Escola de Camins
Escola Tècnica Superior d'Enginyeria de Camins, Canals i Ports
UPC BARCELONATECH

Study of the influence of tunnel excavations on pile founded buildings

Treball realitzat per:
Genís Majoral Oller

Dirigit per:
Dr. Eduardo Alonso Pérez de Ágreda
Dra. Anna Ramon Tarragona

Grau en:
Enginyeria Civil

Barcelona, setembre 2018

Departament d'Enginyeria del Terreny

TREBALL FINAL DE GRAU

Study of the influence of tunnel excavations on pile founded buildings

Genís Majoral Oller

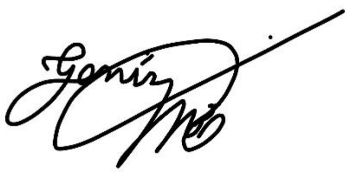
Declaració d'originalitat

L'autor d'aquest treball, Genís Majoral Oller, fa constar que aquest treball s'ha escrit fruit de la seves pròpies paraules i raonament, a excepció d'aquelles parts que estan degudament citades o parafrasejades, sempre amb la pertinent referència d'acord amb la normativa vigent de l'Escola de Camins i sota la tutoria del Dr. Eduardo Alonso Pérez de Ágreda i la Dra. Anna Ramon Tarragona.

Aquesta declaració confirma que:

- En Genís Majoral Oller estudiant del grau en Enginyeria Civil de l'Escola de Camins, ETSECCPB, és l'exclusiu autor d'aquest treball.
- No s'ha comés cap tipus de plagi i s'han seguit les normes de citació dictades per la ETSECCPB, Universitat Politècnica de Barcelona.
- S'han documentat de manera veraç totes les metodologies, dades i processos.
- No s'ha manipulat cap dada.
- S'han citat a totes les persones que han facilitat de manera significativa la elaboració del present treball.

I per a que així en quedi constància signa l'autor



Genís Majoral Oller

Barcelona, a 26 de setembre de 2018.

Abstract

With the development of underground construction in urban areas, tunnelling exerts some unavoidable influence on adjacent buildings, that may become an important problem for many high-rise buildings, or others, that are supported by piled foundations. This may result in additional settlement and lateral displacements, hence a change in internal forces that may lead to structural distress or failure of the piled foundation.

Existing geotechnical software or FEM analysis may be used to solve the tunnel-pile interactions. However, semi-analytical formulation may make it worth solving at a reduced computational cost. To carry out this project, the following fundamental theory has been used: the Sagaseta problem (1987) describing displacements due to an underground soil loss point, the Mindlin problem (1921) for subsurface movements induced by a subsurface force, and the Boussinesq problem (1885) for a surface force. The required variables are the soil elastic modulus and Poisson ratio, the tunnel and piles geometry, the tunnel areal ground loss and the applied forces. The fundamental solutions used are applicable for an elastic, homogeneous, isotropic and incompressible soil.

The project aims to describe and comprehend the existing formulae, while creating a simple semi-analytical procedure implemented in MATLAB language. Throughout the project many subproblems arise such as the effect of a rigid pile cap, how to discretise an advancing tunnel and the applicability of a compensation grouting option in the MATLAB code. Afterwards, several simulations will be run to understand the response of piles to an underground soil volume loss. Likewise, the effects of an advancing tunnel will be studied, for different configurations and throughout the advancement of the tunnel towards and past the foundations. Finally, the capabilities of the developed MATLAB code will be shown with the analysis of the effects of a compensation grouting injection produced on a certain foundations and the study of the Sagrada Família pile wall (Ledesma & Alonso, 2015).

Resum

Amb el desenvolupament de construccions en el subsol de zones urbanes, els túnels exerceixen unes influències inevitables en les edificacions adjacents, que poden derivar en un important problema per molts grata-cels o altres edificis que tenen pilots com a fonamentació. Això es pot traduir en assentaments i moviments laterals addicionals, per tant, un canvi en les forces internes que pot provocar danys estructurals i falles als pilots.

Alguns softwares geotècnics existents, o anàlisis MEF, poden ser emprats per resoldre la interacció túnel-pilots. Tanmateix, els procediments semi-analítics poden resultar d'interès atès el reduït cost computacional que presenten. En base això, la següent teoria proporciona la formulació necessària per desenvolupar el treball: Sagaseta (1987) i el problema de pèrdua de sòl en un punt, Mindlin (1921) i el problema de moviments a causa d'una força a l'interior d'un sòl i Boussinesq (1885) i el problema de moviments a causa d'una força en superfície. Les variables necessàries seran el mòdul elàstic i el coeficient de Poisson del sòl, les geometries del túnel i els pilots, la pèrdua de volum per metre lineal de túnel i les forces aplicades. Les solucions fonamentals emprades són aplicables per sòls elàstics, homogenis, isotròpics i incompressibles.

Aquest projecte se centra en descriure i comprendre la formulació existent, crear un procediment semi-analític i implementar-lo en el llenguatge MATLAB. Durant el projecte apareixen sub-problemes com tenir en compte l'encepat d'un grup de pilots, com discretitzar un túnel i l'aplicabilitat d'una funció de *compensation grouting* al codi MATLAB. Més endavant, diverses simulacions es duran a terme per entendre la resposta dels pilots envers una pèrdua de volum de sòl. A més, els efectes d'un túnel en avenç també seran estudiats, per diferents configuracions de túnel i al llarg de l'avenç d'aquest. Finalment, el potencial del codi MATLAB es veuran amb l'aplicabilitat d'un *compensation grouting* a uns determinats fonaments (i l'estudi dels seus efectes) i amb l'estudi de la pantalla de pilots de la Sagrada Família (Ledesma i Alonso, 2015).

TABLE OF CONTENTS

1 INTRODUCTION

Tunnel excavations threaten building foundations, typically in massified urbanised areas where piled foundations flood the underground space. One of the challenges this poses consists in determining the tunnel-pile interaction, and when the structure is jeopardised, to engineer a plan that mitigates the effects.

The structure of this work starts by detailing, in section 2, the formulation that underpins the analyses carried out in the current project, based on authors like H.G. Poulos and E.H Davis for the general structural approach, R.D. Mindlin and his subsurface force induced movements, J.V. Boussinesq to account for movements due to surface forces and C. Sagaseta the precursor with his fundamental formulation in soil movements due to a ground point volume loss.

Afterwards, section 3 describes the structure of the code and how it flows. While section 4 verifies the output provided by the code against known solutions or commercial software (Plaxis), section 5 actually computes results of interest to examine and learn how the piles behave to different scenarios and to an advancing tunnel. Section 6 analyses a piled foundation through two stages: prior and after applying a compensation grouting injection once a tunnel excavation is completed, it is intended to provide more insight in the response to a tunnel. Eventually, section 6 studies a real case, the Sagrada Família pile wall.

That being said, the main objectives of this dissertation would be classified into:

- Describe and comprehend the existing formulation for pile analysis, tunnel ground loss induced soil movements and the pile-tunnel interaction.
- Create a code that analyses pile-tunnel interaction based on exiting formulation, while learning how to manage and structure a large-sized code.
- Actually simulate the interaction between a group of piles and a tunnel in several situations characterized by different foundation and tunnel designs.

2 PROBLEM MODELLING

This section aims to describe the model being used, both the existing fundamental formulae and its development and adaptation towards creating a functional code.

In the first place, use is made of Poulos and Davis (1980) handbook *Pile and Foundation Analysis Design* which sets the basics on pile foundation analysis. Likewise, Sauter's master thesis and his code (2012), along with the posterior article (Alonso et al., 2015), developed to analyse a swelling soil strata under Pont de Candí has helped and played a reference role in the comprehension process.

To begin with the actual problem, let a certain pile group be bored in a soil. The fundamental equations used constrain the analysis to be applicable for an elastic, undrained, homogeneous, isotropic and incompressible soil. Let us start by saying that the problem uncouples the soil and pile structures, as seen in Figure 1. On the one hand, the pile structure becomes just a matter of elastic structural analysis. It may resemble beam analysis in any building, considering that there may have a pile cap (or not) which will restrain head movements in comparison to the rest of the pile group. The pile structure accounts for the external loads and moments applied at the pile heads, these are known data. Most importantly, the unknowns are both the piles displacements laws and the load distribution acting along the piles shafts.

On the other hand, the soil structure may be conceived as a free field soil where displacements and stresses at the very location of the pile structure are again unknown. In this problem, displacements generated by both ground losses due to a tunnel and/or surface loads are calculated. Parallelly, there is the so-called shielding effect, this is taken into account in the form of influence coefficients calculated by means of the Mindlin subsurface force problem (1937): the displacement of each pile induce some forces that have an effect along the shaft of the neighbouring piles.

As stated earlier, unknowns are both stresses and displacements, which can be related in a simple equation. In order to solve the problem, both displacements at the pile and soil structures will be equalled to work out stresses.

Therefore it is key to comprehend that the current elastic analysis is about relative stiffnesses and interaction between elastic bodies in contact along with a displacement compatibility equation.

Solutions for horizontal and vertical analysis are also uncoupled, i.e. settlements do not influence horizontal displacements and so forth.

