WAVES: Explicit Parallelized Finite Element Solver for Wave Propagation Analysis

Juan Gomez and Juan Carlos Vergara

May 20, 2017

1 Introduction

Topographic effects have been shown to play a major role in the determination of the seismic ground motions at a site. In the idealized case of homogeneous media submitted to horizontally polarized shear waves there is a relatively large family of analytic or semi-analytic frequency domain solutions. However in the more general case of in-plane waves or in actual three-dimensional topographies the problem needs to be solved numerically. In particular, algorithms based upon the finite element method (FEM) have the advantage of allowing the incorporation of arbitrary boundary conditions and material models. This article describes the explicit finite element code **WAVES** which has been created as part of the research program on topographic effects conducted at the research lab Grupo de Mecanica Aplicada at Universidad EAFIT. Although the code has been originally implemented for the solution of plane wave scattering problems in the time domain, it is also possible to conduct analysis of generalized dynamic problems in 2D and 3D domains.

In simple terms the computer program finds the displacement time history for arbitrary two-dimensional and three-dimensional domains discretized into finite elements. As its main features the code allows to conduct plane wave analysis using as input excitation the domain reduction (DRM) technique formulated by **?**, while the time discretization corresponds to a central difference scheme (having diagonal mass matrices) with uncoupled equations for each degree of freedom.

In the first part of the report we focus on the theoretical aspects of the DRM approach, including the formulation of the elastodynamics scattering problem. This theoretical discussion is presented directly in matrix form assuming that the reader is experienced with finite element algorithms. Following the discussion of the DRM method we present the time-domain discretization resulting in an explicit solution scheme. In part 4 of the report we describe the most relevant structural aspects of the program with particular emphasis on the addition of user defined finite elements and material models. In the final part of the report we show the model creation process using as study problem the case of a V-shaped canyon embedded in an elastic half-space. Since the model involves a large number of elements the required input files are created with the aid of third party software..

2 Formulation of the plane wave scattering problem

Classical formulation of the scattering problem

In order to have a general context of the physical basis of the DRM technique formulated by **?** it is convenient to describe the so-called scattering problem in elastodynamics. We will use an integral equation approach based on the elastodynamics representation theorem (**?**). The physical problem is schematized by the top part of fig. 1 which depicts a generalized half-space with domain Ω^+ supporting a scatterer with domain Ω. Both domains are bounded through the perfectly coupled surface Γ and the remote boundary of the half-space Γ^+. Notice that the scatterer also comprises a localized, small-scale topographic feature (e.g., a microzone) embedded into a large topographic irregularity. The scattering problem consists in determining the response of the system when it is submitted to the action of an incident wave.

The domain reduction method, originally formulated by **?** and verified in **?** is a two-step algorithm developed for the simulation of earthquake ground motions in seismic scenarios containing strong variations in wavelength. The fundamental principle underlying the method is the classical partition of the field used in the formulation of scattering problems (**???**) written like ;

$$u = u^0 + w \tag{1}$$

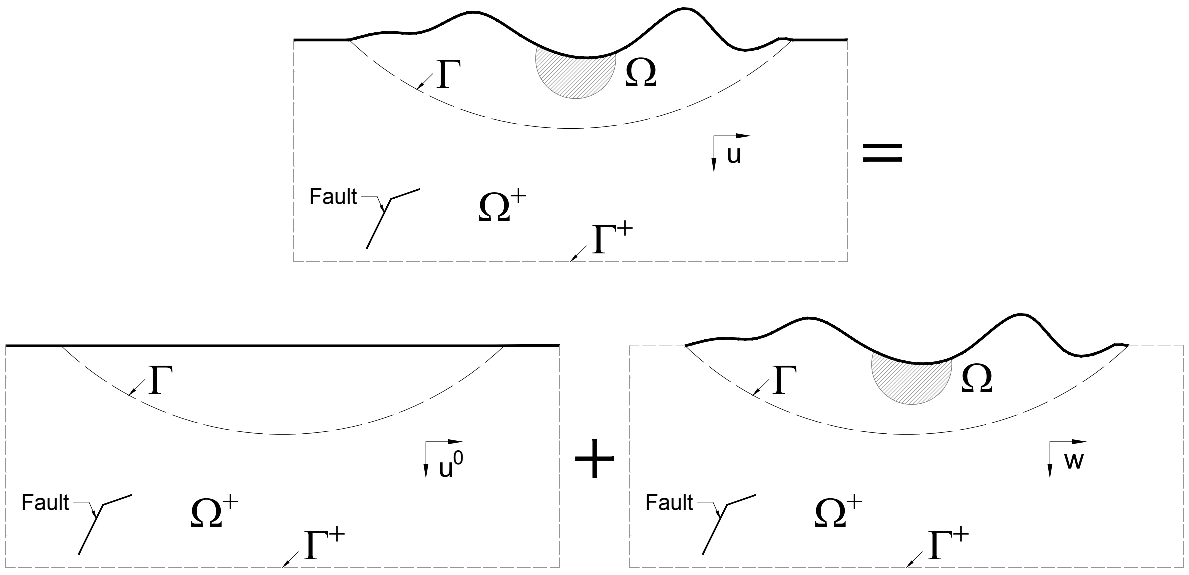


Figure 1: Partition of the field in the classical definition of the scattering problem. The free-field u^0 corresponds to the response of the half-space with the scatterer being removed (bottom left), while the scattered field w would be the additional displacement introduced in the half-space once the scatterer Ω is considered (bottom right).

where u^0 is the free field motion or response of the half-space in the absence of the scatterer and w is a scattered field or relative displacement motion between the total and free-field motion. This free field motion can be obtained in closed-form depending on the nature of the half-space and of the seismic excitation. In the more general case it is also found numerically from the solution of the simpler problem described in the bottom left of fig. 1.

Following the elastodynamics representation theorem the boundary value problem for the total field inside the scatterer is governed by the integral equations;

$$C_{ij}(\vec{\xi})u_j(\vec{\xi}) = \int_{\Gamma} G_{ij}^{FS}(\vec{x};\vec{\xi})t_j(\vec{x})dS(\vec{x}) - \int_{\Gamma^{+}+\Gamma} H_{ij}^{FS}(\vec{x};\vec{\xi})u_j(\vec{x})dS(\vec{x}) \quad \text{for } \vec{\xi} \text{ inside } \Gamma^{+}\cup\Gamma \tag{2}$$

and where G_{ij}^{FS} and H_{ij}^{FS} are the full-space displacements and tractions Green's tensors respectively while C_{ij} is a tensor which depends on the smoothness of the boundary and u_i and t_i are the total displacements and tractions vectors. Similarly, the relative motion in the half-space w_i is governed by;

$$C_{ij}(\vec{\xi})w_j(\vec{\xi}) = \int_{\Gamma} G_{ij}^{HS}(\vec{x};\vec{\xi})t_j^{w}(\vec{x})dS(\vec{x}) - \int_{\Gamma} H_{ij}^{HS}(\vec{x};\vec{\xi})w_j(\vec{x})dS(\vec{x}) \quad \text{for } \vec{\xi} \text{ inside } S_F\cup\Gamma^{+}\cup\Gamma \tag{3}$$

where now G_{ij}^{HS} and H_{ij}^{HS} are the corresponding Green's tensors for the half-space. Notice that the excitation enters into the problem once the surface compatibility conditions;

$$\begin{aligned} u &= u^0 + w \\ t &+ t^0 + t_w = 0 \end{aligned} \tag{4}$$

are imposed along Γ.

Bielak et al(2003) DRM algorithm

The problem is schematized by the top part of fig. 2 which depicts a generalized half-space with domain Ω^{+} supporting a scatterer Ω. Both domains are bounded through the perfectly coupled surface Γ. Notice that the scatterer comprises also a localized, small-scale topographic feature (e.g., a microzone) embedded into a large topographic irregularity. The relevant degrees of freedom have been labeled after **?**. In this work we are interested in conducting SRA at the microzone.

The partitioned equations of motion for the half-space and scatterer (i.e., Ω) read;

$$\begin{bmatrix} M_{ii}^{\Omega} & M_{ib}^{\Omega} \\ M_{bi}^{\Omega} & M_{bb}^{\Omega} \end{bmatrix} \begin{Bmatrix} \ddot{u}_i \\ \ddot{u}_b \end{Bmatrix} + \begin{bmatrix} K_{ii}^{\Omega} & K_{ib}^{\Omega} \\ K_{bi}^{\Omega} & K_{bb}^{\Omega} \end{bmatrix} \begin{Bmatrix} u_i \\ u_b \end{Bmatrix} = \begin{Bmatrix} 0 \\ P_b \end{Bmatrix} \tag{5}$$

$$\begin{bmatrix} M_{bb}^{\Omega^{+}} & M_{be}^{\Omega^{+}} \\ M_{eb}^{\Omega^{+}} & M_{ee}^{\Omega^{+}} \end{bmatrix} \begin{Bmatrix} \ddot{u}_b \\ \ddot{u}_e \end{Bmatrix} + \begin{bmatrix} K_{bb}^{\Omega^{+}} & K_{be}^{\Omega^{+}} \\ K_{eb}^{\Omega^{+}} & K_{ee}^{\Omega^{+}} \end{bmatrix} \begin{Bmatrix} u_b \\ u_e \end{Bmatrix} = \begin{Bmatrix} -P_b \\ P_e \end{Bmatrix} \tag{6}$$

where P_b are nodal forces through the coupling surface Γ, P_e represent the loads induced by a seismic source or by an incident plane wave and M and K are finite element mass and stiffness matrices. Coupling eq. (5) and eq. (6) yields the complete system of equations governing the half-space-scatterer system subjected to an exterior seismic source P_e and solved in one step algorithms (i.e., without DRM);

$$\begin{bmatrix} M_{ii}^{\Omega} & M_{ib}^{\Omega} & 0 \\ M_{bi}^{\Omega} & M_{bb}^{\Omega} + M_{bb}^{\Omega^+} & M_{be}^{\Omega^+} \\ 0 & M_{eb}^{\Omega^+} & M_{ee}^{\Omega^+} \end{bmatrix} \begin{Bmatrix} \ddot{u}_i \\ \ddot{u}_b \\ \ddot{u}_e \end{Bmatrix} + \begin{bmatrix} K_{ii}^{\Omega} & K_{ib}^{\Omega} & 0 \\ K_{bi}^{\Omega} & K_{bb}^{\Omega} + K_{bb}^{\Omega^+} & K_{be}^{\Omega^+} \\ 0 & K_{eb}^{\Omega^+} & K_{ee}^{\Omega^+} \end{bmatrix} \begin{Bmatrix} u_i \\ u_b \\ u_e \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ P_e \end{Bmatrix} \tag{7}$$

