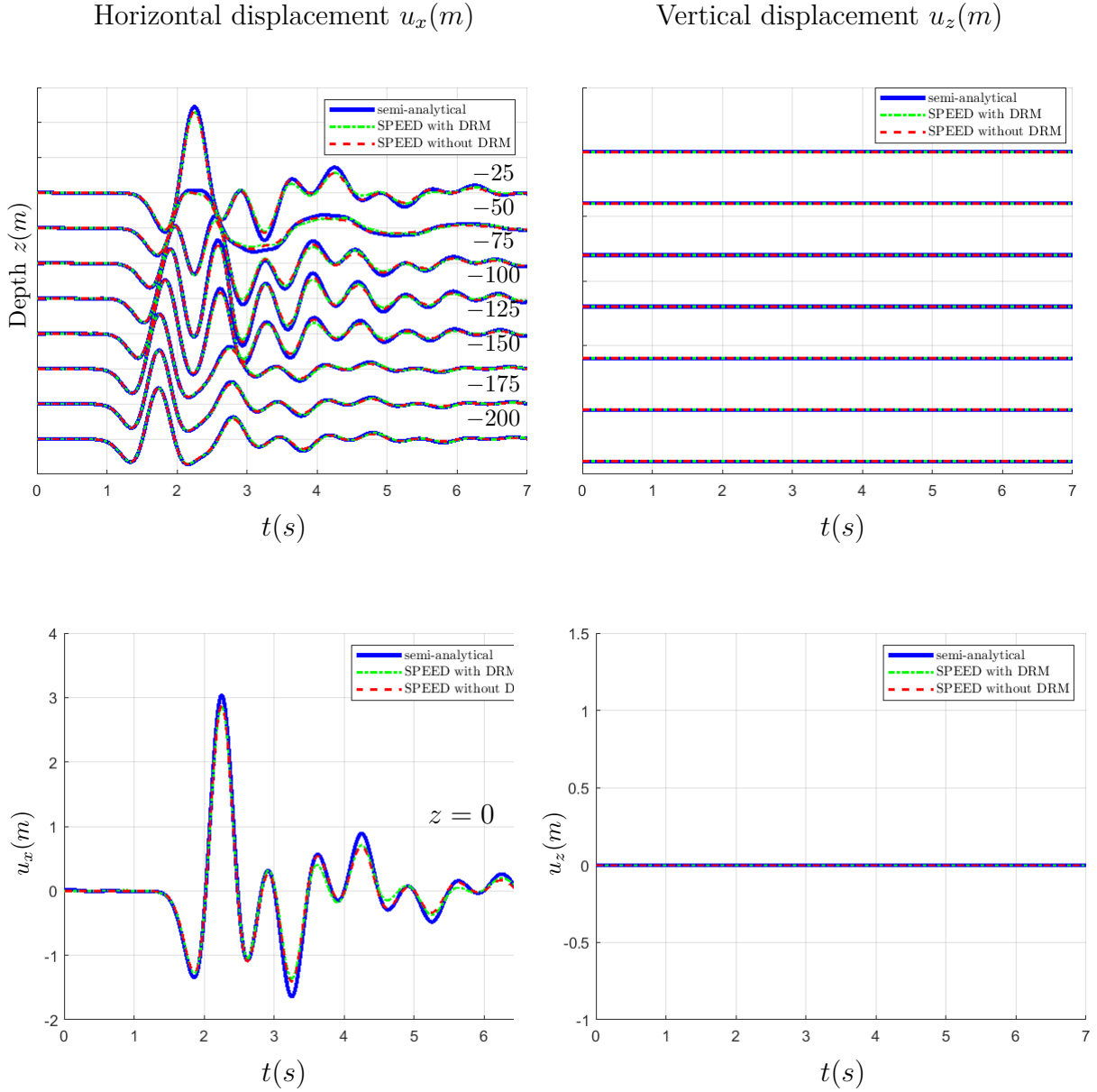Horizontal displacement $u_x(m)$      Vertical displacement $u_z(m)$
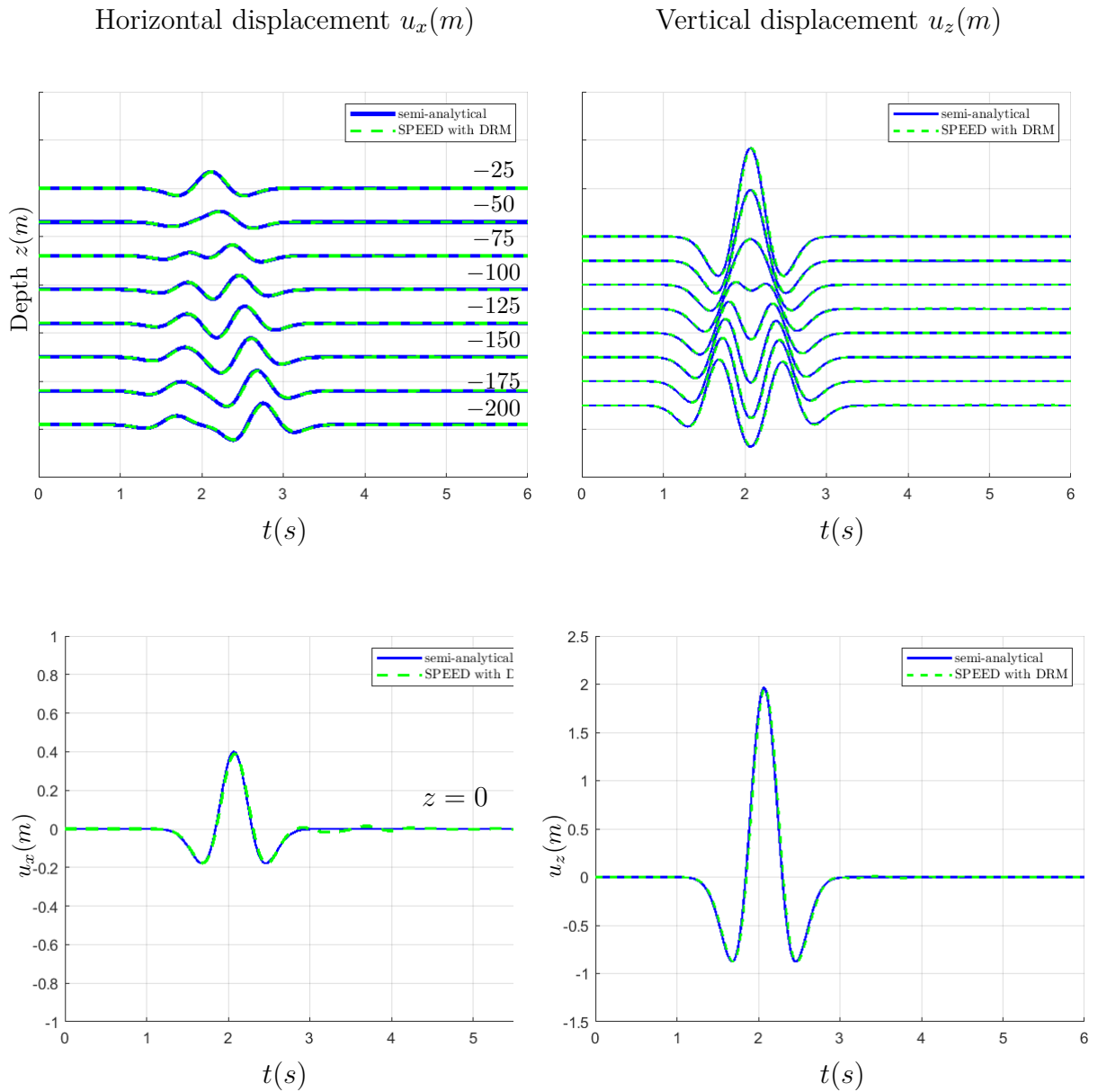


Figure 4.7: **Test case 1**: homogeneous model. Plot of horizontal $u_x$ and vertical $u_z$ displacements corresponding to the internal monitored points (top panel) and to the point located at the free surface (bottom panel) due to the incidence of an inclined ($\gamma = 10°$) plane $P$ wave. Two solutions are compared: the blue line indicates the semi-analytical solution obtained with the H-T method, while the dashed green line represents the solution obtained with SPEED coupled with DRM.

Also for the inclined $P$ wave case, we report in two tables the errors in norm $L^\infty$ of the difference between the semi-analytical solution (denoted as $u$) and the numerical solution obtained with SPEED coupled with DRM (denoted as $u_N$), namely $|u - u_N|_{L^\infty}$. Precisely, Table 4.3 refers to the horizontal displacement $u_x$ (see Figure 4.7, top left panel), while Table 4.4 refers to the vertical displacement $u_z$ (see Figure 4.7, top right panel). The solutions taken into consideration are the one computed in the eight monitor points below the free-surface of the model and the error turns out to be good in each of the eight cases.

| Depth $z(m)$ | $-25$ | $-50$ | $-75$ | $-100$ | $-125$ | $-150$ | $-175$ | $-200$ |
|---|---|---|---|---|---|---|---|---|
| $|u - u_N|_{L^\infty}$ | 0.0127 | 0.0095 | 0.0087 | 0.0113 | 0.0109 | 0.0138 | 0.0169 | 0.021 |

Table 4.3: **Test case 1**. Errors in $L^\infty$ norm of the numerical solution $u_N$ (obtained with SPEED coupled with DRM) with respect to the semi-analytical solution $u$, both referred to the horizontal displacement $u_x$. The measure of the error is given for the case of $P$ inclined ($\gamma = 10°$) incident plane wave. The solutions taken into considerations are the one computed in the eight monitors point below the free-surface of the model.

| Depth $z(m)$ | $-25$ | $-50$ | $-75$ | $-100$ | $-125$ | $-150$ | $-175$ | $-200$ |
|---|---|---|---|---|---|---|---|---|
| $|u - u_N|_{L^\infty}$ | 0.0275 | 0.0204 | 0.0126 | 0.0111 | 0.0126 | 0.0172 | 0.0192 | 0.0186 |

Table 4.4: **Test case 1**. Errors in $L^\infty$ norm of the numerical solution $u_N$ (obtained with SPEED coupled with DRM) with respect to the semi-analytical solution $u$, both referred to the vertical displacement $u_z$. The measure of the error is given for the case of $P$ inclined ($\gamma = 10°$) incident plane wave. The solutions taken into considerations are the one computed in the eight monitors point below the free-surface of the model.

## 4.2.   TEST CASE 2: HETEROGENEOUS MODEL

As a second validation of the numerical code, we consider instead a geological model made of one soft layer (the one above) over a halfspace of harder material. This layers form together a heterogeneous domain. The model is subject to the incidence of both vertical ($\gamma = 0°$) and oblique ($\gamma = 10°$) plane waves from below. The temporal dependence of the incident plane wavefront is given by a Ricker wavelet with maximum frequency equal to $3Hz$, as in the previous case.

Figure 4.8 represents the mesh of the reduced domain used for the analyses. Moreover, always in Figure 4.8, the blocks that form the computational domain are isolated one by one in order to simplify the comprehension of the domain's division. Also in this case, the block on the bottom (block 4) is introduced in order to make the simulation with code SPEED without DRM. Indeed, this is the block in which the plane wave is applied for that specific simulation.