Piles will be discretised into a number of elements and all the equations will be developed in finite differences, which will derive into a system of equations assembled in matrix form, the reason being the use of MATLAB for an easy computational resolution. Afterwards, bending moments, shear stresses and axial compression laws are calculated for each pile.

Concerning the modelling of the tunnel, it will also be discretised into various sections, each containing a number of points that model the associated tunnel volume loss. The induced displacements in the free field soil due to a sink point are found according to Sagaseta (1987).

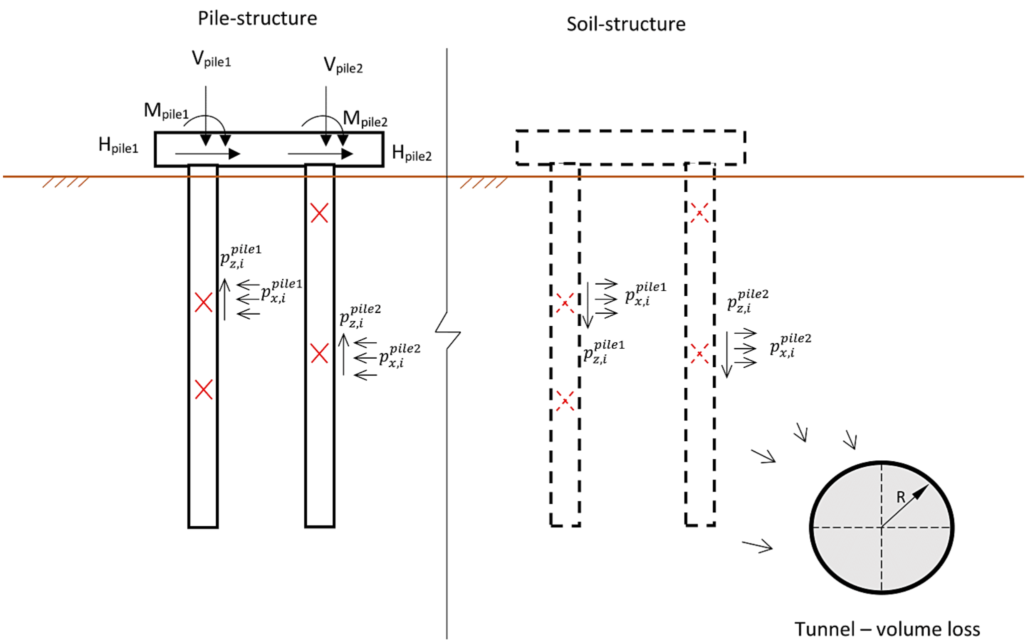


Figure 1. Sketch representation of the problem's approach, both pile (left) and soil (right) structures are depicted.

2.1 PILE MODELLING

2.1.1 Piles lateral response

Over the following lines, the structural analysis of a single pile will be analysed, knowing that it is extendable to a group of piles.

Deep foundations will be treated as beams where external forces and moments act at the top of the pile either for a free or fixed head. The soil exerts an unknown distribution of lateral and vertical stresses along the shaft of the pile, as well as a base vertical stress. Within each discretised element of the pile, the stresses are assumed to remain constant, the smaller the length of each discretisation, the more accurate the assumption is.

A beam's differential bending equation is:

$$E_p I_p \cdot \frac{d^4 w}{dz^4} = -q = pd \qquad (1)$$

Where

E_p is the elastic modulus of the pile

I_p is the moment of inertia of the pile's section

z is the depth in soil

d is the width or diameter of the pile

w is the pile's deflection

p is the horizontal shear stress (kN/m²)

q is the linear load (kN/m)

Applying finite differences the equation for any element i on any pile would then be:

$$-p_i d \cdot \frac{h_i^4}{E_p I_p} = \delta_{i-2} - 4\delta_{i-1} + 6\delta_i - 4\delta_{i+1} + \delta_{i-1} \qquad (2)$$

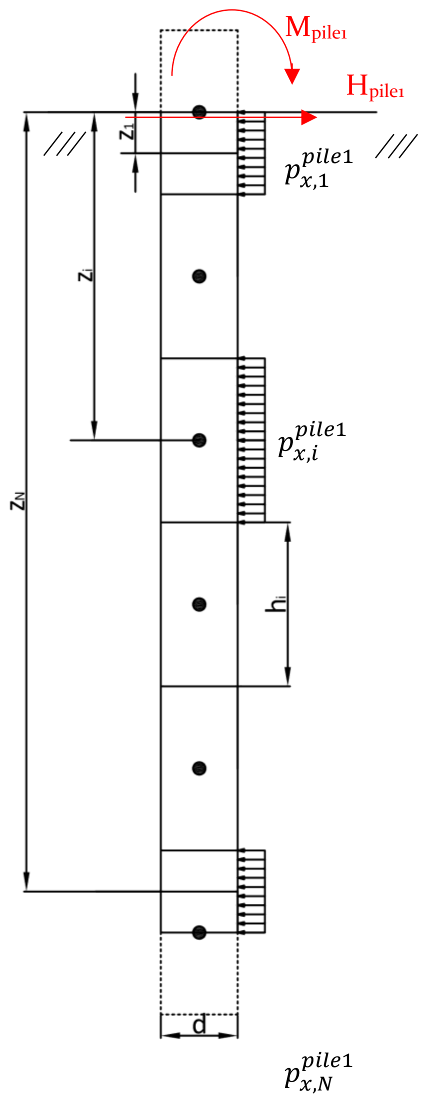


Figure 2. Vertical view sketch of a discretised single pile. Diameter (d), load (p_i), element length (h_i), element depth (z_i) and external forces (H and M) are depicted.

There are N elements discretising the pile, placed so that the nodes are located like shown in Figure 2. Therefore, arising N equations to solve. Note how the first and last nodes fall exactly on the top and tip of the pile. Every element has a length of $h_i = \frac{L}{N-1}$ and notice how the lateral load acts upon the whole length of the elements, except for the first and last ones that only do so in half of it, provided the load acts in half of the length.

Equation (2) is applied to elements 2 to N-1. It would similarly be applied to the first and last elements, but then two new unknowns would come up corresponding to the virtual nodes -2 and N+2.

In order to avoid this, the analysis is completed with the following equilibrium equations:

Force equilibrium

$$\sum_{i=1}^{N} p_i h_i d = H \qquad (3)$$

Where

H	is the external horizontal force applied to the top
p_i	is the load acting upon every element
d	is the piles diameter

h_i is the height of each element $\frac{L}{N-1}$ for each one except the first and last elements (1 and N) where it is halved: $\frac{1}{2}\frac{L}{N-1}$

Moment equilibrium

$$\sum_{i=1}^{N} p_i h_i z_i d + M_E = M_{head} \tag{4}$$

Where the new variables are

M_E is the external moment applied at the top

z_i is the distance of the centre of the load of every discretisation to the head of the pile:

$$z_1 = 0{,}5 \cdot \frac{0{,}5dL}{N-1}$$

$$z_i = 0{,}5 \cdot \frac{0{,}5L}{N-1} \qquad for\ \ i \epsilon\, 2, \dots, N-1$$

$$z_N = 0{,}5 \cdot \frac{0{,}5L}{N-1}$$

M_{head} is the moment at the head of the pile

Additionally, to eliminate virtual nodes -1 and N+1, that appear in the bending equation, depending on the type of pile cap, use is made of the following boundary conditions.

Constrained rotation at the head of the pile

For instance, if there happens to be a pile cap.

$$\left(\frac{d(\delta)}{dz} EI\right)_{z=0} = 0 = \delta_2 - \delta_{-1} \tag{5}$$

Moment at the head of the pile

A free headed pile, for example, will present null moment at the top.

$$M_{head} \frac{\left(\frac{L}{N}\right)^2}{EI} = \delta_2 - 2\delta_1 + \delta_{-1} = 2\delta_2 - 2\delta_1 \tag{6}$$

Moment at the tip of the pile

The pile tip will always be considered free, hence presenting null moment:

$$M_{tip} = \left(\frac{d^2(\delta)}{dz^2}EI\right)_{z=0} = \delta_{n+1} - 2\delta_n + \delta_{n-1} = 0 \qquad (7)$$

Results in:

$$\delta_{n+1} = 2\delta_n - \delta_{n-1}$$

From here, the analysis distinguishes two different pile head cases: free head or fixed head.