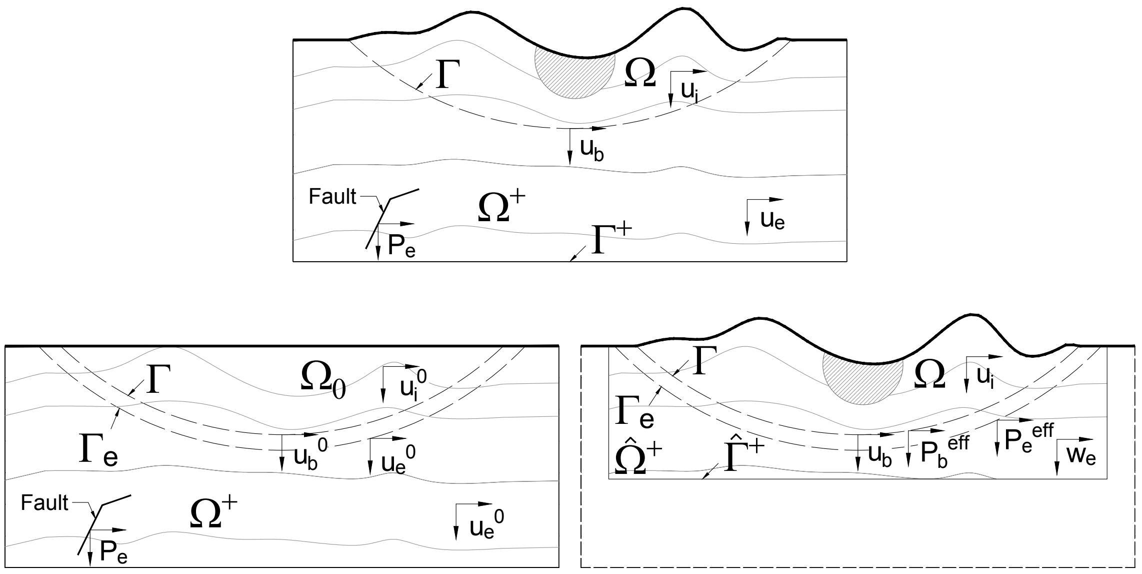


Figure 2: Generalized partition of the total domain and its related fields in the original DRM algorithm. The first analysis step is conducted over the half-space with domain $\Omega^+ \cup \Omega_0$ (bottom left). The resulting motion is then applied as input excitation to the reduced domain $\hat{\Omega}^+ \cup \Omega$ comprising only the localized feature (bottom right).

In the original DRM method the effect of the seismic sources P_e is transferred to the coupling surface Γ using an auxiliary problem (or background structure) constructed after removing from the complete domain the scatterer Ω and replacing it by an arbitrary simplified domain Ω_0 resulting in a generalized half-space $\Omega^+ \cup \Omega_0$ (as shown in the left part of fig. 2). This arbitrary domain is selected in such a way that it is easier to discretize than the original problem. The equations of motion for the domain Ω^+ in the auxiliary problem under the action of the seismic sources P_e yields a generalized free field motion u^0 governed by;

$$\begin{bmatrix} M_{bb}^{\Omega^+} & M_{be}^{\Omega^+} \\ M_{eb}^{\Omega^+} & M_{ee}^{\Omega^+} \end{bmatrix} \begin{Bmatrix} \ddot{u}_b^0 \\ \ddot{u}_e^0 \end{Bmatrix} + \begin{bmatrix} K_{bb}^{\Omega^+} & K_{be}^{\Omega^+} \\ K_{eb}^{\Omega^+} & K_{ee}^{\Omega^+} \end{bmatrix} \begin{Bmatrix} u_b^0 \\ u_e^0 \end{Bmatrix} = \begin{Bmatrix} -P_b^0 \\ P_e \end{Bmatrix} \tag{8}$$

allowing to express the seismic sources like;

$$P_e = M_{eb}^{\Omega^+} \ddot{u}_b^0 + M_{ee}^{\Omega^+} \ddot{u}_e^0 + K_{eb}^{\Omega^+} u_b^0 + K_{ee}^{\Omega^+} u_e^0. \tag{9}$$

The presence of the terms $M_{ee}^{\Omega^+} \ddot{u}_e^0$ and $K_{ee}^{\Omega^+} u_e^0$ in eq. (9) imply that the free field u^0 must be stored throughout the full domain Ω^+. This inconvenient requirement is dealt with after writing the total field in the exterior part of Ω^+ like a superposition of the free field motion u_e^0 and the relative (or scattered) motion w_e as;

$$u_e = u_e^0 + w_e \tag{10}$$

which yields;

$$\begin{aligned}
&\begin{bmatrix} M_{ii}^{\Omega} & M_{ib}^{\Omega} & 0 \\ M_{bi}^{\Omega} & M_{bb}^{\Omega} + M_{bb}^{\Omega+} & M_{be}^{\Omega+} \\ 0 & M_{eb}^{\Omega+} & M_{ee}^{\Omega+} \end{bmatrix} \begin{Bmatrix} \ddot{u}_i \\ \ddot{u}_b \\ \ddot{w}_e \end{Bmatrix} + \begin{bmatrix} K_{ii}^{\Omega} & K_{ib}^{\Omega} & 0 \\ K_{bi}^{\Omega} & K_{bb}^{\Omega} + K_{bb}^{\Omega+} & K_{be}^{\Omega+} \\ 0 & K_{eb}^{\Omega+} & K_{ee}^{\Omega+} \end{bmatrix} \begin{Bmatrix} u_i \\ u_b \\ w_e \end{Bmatrix} \\
&= \begin{Bmatrix} 0 \\ -M_{be}^{\Omega+} \ddot{u}_e^0 - K_{be}^{\Omega+} u_e^0 \\ P_e - M_{ee}^{\Omega+} \ddot{u}_e^0 - K_{ee}^{\Omega+} u_e^0 \end{Bmatrix}
\end{aligned} \tag{11}$$

After substituting for P_e from eq. (9) into eq. (11) the equations of motion are written in terms of degrees of freedom over a single layer of finite elements in Ω^+ adjacent to Γ. This strip of 1-element width lies between Γ and its adjacent surface Γ_e (see fig. 2);

$$\begin{aligned}
&\begin{bmatrix} M_{ii}^{\Omega} & M_{ib}^{\Omega} & 0 \\ M_{bi}^{\Omega} & M_{bb}^{\Omega} + M_{bb}^{\Omega+} & M_{be}^{\Omega+} \\ 0 & M_{eb}^{\Omega+} & M_{ee}^{\Omega+} \end{bmatrix} \begin{Bmatrix} \ddot{u}_i \\ \ddot{u}_b \\ \ddot{w}_e \end{Bmatrix} + \begin{bmatrix} K_{ii}^{\Omega} & K_{ib}^{\Omega} & 0 \\ K_{bi}^{\Omega} & K_{bb}^{\Omega} + K_{bb}^{\Omega+} & K_{be}^{\Omega+} \\ 0 & K_{eb}^{\Omega+} & K_{ee}^{\Omega+} \end{bmatrix} \begin{Bmatrix} u_i \\ u_b \\ w_e \end{Bmatrix} \\
&= \begin{Bmatrix} 0 \\ -M_{be}^{\Omega+} \ddot{u}_e^0 - K_{be}^{\Omega+} u_e^0 \\ M_{eb}^{\Omega+} \ddot{u}_b^0 + K_{eb}^{\Omega+} u_b^0 \end{Bmatrix}
\end{aligned} \tag{12}$$

The actual domain reduction leading to the method in the formulation from **?** is possible after noticing that all the waves in the exterior region Ω^+ are outgoing. Since the primary interest is in the determination of the response for the local site (i.e., a microzone) this fact suggests that the size of Ω^+ can be drastically reduced. This exterior reduced domain in the DRM algorithm is termed $\hat{\Omega}^+$.

The following points regarding the DRM formulation must be highlighted. First the equations of motion given in eq. (7) when written in terms of the free field motion reads:

$$\begin{aligned}
&\begin{bmatrix} M_{ii}^{\Omega} & M_{ib}^{\Omega} & 0 \\ M_{bi}^{\Omega} & M_{bb}^{\Omega} + M_{bb}^{\Omega+} & M_{be}^{\Omega+} \\ 0 & M_{eb}^{\Omega+} & M_{ee}^{\Omega+} \end{bmatrix} \begin{Bmatrix} \ddot{u}_i \\ \ddot{u}_b \\ \ddot{u}_e \end{Bmatrix} + \begin{bmatrix} K_{ii}^{\Omega} & K_{ib}^{\Omega} & 0 \\ K_{bi}^{\Omega} & K_{bb}^{\Omega} + K_{bb}^{\Omega+} & K_{be}^{\Omega+} \\ 0 & K_{eb}^{\Omega+} & K_{ee}^{\Omega+} \end{bmatrix} \begin{Bmatrix} u_i \\ u_b \\ u_e \end{Bmatrix} \\
&= \begin{Bmatrix} 0 \\ 0 \\ M_{eb}^{\Omega+} \ddot{u}_b^0 + M_{ee}^{\Omega+} \ddot{u}_e^0 + K_{eb}^{\Omega+} u_b^0 + K_{ee}^{\Omega+} u_e^0 \end{Bmatrix}.
\end{aligned} \tag{13}$$

However complete transfer of the seismic sources P_e to the coupling surface Γ is only achieved after one eliminates from eq. (13) the terms $M_{ee}^{\Omega+} \ddot{u}_e^0$ and $K_{ee}^{\Omega+} u_e^0$ which is accomplished by introducing the change of variables resulting after writing the total field in the exterior part of Ω^+ in terms of the free-field and relative motion. For this change of variables to remain valid it is required that the reduced exterior domain $\hat{\Omega}^+$ retains the same material properties as the original exterior domain Ω^+. The resulting DRM approach can be summarized in the following two-step algorithm:

- In step-I the complete seismic domain $\Omega \cup \Omega^+$, comprising the seismic source and microzones of soft material properties is replaced by a simpler domain $\Omega_0 \cup \Omega^+$ which results after removing all the surface topography and localized features.

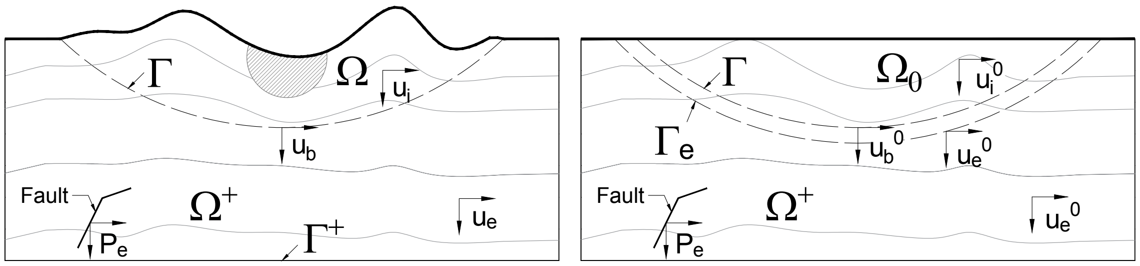


Figure 3: In step I of the DRM algorithm the original domain $\Omega \cup \Omega^+$ (left) is replaced by a simpler domain $\Omega_0 \cup \Omega^+$ (right).