In Table 4.5 we report the mechanical properties of the blocks. Obviously, the two layers of materials present in this case different values. Note that, without loss of generality, we assume $Q_S = Q_P$.

Note that the spectral degree is chosen equal to 4 for all the blocks. However, this is true only for vertical incident plane wave ($\gamma = 0°$). Indeed, for the case of oblique incident plane wave ($\gamma = 0°$) we use spectral degree $N = 2$. The reason for that is explained in the dedicated section.

| | label | degree | $\rho$ ($kg/m^3$) | $V_S$ (m/s) | $V_P$ (m/s) | $Q_S$ | $Q_P$ |
|---|---|---|---|---|---|---|---|
| MATE | 1 | 4 | 2000 | 300 | 519.6 | 60 | 60 |
| MATE | 2 | 4 | 2000 | 300 | 519.6 | 60 | 60 |
| MATE | 3 | 4 | 2200 | 1000 | 1800 | 180 | 180 |
| MATE | 4 | 4 | 2200 | 1000 | 1800 | 180 | 180 |
| MATE | 5 | 4 | 2200 | 1000 | 1800 | 0 | 0 |

Table 4.5: **Test case 2**. Mechanical properties related to the five blocks composing the computational domain. The spectral degree is equal to 4. The properties are the same for the first two blocks, since they are part of the first physical layer, and for the third and fourth blocks, because they are part of the second physical layer. The last block, the one in the bottom, has the same properties related to the second layer, except for the quality factor $Q_S$ and $Q_P$ that are taken equal to 0.

In Figure 4.9 we depict the monitored points of the model during the simulation. These are the points in which we are going to analyse the numerical displacement solutions in the following sections. They are nine points taken along the $z$ axis, starting from the top ($z = 0$). Note that these are the same points used for the homogeneous test case. This time, however, the seven points on the top are located in the first layer, while the other two points are located in the halfspace.

Computational domain

| Block 1: layer 1 (inside DRM) | Block 1: layer 1 (outside DRM) | Block 2: DRM boundary in layer 1 (MDRM) |



| Block 3: halfspace (inside DRM) | Block 3: halfspace (outside DRM) | Block 4: DRM boundary in halfspace (MDRM) |



| Block 5: plane wave | Block 6: Absorbing BCs | Block 7: internal boundary (BDRM) |
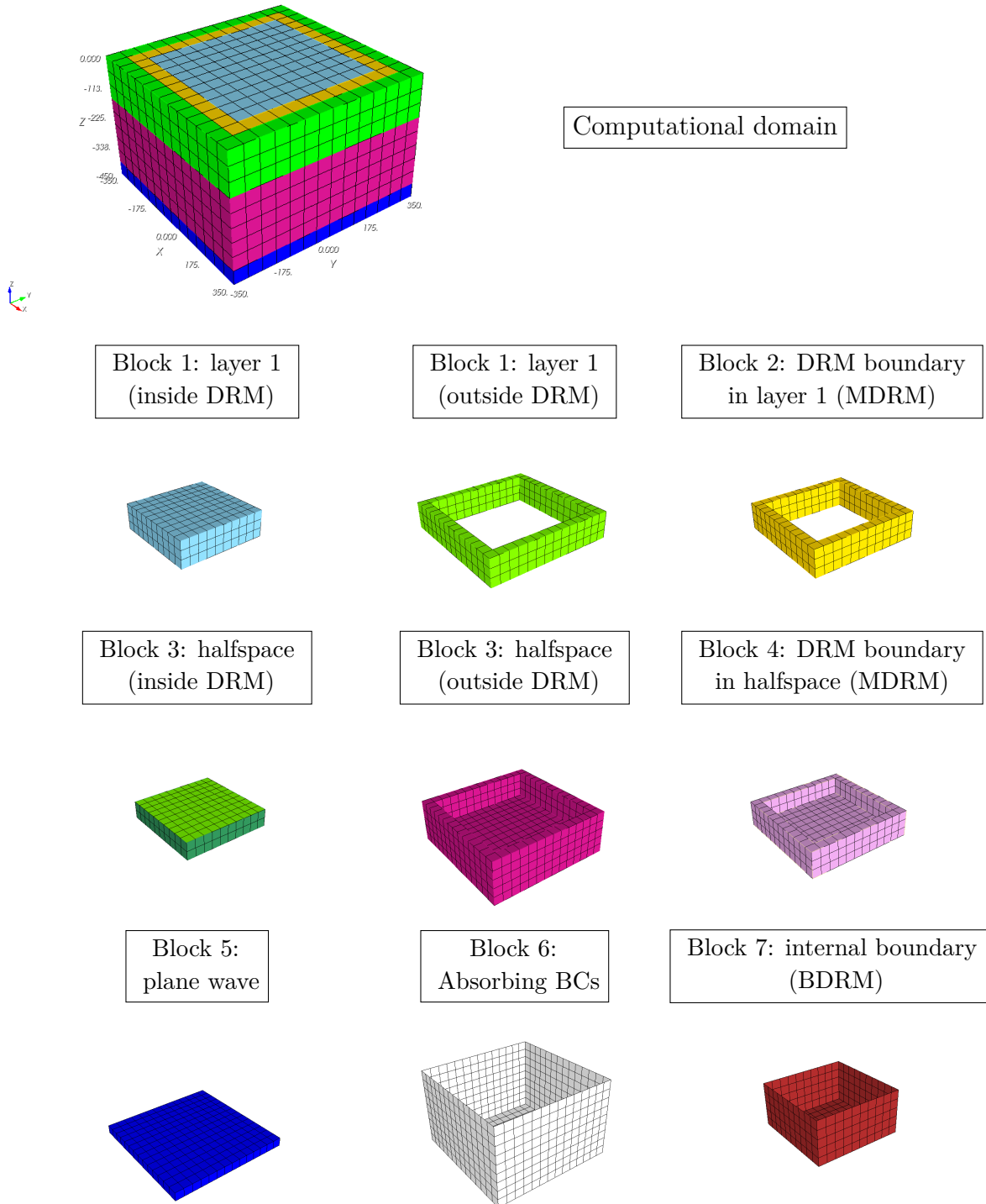


Figure 4.8: **Test case 2**. Blocks representation of the heterogeneous model. The figure in the top panels represents the mesh of the entire computational model. Then, all the blocks are isolated one by one, included those regarding the boundary conditions of the model.
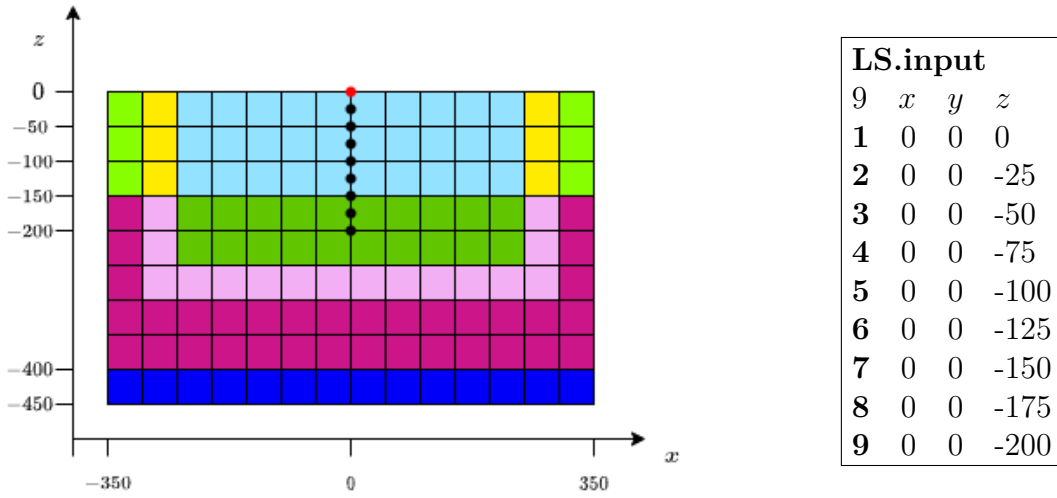
Figure 4.9: **Test case 2**. Monitored points of the simulation. The picture on the left indicates where the points are located inside the computational domain. Note that the sketch represents the $y$ section of the entire domain. On the right, instead, it is provided an entire list of the monitored points with their respective coordinates, namely the file LS.input. These points are all taken inside the DRM block of elements: the first seven points are inside the first layer, while the other two are taken in the halfspace.

## 4.2.1.  Vertical plane wave: $\gamma = 0°$

As we did for the homogeneous case, we analyse the results relative to the vertical ($\gamma = 0°$) plane wave, both in the case of $P$ and $SV$ waves.

The resulting displacements are plotted in two different figures. Figure 4.10 shows the horizontal $u_x$ and vertical $u_z$ displacement time histories obtained due to the incidence of plane $SV$ wave. Precisely, the top panel shows the displacements corresponding to all the monitored point except the one on the free surface (see red point in Figure 4.9), that, instead, is plotted in the bottom panel.

Then, in Figure 4.11 we report the resulting displacements obtained in the case of $P$ vertical incident plane wave.

In both cases we make a comparison between the semi-analytical solution (computed with H-T method and depicted as a blue continuous line), the numerical solution obtained with SPEED coupled with Domain Reduction Method (dashed green line), and the numerical solution obtained with SPEED without the implementation of DRM (dashed red line).