Free head pile

For this case, the following boundary conditions apply:

- Null moment at the head of the pile
- Null moment at the tip of the pile

The first and last rows are used to fit in the two equilibrium equations, whereas the rest correspond to the finite difference form of the bending equations:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \dots & & & & 0 & 0 \\ 2 & 5 & -4 & 1 & 0 & & & & & & \\ 1 & -4 & 6 & -4 & 1 & & & & & & \\ & & & & & \dots & & & & & \\ & & & & & 1 & -4 & 6 & -4 & 1 & 0 \\ & & & & & 0 & 0 & 1 & -4 & 5 & -2 \\ 0 & 0 & & & & \dots & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ \dots \\ \dots \\ \delta_{n-2} \\ \delta_{n-1} \\ \delta_n \end{bmatrix} + \begin{bmatrix} BC \\ BC2 \\ 0 \\ \dots \\ 0 \\ BC3 \end{bmatrix} =$$

$$= \begin{bmatrix} \frac{0,5dL}{N-1} & \frac{dL}{N-1} & & \dots & & \frac{dL}{N-1} & \frac{0,5dL}{N-1} \\ 0 & \frac{d}{EI}\left(\frac{L}{N}\right)^4 & 0 & & & & \\ & & & \dots & & & \\ & & & & 0 & \frac{d}{EI}\left(\frac{L}{N}\right)^4 & 0 \\ \frac{0,5dL}{N-1}z_1 & \frac{dL}{N-1}z_2 & & \dots & & \frac{dL}{N-1}z_{N-1} & \frac{0,5dL}{N-1}z_N \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ \dots \\ \dots \\ p_{n-2} \\ p_{n-1} \\ p_n \end{bmatrix} \qquad (8)$$

Where:

$$BC1 = H_E$$

$$BC2 = \frac{M}{EI}\left(\frac{L}{N}\right)^2$$

$$BC3 = M_E$$

Put it in matrix form:

$$[D_{pile}]\{\delta_{pile}\} + [BC_{pile}] = [A_{pile}]\{p_{pile}\} \tag{9}$$

Where

D_{ij}	$N \times N$ *stiffness* matrix
δ_j	N size pile displacements column vector
BC_j	N size load column vector
p_j	N size column vector
D_{ij}	$N \times N$ load coefficients matrix

Constrained or casted pile

For this case, which tends to be more common, piles are casted into a pile cap. This has an effect on the piles of a group that compels their heads to remain in a plane and maintain the spacing at the top (see section 2.5). Concerning the assembly of equations for a single pile the next boundary conditions are used:

- Rotation at the head is zero.
- Moment at the end of the pile is zero.

Furthermore, the two equilibrium equations are placed in the first and last rows. In this case, unlike the free headed problem where M_{head} is zero, there is some bending moment, namely fixing moment. It is taken into account with the following.

$$M_{head} = \left(\frac{d^2(\delta)}{dz^2}EI\right)_{z=0} = -2\delta_1 + 2\delta_2$$

Therefore, the moment equilibrium equation is:

$$\sum_{i=1}^{N} p_i h_i z_i d + M_E = M_{head} = -2\delta_1 + 2\delta_2 \tag{10}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & & & & & & \\ -4 & 7 & -4 & 1 & & & & & & \\ 1 & -4 & 6 & -4 & 1 & & & & & \\ & & & & \dots & & & & & \\ & & & & 1 & -4 & 6 & -4 & 1 & \\ & & & & & & 1 & -4 & 5 & -2 \\ -2 & 2 & 0 & & \dots & & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \\ \dots \\ \dots \\ \delta_{n-2} \\ \delta_{n-1} \\ \delta_n \end{bmatrix} + \begin{bmatrix} BC1 \\ BC2 \\ 0 \\ \\ \dots \\ \\ 0 \\ BC3 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{0{,}5dL}{N-1} & \frac{dL}{N-1} & \dots & \frac{dL}{N-1} & \frac{0{,}5dL}{N-1} \\ 0 & \frac{d}{EI}\left(\frac{L}{N}\right)^4 & & & \\ & & \dots & & \\ & & & \frac{d}{EI}\left(\frac{L}{N}\right)^4 & 0 \\ \frac{0{,}5dL}{N-1} z_1 & \frac{dL}{N-1} z_2 & \dots & \frac{dL}{N-1} z_{N-1} & \frac{0{,}5dL}{N-1} z_N \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \\ \dots \\ \\ p_{N-1} \\ p_N \end{bmatrix}$$

Where

$$BC1 = H_E$$

$$BC2 = \frac{M}{EI}\left(\frac{L}{N}\right)^2$$

$$BC3 = M_E$$

Let this be put as:

$$[D_{pile}]\{\delta_{pile}\} + [BC_{pile}] = [A_{pile}]\{p_{pile}\} \qquad (11)$$

2.1.2 Piles axial response

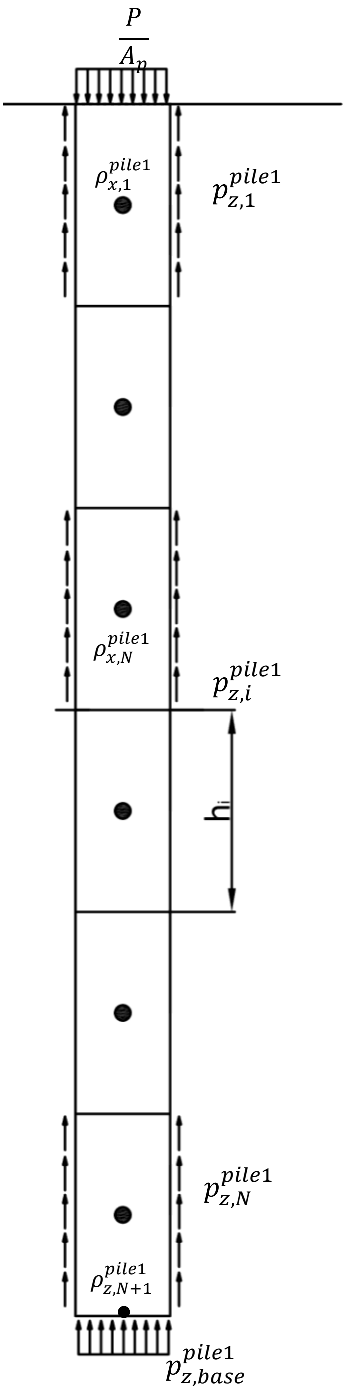


Figure 3. Vertical view of single pile subjected to settlement.

For vertical pile modelling, each element measures now $h_i = \frac{L}{N}$. Nodes have to be defined so that they are located like shown in Figure 3. In this problem, N skin stresses and 1 base stress make N+1 unknowns. Defining σ_i as the axial stress at node *i*, it can be derived from axial equilibrium of an element that:

$$\sigma_i = \sigma_{i-1} + \frac{\partial\sigma}{\partial z} \cdot h_i = \sigma_{i-1} - \frac{p\pi d}{A_p} h_i$$

$$\frac{\partial\sigma}{\partial z} = -\frac{p\pi d}{A_p} = -\frac{4p}{R_A d} \qquad (12)$$

Where:

$$R_A = \frac{A_p}{\frac{\pi d^2}{4}}$$

As defined in Poulos and Davis (1980), for circular massive piles $R_A = 1$.

On the other hand, from elastic theory:

$$\frac{\partial\rho}{\partial z} = -\frac{\sigma}{E_p}$$

$$\frac{\partial^2\rho}{\partial z^2} = -\frac{1}{E_p}\frac{\partial\sigma}{\partial z} = \frac{1}{E_p}\frac{4p}{R_A d} \qquad (13)$$

Having found a relationship for shaft stress p as a function of vertical node displacements, now, by means of finite differences:

$$p_i = \frac{R_A d}{4E_s} \cdot \frac{\rho_{i+1} - 2\rho_i + \rho_{i-1}}{\left(\frac{L}{N}\right)^2} \qquad (14)$$

From now on:

$$p_i = K \cdot [\rho_{i+1} - 2\rho_i + \rho_{i-1}]$$

The final form of the (N+1) system of equations for a single pile is:

$$\begin{bmatrix} -2 & 2 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & & & & & \\ 0 & 1 & -2 & 1 & 0 & & & & & \\ & & & & \dots & & & & & \\ & & & & & \dots & & & & \\ & & & & & 0 & 1 & -2 & 1 & 0 \\ & & & & & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \\ \dots \\ \dots \\ \delta_{n-2} \\ \delta_{n-1} \\ \delta_n \end{bmatrix} + \begin{bmatrix} BC1 \\ 0 \\ 0 \\ \\ \dots \\ \\ 0 \\ P \end{bmatrix} =$$

$$\begin{bmatrix} 1 + K \cdot D1 & 0 & 0 & \dots & \dots & 0 \\ 0 & 1/K & 0 & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1/K & 0 & 0 \\ 0 & 0 & \dots & 0 & D2 & 1/2 \cdot L/N \cdot 1/E_p \\ \frac{pidL}{N} & \dots & \dots & \dots & \frac{pidL}{N} & pi\left(\frac{d}{2}\right)^2 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ \\ \dots \\ \\ p_{N-1} \\ p_N \\ p_{N+1} \end{bmatrix} \tag{15}$$

Let this be put as:

$$[D]\{\delta_{pile}\} + [BC] = [A]\{p\} \tag{16}$$

Where

$$BC1 = K\frac{P}{A_p E_s} \cdot \frac{L}{N}$$

$$D1 = \left(\frac{L}{N}\right)^2 \cdot pi \cdot \frac{\frac{d}{Ep}}{pi \cdot \frac{d^2}{4}}$$

$$D2 = \left(\frac{1}{2} \cdot \frac{L}{N}\right)^2 \cdot pi \cdot \frac{\frac{d}{Ep}}{pi \cdot \frac{d^2}{4}}$$

D1 and D2 appear once imposed the equilibrium of axial forces at the first element (see Figure 4) and displacements compatibility at the last element (see Figure 5).