This simpler domain is analyzed, under the action of seismic sources P_e, in order to determine the free field response leading to effective loads located over a boundary adjacent to the local site given by;

$$P^{eff} = \left\{ \begin{array}{c} 0 \\ -M_{be}^{\Omega^+} \ddot{u}_e^0 - K_{be}^{\Omega^+} u_e^0 \\ M_{eb}^{\Omega^+} \ddot{u}_e^0 + K_{ee}^{\Omega^+} u_e^0 \end{array} \right\}. \tag{14}$$

- In step-II of the algorithm, ground response analysis is performed at desired microzones using as excitation the free-field motion extracted from the data base created during step-I. In the case of a plane wave analysis the calculation of the incoming motion through the solution of the FE-system specified in step-I is unnecessary as the field can be obtained analytically. This computation is performed automatically by **WAVES** in the elemental subroutines used to define the elements along the strip (see fig. 4).

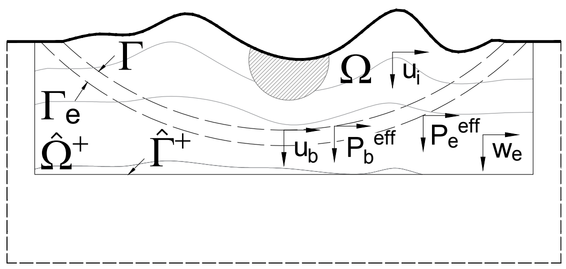


Figure 4: Reduced domain subjected to the action of effective forces P^{eff} and equivalent to the seismic sources.

The local model can take into account a highly detailed description of the small-scale topography. This local step can be performed using computing resources typically available at a consulting office and using an independent numerical technique.

A key aspect in the DRM technique is the fact that in the reduced domain $\hat{\Omega}^+ \cup \Omega$, the domain $\hat{\Omega}^+$ is of the same material properties as in the original Ω^+ (right part of fig. 2). To reflect this fact the mass and stiffness coefficients pertaining to the domain $\hat{\Omega}^+$ have retained the superscript Ω^+.

3 Explicit solution scheme

Consider the discrete dynamic equilibrium equations at time t

$$M^tA + C^tV + K^tU =^t F \tag{15}$$

and where M, C, K are the assembled mass, damping and stiffness matrices respectively while ${}^tA$, ${}^tV$, ${}^tU$ and ${}^tF$ are the nodal accelerations, velocities, displacements and external loads vectors at time t. In **WAVES** the external forces may be the result of particular point loads or effective forces consistent with a plan wave analysis.

In terms of nodal forces, eq. (15) can be written like;

$${}^tF^I +^t F^D +^t F^s =^t F \tag{16}$$

where ${}^tF^I$, ${}^tF^D$ and ${}^tF^s$ are inertial, damping, elastic and external nodal forces respectively.

Expanding the acceleration and velocity terms at time t in a consistent finite central differences scheme we have;

$$\begin{aligned} {}^tA &= \frac{1}{\Delta t^2}\left({}^{t-\Delta t}U - 2^tU +^{t+\Delta t} U\right) \\ {}^tV &= \frac{1}{2\Delta t}\left(-^{t-\Delta t}U +^{t+\Delta t} U\right). \end{aligned} \tag{17}$$

Consider now the trial states

$$\begin{aligned} {}^t\hat{A} &= \frac{1}{\Delta t^2}\left({}^{t-\Delta t}U - 2^tU\right) \\ {}^t\hat{V} &= -\frac{1}{2\Delta t}{}^{t-\Delta t}U \end{aligned} \tag{18}$$

which allows us to write eq. (17) like;

$$\begin{aligned} {}^tA &= {}^t\hat{A} + \frac{1}{\Delta t^2}{}^{t+\Delta t}U \\ {}^tV &= {}^t\hat{V} + \frac{1}{2\Delta t}{}^{t+\Delta t}U \end{aligned} \tag{19}$$

Notice that the terms ${}^t\hat{A}$ and ${}^t\hat{V}$ result after assuming that ${}^{t+\Delta t}U = 0$ in eq. (17) thus they are commonly referred to as predictors. Notice also that after the displacements at $t + \Delta t$ have been found the corrected values of ${}^tA$ and ${}^tV$ can be obtained via eq. (19). In this sense the terms $\frac{1}{\Delta t^2}{}^{t+\Delta t}U$ and $\frac{1}{2\Delta t}{}^{t+\Delta t}U$ play the role of correctors to the initial predictors. The resulting algorithm is commonly referred to, by obvious reasons, as a predictor-corrector scheme.

Using eq. (19) in eq. (15) results in the following equation governing the displacements at time $t + \Delta t$;

$$\left(\frac{1}{\Delta t^2}M + \frac{1}{2\Delta t}C\right)^{t+\Delta t}U =^t F - M^t\hat{A} - C^t\hat{V} - K^tU \tag{20}$$

The dynamic finite element equilibrium equations given in eq. (20) can be written in the standard static form:

$$\hat{K}^{t+\Delta t}U = {}^t\hat{F}$$

after letting;

$$\hat{K} = \frac{1}{\Delta t^2}M + \frac{1}{2\Delta t}C$$

and

$${}^t\hat{F} = {}^tF - M^t\hat{A} - C^t\hat{V} - K^tU$$

.

It is convenient, and physically appealing, to write eq. (20) in terms of forces like:

$${}^{t+\Delta t}F^I +^{t+\Delta t} F^D =^t F -^t \hat{F}^I -^t \hat{F}^D -^t F^s \tag{21}$$

- Equation (21) is an equilibrium equation at time $t = t$ allowing to predict the displacements at time $t = t + \Delta t$ in terms of previously known values at times t and $t = t - \Delta t$ as schematically shown in fig. 5

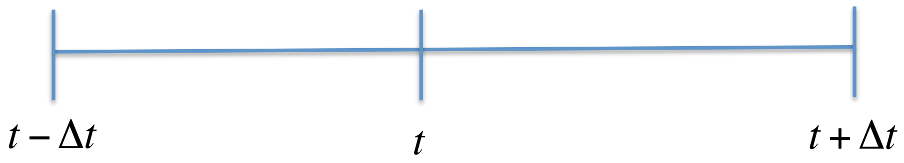


Figure 5: Definition of the general iteration giving displacements at $t = t+\Delta t$ in terms of previously known values at times t and $t = t - \Delta t$.

- The equation is exact within the error introduced by the expansion used in eq. (17).
- The first predicted solution is at $t = \Delta t$ which implies that we require data at $t = -\Delta t$ and at $t = 0$ (i.e., initial conditions) as schematically shown in fig. 6.

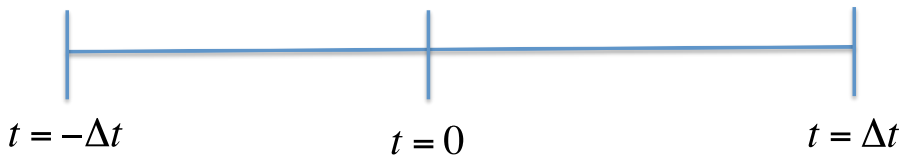


Figure 6: Initial iteration predicting values at time $t = \Delta t$ in terms of the artifitial values at $t = -\Delta t$ and the initial conditions at $t = 0$.

Damping Assumptions

The damping matrix C appearing in eq. (20) is actually not assembled in the standard finite element sense but it is rather built from the mass and stiffness matrices. Moreover we can consider the following possibilities regarding damping.

(i) Neglect damping (This is however inconvenient for finite domains) which gives:

$$^{t+\Delta t}F^I +^{t+\Delta t} F^D =^t F -^t \hat{F}^I -^t F^s \tag{22}$$

(ii) Use Rayleigh Damping and retain the velocity expansion used in (17). That is;

$$C = \alpha M + \beta K \tag{23}$$

then we have (in terms of forces);

$$(1+\beta\Delta t^2)^{t+\Delta t}F^I + \frac{\alpha}{2\Delta t}{}^{t+\Delta t}F^S =^t \hat{F} \tag{24}$$

where;

$$^t\hat{F} =^t F -^t \hat{F}^I -^t \hat{F}^D -^t F^s$$

Solution in equation eq. (24) requires the full assembly and factorization of an effective stiffness matrix.

(iii) Use Rayleigh damping but modify the velocity expansion introduced in eq. (17). Using

$$^tV = \frac{1}{\Delta t}(^tU -^{t-\Delta t} U) \tag{25}$$

yielding;

$$^{t+\Delta t}F^I =^t F -^t \hat{F}^I -^t \hat{F}^D -^t F^s \tag{26}$$

where now the velocity predictor and corrector terms are defined like:

$$^t\hat{V} = \frac{1}{\Delta t}\left(^tU - {}^{t-\Delta t}U\right)$$

and

$$^tV = \frac{1}{2\Delta t}\left(-{}^{t-\Delta t}U + {}^{t+\Delta t}U\right)$$

respectively, giving the equation:

$$^{t+\Delta t}F^I =^t F -^t \hat{F}^I -^t F^s -^t \hat{F}^D \tag{27}$$

4 Algorithm implemented in WAVES (damping assumption 3)

Let us write eq. (27) like

$$^{j+1}F^I =^j F -^t \hat{F}^I -^j F^s -^j \hat{F}^D. \tag{28}$$

The first prediction corresponding to $t = \Delta t$ is given by;

$$^{\Delta t}F^I = {^0F} - {^0\hat{F}^I} - {^0\hat{F}^S} - {^0\hat{F}^D}$$

which results after applying eq. (27) at $t = 0$ and where the term

$$^0\hat{F}^D = C\frac{1}{\Delta t}\left({^0U} - {^{-\Delta t}U}\right)$$

requires knowing the fictitious value $-^{\Delta t}U$. This last term can be computed after applying the central difference expansion at $t = 0$ and solving for $-^{\Delta t}U$ giving;

$$^{-\Delta t}U =^0 U - \Delta t^0 V + \frac{\Delta t^2}{2}{^0A} \tag{29}$$

Finally using eq. (29) in eq. (28) allows us to start up the algorithm. The general and initial iteration are schematized in fig. 7 and fig. 8 respectively.