SV WAVES

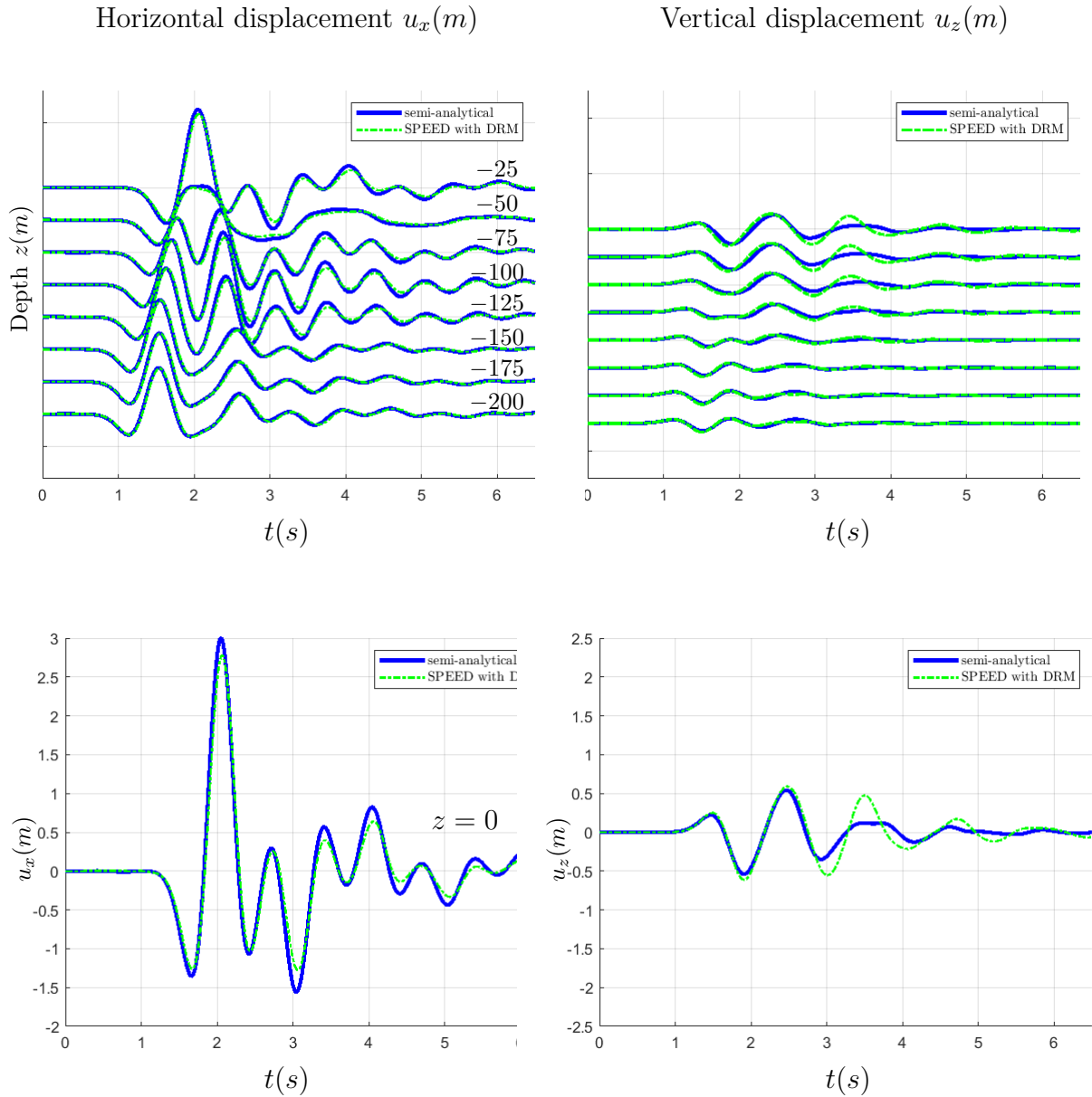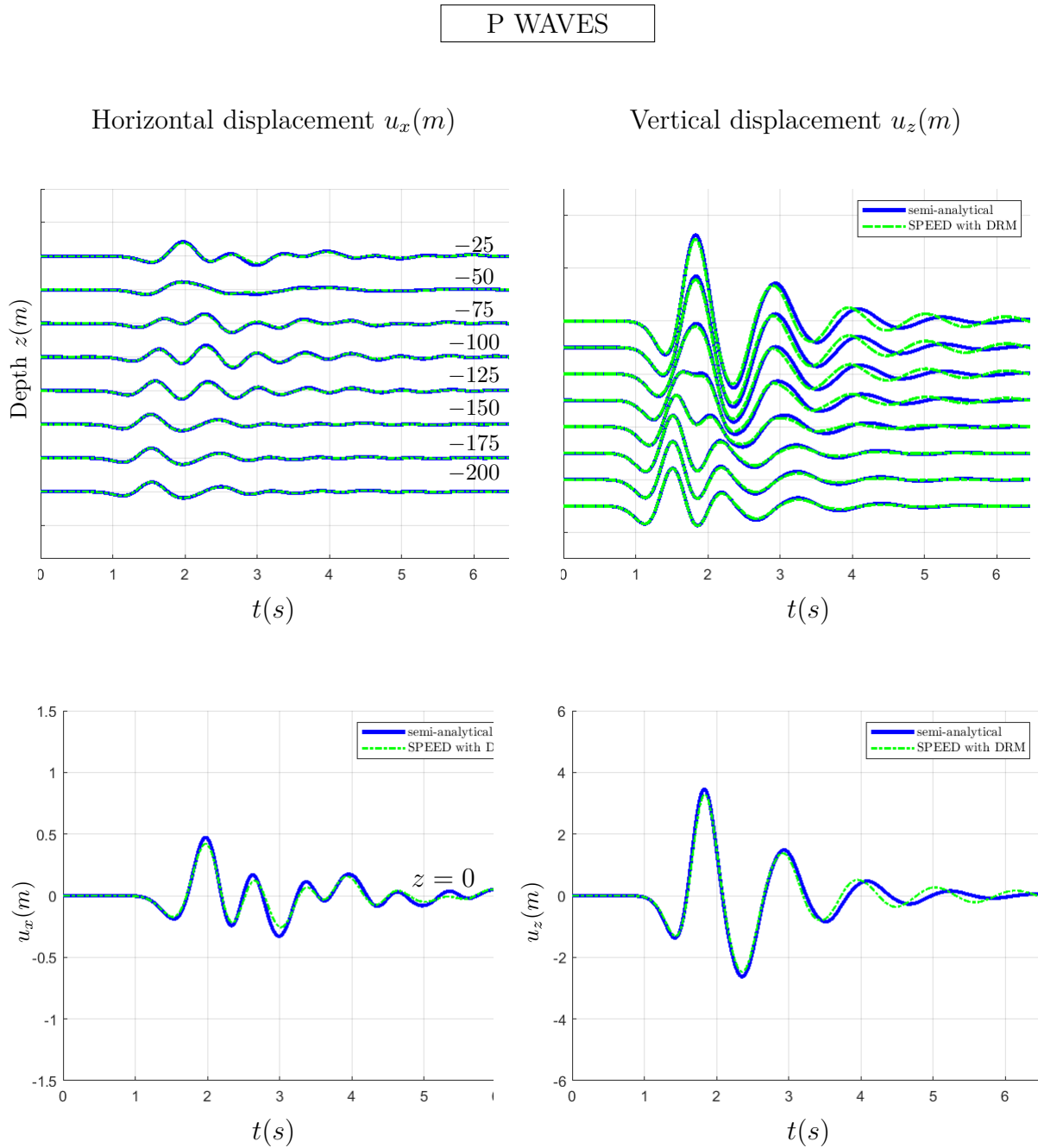Horizontal displacement $u_x(m)$                    Vertical displacement $u_z(m)$



Figure 4.10: **Test case 2**: heterogeneous model. Plot of horizontal $u_x$ and vertical $u_z$ displacements corresponding to the internal monitored points (top panel) and to the point located at the free surface (bottom panel) due to the incidence of a vertical ($\gamma = 0°$) plane $SV$ wave. Three solutions are compared: the blue line indicates the semi-analytical solution obtained with the H-T method, the dashed green line represents the solution obtained with SPEED coupled with DRM, while the red dashed line the one obtained with SPEED without DRM.

P WAVES

Horizontal displacement $u_x(m)$                Vertical displacement $u_z(m)$
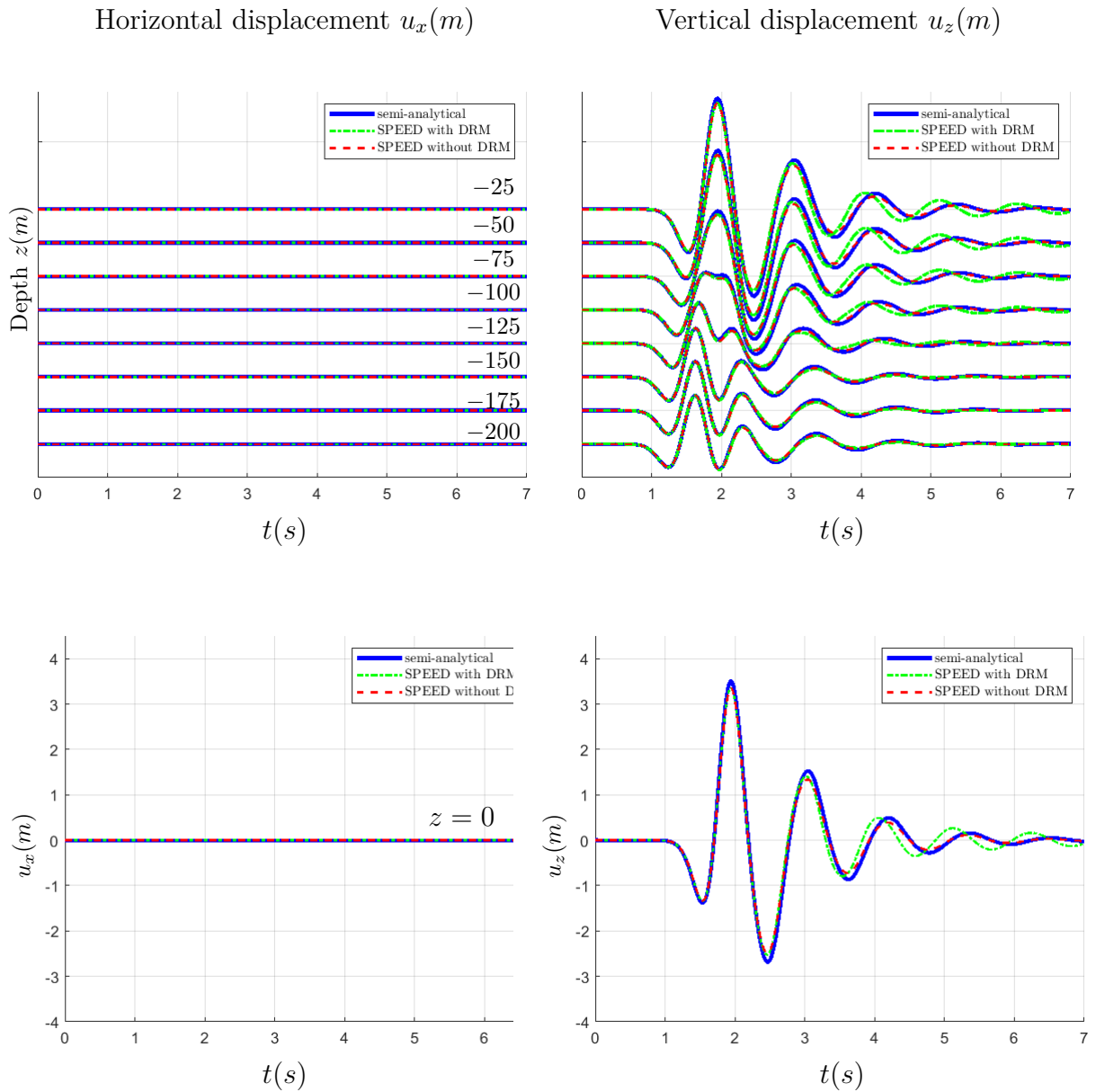


Figure 4.11: **Test case 2:** heterogeneous model. Plot of horizontal $u_x$ and vertical $u_z$ displacements corresponding to the internal monitored points (top panel) and to the point located at the free surface (bottom panel) due to the incidence of a vertical ($\gamma = 0°$) plane $P$ wave. Three solutions are compared: the blue line indicates the semi-analytical solution obtained with the H-T method, the dashed green line represents the solution obtained with SPEED coupled with DRM, while the red dashed line the one obtained with SPEED without DRM.