For the first element:

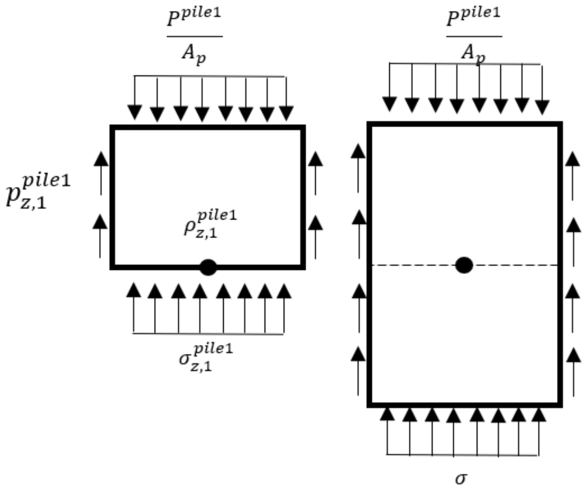


Figure 4. Sketch of stresses upon element 1, analysis extendable to any pile.

$$\sigma_1 A_p = P - p_1 \left(pi \cdot d \cdot \frac{1}{2} \cdot \frac{L}{N} \right)$$

$$-\frac{\sigma_1}{E_p} = \frac{\partial \rho}{\partial z} = \frac{-\rho_2 + \rho_{-1}}{2 \cdot \frac{L}{N}}$$

Thus, the equation inserted in the first row is:

$$p_1 \left(1 + Kpi \cdot d \cdot \frac{L^2}{N^2} \cdot \frac{1}{E_s A_p} \right) = K\{-2\rho_1 + 2\rho_2\} + K \frac{P}{A_p E_s} 2 \cdot \frac{L}{N}$$

For the last element:

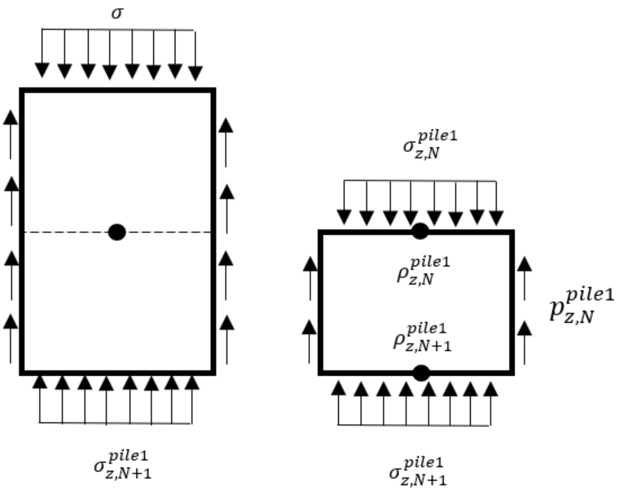


Figure 5. Sketch of actions upon element N.

$$\rho_N = \rho_B - \frac{\partial \rho}{\partial z} \cdot \frac{1}{2} \cdot \frac{L}{N}$$

$$\sigma_N A_p = \sigma_B A_p + p_N \left(pi \cdot d \cdot \frac{1}{2} \cdot \frac{L}{N} \right)$$

The previous equations come to give the equation inserted in the last row of matrix $[D]$:

$$\rho_N - \rho_B = \sigma_B \cdot \frac{1}{E_p} \cdot \frac{1}{2} \cdot \frac{L}{N} + p_N \cdot \frac{1}{E_p} \cdot \left(\frac{1}{2} \cdot \frac{L}{N} \right)^2 \cdot (pi \cdot d) \cdot \frac{1}{A_p}$$

2.1.3 Pile structure matrix assembly

Based on the previously explained equations, notation does not distinguish horizontal or vertical analysis, so the following generalised form is valid for each direction, always bearing in mind the intrinsic differences. Just as a reminder, in the horizontal analysis the number of equations is $(N_{elements}) \cdot (N_{piles})$, whereas in the vertical analysis it is $(N_{elements} + 1) \cdot (N_{piles})$.

$$[D_{pile}]\{\delta_{pile}\} + [BC_{pile}] = [A_{pile}]\{p_{pile}\}$$

Assembling a system of equations for N_{piles} piles would give:

$$\begin{bmatrix} [D]_{pile1} & & \\ & \cdots & \\ & & [D]_{pileN_p} \end{bmatrix} \begin{Bmatrix} \delta_{pile1} \\ \cdots \\ \delta_{pileN_p} \end{Bmatrix} + \begin{Bmatrix} BC_{pile1} \\ \cdots \\ BC_{pileN_p} \end{Bmatrix} = \begin{bmatrix} [A]_{pile1} & & \\ & \cdots & \\ & & [A]_{pileN_p} \end{bmatrix} \begin{Bmatrix} p_{pile1} \\ \cdots \\ p_{pileN_p} \end{Bmatrix}$$

The final form may be written as:

$$[D]\{\delta\} + [BC] = [A]\{p\} \tag{17}$$

2.2 Soil modelling

The soil is modelled as an elastic continuum, homogeneous, isotropic and incompressible. As stated earlier, the soil displacements come from 3 sources: the tunnel excavation volume loss, the displacements due to subsurface forces and from surface forces. Each sort of movements is solved based on the problems presented by the following authors.

2.2.1 Sagaseta

Sagaseta (1987) presented the following solution for 3D elastic, homogeneous, isotropic and incompressible soil. Figure 6 depicts the problem.

Let the following definitions be:

$$\Delta x = X_{afected} - X_{vloss}$$

$$\Delta y = Y_{afected} - Y_{vloss}$$

$$z = Z_{afected}$$

$$x = sqrt(\Delta x^2 + \Delta y^2)$$

$$r_1 = (x^2 + (z-h)^2)^{\frac{1}{2}}$$

$$r_2 = (x^2 + (z+h)^2)^{\frac{1}{2}}$$

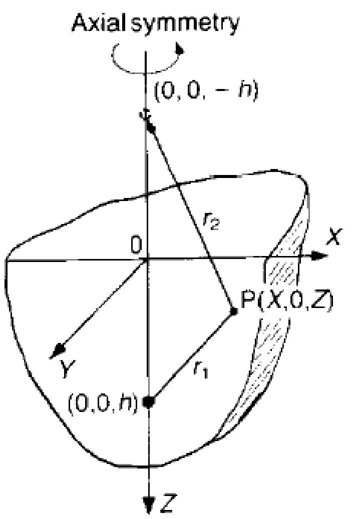


Figure 6. Point sink, problem definition as shown in Fig. 9 by Sagaseta (1987).

Sagaseta (1987) defined the soil displacement at a point due to a localised volume loss (or sink point) as the sum of:

$$\rho_x = \rho_{x,paved} + \rho_{x,free} \qquad (18)$$

This comes from the solving process described in his article (Figure 7). Imposing a free surface condition, it starts by computing displacements in an infinite medium. Then add the contribution of a negative image source to cancel out normal stresses at the surface. Finally, to achieve the free surface condition, the shear stresses due to steps 1 and 2 are removed in step 3. As formulated in equation (18) $\rho_{x,paved}$ corresponds to the sum of steps 1 and 2, called paved because as Sagaseta explains in step 2 there are no surface normal stresses but there are shear stresses constraining surface horizontal movement. This may resemble a situation where there is an inextensible membrane, like an urban pavement, hence the term paved.

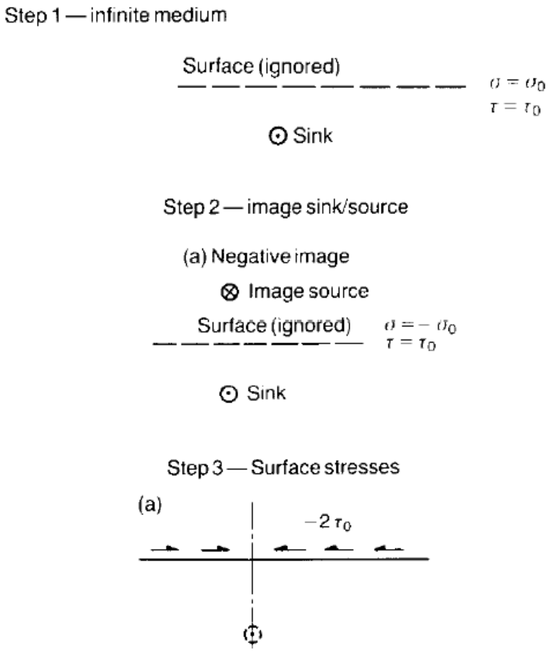


Figure 7. Steps in the Sagaseta analysis extracted from Sagaseta (1987)

Equation (19) is the result of step 2,

$$\rho_{x,paved} = -\frac{r^3}{3}\left(\frac{\Delta x}{r_1^3} - \frac{\Delta x}{r_2^3}\right) \tag{19}$$

$$\rho_{x,free} = \frac{2}{\pi} rad^3 \frac{h}{x} \int_0^{+\infty} r_b \cdot \frac{\alpha}{(h^2 + \alpha^2)^{\frac{5}{2}}} \cdot [I_E \cdot E(k) + \ I_F F(k)] d\alpha \tag{20}$$

Equations above are valid for horizontal displacements, coordinate y displacement may be similarly calculated thanks to axial symmetry.