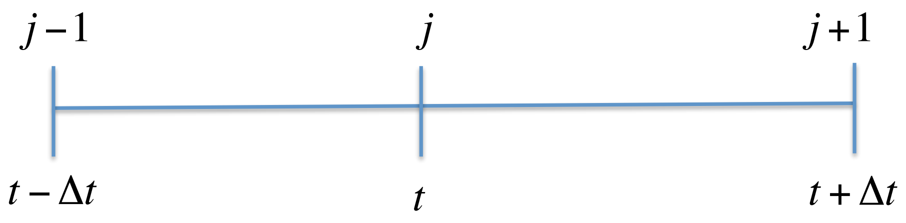


Figure 7: Definition of the general iteration

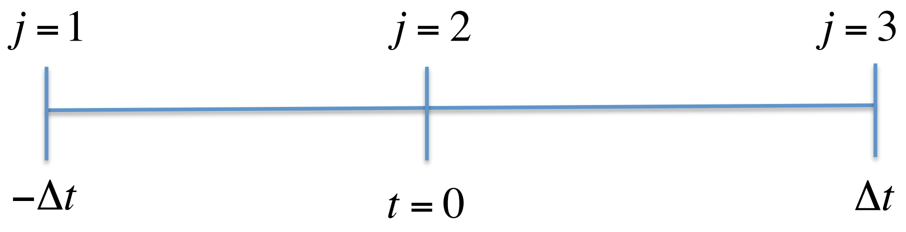


Figure 8: Definition of the initial iteration

4.1 Decoupling

The equilibrium equations discussed so far can be decoupled whenever the coefficient matrix is diagonal. As a result, the algorithm can proceed one degree of freedom at a time without the need for a complete assembly process and with large memory savings. Consider the particular case of damping assumption 3 and a lumped mass matrix. Writing eq. (28) for the i-th degree of freedom (i.e., i is kept fixed) results in:

$$\frac{1}{\Delta t^2} M_{ij}^{t+\Delta t} U_j =^t F_i - K_{ij} {}^t U_j - M_{ij} {}^t \hat{A}_j - C_{ij} {}^t \hat{V}_j \tag{30}$$

Now, if the lumped mass matrix is written like;

$$M_{ij} = m_I \delta_{ij}$$

we have:

$$\frac{1}{\Delta t^2} m_I^{t+\Delta t} U_i =^t F_i - K_{ij} {}^t U_j - m_I {}^t \hat{A}_i - C_{ij} {}^t \hat{V}_j \tag{31}$$

producing the following recursive equation;

$${}^{t+\Delta t} F_i^I =^t F_i -^t \hat{F}_i^S - {}^t \hat{F}_i^I - {}^t \hat{F}_i^D. \tag{32}$$

To initialize the algorithm we apply the FDs equations at $t = 0$ leading to:

$$\frac{1}{\Delta t^2} m_I^{\Delta t} U_i =^0 F_i - K_{ij} {}^0 U_j - m_I {}^0 \hat{A}_i - C_{ij} {}^0 \hat{V}_i.$$

In the above we require:

$${}^0 \hat{V} = \frac{1}{\Delta t} \left({}^0 U_i - {}^{-\Delta t} U_i \right)$$

which at the same time implies prior knowledge of the artificial term ${}^{-\Delta t} U_i$. This can be obtained from (29) as follows:

$${}^{-\Delta t} U_i =^0 U_i - \Delta t^0 V_i + \frac{\Delta t^2}{2} {}^0 A_i. \tag{33}$$

The initial acceleration is obtained after assuming homogeneous ICs;

$$m_I {}^0 A_i + C_{ij} {}^0 V_j + K_{ij} {}^0 U_j =^0 F_i$$

therefore

$$m_I^0 A_i = \frac{{}^0 F_i}{m_I}$$

and

$${}^{-\Delta t} U_i = \frac{\Delta t^2}{2 m_I} {}^0 F_i.$$

Moreover, neglecting the damping effects on the prediction of ${}^{\Delta t} U_i$ yields;

$${}^{\Delta t} U_i = \frac{\Delta t^2}{2 m_I} {}^0 F_i$$

Algorithm 1: Full Algorithm

```
Data: Time span, Geometry, Material Parameters
Result: Displacements, Velocity and Acceleration time histories
Initialize solution vectors ($j = 1$ corresponds to ICs: 0 superscript indicates $t = 0$);
$^{0}U_i \longleftrightarrow ^{j=1}U_i = 0,\ ^{0}V_i = 0,\ ^{j=1}A_i = \frac{^{1}R_i}{m_I}$ ;
Select $\Delta t$ and damping coefficients $\alpha$, $\beta$;
Fix 1-st predicted value (let $j = 2$);
    $^{\Delta t}U_i \longleftarrow \frac{\Delta t^2}{2m_I} {}^{0}F_i \longleftrightarrow \left[ ^{j=2}U_i \longleftarrow \frac{\Delta t^2}{2m_I} {}^{j=1}F_i \right]$
Time Integration Phase;
while $j \leq N$ do
    Predictors
        $^{j}\hat{A}_i \longleftarrow \frac{1}{\Delta t^2} \left( ^{j-1}U_i - 2 {}^{j}U_i \right)$
        $^{j}\hat{V}_i \longleftarrow \frac{1}{\Delta t} \left( ^{j}U_i - {}^{j-1}U_i \right)$
        $^{j+1}F_i^I \longleftarrow {}^{j}F_i - K_{ij} {}^{j}U_j - m_I {}^{j}\hat{A}_j - C_{ij} {}^{j-1}\hat{V}_j$
    Solve
        $^{j+1}U_i \longleftarrow \frac{\Delta t^2}{m_I} {}^{j+1}F_i^I$
    Correctors
        $^{j}A_i \longleftarrow {}^{j}\hat{A}_i + \frac{1}{\Delta t^2} {}^{j+1}U_i$
        $^{j}V_i \longleftarrow {}^{j}\hat{V}_i + \frac{1}{2\Delta t} {}^{j+1}U_i$
        $j \longleftarrow j + 1$
end
```

5 Program structure

In the explicit integration scheme the solution corresponding to the i-th degree of freedom at time $t + \Delta t$ is found from eq. (34):

$$^{j+1}U_i \longleftarrow \left(\frac{\Delta t^2}{m_I} \right) {}^{j+1}F^I{}_i \tag{34}$$

where m_I is the total mass associated to the i-th degree of freedom and $^{j+1}F_i^I$ is the total force considering the contribution from the external, elastic, inertial and damping terms at times t and $t - \Delta t$. This force is given by eq. (35):

$$^{j+1}F_i^I \longleftarrow {}^{j}F_i - K_{ij} {}^{j}U_j - m_I {}^{j}\hat{A}_i - C_{ij} {}^{j}\hat{V}_j. \tag{35}$$

Accordingly, in the uncoupled explicit finite element formulation the equation solving process proceeds one degree of freedom at a time. This implies a different assembly process to the one used in an implicit algorithm where a formal coefficient matrix is assembled and inverted. Now the mass, damping and stiffness elemental matrices are used to obtain effective nodal forces at each degree of freedom. In summary the mesh is not covered in an element by element basis, but in a node by node basis. In the following algorithm we discuss this nodal assembly process where in order to solve the displacement at a given degree of freedom prior knowledge of the elements contributing to the given node is necessary.

In particular in **WAVES** the total mass m_I and the total force ${}^{j+1}F_i^I$ is built after considering the contribution from the different elements connected to the node. This nodal assembly process is schematized in fig. 9 in which 4 bi-lineal elements are connected by a central node.

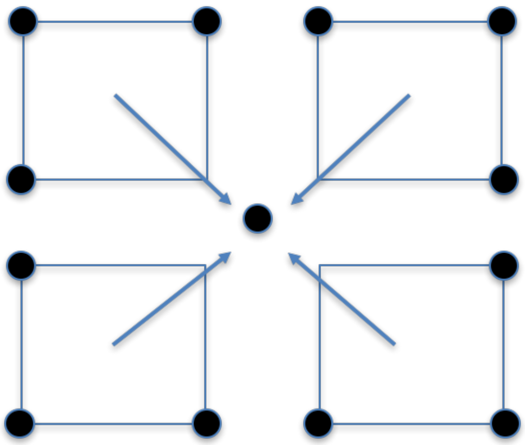


Figure 9: Nodal assembly

The nodal assembly is described in the following algorithm:

Algorithm 2: Nodal assembly

```
Data: Number of elements connected to the node
Result: Total mass and force contribution
for $i \longleftarrow 1$ to $Numnp$ do
    $k \longleftarrow NIEL_i$
    for $j \longleftarrow 1$ to $k$ do
        Retrieve element parameters
        Compute predictors
        Compute element contribution (UEL.for)
        Assembly element contribution into the force vector ${}^{j+1}F_i^I$
    end
    Solve for the current d.o.f
    Perform corrections.
end
```

The subroutine **UEL.for** in the algorithm computes the contribution from the current element to the degree of freedom being solved. The library of available elements in the code corresponds precisely to a set of **UELs** subroutines. Additional elements can be easily implemented and added to the code by just coding new **UEL** subroutines as will be specified later.

6 Elements

The following elements are currently available in **WAVES** :

```
UELEXP8: 8-noded 2D quad element.
UELEXP4: 4-noded 2D quad element.
UELEXP9: 9-noded 2D quad element.
UELEXP6: 6-noded 2D triangular element.
UEL8INCOT: 8-noded 2D quad element with plane-wave incoming effective DRM forces.
UEL9INCOT: 9-noded 2D quad element with plane-wave incoming effective DRM forces.
UELEDASH3: 3-noded 2D Lysmer and Kuhlemeyer absorbing boundary.
UELDASHINC: 3-noded 2D transmitting boundary.
UEL3DEXP8: 8-noded bilineal 3D tetrahedarl elment.
```

6.1 Adding elements

User element subroutines may be added to the code by implementing a **UEL** subroutine with the following interface.

```
      SUBROUTINE UELXXX(RHS,ANMASS,NDOFEL,PROPS,NPROPS,IPROPS,NIPROPS,
     1                   AWAVE,IWAVE,COORDS,MCRD,NNODE,U,DU,V,A,JELEM,
     2                   DT,KINC,NINCR)

      IMPLICIT REAL*8(A-H,O-Z)

      PARAMETER (ZERO=0.D0,HALF=0.5D0,ONE=1.D0,NTENS=4,TWO=2.D0,NGPTS=9,
     1           THREE=3.D0, NDI=3)

C     Parameter arrays from UEL.f

      DIMENSION RHS(NDOFEL),ANMASS(NDOFEL),PROPS(NPROPS),
     1          IPROPS(NIPROPS),AWAVE(5),COORDS(MCRD,NNODE),U(NDOFEL),
     2          DU(NDOFEL),V(NDOFEL),A(NDOFEL),AMATRX(NDOFEL,NDOFEL),
     3          AMASS(NDOFEL,NDOFEL),CDMAT(NDOFEL,NDOFEL),UP(NDOFEL),
     4          DDSDDE(NTENS,NTENS),B(NTENS,NDOFEL),BT(NDOFEL,NTENS),
     5          FRST2(NDOFEL,NDOFEL),FRST1(NDOFEL,NDOFEL),XP(2,NGPTS),
     6          XW(NGPTS),AUX1(NTENS,NDOFEL)
```

```
    RETURN
```

The subroutine must return the following parameters to the main program:

- RHS: Contribution from the current element to the i-th degree of freedom total external force as given by eq. (35).
- ANMASS: Contribution from the current element to the i-th degree of freedom total mass.

On the other hand the input parameters are defined as follows:

- NDOFEL: Total number of degrees of freedom of the elemnt.
- PROPS: Array containing the real material properties needed to compute the forces in the current element.
- NPROPS: Dimension of the PROPS array.
- IPROPS: Array containing the integer material properties needed to compute the forces in the current element. This arrays is mainly used in the definition of incoming elements in order to specify which face of the element is in direct contact with the scatterer.
- NIPROPS: Dimension of the IPROPS array.
- AWAVE: Plan wave parameters.
- IWAVE: Dimension of the AWAVE array.
- COORDS: Array with the nodal coordinates of the element.
- MCRD: Parameter defining the dimensionality of the problem.
- NNODE: Total number of nodes in the element.
- U: Nodal displacements at time t.
- DU: Displacement increment defined as $\Delta U = {}^{t}U - {}^{t-\Delta t}U$.
- V: Trial velocity defined as ${}^{t}\hat{V} = -\frac{1}{2\Delta t}{}^{t-\Delta t}U$.
- A: Trial acceleration defined as ${}^{t}\hat{A} = \frac{1}{\Delta t^2}\left({}^{t-\Delta t}U - 2{}^{t}U\right)$.
- JELEM: Current element identifier.
- DT: Size of the time step.
- KINC: Current increment number.
- NINCR: Total number of increments.