The solution obtained with SPEED coupled with DRM coincides with the semi-analytical solution mostly in the first part of the simulation (before $\approx 3.5sec$), after that it tends to dampen more slowly. In particular, this happens more for the monitored points located closest to $z = 0$ and this is probably due to the change of values of the quality factors between the two layers. In general, especially for $SV$ plane wave, we can say that the comparison between the two solutions is quite good.

The displacements obtained with SPEED without DRM, instead, seem to fit better the semi-analytical solution in the second part of the simulation (after $\approx 3.5sec$). However, it is worth making an important remark on the model used for the simulation of SPEED without the DRM, particularly regarding the boundary conditions that are applied on it.

Indeed, if from one side the numerical solution obtained with SPEED coupled with DRM experienced very low spurious effects due to the absorbing boundary conditions, from the other side this is no longer true for the numerical solution obtained with SPEED without DRM.

For this reason, in order to obtain the results depicted in Figure 4.10 and in Figure 4.11, Dirichlet boundary conditions have been applied to the lateral surface of the computational domain.

In spite of that, if we consider to apply the same boundary conditions, namely ABCs to the bottom and lateral surface, to both the computational domains, the numerical result obtained with SPEED without DRM presents lots of spurious effects. To clarify the different setting of boundary conditions (BCs), see Figure 4.12, in which, for the case of $P$ vertical incident wave, it is shown in which blocks and what types of BCs are applied.

Finally, as an example of the different solutions we obtain, in Figure 4.13 we show a comparison of the vertical displacement get by the same monitored point (the one on the free-surface) when the two models undergo the incidence of a vertical $P$ plane wave. From the latter, it is quite evident how, applying the same boundary conditions (ABCs) SPEED coupled with DRM provides a much better solution.

Block 6: Dirichlet BCs    Block 7: ABCs

**SPEED.input**
\# Materials' properties
...
\# Absorbing BCs
ABSO 7

\# Dirichlet BCs
DIRX 6 8 0 0 0 0
DIRY 6 8 0 0 0 0
FUNC 8 0 0

...

Block 6: ABCs

**SPEED.input:**
\# Materials' properties
...

\# Absorbing BCs
ABSO 6
...
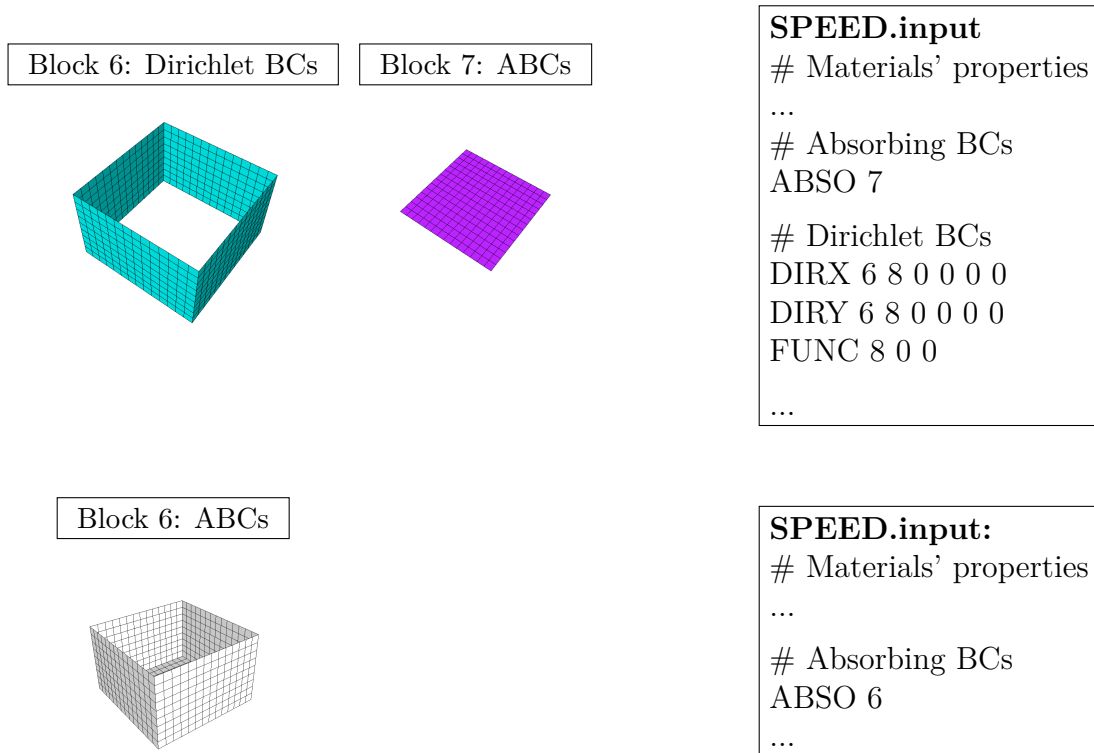
Figure 4.12: Boundary conditions for SPEED simulation of heterogeneous case. In the top panel are shown the blocks in which Dirichlet boundary conditions (block 6, lateral surface) and absorbing boundary conditions (block 7, bottom surface) are applied together with their setting on the file SPEED.input. In the bottom panel, instead, it is shown the unique block (block 6) in which absorbing boundary conditions are applied.

## 4.2.2.  Inclined plane wave: $\gamma = 10°$

In this section, we analyse what happens when the heterogeneous model is subject to the incidence of an oblique ($\gamma = 10°$) plane wave. This choice leads to a number of degrees of freedom for spectral element equal to $3(N+1)^3 = 81$.

As remarked above, we consider the spectral degree $N$ equal to 2. This is done in order to reduce the computational cost of the simulation and also because we noticed that increasing the spectral degree do not improve the goodness of the results in this particular situation.

As done before, we show in Figure 4.14 the horizontal $u_x$ and vertical $u_z$ displacement time histories obtained due to the incidence of plane $SV$-wave, and, in Figure 4.15, the same results but due to the incidence of a plane $P$-wave.

Once again, the top panel shows the displacements corresponding to all the monitored point except the one on the free surface (see red point in Figure 4.9), that, instead, is

P WAVE

Vertical displacement $u_z$
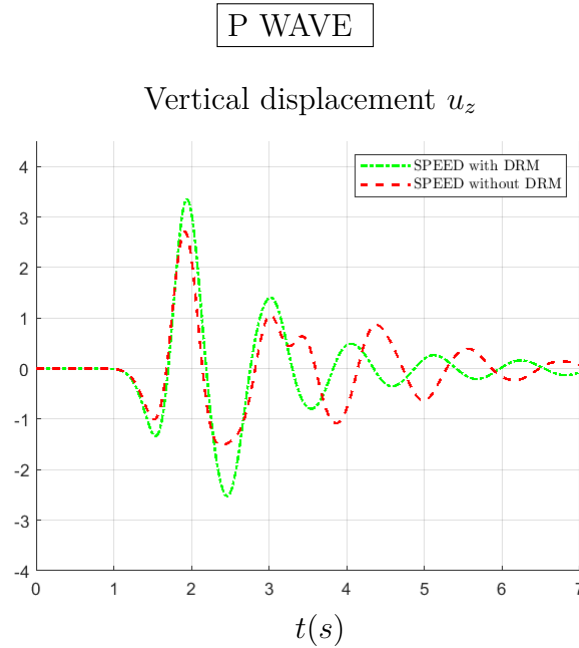


Figure 4.13: **Test case 2**. Comparison between the solution obtained with SPEED coupled with DRM (dashed green line) and the solution obtained with SPEED without DRM (dashed red line) for $z = 0$ due to the incidence of vertical $P$ plane wave. The solution obtained without DRM presents lots of spurious effects due to the presence of absorbing boundary conditions in the bottom surface and in the lateral surface.

plotted in the bottom panel.

Like TEST CASE 1, the comparison occurs only between the semi-analytical solution (computed with H-T method and depicted as a blue continuous line) and the numerical solution obtained with SPEED coupled with the DRM (dashed green line), because it is not possible to compute it with SPEED without DRM.

Similarly to what happens in the analogous case for the homogeneous model, the results present some spurious effects. The reason for this is probably twofold: minimally, they are due to ABCs, but mostly they are due to the change of the quality factors $Q_S$ and $Q_P$ in passing from the harder layer to the softer layer. This is especially visible when the monitored points get closer to the free-surface $z = 0$.

Once again, this is more evident in the vertical displacement $u_z$ for $SV$ plane wave case, and in the horizontal displacement $u_x$ for $P$ plane wave case, the displacements arising because of the inclination of the plane wave. Here, specially starting from $t = 3.5s$, the numerical solution results less damped than the analytical one.

SV WAVES

Horizontal displacement $u_x(m)$

Vertical displacement $u_z(m)$



Figure 4.14: **Test case 2**: heterogeneous model. Plot of horizontal $u_x$ and vertical $u_z$ displacements corresponding to the internal monitored points (top panel) and to the point located at the free surface (bottom panel) due to the incidence of an oblique ($\gamma = 10°$) plane $SV$ wave. Two solutions are compared: the blue line indicates the semi-analytical solution obtained with the H-T method, while the dashed green line represents the solution obtained with SPEED coupled with DRM.

P WAVES

Horizontal displacement $u_x(m)$                Vertical displacement $u_z(m)$



Figure 4.15: **Test case 2**: heterogeneous model. Plot of horizontal $u_x$ and vertical $u_z$ displacements corresponding to the internal monitored points (top panel) and to the point located at the free surface (bottom panel) due to the incidence of an inclined ($\gamma = 10°$) plane $P$ wave. Two solutions are compared: the blue line indicates the semi-analytical solution obtained with the H-T method, while the dashed green line represents the solution obtained with SPEED coupled with DRM.

## 4.3.   TEST CASE 3: "CROISSANT" VALLEY

The following test, proposed by Sanchez-Sesma and Luzon [1995], concerns the analysis of a three-dimensional alluvial valley under the action of different plane waves' types. Precisely, in this section we analyse the effects of the alluvial basin, here referred as *Croissant valley*, under the action of a vertical ($\gamma = 0°$) $SH$ plane wave.

The temporal dependence of the incident plane wavefront is now given by a Ricker wavelet with maximum frequency equal to $1Hz$.

After a brief presentation of the geological features of the model, we analyse the results in terms of $x$ and $y$ displacements components for a series of monitored points located on the free-surface of the model. Precisely, 98 points laying on the $x - y$ plane for $z = 0$ have been taken into considerations: 49 along the $x$-axis and 49 along the $y$-axis. Each point is distant from the adjacent one on the same axis by a distance of 320 m. However, for the sake of clarity, they are not reported here.

It is made a comparison between the semi-analitycal solution obtained by indirect boundary element method (IBEM) (see Sanchez-Sesma and Luzon [1995]), the numerical solution computed by SPEED coupled with DRM and the one calculated by SPEED without DRM.

In Figure 4.16 is shown the mesh of the reduced domain used for the simulation of SPEED with DRM. Here, one can also find the blocks that form the computational domain isolated one by one.