Vertical displacements are the sum of equations (23) and (22):

$$\rho_z = \rho_{z,paved} + \rho_{z,free} \tag{21}$$

$$\rho_{z,paved} = -\frac{rad^3}{3} \cdot \left(\frac{z-h}{r_1^3} - \frac{z+h}{r_2^3}\right) \tag{23}$$

$$\rho_{z,free} = \frac{2}{\pi} rad^3 hz \int_0^{+\infty} \frac{1}{r_b} \cdot \frac{\alpha}{(h^2 + \alpha^2)^{\frac{5}{2}}} \cdot [J_E \cdot E(k) + \ F(k)] d\alpha \tag{22}$$

Where for both cartesian directions, $F(k)$ and $E(k)$ are complete elliptic functions of first and second kind respectively with;

$$k = \left(1 - \frac{r_a^2}{r_b^2}\right)^{0,5}$$

And

$$r_a = \sqrt{(\alpha - x)^2 + z^2}$$

$$r_b = \sqrt{(\alpha + x)^2 + z^2}$$

$$IE = 1 + \frac{1}{2} z^2 \left(\frac{1}{{r_a}^2} + \frac{1}{r_b^2} \right)$$

$$I_F = -\frac{1}{r_b^2}(\alpha^2 + x^2 + 2z^2)$$

$$J_E = -1 + 2\big(\alpha \cdot (\alpha - x)\big) \cdot \frac{1}{r_a^2}$$

2.2.2 Boussinesq

The well-known Boussinesq solution (1885) for a surface horizontal point load:

$$\rho_x = \frac{P(1+v_s)}{2\pi E_s R} \cdot \left(1 + \frac{x^2}{R^2} + (1 - 2v_s)\left(\frac{R}{R+z} - \frac{x^2}{(R+z)^2}\right)\right) \tag{24}$$

Where

$$R = \sqrt{x^2 + y^2 + z^2}$$

The Y displacement is similarly calculated by means of axial symmetry.

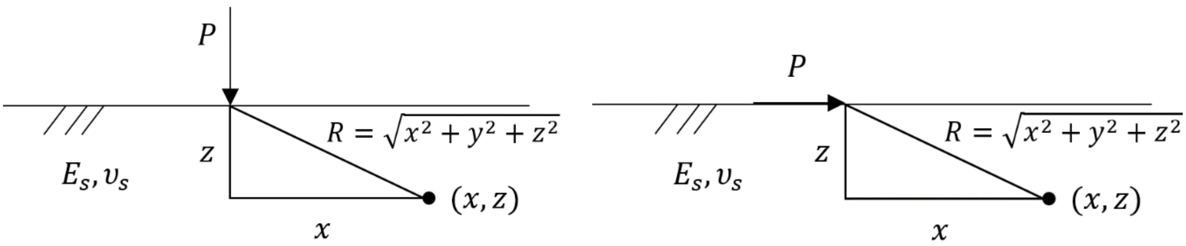


Figure 8. Boussinesq surface force problem. Vertical (left), horizontal (right).

Similarly, for a surface vertical point load:

$$\rho_z = \frac{P(1+v_s)}{2\pi E_s R}\left(2(1 - v_s) + \frac{z^2}{R^2}\right) \tag{25}$$

$$R = \sqrt{x^2 + y^2 + z^2}$$

2.2.3 Mindlin

The Mindlin solution provides a subsurface displacement generated by a subsurface point force. This is used to take into account the influence that in-pile displacements have in neighbouring piles. Since the Mindlin solution is valid for point forces and the problem is relating loads and displacements, by integrating over the area of each pile element where the load is acting, new influence coefficients are found.

Horizontal Loading

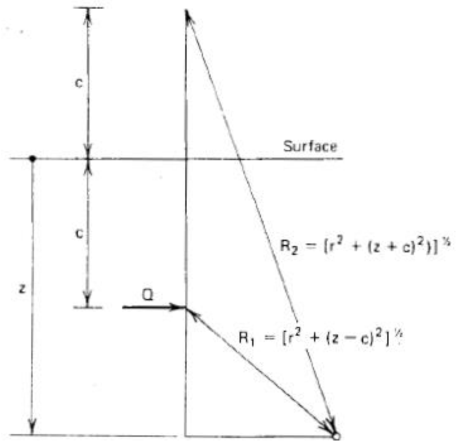


Figure 9. Horizontal point-load Mindlin problem. Extracted from Poulos and Davis (1980).

The Mindlin problem for a horizontal load Q relates:

$$\rho_x = Q \cdot \frac{1+v_s}{8E_s\pi(1-v_s)}\left(\frac{3-4v_s}{R1} + \frac{1}{R2} + \frac{x^2}{R1^3} + \frac{(3-4v_s)x^2}{R2^3} + \frac{2cz}{R2^3}\left(1 - 3\frac{x^2}{R2^2}\right) + 4(1-v_s)\frac{1-2v_s}{R2+z+c} \cdot \left(1 - \frac{x^2}{R2(R2+z+c)}\right)\right) \quad (26)$$

Where

$$R1 = \sqrt{x^2 + y^2 + (z-c)^2}$$

$$R2 = \sqrt{x^2 + y^2 + (z+c)^2}$$

$$R2 = \sqrt{x^2 + y^2 + (z+c)^2}$$

$$E_s = soil\ elastic\ modulus$$

$$v_s = soil\ Poisson\ ratio$$

Let I be defined as the so-called influence coefficient, from equation (26):

$$\rho_x = \frac{Q}{E_s} I$$

Now, the integration over the area where the load acts is only done for in-pile coefficients, i.e. for the influence that elements of one pile exert in the same pile. Let these prima coefficients be:

$$I'_{i,j} = 2 \cdot \int_{c_{inf}}^{c_{sup}} \int_{0}^{\frac{d}{2}} I \, dydc$$

As for pile to pile influences, it is accurately enough to account for a load by simply multiplying coefficients I by the area of the element (benefitting from a reduced computational time):

$$\rho_x = \frac{1}{E_s} I \cdot (dh_i)$$

Vertical Loading

Quite like the process developed in the previous section for horizontal loading, let's define:

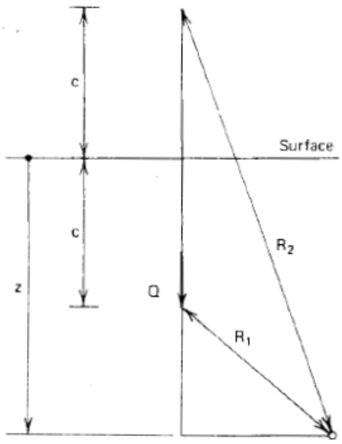


Figure 10. Vertical Mindlin problem. Adapted from Poulos and Davis (1980).

$$z = h + c$$

$$z_1 = h - c$$

$$R_1^2 = x^2 + y^2 + z_1^2$$

$$R_2^2 = x^2 + y^2 + z^2$$

Let the vertical displacement be defined in compact form, for a vertical subsurface force Q:

$$\rho_z = \frac{Q}{Es} I$$

Which is:

$$\rho_z = \mathrm{Q} \frac{(1+v)}{8Es\pi(1-v)} \cdot \left[\frac{z_1^2}{R_1^3} + \frac{3-4v}{R_1} + \frac{5-12v+8v^2}{R_2} + \frac{(3-4v)z^2 - 2cz + 2c^2}{R_2^3} + \frac{6cz^2(z-c)}{R_2^5} \right] \quad (27)$$

For the vertical problem, the pile is formed of N nodes all of length $\frac{L}{N}$. Like in the horizontal problem, vertical influence coefficients will need to be integrated over the acting area of the load. Regarding the used notation, integrated coefficients will be labelled as I'.

For elements $I'_{i,j}$, when the force is and acts on the shaft, $i, j \in [1, \dots, N]$:

$$I'_{i,j} = 2 \cdot \int_{c_{inf,j}}^{c_{sup,j}} \int_{0}^{\frac{\pi}{2}} I \, d\theta dc$$

$$c_{sup,j} = z_j - \frac{1}{2} \cdot \frac{L}{N}$$

$$c_{sup,j} = z_j + \frac{1}{2} \cdot \frac{L}{N}$$

Where z_j is the depth of each node.

The influence coefficient resulting from the force acting on the base affecting elements on the shaft of the pile is:

$$I'_{i,b} = \frac{1}{d} \int_{0}^{2\pi} \int_{0}^{\frac{d}{2}} r \cdot I \, dr d\theta$$

Finally, for the base acting upon the base itself:

$$I'_{b,b} = \frac{\pi}{4} \int_{0}^{2\pi} \int_{0}^{\frac{d}{2}} r \cdot I \, dr d\theta$$

As stated by Poulos and Davis (1980) a factor $\frac{\pi}{4}$ must be allowed to account for an effect of rigidity of the base. These authors remark that this corresponds to the ratio of the surface displacement of a rigid circle on the surface of a half-space.