7 Creating a model

In **WAVES** a finite element model is defined through an input data file with extension **.inp** as described in the file named **input_file_template**. The file contains the following general data blocks.

```
PROBLEM DEFINITION BLOCK
NODAL DEFINITION BLOCK
MATERIAL DEFINITION BLOCK
WAVE DEFINITION BLOCK
ELEMENTS DEFINITION BLOCK
POINT LOADS DEFINITION BLOCK
NODAL OUTPUT DEFINITION BLOCK
```

The template file provided with the code and named **input_file_template** describes the composition of each one of the fields.

Creating a model with Gmesh

Although input data files for simple models can be created manually the process becomes cumbersome in large problems. In this section we describe how to define a model using the third party software **Gmesh**. On the other hand, although the code can be used to solve virtually any dynamic loading problem, in the following example we focus in a plane wave analysis typically found in earthquake engineering.

The creation of a **WAVES** model in **Gmesh** involves a three-step procedure:

(i) Define the model geometry through the ***.geo** file.

(ii) Define the finite element mesh through the ***.msh** file.

(iii) Define the problem parameters through an input ***.txt** file.

(iv) Define the ***.inp** file using the auxiliary program **mesher.for**.

Problem definition

To facilitate the model creation process the geometric template shown in fig. 10 contains an open rectangular scatterer defined by the key-points labeled 3, 4, 5 and 6. This template is useful in the definition of models with a single scatterer. The geometry of the single scatterer can later be added to the square box.

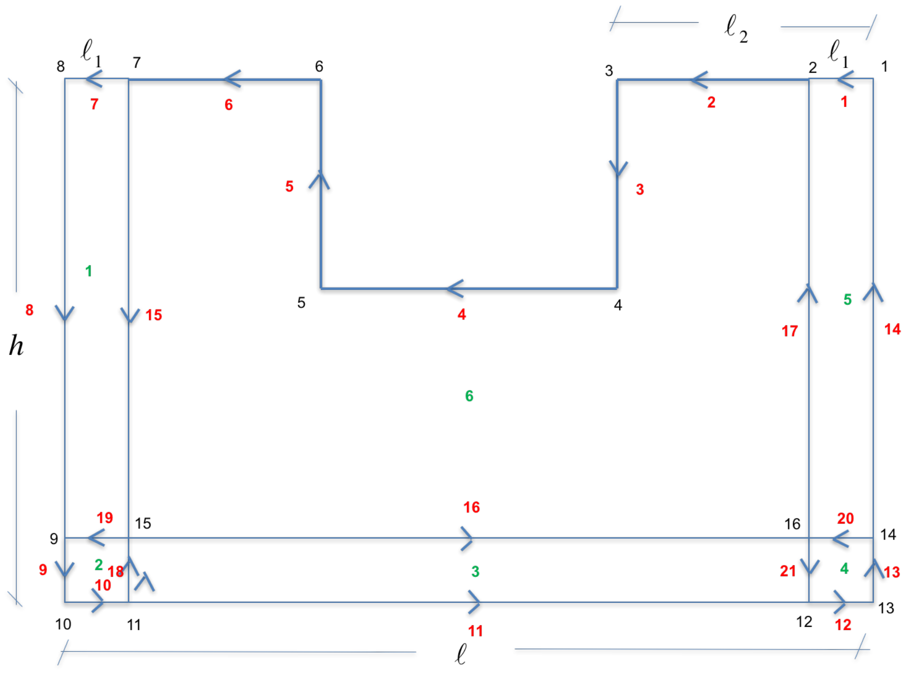


Figure 10: Geometry of the general template with a rectangular scatterer. Single, scatterers can be added to the model.

Model geometry

We first describe the geometric definition file **.geo** corresponding to the generalized rectangular scatterer. The model creation process is later completed by the introduction of additional data lines corresponding to the particular scatterer. A **gmesh** geometry is defined by fundamental entities prescribed in the following hierarchical order in a ***.geo** file :

- Parameters.
- Keypoints.
- Keylines.
- Surfaces.
- Physical surfaces.

The specific file for the generalized rectangular scatterer is shown in fig. 11. The first part of the file contains data to define the maximum element size associated to the nodes [1]. The template has been built for a rectangular scatterer defined in terms of parameters l and h corresponding to the total domain length and height respectively.

On the other hand, the dimension specified as *l1* corresponds to the width of the strip of elements where the incoming motion is to be prescribed. This line should not be modified since

[1]The size of the elements pertaining to the strip where the incoming motion is to be applied must remain unmodified

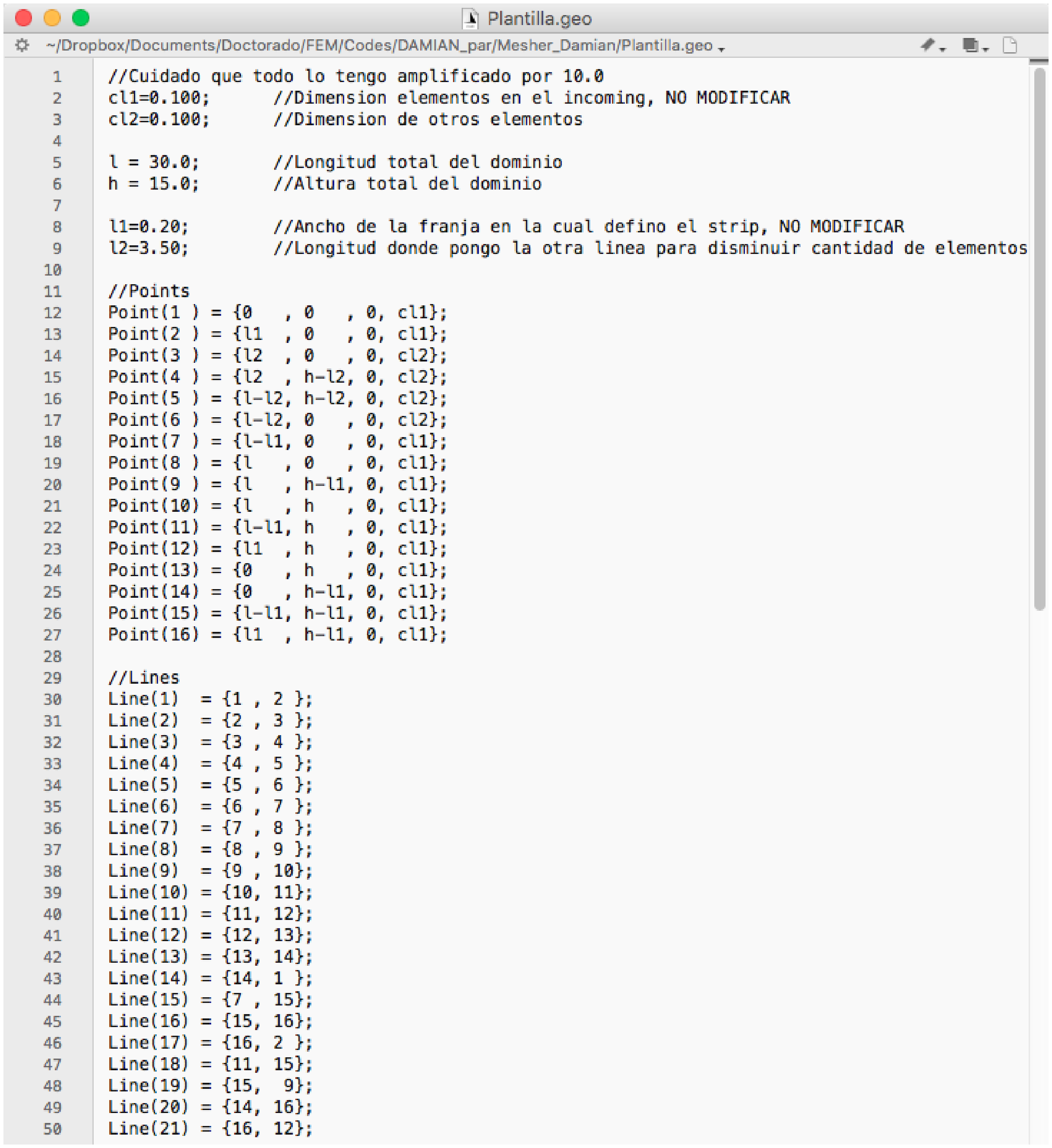
this value guarantees that the maximum element size for these strip elements is always equal to 2 and that the elements remain square shaped.

The point and line data appears next with the template containing nodes from 1 to 16 and lines from 1 to 21. Similarly, surfaces 1 through 5 are those corresponding to the strip for the incoming motion in a plane wave analysis, while surface 6 is a transition surface that is used in order to decrease the number of elements in the complete domain.

Physical surfaces for the template can now be defined as follows:

- **Physical Surface(10000):** Surfaces conforming the strip.
- **Physical Surface(11000):** Contains the transition surface 6 but it can contain additional surfaces. Each one of the physical surfaces must be associated to a material profile in the input file.
- **Additional physical surfaces:** The value of 1000 must be added to each additional physical surface.