It is worth underlying that in this case, in order to avoid the spurious effects due to the ABCs, the computational domain used for the simulation by SPEED without the DRM is different from the one above. Precisely, the length of the domain along the $x$ and $y$ directions goes from $-4.2 \cdot 10^4$ $m$ to $4.2 \cdot 10^4$ $m$, differently from the DRM computational domain, in which $x$ and $y$ dimensions go from $-1.2 \cdot 10^4$ $m$ to $1.2 \cdot 10^4$ $m$. Figure 4.17 shows the computational domain used for traditional SPEED simulation.

In Table 4.6 we report the mechanical properties of the blocks. Obviously, all the four volumes blocks have the same properties since the model is homogeneous.

Note that the spectral degree $N$ is chosen equal to $N = 3$ for all the block, thus leading to have a number of degrees of freedom for spectral element equal to $3(N + 1)^3 = 192$.
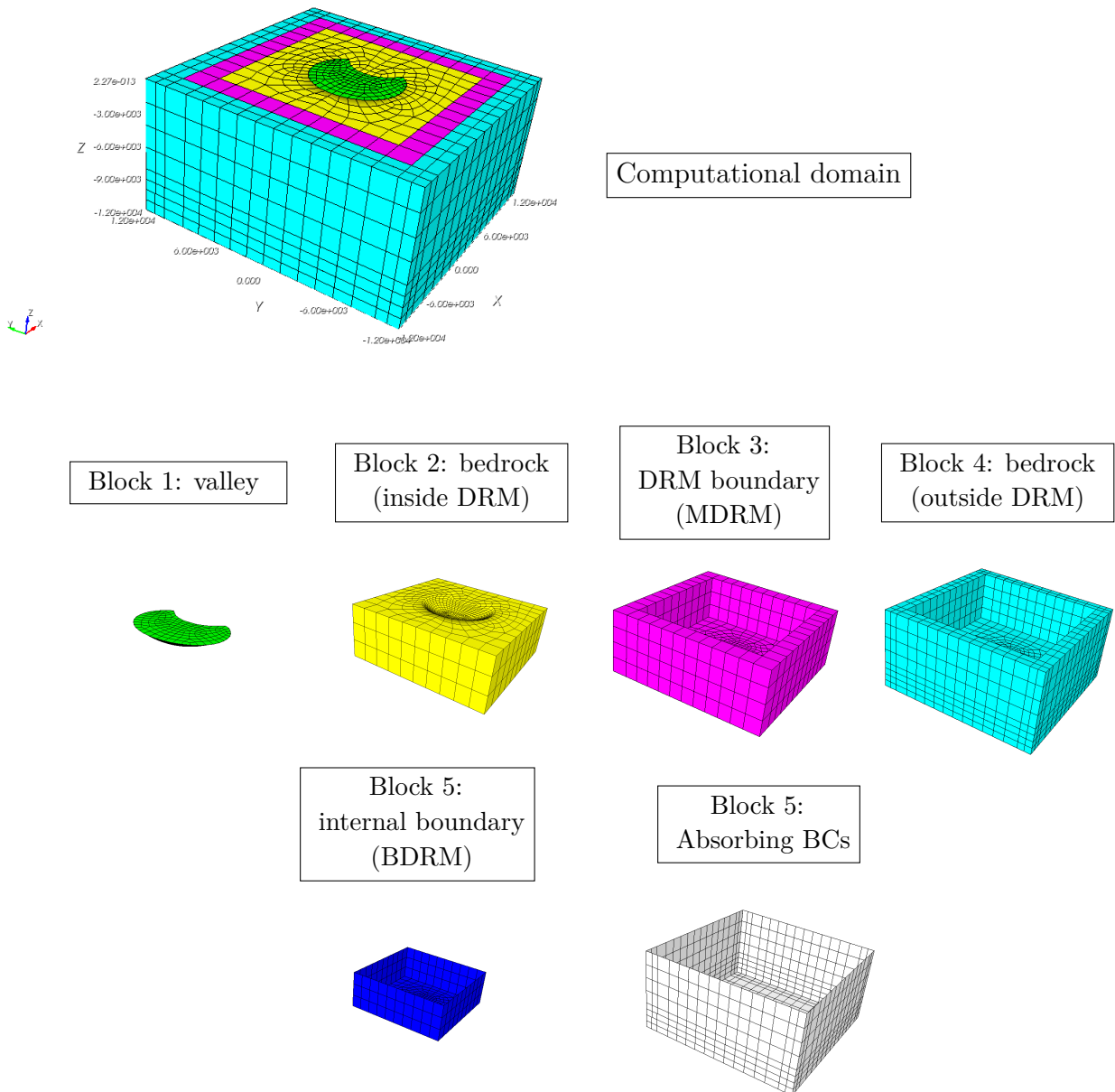
Figure 4.16: **Test case 3**. Blocks representation of "croissant" valley. The figure in the top panels represents the mesh of the entire DRM computational model. Then, all the blocks are isolated one by one, included those regarding the boundary conditions of the model.
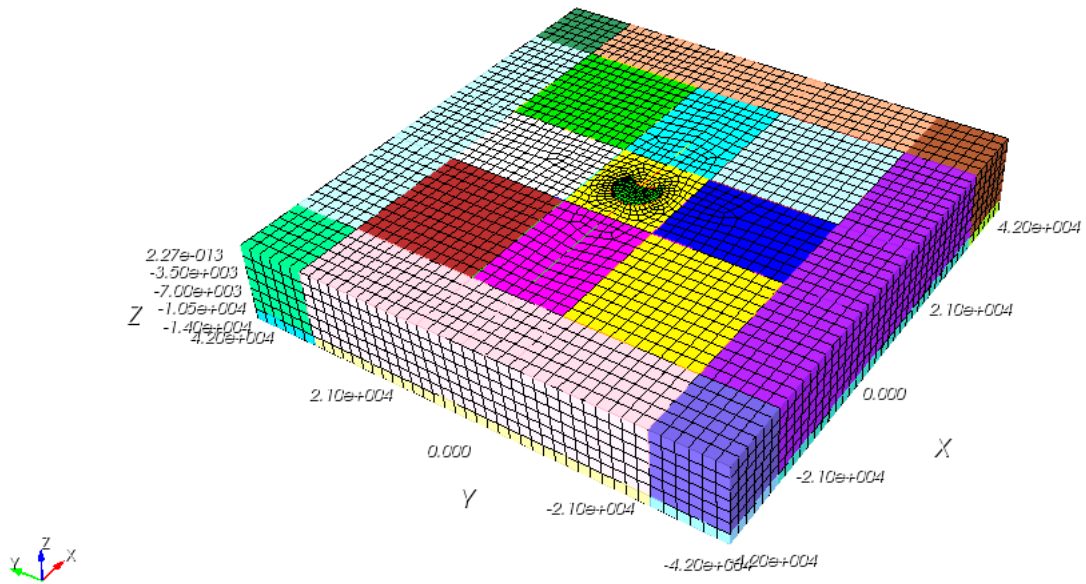
Figure 4.17: **Test case 3**. Computational domain used for the simulation of SPEED without the implementation of the DRM method. In $x$ and $y$ direction, this domain is more than three times as much as the DRM computational domain. This choice is made in order to avoid the spurious effects due to ABCs.

| | label | degree | $\rho$ $(kg/m^3)$ | $V_S$ (m/s) | $V_P$ (m/s) | $Q_S$ | $Q_P$ |
|---|---|---|---|---|---|---|---|
| **MATE** | 1 | 3 | 2000 | 1000 | 2081.9942 | 100 | 100 |
| **MATE** | 2 | 3 | 2500 | 2000 | 3463.9976 | 200 | 200 |
| **MATE** | 3 | 3 | 2500 | 2000 | 3463.9976 | 200 | 200 |
| **MATE** | 4 | 3 | 2500 | 2000 | 3463.9976 | 200 | 200 |

Table 4.6: **Test case 3**. Mechanical properties related to the four blocks composing the computational domain: the valley and the halfspace bedrock. The spectral degree is equal to 3. The bedrock outside the valley is homogeneous.

In Figure 4.18 we plot the results obtained for the $x$ component (top panel) and $y$ component (bottom panel) of the $u_y$ displacement.

As said before, three solutions are compared: semi-analytical solution obtained by IBEM (blue line), numerical solution obtained with SPEED coupled with DRM (green line), and numerical solution obtained with SPEED without DRM (red line).

Before making some comments on the results, we make a remark on the choice of the spectral degree $N$.

For SPEED coupled with DRM, $N$ is chosen equal to 3 in order to reduce the computational cost of the simulation. SPEED without DRM, instead, adopts $N = 4$ (number of degrees of freedom per spectral element $= 375$), since it is not possible to perform plane wave analysis with odd spectral degrees.

Now, if we look at the results, we can state that there is a good correspondence in the three solutions, specially for what concerns the $x$ component of the displacement. Once again, the good fit between the two numerical solutions obtained with SPEED proves the capability of the DRM from not being affected from impurities caused by ABCs.

SH WAVES

$u_y(x, 0, 0, t)$



$t(s)$

$u_y(0, y, 0, t)$



$t(s)$
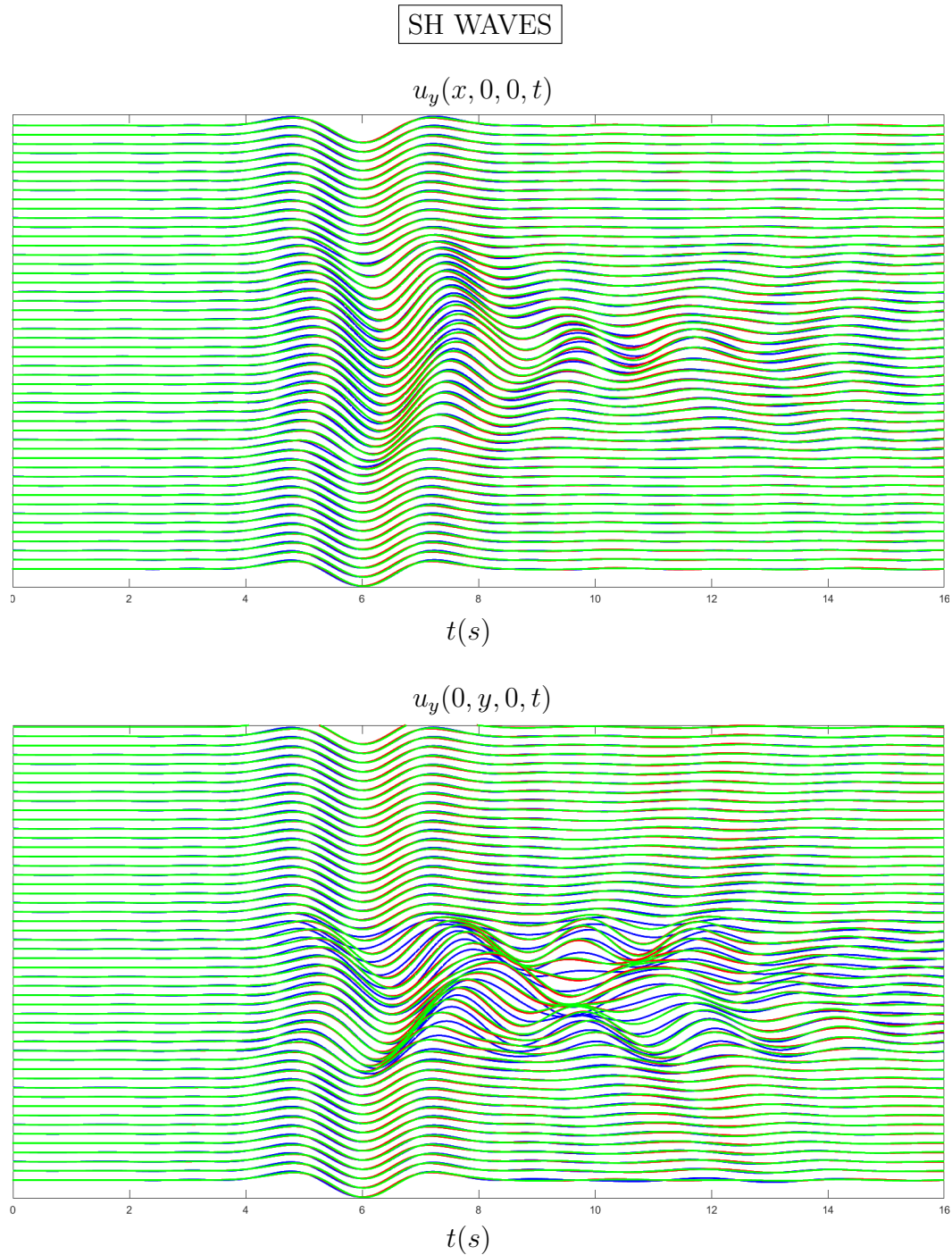
Figure 4.18: **Test case 3**: "croissant valley". Plot of $x$ component (top panel) and $y$ component (bottom panel) of the $u_y$ displacement corresponding to the surface monitored points along $x$ axis (top panel) and $y$ axis (bottom panel) due to the incidence of a vertical ($\gamma = 0°$) plane $SH$ wave. Three solutions are compared: the blue line indicates the semi-analytical solution obtained with IBEM, while the green line represents the solution obtained with SPEED coupled with DRM and, last, the red line shows the solution obtained with SPEED without DRM.