Again, for influence coefficients relating two piles, these are acceptable enough if coefficients I are multiplied by the area of the element $d \cdot \frac{L}{N}$ (where d is pile diameter).

$$I_{pile\ i, pile\ j} = \frac{1}{E_s} \cdot (d \cdot \frac{L}{N})$$

2.2.4 Soil structure matrix assembly

Horizontal analysis

Finally, one can build the following system of equations of size $N_{elements}$ for a single pile problem. Based on the addition principle, volume loss (Sagaseta) and surface loading (Boussinesq) displacements need to be added to the total soil displacements.

$$\begin{Bmatrix} \rho_{x,1} \\ \rho_{x,2} \\ \dots \\ \rho_{x,N} \end{Bmatrix} = \frac{1}{E_s} \begin{bmatrix} I_{1,1}' & I_{1,2}' & \dots & I_{1,N}' \\ I_{2,1}' & I_{2,2}' & \dots & I_{2,N}' \\ \dots & \dots & \dots & \dots \\ I_{N,1}' & I_{N,2}' & \dots & I_{N,N}' \end{bmatrix} \cdot \begin{Bmatrix} p_{x,1} \\ p_{x,2} \\ \dots \\ p_{x,N} \end{Bmatrix} + \begin{Bmatrix} \rho_{x,1}^{sagaseta} \\ \rho_{x,2}^{sagaseta} \\ \dots \\ \rho_{x,N}^{sagaseta} \end{Bmatrix} + \begin{Bmatrix} \rho_{x,1}^{boussinesq} \\ \rho_{x,2}^{boussinesq} \\ \dots \\ \rho_{x,N}^{boussinesq} \end{Bmatrix} \tag{28}$$

Extending the problem to a group of piles, there are $(N_{elements}) \cdot (N_{piles})$ equations, note the use of coefficients I' or I:

$$\begin{Bmatrix} \{\rho_{x,pile1}\} \\ \{\rho_{x,pile2}\} \\ \dots \\ \{\rho_{x,pileN}\} \end{Bmatrix} = \frac{1}{E_s} \begin{bmatrix} [I']_{1,1} & [I]_{1,2} & \dots & [I]_{1,N} \\ [I]_{2,1} & [I']_{2,2} & \dots & [I]_{2,N} \\ \dots & \dots & \dots & \dots \\ [I]_{N,1} & [I]_{N,2} & \dots & [I']_{N,N} \end{bmatrix} \cdot \begin{Bmatrix} \{p_{x,pile1}\} \\ \{p_{x,pile2}\} \\ \dots \\ \{p_{x,pileN}\} \end{Bmatrix} + \begin{Bmatrix} \{\rho_{x,pile1}^{sagaseta}\} \\ \{\rho_{x,pile2}^{sagaseta}\} \\ \dots \\ \{\rho_{x,pileN}^{sagaseta}\} \end{Bmatrix} + \begin{Bmatrix} \{\rho_{x,pile1}^{boussinesq}\} \\ \{\rho_{x,pile2}^{boussinesq}\} \\ \dots \\ \{\rho_{x,pileN}^{boussinesq}\} \end{Bmatrix} \tag{29}$$

Vertical analysis

Eventually, similar to the horizontal direction, the final vertical displacements in a pile group problem may be defined as:

$$\begin{Bmatrix} \{\rho_{x,pile1}\} \\ \{\rho_{x,pile2}\} \\ \dots \\ \{\rho_{x,pileN}\} \end{Bmatrix} = \frac{1}{E_s} \begin{bmatrix} [I']_{1,1} & [I]_{1,2} & \dots & [I]_{1,N} \\ [I]_{2,1} & [I']_{2,2} & \dots & [I]_{2,N} \\ \dots & \dots & \dots & \dots \\ [I]_{N,1} & [I]_{N,2} & \dots & [I']_{N,N} \end{bmatrix} \cdot \begin{Bmatrix} \{p_{x,pile1}\} \\ \{p_{x,pile2}\} \\ \dots \\ \{p_{x,pileN}\} \end{Bmatrix} + \begin{Bmatrix} \{\rho_{x,pile1}^{sagaseta}\} \\ \{\rho_{x,pile2}^{sagaseta}\} \\ \dots \\ \{\rho_{x,pileN}^{sagaseta}\} \end{Bmatrix} + \begin{Bmatrix} \{\rho_{x,pile1}^{boussinesq}\} \\ \{\rho_{x,pile2}^{boussinesq}\} \\ \dots \\ \{\rho_{x,pileN}^{boussinesq}\} \end{Bmatrix} \tag{30}$$

Where in this case there are $(N_{elements} + 1) \cdot (N_{piles})$ equations due to the extra unknown in the base of the pile.

Always bearing in mind that there are differences between the definition of some parameters depending whether it is horizontal or vertical analysis.

2.3 Problem solving

The matrix form in which the problem has been conceived allows MATLAB to easily solve for load vector $\{p\}$ containing a number of N_{piles} load vectors, when pile and soil displacements are equated. The following structure can analogously be applied either to horizontal or vertical directions (bearing in mind the intrinsic differences between each analysis as the vectors and matrices in equations (32)and (33) have been constructed in previous sections):

$$[D]\{\delta\} + [BC] = [A]\{p\} \tag{32}$$

$$\{\rho_s\} = [I_s]\{p\} + \{\rho_{sagaseta}\} + \{\rho_{boussinesq}\} \tag{33}$$

$$\{p\} = \left([A] - \frac{1}{E_s}[D][I_s]\right)^{-1} \cdot \left([BC] + [D]\left[\{\rho_{sagaseta}\} + \{\rho_{boussinesq}\}\right]\right) \tag{31}$$

Afterwards pile deflection or settlement may be calculated, reusing equation (33).

Then, the corresponding bending moment may be found by finite differences at the Euler-Bernoulli bending theory and boundary conditions presented in equations (5), (6) and (7). By using δ as horizontal displacements:

$$M_{i,pilej} = \frac{E_p I_p}{\left(\frac{L}{N}\right)^2}\left[\delta_{i+1}^{pilej} - 2\delta_i^{pilej} + \delta_{i-1}^{pilej}\right] \tag{34}$$

Using standard notation, displacements in the X direction generate a law of M_y bending moments. Likewise, displacements in the Y direction generate a law of M_x bending moments. For this project an assumption is made in that no torsion M_z is considered, nor that it is applied as an external action.

On the other hand, the axial force, now by using δ vertical displacements, can be calculated for any element $i = 2, \dots, N$ with equations (35):

$$N_{i,pilej} = -\frac{E_p A_p}{2\frac{L}{N}}\left[\delta_{i+1}^{j} - \delta_{i-1}^{j}\right] \tag{35}$$

For the first element:

$$N_{1,pilej} = P_{pilej} - p_{z,1}^{pilej}\frac{1}{2}\cdot\frac{L}{N}\pi d$$

For the last element:

$$N_{N,pilej} = p_{z,N+1}^{pilej} \cdot \frac{\pi d^2}{4} + p_{z,N}^{pilej} \frac{1}{2} \cdot \frac{L}{N} \pi d$$

Hereafter, according to how it has been defined, all results showing positive axial force correspond to compression while negative refer to tensile stresses.

2.4 PILE CAP EFFECT

The pile cap acts as a rigid plate (rigid pile cap) that constrains horizontal movements, by making pile head displacements equal, i.e. the pile heads may present some head displacements albeit all the same. Differential displacements are not allowed, that is the purpose of a rigid plate, or rafted foundation. In a similar way, vertical movements at the head are imposed to be in a plane, i.e. heads may present different settlements as long as they are coplanar. In order to achieve this, Sauter (2012) proposed to modify the force acting on each pile, as if the pile group distributed the total load among its piles (which is what in reality occurs). The procedure is iterative, where it is applicable to any time step *j* (as in reference to tunnel advance steps).

Horizontal analysis

1. Calculate the mean value of the displacements at time *j* at the piles' head:

$$\delta_{x,mean}^{head,j}$$

2. Change the new pile external applied force ($P_{x,i}^{j}$) with the following weighted function, for pile i and actual head displacements at current time step being $\delta_{x,i}^{head,j}$:

$$P_{x,i}^{j} = P_{x,i}^{j-1} + \left(\delta_{x,mean}^{head,j} - \delta_{x,i}^{head,j}\right) \cdot \frac{\left|\sum_{i=1}^{N_{piles}} P_{x,i}^{j-1}\right|}{N_{piles}} \cdot 100 \qquad (36)$$

3. Recalculate piles deflection and stresses, with the procedure developed in section 2.3 and the new forces having changed the matrix $[BC]$.
4. Iterate whether the following conditions is not fulfilled, for every pile head i:

$$-5 \cdot 10^{-6} < \left[max\left(\delta_{x,i}^{head,j}\right) \; - \; \min\left(\delta_{x,i}^{head,j}\right)\right] < 5 \cdot 10^{-6}$$

$$i\epsilon\left\{1, \dots, N_{piles}\right\};\; j\epsilon\mathbb{N}$$

For the horizontal Y direction the procedure is similarly applied.