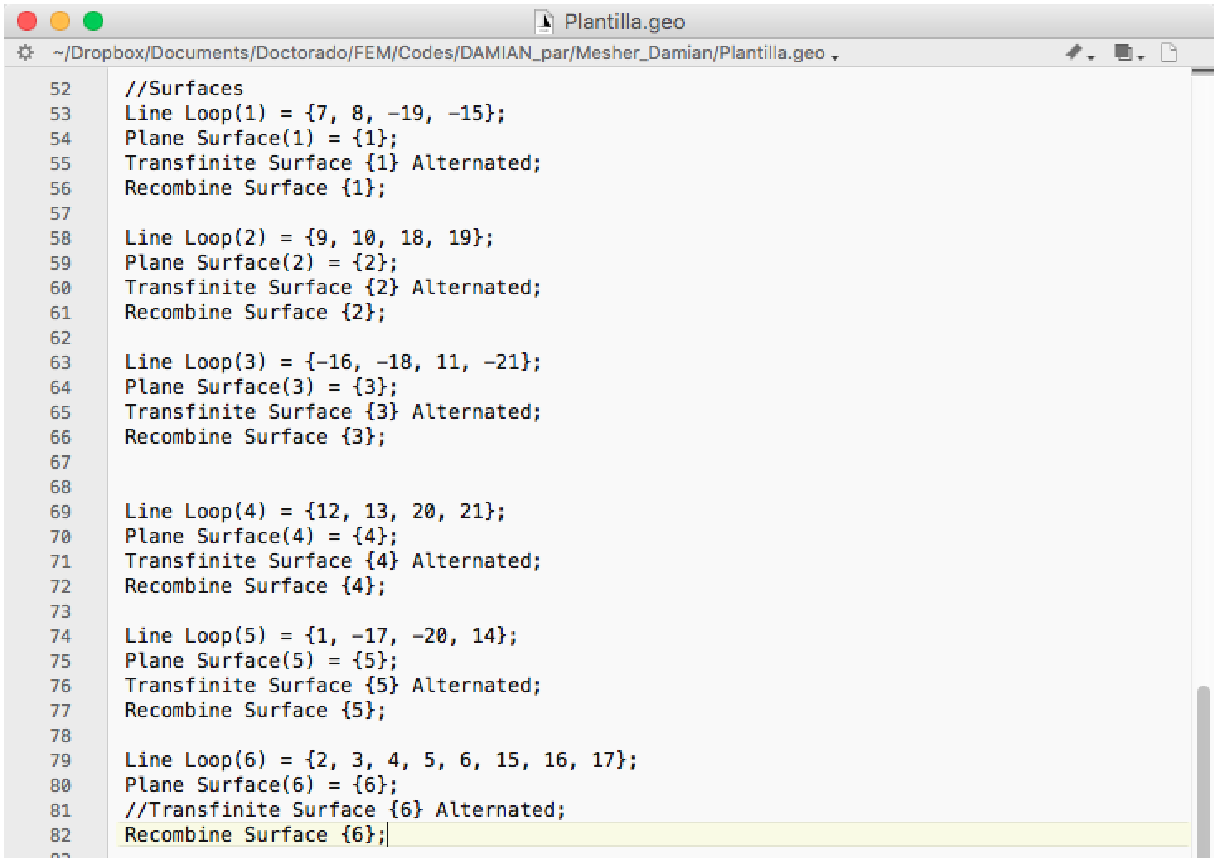
The physical lines defining the absorbing boundaries are defined next. These are also useful to identify the incoming elements and they all must have the label **Physical Line(1)**. Along the strip, only the external elements conform the incoming elements, the remaining ones are added since **Gmsh** is unable to add a single element over a strip. These lines must be defined from left to right.


Plantilla.geo

~/Dropbox/Documents/Doctorado/FEM/Codes/DAMIAN_par/Mesher_Damian/Plantilla.geo

```
//Cuidado que todo lo tengo amplificado por 10.0
cl1=0.100;      //Dimension elementos en el incoming, NO MODIFICAR
cl2=0.100;      //Dimension de otros elementos

l = 30.0;       //Longitud total del dominio
h = 15.0;       //Altura total del dominio

l1=0.20;        //Ancho de la franja en la cual defino el strip, NO MODIFICAR
l2=3.50;        //Longitud donde pongo la otra linea para disminuir cantidad de elementos

//Points
Point(1 ) = {0   , 0   , 0, cl1};
Point(2 ) = {l1  , 0   , 0, cl1};
Point(3 ) = {l2  , 0   , 0, cl2};
Point(4 ) = {l2  , h-l2, 0, cl2};
Point(5 ) = {l-l2, h-l2, 0, cl2};
Point(6 ) = {l-l2, 0   , 0, cl2};
Point(7 ) = {l-l1, 0   , 0, cl1};
Point(8 ) = {l   , 0   , 0, cl1};
Point(9 ) = {l   , h-l1, 0, cl1};
Point(10) = {l   , h   , 0, cl1};
Point(11) = {l-l1, h   , 0, cl1};
Point(12) = {l1  , h   , 0, cl1};
Point(13) = {0   , h   , 0, cl1};
Point(14) = {0   , h-l1, 0, cl1};
Point(15) = {l-l1, h-l1, 0, cl1};
Point(16) = {l1  , h-l1, 0, cl1};

//Lines
Line(1)  = {1 , 2 };
Line(2)  = {2 , 3 };
Line(3)  = {3 , 4 };
Line(4)  = {4 , 5 };
Line(5)  = {5 , 6 };
Line(6)  = {6 , 7 };
Line(7)  = {7 , 8 };
Line(8)  = {8 , 9 };
Line(9)  = {9 , 10};
Line(10) = {10, 11};
Line(11) = {11, 12};
Line(12) = {12, 13};
Line(13) = {13, 14};
Line(14) = {14, 1 };
Line(15) = {7 , 15};
Line(16) = {15, 16};
Line(17) = {16, 2 };
Line(18) = {11, 15};
Line(19) = {15,  9};
Line(20) = {14, 16};
Line(21) = {16, 12};
```

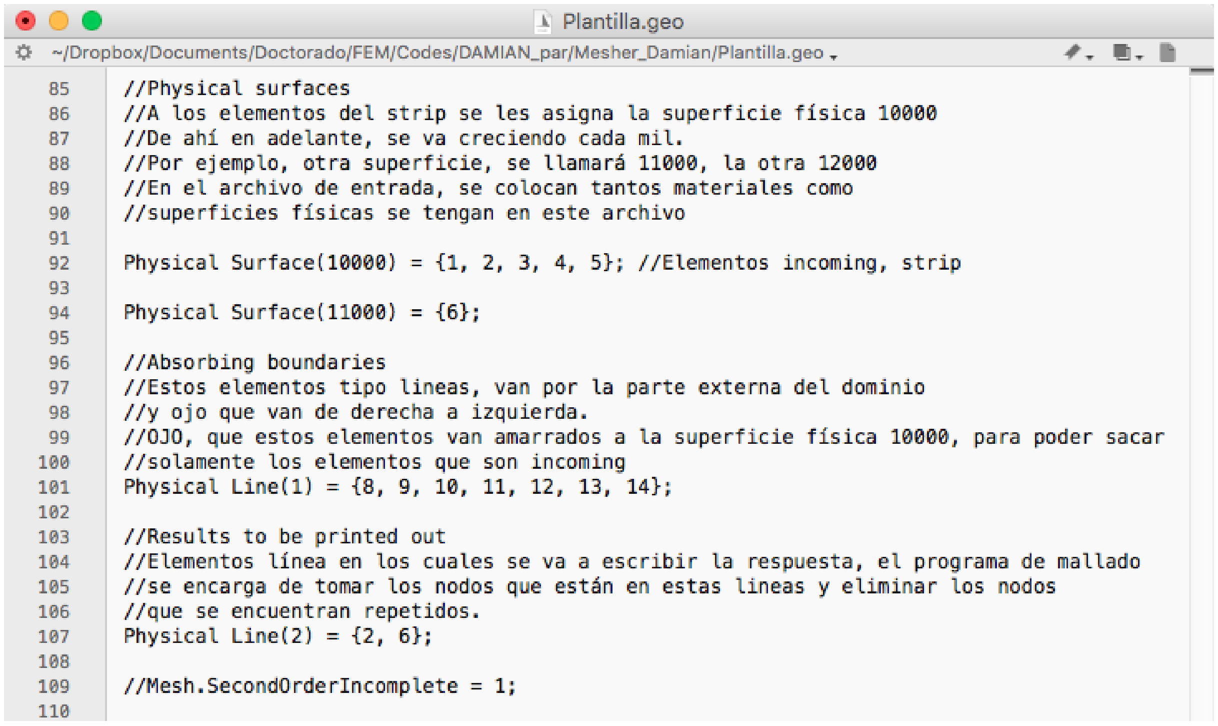
Figure 11: First part of the template file defining points, lines and initial model parameters for the rectangular scatterer.


Plantilla.geo
~/Dropbox/Documents/Doctorado/FEM/Codes/DAMIAN_par/Mesher_Damian/Plantilla.geo

```
//Surfaces
Line Loop(1) = {7, 8, -19, -15};
Plane Surface(1) = {1};
Transfinite Surface {1} Alternated;
Recombine Surface {1};

Line Loop(2) = {9, 10, 18, 19};
Plane Surface(2) = {2};
Transfinite Surface {2} Alternated;
Recombine Surface {2};

Line Loop(3) = {-16, -18, 11, -21};
Plane Surface(3) = {3};
Transfinite Surface {3} Alternated;
Recombine Surface {3};


Line Loop(4) = {12, 13, 20, 21};
Plane Surface(4) = {4};
Transfinite Surface {4} Alternated;
Recombine Surface {4};

Line Loop(5) = {1, -17, -20, 14};
Plane Surface(5) = {5};
Transfinite Surface {5} Alternated;
Recombine Surface {5};

Line Loop(6) = {2, 3, 4, 5, 6, 15, 16, 17};
Plane Surface(6) = {6};
//Transfinite Surface {6} Alternated;
Recombine Surface {6};
```

Figure 12: Second part of the template file defining surfaces for the rectangular scatterer.

The physical surfaces defined as a line can be used to specify the points used for response output. To identify these lines all the lines with the label **Physical Line(2)** must be included. This part of the template must be added at the end of the file since it is possible that the response is required along points still undefined.


Plantilla.geo
~/Dropbox/Documents/Doctorado/FEM/Codes/DAMIAN_par/Mesher_Damian/Plantilla.geo

```
//Physical surfaces
//A los elementos del strip se les asigna la superficie física 10000
//De ahí en adelante, se va creciendo cada mil.
//Por ejemplo, otra superficie, se llamará 11000, la otra 12000
//En el archivo de entrada, se colocan tantos materiales como
//superficies físicas se tengan en este archivo

Physical Surface(10000) = {1, 2, 3, 4, 5}; //Elementos incoming, strip

Physical Surface(11000) = {6};

//Absorbing boundaries
//Estos elementos tipo lineas, van por la parte externa del dominio
//y ojo que van de derecha a izquierda.
//OJO, que estos elementos van amarrados a la superficie física 10000, para poder sacar
//solamente los elementos que son incoming
Physical Line(1) = {8, 9, 10, 11, 12, 13, 14};

//Results to be printed out
//Elementos línea en los cuales se va a escribir la respuesta, el programa de mallado
//se encarga de tomar los nodos que están en estas lineas y eliminar los nodos
//que se encuentran repetidos.
Physical Line(2) = {2, 6};

//Mesh.SecondOrderIncomplete = 1;
```

Figure 13: Third part of the template file defining physical surfaces (i.e., those to be meshed) for the rectangular scatterer.

We now explain how to incorporate a particular scattterer, in this case with a triangular shape, into the rectangular box previously defined. The V-shaped canyon is shown in fig. 14. The canyon is implemented through the addition of points 17, 18 and 19 forming the lines 22, 23, 24 and 25 shown in thick black line.