# Conclusions and future developments

In this thesis we have presented the Domain Reduction Method (DRM) starting from its theoretical formulation toghether with its implementation within the code SPEED (*SPectral Elements in Elastodynamics with Discontinuous Galerkin*) in the plane wave scenario. This has been achieved by exploiting the subroutines created on purpose by Smerzini [2010] in the code GeoELSE (*GeoELastodynamics by Spectral Elements*), that, however, is no longer mantained.

Particular attention has been paid to the creation of the input files needed to start the code. To this aim, we make use of the Matlab scripts already implemented by Smerzini [2011] and we reconstruct the passages necessary for the correct creation of the desired files.
In particular, differently from the previous usage of the code, it is not necessary to run SPEED twice in order to create the Legendre-Gauss-Lobatto nodes in which the effective forces are calculated. Indeed, this can be simply done with a Matlab function, thus allowing to change the code environment only once (from Matlab to SPEED).

Then, particular attention is made on the validation of three-dimensional test cases. In order to verify the validity of the method, we have compared the numerical solution obtained with SPEED coupled with DRM with the semi-analytical solution (obtained differently depending on the case). A second comparison has been made with the numerical solution obtained with the traditional implementation of SPEED, namely without DRM.

The analysis of the results leads to highlight some main advantages of using the DRM:

1. Capability of the DRM from not being affected almost entirely from impurities caused by absorbing boundary conditions. In particular, this allows to reduce a lot the size of the computational domain, thus permitting to decrease the computational cost of the simulation.

2. Possibility to perform simulations in which the plane wave is not vertical.

3. Possibility to perform parametric analyses with respect to different input motions,

including accelerograms derived from seismic hazard analysis.

Further developments include the following improvements.

We consider to analyze further cases, for example with a different seismic input type or considering more complex geometry.

Moreover, in the light of the obtained results, some precautions need to be made in the implementation of the method since there are cases that are not completely satisfactory. Indeed, especially if we look at the results obtained for the heterogeneous test case, the numerical solution obtained with SPEED coupled with the DRM is not perfectly capable of modeling the transition from one material to another, and this produces spurious effects. Another important future development regards the implementation of the three-dimensional parallel version of the code, currently implemented only in sequential version. This will lead, above all, to an advantageous reduction of the computational time.

# Bibliography

B. T. Aagaard, J. F. Hall, and T. H. Heaton. Characterization of near-source ground motions with earthquake simulations. *Earthquake Spectra*, 17(2):177–207, 2001.

M. Abramovitz and I. Stegun. Handbook of mathematical formulas graphs and mathematical tables. *US Department of Commerce, National Bureau of Standards, Washington, DC*, 1966.

K. Aki and P. G. Richards. *Quantitative seismology*. 2002.

R. M. Alford, K. R. Kelly, and D. M. Boore. Accuracy of finite-difference modeling of the acoustic wave equation. *Geophysics*, 39(6):834–842, 1974.

D. N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM journal on numerical analysis*, 19(4):742–760, 1982.

D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous galerkin methods for elliptic problems. *SIAM journal on numerical analysis*, 39(5): 1749–1779, 2002.

M. N. Aydinoğlu. Consistent formulation of direct and substructure methods in nonlinear soil-structure interaction. *Soil Dynamics and Earthquake Engineering*, 12(7):403–410, 1993.

I. Babuška and M. Suri. The *hp* version of the finite element method with quasiuniform meshes. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 21(2):199–238, 1987.

I. Babuska, B. A. Szabo, and I. N. Katz. The p-version of the finite element method. *SIAM journal on numerical analysis*, 18(3):515–545, 1981.

H. Bao. *Finite element simulation of earthquake ground motion in realistic basins*. PhD thesis, Carnegie Mellon University, 1998.

H. Bao, J. Bielak, O. Ghattas, L. F. Kallivokas, D. R. O'Hallaron, J. R. Shewchuk, and J. Xu. Large-scale simulation of elastic wave propagation in heterogeneous media on

parallel computers. *Computer methods in applied mechanics and engineering*, 152(1-2): 85–102, 1998.

J. Bielak and P. Christiano. On the effective seismic input for non-linear soil-structure interaction systems. *Earthquake engineering & structural dynamics*, 12(1):107–119, 1984.

J. Bielak, R. C. MacCamy, D. S. McGhee, and A. Barry. Unified symmetric bem-fem for site effects on ground motion—sh waves. *Journal of engineering mechanics*, 117(10): 2265–2285, 1991.

J. Bielak, K. Loukakis, Y. Hisada, and C. Yoshimura. Domain reduction method for three-dimensional earthquake modeling in localized regions, part i: Theory. *Bulletin of the seismological Society of America*, 93(2):817–824, 2003.

M. Board. Flac (fast lagrangian analysis of continua), version 2. 20. Technical report, Nuclear Regulatory Commission, Washington, DC (USA). Div. of High-Level . . . , 1989.

D. M. Boore. Love waves in nonuniform wave guides: Finite difference calculations. *Journal of Geophysical Research*, 75(8):1512–1527, 1970.

J. P. Boyd. *Chebyshev and Fourier spectral methods.* Courier Corporation, 2001.

C. Canuto, M. Y. Hussaini, A. Quarteroni, T. A. Zang, C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. Fundamentals of spectral methods for pdes. *Spectral methods in fluid dynamics*, pages 76–93, 1988.

C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral methods: fundamentals in single domains.* Springer Science & Business Media, 2007.

F. Casadei and E. Gabellini. *Implementation of a 3D coupled spectral element/finite element solver for wave propagation and soil-structure interaction simulations: numerical examples.* Office for Official Publications of the European Communities, 1998.

E. T. Chung and B. Engquist. Optimal discontinuous galerkin methods for wave propagation. *SIAM Journal on Numerical Analysis*, 44(5):2131–2158, 2006.

R. Clayton and B. Engquist. Absorbing boundary conditions for acoustic and elastic wave equations. *Bulletin of the seismological society of America*, 67(6):1529–1540, 1977.

B. Cockburn, G. E. Karniadakis, and C. Shu. *Discontinuous Galerkin methods: theory, computation and applications*, volume 11. Springer Science & Business Media, 2012.

M. G. Cremonini, P. Christiano, and J. Bielak. Implementation of effective seismic input

for soil-structure interaction systems. *Earthquake engineering & structural dynamics*, 16(4):615–625, 1988.

J. D. De Basabe and M. K. Sen. Grid dispersion and stability criteria of some common finite-element methods for acoustic and elastic wave equations. *Geophysics*, 72(6):T81–T95, 2007.

J. D. De Basabe and M. K. Sen. Stability of the high-order finite elements for acoustic or elastic wave propagation with high-order time stepping. *Geophysical Journal International*, 181(1):577–590, 2010.

M. Dumbser, M. Käser, and E. F. Toro. An arbitrary high-order discontinuous galerkin method for elastic waves on unstructured meshes-v. local time stepping and p-adaptivity. *Geophysical Journal International*, 171(2):695–717, 2007.

E. Faccioli, F. Maggio, R. Paolucci, and A. Quarteroni. 2d and 3d elastic wave propagation by a pseudo-spectral domain decomposition method. *Journal of seismology*, 1:237–251, 1997.

E. Faccioli, M. Vanini, R. Paolucci, and M. Stupazzini. Comment on "domain reduction method for three-dimensional earthquake modeling in localized regions, part i: Theory," by j. bielak, k. loukakis, y. hisada, and c. yoshimura, and "part ii: Verification and applications," by c. yoshimura, j. bielak, y. hisada, and a. fernandez. *Bulletin of the Seismological Society of America*, 95(2):763–769, 2005.

E. Faccioli, M. Vanini, M. Villani, C. Cauzzi, C. Smerzini, et al. Mapping seismic hazard to account for basin amplification effects. 2010.

D. Fäh, P. Suhadolc, and G. F. Panza. A hybrid method for the estimation of ground motion in sedimentary basins: quantitative modeling for mexico city. *Bulletin of the Seismological Society of America*, 84(2):383–399, 1994.

R. S. Falk and G. R. Richter. Explicit finite element methods for symmetric hyperbolic equations. *SIAM Journal on Numerical Analysis*, 36(3):935–952, 1999.

A. Frankel. Three-dimensional simulations of ground motions in the san bernardino valley, california, for hypothetical earthquakes on the san andreas fault. *Bulletin of the Seismological Society of America*, 83(4):1020–1041, 1993.

A. Frankel and J. Vidale. A three-dimensional simulation of seismic waves in the santa clara valley, california, from a loma prieta aftershock. *Bulletin of the Seismological Society of America*, 82(5):2045–2074, 1992.

T. Furumura and T. Hayakawa. Anomalous propagation of long-period ground motions recorded in tokyo during the 23 october 2004 m w 6.6 niigata-ken chuetsu, japan, earthquake. *Bulletin of the Seismological Society of America*, 97(3):863–880, 2007.