Vertical analysis

1. Calculate the mean plane using a least-square approximation and the theoretical head settlement:

$$\delta_{z,mean}^{head}$$

2. Compute the difference between the calculated pile heads and the mean plane pile heads:

$$\delta_{z,dif,i}^{head} = \delta_{z,i}^{head} - \delta_{z,mean,i}^{head}$$

3. Average the pile heads differences:

$$\delta_{z,dif,mean}^{head}$$

4. Change new pile external applied force with this weighted function:

$$P_{z,i}^{j} = P_{z,i}^{j-1} + \left(\delta_{z,dif,mean}^{head,j} - \delta_{z,dif,i}^{head,j}\right) \cdot \frac{\left|\sum_{i=1}^{N_{piles}} P_{x,i}^{j-1}\right|}{N_{piles}} \cdot 100 \tag{37}$$

5. Recalculate piles settlement and stresses, with the procedure developed in section 2.3 and the new forces having changed the matrix $[BC]$.
6. Iterate whether the following condition is not fulfilled (limits in meters), for every pile head i:

$$\left[-5 \cdot 10^{-5} < min\left(\delta_{z,dif,i}^{head}\right)\right] \vee \left[\max\left(\delta_{z,dif,i}^{head}\right) < 5 \cdot 10^{-5}\right]$$

$$i\epsilon\{1, \dots, N_{piles}\};\ j\epsilon\mathbb{N}$$

As it can be appreciated that: for the horizontal analysis the head displacements are compared to the mean head displacements, whereas the vertical procedure compares each head settlement to the corresponding theoretical settlement in a mean plane. With this is made clear that the horizontal head displacements are constraint to be all equal, in contrast with the vertical procedure that allows differential settlement as long as it is in a plane.

2.5 TUNNEL VOLUME LOSS

González and Sagaseta (2001) thoroughly described the crossectional deformation undergone by a tunnel (see Figure 11). In earlier works, Lee et al. (1992) defined the gap parameter, which is nothing but the maximum radial distance from the final tunnel cross section to the original, when it undergoes ground loss and vertical movement (marked as $d_0 - d_1$ in Figure 12), in other words, the maximum measurable settlement at the tunnel crown. From Figure 11, the gap parameter would imply the consideration of the sum of ground loss and vertical movement.

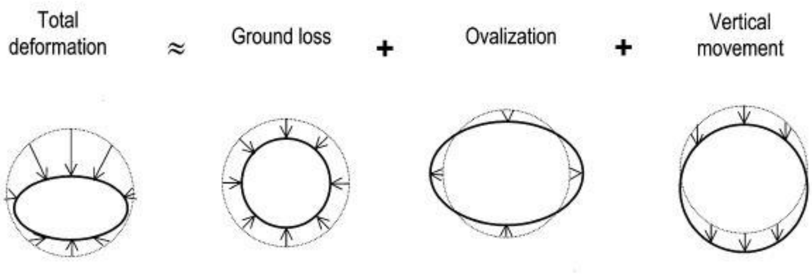


Figure 11. Adapted from González and Sagaseta (2001). Components of deformation of the tunnel.

To show insight about the sources of deformation, the Lee et al. (1992) gap parameter definition is:

$$2r_0 = 2r_1 + g$$

$$g = G_p + U_{3D}^{*} + \omega$$

Where:

G_p	Physical gap. The difference between the maximum outside diameter of the tunneling machine and the outside diameter of the lining for a circular tunnel.
U_{3D}^{*}	3D elastoplastic deformation into the tunnel face.
ω	Workmanship factor.

For this project, ovalisation as it is described by González and Sagaseta (2001) will be neglected, therefore, the ground loss and vertical movement contributions are the considered types, just like Lee et al. (1992) in the gap parameter. From here, equations below were worked out to discretise a tunnel deformation in various points. Based only on the initial and final geometries (see Figure 12), let's define the following as the areal loss ratio:

$$\varepsilon_0 = \frac{A_{initial} - A_{final}}{A_{final}} = \frac{\pi r_0^2 - \pi r_1^2}{\pi r_1^2} = \frac{r_0^2}{r_1^2} - 1 \qquad (39)$$

ε_0 is nothing but the relationship between the final and initial tunnel cross section areas. In general, this ratio remains around 1% strongly dependant on machine shield technology, lining characteristics and workmanship skills (Lee et al., 1992). With that being said, the MATLAB code developed for this project only asks to input the ε_0 ratio and the tunnel diameter which is $2r_1$. Then it computes r_0 from equation (39), rearranging terms:

$$r_0 = \left(r_1^2 \cdot (\varepsilon_0 + 1)\right)^{1/2} \qquad (38)$$

As stated earlier, the tunnel cross section undergoing ground loss and vertical settlement results in the cross section depicted in blue in Figure 12. The perimetral areal loss will be discretised into various portions (depending on the required number of elements). If one sets a local polar system of coordinates $(Y' - Z')$, datum on the biggest circle's centre (black cross), equations describing the two circumferences that represent the original and the final tunnel cross sections may be expressed as:

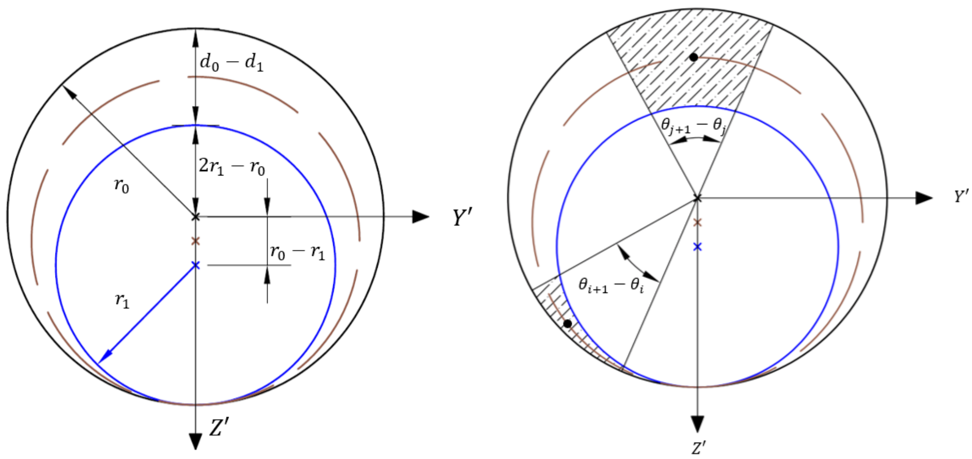


Figure 12. Tunnel volume loss cross section representation (exagerated), local coordinate system.

$$r_{initial} = r_0$$

$$r_{final}(\theta) = r_{c,final}\cos(\theta - \varphi) + \sqrt{a^2 - r_{c,final}\sin(\theta - \varphi)} \qquad (40)$$

Where a is the final circle's radius, φ is 270º and:

$$r_{c,final} = r_1 - r_0$$

Now, each of the elements that define a portion shaded in Figure 12, have varying area. The area for each element is calculated as:

$$(\Delta A)_j = A_{initial}^{\theta_{i+1}-\theta_i} - A_{final}^{\theta_{i+1}-\theta_i}$$

$$A_{initial}^{\theta_{i+1}-\theta_i} = pi \cdot \frac{r_{initial}^2}{n_{disc}}$$

$$A_{final}^{\theta_{i+1}-\theta_i} = \int_{\theta_i}^{\theta_{i+1}} r_{final}(\theta)\frac{1}{2}r_{final}(\theta)d\theta \qquad (41)$$

For:

$$i\epsilon[0, n_{discr}];\ j\epsilon[1, n_{discr}];\ \theta_i\epsilon[0, \frac{2pi}{n_{discr}}, \dots, 2pi]$$

Once the areal loss in known for every discretisation, these are associated to corresponding sink points located in the middle circumference, just as marked in Figure 12. With this, proportionality is already taken into account at each point.

To recap, for any problem the variables required for the MATLAB code are ε_0, r_0 and the number of points. For the calculations developed further on (sections 5.2, and 6), these are specified.

An example below, similar to the sketch in Figure 12, shows the discretisation in 10 points, for a $\varepsilon_0 = 1\%$ tunnel. Since the deformation is normally almost imperceptible the zoomed-in plots show the discretisation sink points in between the two cross sections. Note that the bottom point coincides with both curves, whereas the maximum difference is at the crown, that is a result of the vertical movement.

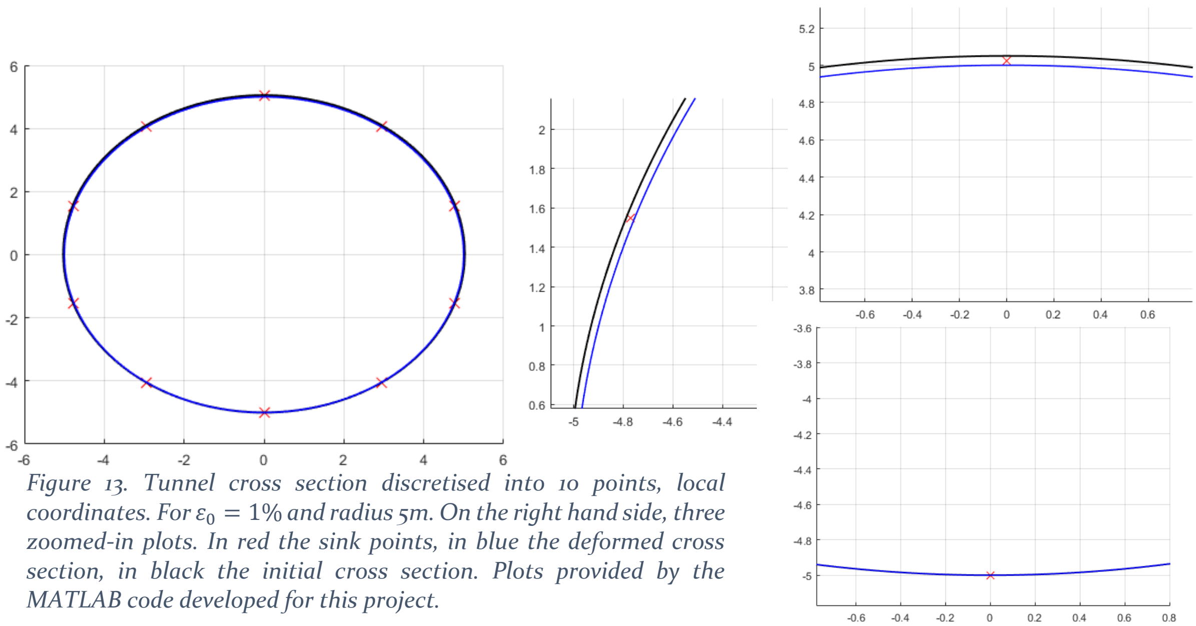


Figure 13. Tunnel cross section discretised into 10 points, local coordinates. For $\varepsilon_0 = 1\%$ and radius 5m. On the right hand side, three zoomed-in plots. In red the sink points, in blue the deformed cross section, in black the initial cross section. Plots provided by the MATLAB code developed for this project.