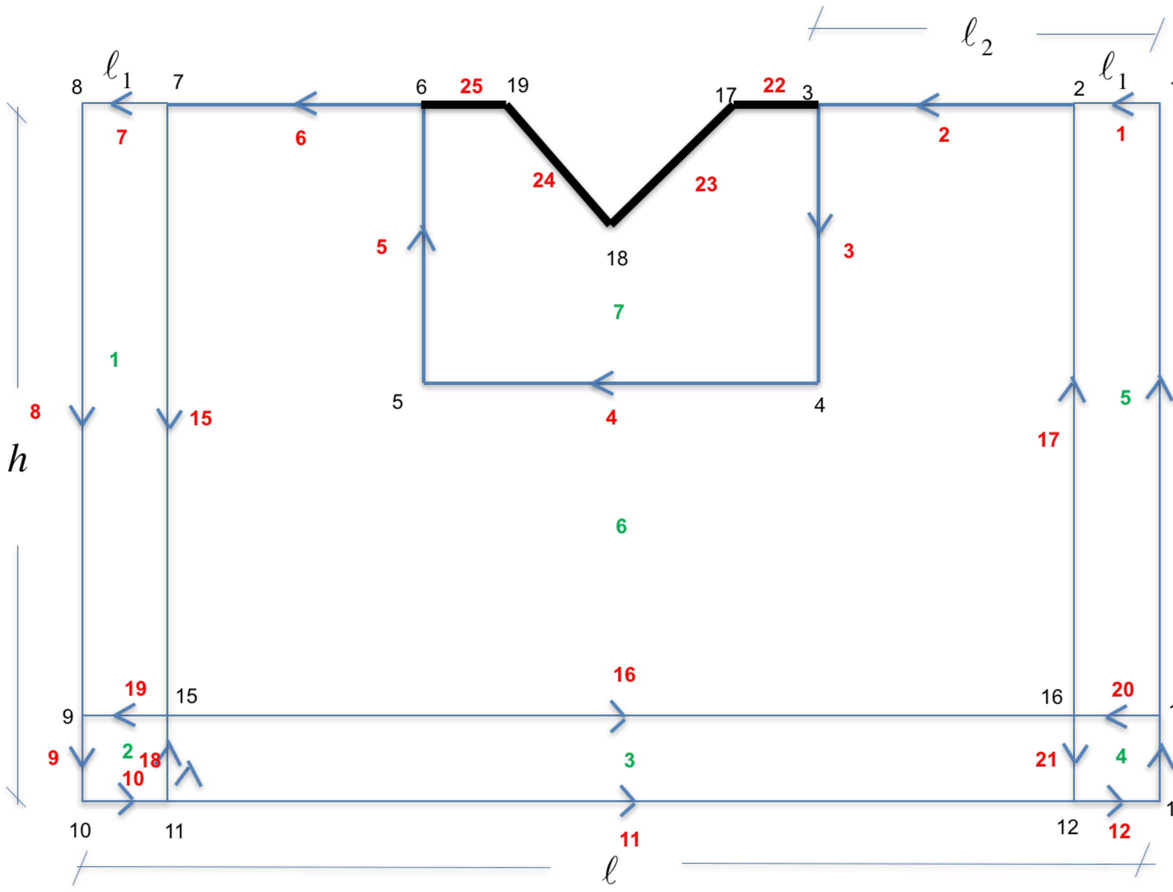
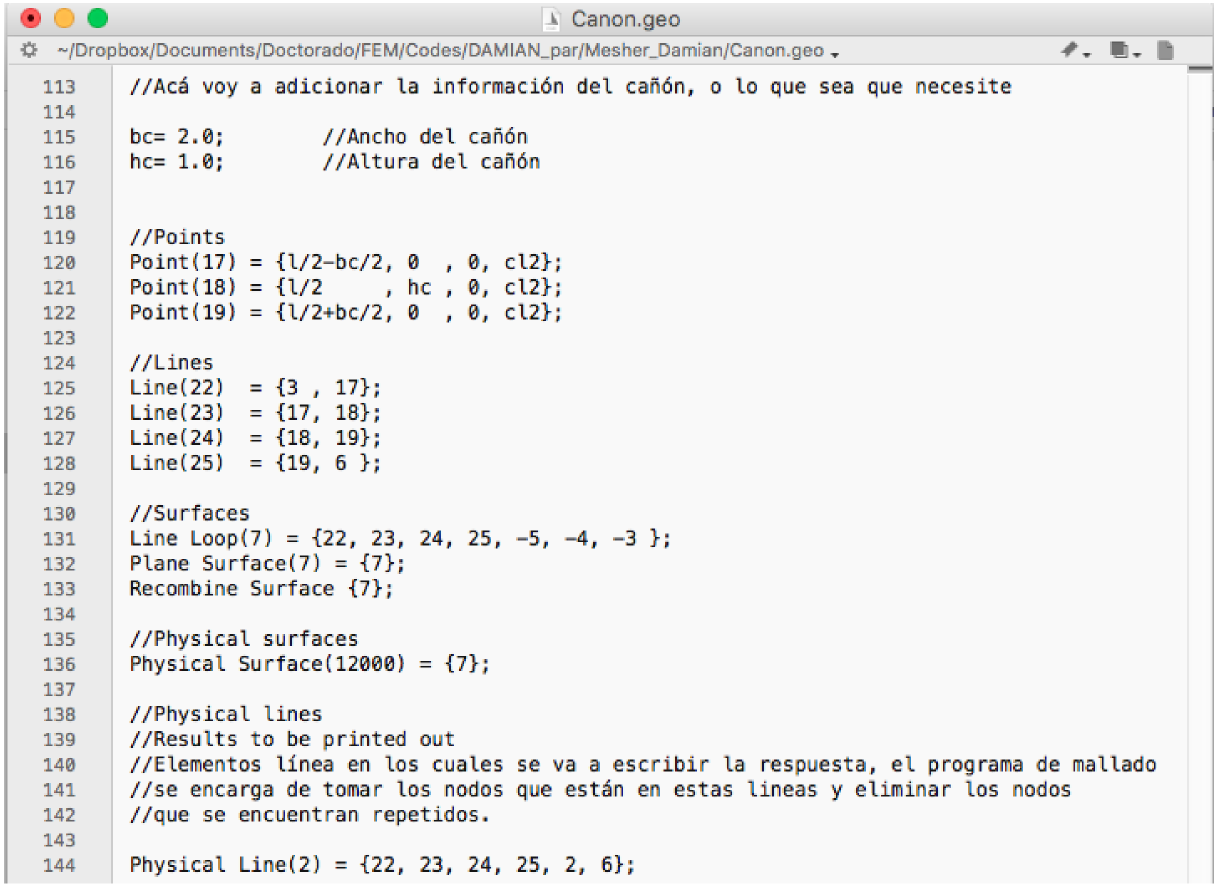


Figure 14: Particular scatterer in the form of a V-shaped canyon.


Canon.geo
~/Dropbox/Documents/Doctorado/FEM/Codes/DAMIAN_par/Mesher_Damian/Canon.geo

```
//Acá voy a adicionar la información del cañón, o lo que sea que necesite

bc= 2.0;        //Ancho del cañón
hc= 1.0;        //Altura del cañón


//Points
Point(17) = {l/2-bc/2, 0  , 0, cl2};
Point(18) = {l/2     , hc , 0, cl2};
Point(19) = {l/2+bc/2, 0  , 0, cl2};

//Lines
Line(22)  = {3 , 17};
Line(23)  = {17, 18};
Line(24)  = {18, 19};
Line(25)  = {19, 6 };

//Surfaces
Line Loop(7) = {22, 23, 24, 25, -5, -4, -3 };
Plane Surface(7) = {7};
Recombine Surface {7};

//Physical surfaces
Physical Surface(12000) = {7};

//Physical lines
//Results to be printed out
//Elementos línea en los cuales se va a escribir la respuesta, el programa de mallado
//se encarga de tomar los nodos que están en estas lineas y eliminar los nodos
//que se encuentran repetidos.

Physical Line(2) = {22, 23, 24, 25, 2, 6};
```

Figure 15: Additional data lines required in the definition of the V-shaped canyon.

Figure 15 shows the additional data lines required in order to add the V-shaped canyon to the generalized rectangular scatterer. The additional lines are defined as described next.

First, we add the geometric parameters definig the scatterer.

- **bc:** Canyon width.
- **hc:** Canyon height.

The data lines corresponding to the new points defining the canyon appear next. These points are numbered immediately after the last point in the template geometry. In the current case the new points are those from 17 to 19. The lines forming the scatterer are now formed and correspond to lines from 22 to 25. There is also a new surface labeled as 7 in the figure. Similarly, there is now a new physical surface which must be assigned a material profile. In the current example there is only one new physical surface labeled **Physical Surface(12000)**. It must be remembered that the definition of the new physical surface implies the definition of a new material profile. In the last part of the file we now define the line **Physical Line(2)** used to indicate points for response output. The resulting canyon model as printed out by **Gmesh** is shown in fig. 16.

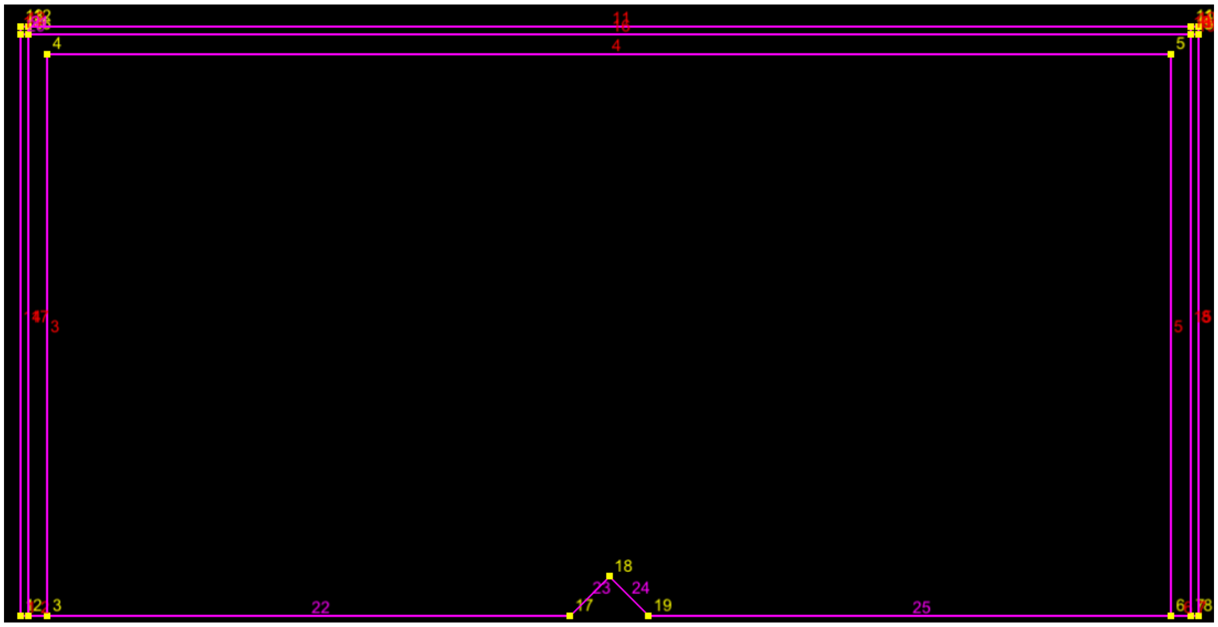


Figure 16: Canyon model as printed out by **Gmsh**. The model is shown upside down as originally created in **Gmsh** with the y axis pointing towards the interior of the half-space.