P. Gervasio. Spectral element library-chqz lib release 1.0. 2007.

F. X. Giraldo, J. S. Hesthaven, and T. Warburton. Nodal high-order discontinuous galerkin methods for the spherical shallow water equations. *Journal of Computational Physics*, 181(2):499–525, 2002.

K. F. Graff. *Wave motion in elastic solids.* Courier Corporation, 2012.

R. W. Graves. Modeling three-dimensional site response effects in the marina district basin, san francisco, california. *Bulletin of the Seismological Society of America*, 83(4): 1042–1063, 1993.

R. W. Graves. Simulating seismic wave propagation in 3d elastic media using staggered-grid finite differences. *Bulletin of the seismological society of America*, 86(4):1091–1106, 1996.

N. A. Haskell. The dispersion of surface waves on multilayered media. *Vincit Veritas: A Portrait of the Life and Work of Norman Abraham Haskell, 1905–1970*, 30:86–103, 1990.

J. S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications.* Springer Science & Business Media, 2007.

Y. Hisada. An efficient method for computing green's functions for a layered half-space with sources and receivers at close depths. *Bulletin of the Seismological Society of America*, 84(5):1456–1472, 1994.

Y. Hisada. An efficient method for computing green's functions for a layered half-space with sources and receivers at close depths (part 2). *Bulletin of the Seismological Society of America*, 85(4):1080–1093, 1995.

Y. Hisada and J. Bielak. A theoretical method for computing near-fault ground motions in layered half-spaces considering static offset due to surface faulting, with a physical interpretation of fling step and rupture directivity. *Bulletin of the Seismological Society of America*, 93(3):1154–1168, 2003.

Y. Hisada, K. Aki, and T. Teng. 3-d simulations of surface wave propagation in the kanto sedimentary basin, japan part 2: Application of the surface wave bem. *Bulletin of the Seismological Society of America*, 83(6):1700–1720, 1993.

M. Käser and M. Dumbser. An arbitrary high-order discontinuous galerkin method for elastic waves on unstructured meshes—i. the two-dimensional isotropic case with external source terms. *Geophysical Journal International*, 166(2):855–877, 2006.

K. Kato, K. Aki, and T. Teng. 3-d simulations of surface wave propagation in the kanto sedimentary basin, japan—part 1: Application of the surface wave gaussian beam method. *Bulletin of the Seismological Society of America*, 83(6):1676–1699, 1993.

H. Kawase and K. Aki. Topography effect at the critical sv-wave incidence: possible explanation of damage pattern by the whittier narrows, california, earthquake of 1 october 1987. *Bulletin of the Seismological Society of America*, 80(1):1–22, 1990.

D. Komatitsch and J. Tromp. Introduction to the spectral element method for three-dimensional seismic wave propagation. *Geophysical journal international*, 139(3):806–822, 1999.

D. Komatitsch and J. Vilotte. The spectral element method: an efficient tool to simulate the seismic response of 2d and 3d geological structures. *Bulletin of the seismological society of America*, 88(2):368–392, 1998.

D. Komatitsch, S. Tsuboi, C. Ji, and J. Tromp. A 14.6 billion degrees of freedom, 5 teraflops, 2.5 terabyte earthquake simulation on the earth simulator. In *Proceedings of the 2003 ACM/IEEE conference on Supercomputing*, page 4, 2003.

D. Komatitsch, Q. Liu, J. Tromp, P. Suss, C. Stidham, and J. H. Shaw. Simulations of ground motion in the los angeles basin based upon the spectral-element method. *Bulletin of the Seismological Society of America*, 94(1):187–206, 2004.

D. Kosloff and E. Baysal. Forward modeling by a fourier method. *Geophysics*, 47(10): 1402–1412, 1982.

D. Kosloff, D. Kessler, A. Q. Filho, E. Tessmer, A. Behle, and R. Strahilevitz. Solution of the equations of dynamic elasticity by a chebychev spectral method. *Geophysics*, 55 (6):734–748, 1990.

K. Loukakis. *Transient response of shallow layered valleys for inclined incident SV waves calculated by the finite element method*. PhD thesis, 1988.

K. Loukakis. Sesimic response of two-dimensional sediment-filled valleys to oblique incident sv-waves calculated by the finite element method. In *Proceedings of the fifth US national conference on earthquake engineering, July 10-14, 1994, Chicago*, pages 25–34, 1994.

K. Loukakis and J. Bielak. Layering and damping effects on seismic response of sedimentary valleys to oblique excitation. In *Proceedings of the Second International Conference on Earthquake Resistant Construction and Design*, pages 15–17, 1994.

J. Lysmer and L. A. Drake. The propagation of love waves across nonhorizontally layered structures. *Bulletin of the Seismological Society of America*, 61(5):1233–1251, 1971.

Y. Maday and A. T. Patera. Spectral element methods for the incompressible navier-stokes equations. *IN: State-of-the-art surveys on computational mechanics (A90-47176 21-64). New York*, pages 71–143, 1989.

I. Mazzieri. Non-conforming high order methods for the elastodynamics equation. 2012.

I. Mazzieri, M. Stupazzini, R. Guidotti, and C. Smerzini. Speed: Spectral elements in elastodynamics with discontinuous galerkin: A non-conforming approach for 3d multiscale problems. *International Journal for Numerical Methods in Engineering*, 95(12): 991–1010, 2013.

A. Mita and J. E. Luco. Dynamic response of embedded foundations: a hybrid approach. *Computer methods in applied mechanics and engineering*, 63(3):233–259, 1987.

P. Moczo, M. Lucká, J. Kristek, and M. Kristeková. 3d displacement finite differences and a combined memory optimization. *Bulletin of the Seismological Society of America*, 89 (1):69–79, 1999.

T. K. Mossessian and M. Dravinski. Application of a hybrid method for scattering of p, sv, and rayleigh waves by near-surface irregularities. *Bulletin of the Seismological Society of America*, 77(5):1784–1803, 1987.

K. B. Olsen and R. J. Archuleta. Three-dimensional simulation of earthquakes on the los angeles fault system. *Bulletin of the Seismological Society of America*, 86(3):575–596, 1996.

K. B. Olsen, J. C. Pechmann, and G. T. Schuster. Simulation of 3d elastic wave propagation in the salt lake basin. *Bulletin of the Seismological Society of America*, 85(6): 1688–1710, 1995.

I. Oprsal and J. Zahradnik. Three-dimensional finite difference method and hybrid modeling of earthquake ground motion. *Journal of Geophysical Research: Solid Earth*, 107 (B8):ESE–2, 2002.

A. T. Patera. A spectral element method for fluid dynamics: laminar flow in a channel expansion. *Journal of computational Physics*, 54(3):468–488, 1984.

A. Pitarka. 3d elastic finite-difference modeling of seismic motion using staggered grids with nonuniform spacing. *Bulletin of the Seismological Society of America*, 89(1):54–68, 1999.

E. Por, M. Van Kooten, and V. Sarkovic. Nyquist–shannon sampling theorem. *Leiden University*, 1(1), 2019.

E. Priolo, J. M. Carcione, and G. Seriani. Numerical simulation of interface waves by high-order spectral modeling techniques. *The Journal of the Acoustical Society of America*, 95(2):681–693, 1994.

A. Quarteroni, R. Sacco, F. Saleri, and P. Gervasio. I fondamenti della matematica numerica. *Matematica Numerica*, pages 33–58, 2014.

W. H. Reed and T. R. Hill. Triangular mesh methods for the neutron transport equation. Technical report, Los Alamos Scientific Lab., N. Mex.(USA), 1973.

B. Riviere, S. Shaw, M. F. Wheeler, and J. R. Whiteman. Discontinuous galerkin finite element methods for linear elasticity and quasistatic linear viscoelasticity. *Numerische Mathematik*, 95:347–376, 2003.

F. J. Sanchez-Sesma and F. Luzon. Seismic response of three-dimensional alluvial valleys for incident p, s, and rayleigh waves. *Bulletin of the Seismological Society of America*, 85(1):269–284, 1995.

L. Scandella. *Numerical evaluation of transient ground strains for the seismic response analysis of underground structures.* PhD thesis, PhD Thesis, Politecnico di Milano, 2007.

C. Smerzini. *The earthquake source in numerical modeling of seismic wave propagation in heterogeneous earth's media.* PhD thesis, PhD thesis, ROSE School, IUSS, Pavia, Italy, 2010.

C. Smerzini. Geoelse2d user guide for plane wave propagation. 2011.

R. Stacey. Improved transparent boundary formulations for the elastic-wave equation. *Bulletin of the Seismological Society of America*, 78(6):2089–2097, 1988.

M. Stupazzini. A spectral element approach for 3d dynamic soil-structure interaction problems (phd thesis). *Politecnico di Milano*, 2004.

M. Stupazzini, C. Zambelli, et al. Geo-elsevp: a spectral element approach for 2d or 3d dynamic elasto-viscoplastic problems. *Rivista Italiana di Geotecnica*, 39(4):70–82, 2005.

W. T. Thomson. Transmission of elastic waves through a stratified solid medium. *Journal of applied Physics*, 21(2):89–93, 1950.

M. Tomasin. Non-linear soil-structure interaction for 3d embedded systems under seismic excitation. 2018.

T. Toshinawa and T. Ohmachi. Love-wave propagation in a three-dimensional sedimentary basin. *Bulletin of the Seismological Society of America*, 82(4):1661–1677, 1992.

D. J. Wald and R. W. Graves. The seismic response of the los angeles basin, california. *Bulletin of the Seismological Society of America*, 88(2):337–356, 1998.

M. F. Wheeler. An elliptic collocation-finite element method with interior penalties. *SIAM Journal on Numerical Analysis*, 15(1):152–161, 1978.

L. C. Wilcox, G. Stadler, C. Burstedde, and O. Ghattas. A high-order discontinuous galerkin method for wave propagation through coupled elastic–acoustic media. *Journal of Computational Physics*, 229(24):9373–9396, 2010.

C. Yoshimura, J. Bielak, Y. Hisada, and A. Fernández. Domain reduction method for three-dimensional earthquake modeling in localized regions, part ii: Verification and applications. *Bulletin of the seismological Society of America*, 93(2):825–841, 2003.

J. Zahradník and P. Moczo. Hybrid seismic modeling based on discrete-wave number and finite-difference methods. *Seismic Waves in Laterally Inhomogeneous Media: Part 1*, pages 21–38, 1996.

O. C. Zienkiewicz and R. L. Taylor. The finite element method, 4th edn., vol. 1. *Basic Formulation and Linear*, 1989.

# A | Appendix A

In this section we report the implementation of the stiffness DRM matrix, called $K_{DRM}$ (see Section 2.2.1). The subroutine shown below was implemented in SPEED making reference to the one implemented in GeoELSE by Scandella [2007].

We underline that the programming language used is Fortran90.

```fortran
call lgl(nn,ct,ww,dd)

do el = 1,n_el_DRM
   ie = el_DRM(el,1)
  do  imDRM = 1,nMDRM
    if (cs_loc(cs_loc(ie -1) +0).eq.tag_MDRM(imDRM)) then
       do im = 1,nm   !For each material
         if (tag_mat(im).eq.tag_MDRM(imDRM)) then

            lambda = prop_mat(im,2)
            mu = prop_mat(im,3)
          endif
       enddo

       allocate(stiff_el(3*nn*nn*nn,3*nn*nn*nn))
       stiff_el = 0.0

       do lk = 1,nn
         do lj = 1,nn
           do li = 1,nn
             !Node number in the reference element
             l = (lk-1)*nn*nn+(lj-1)*nn+li
             do mk =1,nn
               do mj =1,nn
                 do mi =1,nn
                   !Node number in the reference element
                   m = (mk-1)*nn*nn+(mj-1)*nn+mi

                   !Partial stiff matrix initialization
                   do i=1,3
```

```
30                      do j= 1,3
31                        stiff_ij(i,j) = 0
32                      enddo
33                    enddo
34
35                !Integration loop on nodes K
36                do kk = 1,nn
37                  do kj = 1,nn
38                    do ki = 1,nn
39                      !Node number in the reference element
40                      k = (kk-1)*nn*nn+(kj-1)*nn+ki
41
42                      dxdx = alfa11(ie) + beta12(ie)*ct(kk) &
43                              + beta13(ie)*ct(kj) &
44                              + gamma1(ie)*ct(kj)*ct(kk)
45                      dydx = alfa21(ie) + beta22(ie)*ct(kk) &
46                              + beta23(ie)*ct(kj) &
47                              + gamma2(ie)*ct(kj)*ct(kk)
48                      dzdx = alfa31(ie) + beta32(ie)*ct(kk) &
49                              + beta33(ie)*ct(kj) &
50                              + gamma3(ie)*ct(kj)*ct(kk)
51
52                      dxdy = alfa12(ie) + beta11(ie)*ct(kk) &
53                              + beta13(ie)*ct(ki) &
54                              + gamma1(ie)*ct(kk)*ct(ki)
55                      dydy = alfa22(ie) + beta21(ie)*ct(kk) &
56                              + beta23(ie)*ct(ki) &
57                              + gamma2(ie)*ct(kk)*ct(ki)
58                      dzdy = alfa32(ie) + beta31(ie)*ct(kk) &
59                              + beta33(ie)*ct(ki) &
60                              + gamma3(ie)*ct(kk)*ct(ki)
61
62                      dxdz = alfa13(ie) + beta11(ie)*ct(kj) &
63                              + beta12(ie)*ct(ki) &
64                              + gamma1(ie)*ct(ki)*ct(kj)
65                      dydz = alfa23(ie) + beta21(ie)*ct(kj) &
66                              + beta22(ie)*ct(ki) &
67                              + gamma2(ie)*ct(ki)*ct(kj)
68                      dzdz = alfa33(ie) + beta31(ie)*ct(kj) &
69                              + beta32(ie)*ct(ki) &
70                              + gamma3(ie)*ct(ki)*ct(kj)
71
72
73
```

```fortran
                      !Jacobian determinant
                      det_j = dxdz * (dydx*dzdy - dzdx*dydy) &
                               - dydz * (dxdx*dzdy - dzdx*dxdy) &
                               + dzdz * (dxdx*dydy - dydx*dxdy)

                      !Derivation of shape functions respect to local
                         riferiment
                      !(equal to spectral derivate if no zero)
                      if ((kj.eq.lj).and.(kk.eq.lk)) then
                         dfi_i_dcsi_l = dd(ki,li)
                      else
                         dfi_i_dcsi_l = 0
                      endif
                      if ((ki.eq.li).and.(kk.eq.lk)) then
                         dfi_j_deta_l = dd(kj,lj)
                      else
                         dfi_j_deta_l = 0
                      endif
                      if ((ki.eq.li).and.(kj.eq.lj)) then
                         dfi_k_zeta_l = dd(kk,lk)
                      else
                         dfi_k_zeta_l = 0
                      endif

                      if ((kj.eq.mj).and.(kk.eq.mk)) then
                         dfi_i_dcsi_m = dd(ki,mi)
                      else
                         dfi_i_dcsi_m = 0
                      endif
                      if ((ki.eq.mi).and.(kk.eq.mk))  then
                         dfi_j_deta_m = dd(kj,mj)
                      else
                         dfi_j_deta_m = 0
                      endif
                      if ((ki.eq.mi).and.(kj.eq.mj))  then
                         dfi_k_zeta_m = dd(kk,mk)
                      else
                         dfi_k_zeta_m = 0
                      endif

                      !Derivation of shape functions respect to
                      !global riferiment
                       dfi_i_dx = (dydx*(dzdy*dfi_k_zeta_l - &
                                  dzdz*dfi_j_deta_l) &
```

```
117                                        - dydy*(dzdx*dfi_k_zeta_l - dzdz*
                                                dfi_i_dcsi_l) &
118                                        + dydz*(dzdx*dfi_j_deta_l - dzdy*
                                                dfi_i_dcsi_l))/det_j
119
120                      dfi_i_dy = (dzdx*(dxdy*dfi_k_zeta_l - &
121                                 dxdz*dfi_j_deta_l) &
122                                        - dzdy*(dxdx*dfi_k_zeta_l - dxdz*
                                                dfi_i_dcsi_l)&
123                                        + dzdz*(dxdx*dfi_j_deta_l - dxdy*
                                                dfi_i_dcsi_l))/det_j
124
125                      dfi_i_dz = (dxdx*(dydy*dfi_k_zeta_l - &
126                                 dydz*dfi_j_deta_l) &
127                                        - dxdy*(dydx*dfi_k_zeta_l - dydz*
                                                dfi_i_dcsi_l) &
128                                        + dxdz*(dydx*dfi_j_deta_l - dydy*
                                                dfi_i_dcsi_l))/det_j
129
130                      dfi_j_dx = (dydx*(dzdy*dfi_k_zeta_m - &
131                                 dzdz*dfi_j_deta_m) &
132                                        - dydy*(dzdx*dfi_k_zeta_m - dzdz*
                                                dfi_i_dcsi_m) &
133                                        + dydz*(dzdx*dfi_j_deta_m - dzdy*
                                                dfi_i_dcsi_m))/det_j
134
135                      dfi_j_dy = (dzdx*(dxdy*dfi_k_zeta_m - &
136                                 dxdz*dfi_j_deta_m) &
137                                        - dzdy*(dxdx*dfi_k_zeta_m - dxdz*
                                                dfi_i_dcsi_m) &
138                                        + dzdz*(dxdx*dfi_j_deta_m - dxdy*
                                                dfi_i_dcsi_m))/det_j
139
140                      dfi_j_dz = (dxdx*(dydy*dfi_k_zeta_m - &
141                                 dydz*dfi_j_deta_m) &
142                                        - dxdy*(dydx*dfi_k_zeta_m - dydz*
                                                dfi_i_dcsi_m) &
143                                        + dxdz*(dydx*dfi_j_deta_m - dydy*
                                                dfi_i_dcsi_m))/det_j
144
145
146
147
148
```

```fortran
                         !Stiff matrix components
                         stiff_ij(1,1) = stiff_ij(1,1) + &
                                         det_j*ww(ki)*ww(kj)*ww(kk) &
                                         *((lambda+2*mu)*dfi_i_dx*
                                             dfi_j_dx &
                                         + mu*dfi_i_dy*dfi_j_dy + & mu*
                                             dfi_i_dz*dfi_j_dz)

                         stiff_ij(1,2) = stiff_ij(1,2) + &
                                         det_j*ww(ki)*ww(kj)*ww(kk) &
                                         *(lambda*dfi_i_dx*dfi_j_dy &
                                         + mu*dfi_i_dy*dfi_j_dx)

                         stiff_ij(1,2) = stiff_ij(1,2) + &
                                         det_j*ww(ki)*ww(kj)*ww(kk) &
                                         *(lambda*dfi_i_dx*dfi_j_dy &
                                         + mu*dfi_i_dy*dfi_j_dx)

                         stiff_ij(1,3) = stiff_ij(1,3) + &
                                         det_j*ww(ki)*ww(kj)*ww(kk) &
                                         *(lambda*dfi_i_dx*dfi_j_dz &
                                         + mu*dfi_i_dz*dfi_j_dx)

                         stiff_ij(2,1) = stiff_ij(2,1) + &
                                         det_j*ww(ki)*ww(kj)*ww(kk) &
                                         *(lambda*dfi_i_dy*dfi_j_dx &
                                         + mu*dfi_i_dx*dfi_j_dy)

                         stiff_ij(2,2) = stiff_ij(2,2) + &
                                         det_j*ww(ki)*ww(kj)*ww(kk) &
                                         *((lambda+2*mu)*dfi_i_dy*
                                             dfi_j_dy &
                                         + mu*dfi_i_dx*dfi_j_dx + mu*
                                             dfi_i_dz*dfi_j_dz)

                         stiff_ij(2,3) = stiff_ij(2,3) + &
                                         det_j*ww(ki)*ww(kj)*ww(kk) &
                                         *(lambda*dfi_i_dy*dfi_j_dz &
                                         + mu*dfi_i_dz*dfi_j_dy)

                         stiff_ij(3,1) = stiff_ij(3,1) + &
                                         det_j*ww(ki)*ww(kj)*ww(kk) &
                                         *(lambda*dfi_i_dz*dfi_j_dx &
                                         + mu*dfi_i_dx*dfi_j_dz)
```

```
189
190                              stiff_ij(3,2) = stiff_ij(3,2) + &
191                                      det_j*ww(ki)*ww(kj)*ww(kk) &
192                                      *(lambda*dfi_i_dz*dfi_j_dy &
193                                      + mu*dfi_i_dy*dfi_j_dz)
194
195                              stiff_ij(3,3) = stiff_ij(3,3) + &
196                                      det_j*ww(ki)*ww(kj)*ww(kk) &
197                                      *((lambda+2*mu)*dfi_i_dz*
                                             dfi_j_dz &
198                                      + mu*dfi_i_dx*dfi_j_dx + mu*
                                             dfi_i_dy*dfi_j_dy)
199                        enddo
200                      enddo
201                    enddo
202
203                    !Element stiff matrix
204                    stiff_el(3*l-2,3*m-2) = stiff_ij(1,1)
205                    stiff_el(3*l-2,3*m-1) = stiff_ij(1,2)
206                    stiff_el(3*l-2,3*m)   = stiff_ij(1,3)
207                    stiff_el(3*l-1,3*m-2) = stiff_ij(2,1)
208                    stiff_el(3*l-1,3*m-1) = stiff_ij(2,2)
209                    stiff_el(3*l-1,3*m)   = stiff_ij(2,3)
210                    stiff_el(3*l,3*m-2)   = stiff_ij(3,1)
211                    stiff_el(3*l,3*m-1)   = stiff_ij(3,2)
212                    stiff_el(3*l,3*m)     = stiff_ij(3,3)
213                      enddo
214                    enddo
215                  enddo
216                enddo
217              enddo
218            enddo
219
220          K_DRM(1+(el-1)*3*nn*nn*nn:el*3*nn*nn*nn,1:3*nn*nn*nn) =
                stiff_el
221        deallocate(stiff_el)
222      endif
223    enddo
224 enddo
225
226 deallocate(ct,ww,dd)
```

# List of Figures

# List of Tables