Meshing the model

The actual mesh for the problem can now be created using the **.geo** file executing the following command from the terminal:

root$ /Applications/Gmsh.app/Contents/MacOS/gmsh canon.geo -2 -order 2 where

- **gmsh:** Command that execute **Gmsh**.
- **canon.geo:** Name of the **Gmsh** file being processed.
- **-2:** Integer flag indicating **Gmsh** that the problem is defined in two dimensions $2 - D$.
- **-order 2:** Integer flag indicating **Gmsh** that the mesh must use 9-noded quads and 6-noded triangles.

After execution of **Gmsh** the file **Canon.msh** containing the actual mesh must be available for viewing. The result is shown in fig. 17.

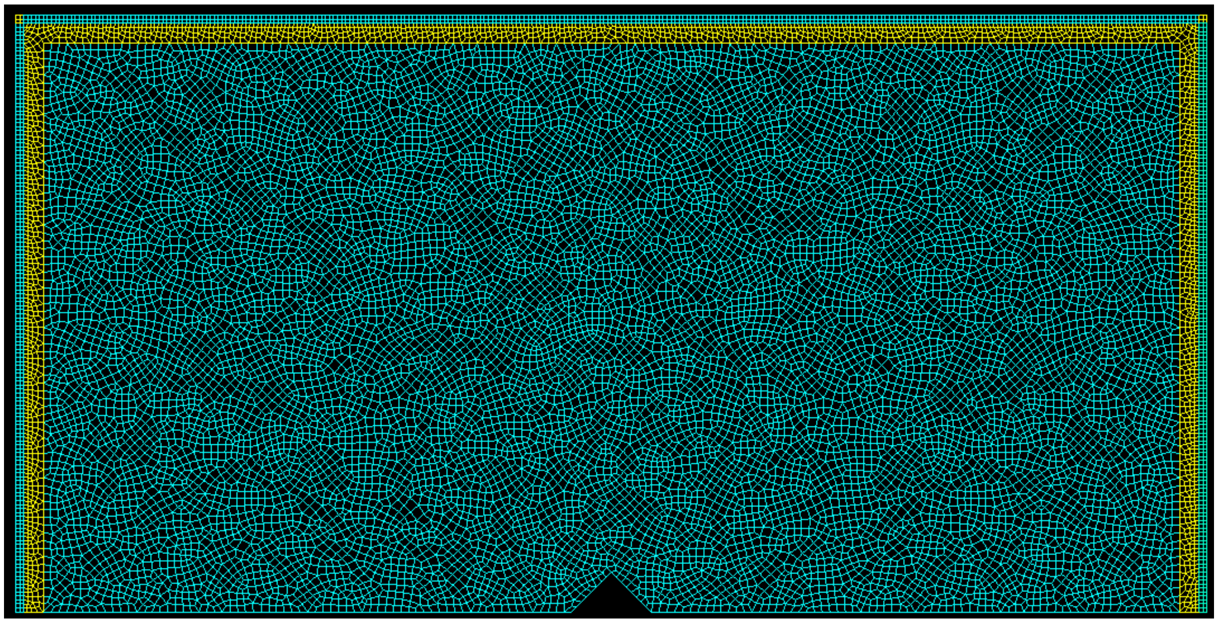
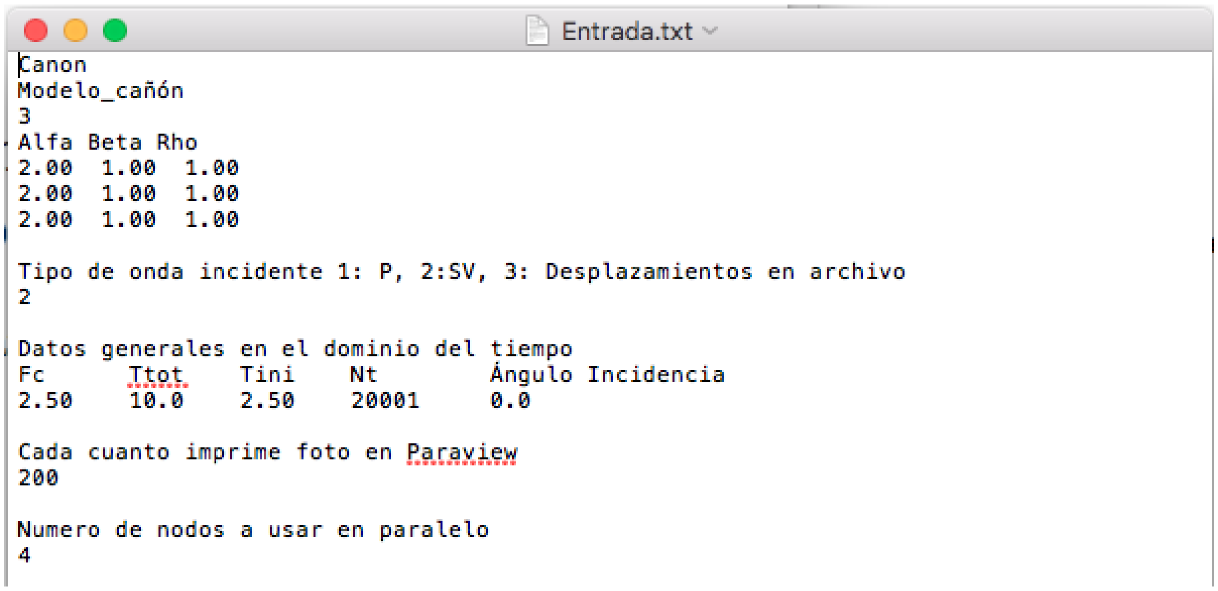


Figure 17: Mesh for the V-shaped canyon generated by **Gmsh**. The model is shown upside down as originally created in **Gmsh** with the y axis pointing towards the interior of the half-space.

Creating the *.inp file

In order to generate the ***.inp** file out of the ***.msh** file we use the Fortran code **mesher.for**. The code uses as input the file **entrada.txt** defining the problem paramters (see fig. 18)



```
Canon
Modelo_cañón
3
Alfa Beta Rho
2.00  1.00  1.00
2.00  1.00  1.00
2.00  1.00  1.00

Tipo de onda incidente 1: P, 2:SV, 3: Desplazamientos en archivo
2

Datos generales en el dominio del tiempo
Fc      Ttot    Tini    Nt       Ángulo Incidencia
2.50    10.0    2.50    20001    0.0

Cada cuanto imprime foto en Paraview
200

Numero de nodos a usar en paralelo
4
```

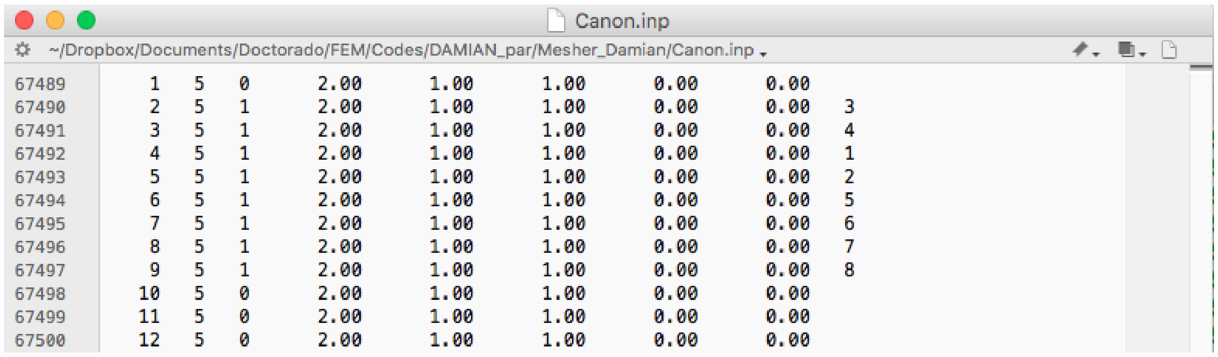
Figure 18: Input file to be processed by mesher.f90 in oder to create the **.inp** file.

This data file is defined as follows:

- **Plantilla:** This is th name of the **Gmsh** file to b processed. This name will also be assigned to the ***.inp WAVES** file.

- **Plantilla_para_generar_archivo_de_entrada_inp:** This is just the problem title.
- **3:** Number of materials to use in the model. The first one corresponds to the elements over the half-space, including the absorbing boundaries. This material is assigned toall the elements located from physical surface 1 to 6.
- **2:** Integer flag indicating the type of wave to be propagated:
  - **1:** Incident **P** wave.
  - **2:** Incidnt **SV** wave.
  - **3:** Imposed displecements defined by a data file.
- **Time domain data (shown here for a Ricker pulse):**
  - **Fc:** Characteristic frquency of the Ricker pulse.
  - **T_tot:** Total size of the time window for the signal.
  - **T_ini:** Central time of the Ricker pulse.
  - **Nt:** Number of increments for the signal.
  - **Incidence angle:** Incidence angle for the plane wave.
- **Incremental time step at which a Paraview snapshot is taken:** Integer flag indicating **WAVES** the time step for snapshot output.
- **Number of computational nodes:** Number of computational nodes to be used when the code is executed in parallel.

Once the file mesher.f90 has been compiled and executed the file **Canon.inp** is generated. If the user needs to define a perfect incoming field then it is necessary to search for those elements having a single node in contact with the scatterer as these elements must be assigned a different element type. The materials in the **.inp** file are defined as follows (see fig. 19):


Canon.inp
~/Dropbox/Documents/Doctorado/FEM/Codes/DAMIAN_par/Mesher_Damian/Canon.inp

```
 1  5  0    2.00    1.00    1.00    0.00    0.00
 2  5  1    2.00    1.00    1.00    0.00    0.00   3
 3  5  1    2.00    1.00    1.00    0.00    0.00   4
 4  5  1    2.00    1.00    1.00    0.00    0.00   1
 5  5  1    2.00    1.00    1.00    0.00    0.00   2
 6  5  1    2.00    1.00    1.00    0.00    0.00   5
 7  5  1    2.00    1.00    1.00    0.00    0.00   6
 8  5  1    2.00    1.00    1.00    0.00    0.00   7
 9  5  1    2.00    1.00    1.00    0.00    0.00   8
10  5  0    2.00    1.00    1.00    0.00    0.00
11  5  0    2.00    1.00    1.00    0.00    0.00
12  5  0    2.00    1.00    1.00    0.00    0.00
```

Figure 19: Proper ordering of the material profiles required in the **WAVES .inp** file.

1. Material to be used by the absorbing boundaries and by those elements belonging to the strip but not defining the incoming motion.

2. Material to be used by the incoming element with surface 1 in contact with the scatterer.
3. Material to be used by the incoming element with surface 2 in contact with the scatterer.
4. Material to be used by the incoming element with surface 3 in contact with the scatterer.
5. Material to be used by the incoming element with surface 4 in contact with the scatterer.
6. Material to be used by the incoming element with nodal point 1 in contact with the scatterer.
7. Material to be used by the incoming element with nodal point 2 in contact with the scatterer.
8. Material to be used by the incoming element with nodal point 3 in contact with the scatterer.
9. Material to be used by the incoming element with nodal point 4 in contact with the scatterer.

The material profiles additionally defined are those corresponding to the remaining physical surfaces.

8 Running the program

To execute **WAVES** use the following command from the terminal:

```
./waves.out
```

and input the file name, in this case **Canon.inp**