2.6 ADVANCING TUNNEL PROBLEM

The problem of an advancing tunnel is solved based on the development in sections 2.1, 2.2, 2.3, 2.4 and 2.5.

Firstly, the tunnel will be discretised into various sections of a certain length (Figure 14). It is assumed that along the length of the tunnel the areal loss remains the same. As described in section 2.6, the tunnel cross section areal loss is simulated by a number of sink points. To account for volume, each areal loss at each point is multiplied by the corresponding length of the section, as seen in Figure 14 it is $\frac{L_{tun}}{N_{disc}}$, where L_{tun} is the input length of the tunnel in the code.

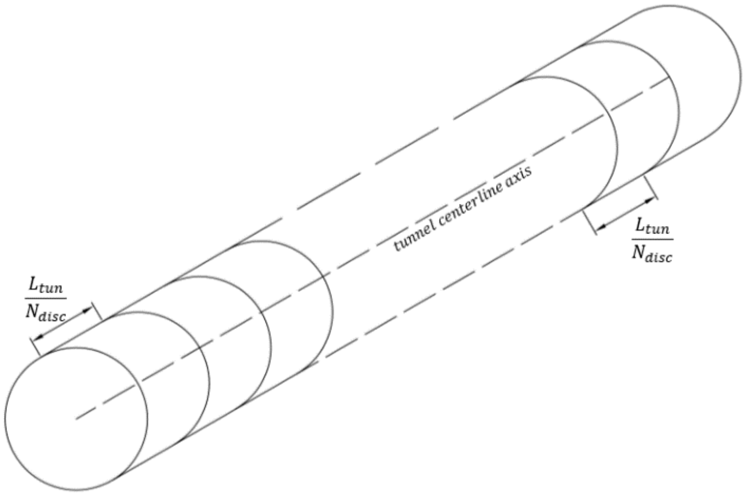


Figure 14. Sketch of the tunnel discretised into several segments, each segment is discretised into several points.

Once the location and volume loss of each point in known, soil movements at the piles are computed for each advancing tunnel discretisation. Therefore, if N_{disc} is the number of tunnel discretisations, there appear N_{disc} column vectors of size $N_{elem} \cdot N_{piles}$ that are stored in a matrix.

For any time step j (tunnel advance) the final soil displacements are the cumulative soil displacements generated by the advances 1 to j. For instance, in the X direction:

$$\begin{Bmatrix} \left\{\rho_{x,pile1}^{sagaseta}\right\} \\ \left\{\rho_{x,pile2}^{sagaseta}\right\} \\ \dots \\ \left\{\rho_{x,pileN}^{sagaseta}\right\} \end{Bmatrix}_{cumulative,j} = \sum_{i=1}^{j} \begin{Bmatrix} \left\{\rho_{x,pile1}^{sagaseta}\right\} \\ \left\{\rho_{x,pile2}^{sagaseta}\right\} \\ \dots \\ \left\{\rho_{x,pileN}^{sagaseta}\right\} \end{Bmatrix}_{i} \tag{42}$$

This may be generalised to any coordinate.

Then, for the said j tunnel advance, the problem is solved using *cumulative* soil displacements described above and the standard solutions presented in previous sections. Equation (31) is then, at time step j:

$$\{p\}_j = \left([A] - \frac{1}{E_s}[D][I_s]\right)^{-1} \cdot \left([BC] + [D]\left[\{\rho_{sagaseta}\}_{cumulative,j} + \{\rho_{boussinesq}\}\right]\right)$$

Below, a sketch of the problem geometry in MATLAB, in this example, containing 10 points per discretised tunnel element. The number of points is deemed as a choice for the user of the code, however, it has been found that calculations for more than 10 points provide similar results.

The length of the tunnel is a choice of the user of the code, where they have to find a reasonable computational time in relation to a representative simulation.

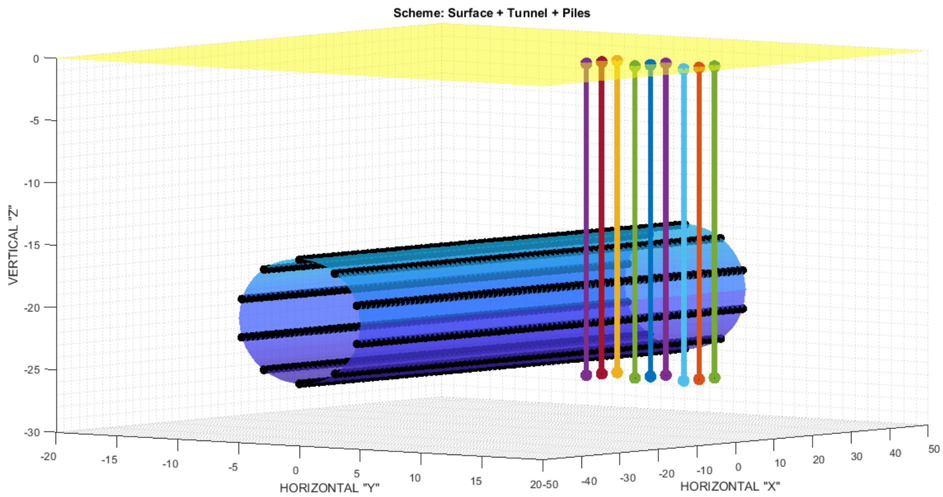


Figure 15. Sample 3D scheme output provided by the code. Black nodes represent the tunnel volume loss discretisation points.

2.7 COMPENSATION GROUTING

Compensation grouting is a technique used to compensate or reverse displacements on a structure. Injection of material expands so that its induced movements counteract the effects of the source of unwanted displacements. In this case, the compensation grouting nested in the code is built so that the user specifies its characteristics, it can be injected in a strip or more than one and vary the total expansive volume, which will be negative to account for swelling.

Each strip is formed by N_{disc} points that add up to the specified total expansion volume. The problem is similarly solved to what has been described in previous sections. The new volume expansion creates soil movements along the piles, both volume loss and swelling are added, thus these final soil displacements are used to solve the problem.

Equation (42) becomes then, at time step j:

$$\begin{Bmatrix} \left\{\rho_{x,pile1}^{sagaseta}\right\} \\ \left\{\rho_{x,pile2}^{sagaseta}\right\} \\ \cdots \\ \left\{\rho_{x,pileN}^{sagaseta}\right\} \end{Bmatrix}_{cumulative,j} = \sum_{i=1}^{j} \left(\begin{Bmatrix} \left\{\rho_{x,pile1}^{sagaseta}\right\} \\ \left\{\rho_{x,pile2}^{sagaseta}\right\} \\ \cdots \\ \left\{\rho_{x,pileN}^{sagaseta}\right\} \end{Bmatrix}_{i}^{vol.\ loss} + \begin{Bmatrix} \left\{\rho_{x,pile1}^{sagaseta}\right\} \\ \left\{\rho_{x,pile2}^{sagaseta}\right\} \\ \cdots \\ \left\{\rho_{x,pileN}^{sagaseta}\right\} \end{Bmatrix}_{i}^{comp.\ grout} \right)$$

Finally, the problem is again solved with Equation (31), although it can be solved for a certain time step as if each point compensates a tunnel advance step, it is more relevant to find the final solution, for the last time step $j = N_{disc}$:

$$\{p\}_j = \left([A] - \frac{1}{E_s}[D][I_s]\right)^{-1} \cdot \left([BC] + [D]\left[\{\rho_{sagaseta}\}_{cumulative,j} + \{\rho_{boussinesq}\}\right]\right)$$

3 MATLAB CODE

Section 3 provides an insight into the structure of the code. Having started from scratch, the code has been inspired by the 2012 Master's Thesis by Sauter, where the author studied the effects of an expanding layer of soil underneath a pile group in Pont del Candí (Sauter, 2012). It must be said that although the main goal of this project was to study the tunnel-pile interaction, a collateral consequence have been that of learning how to create and manage a large-sized code.

After a first stage of input data, the piles' and tunnel geometries are created. Then, a MATLAB function computes the soil movements induced by the points discretising the tunnel, as well as displacements due to surface loads being computed. In terms of computational cost, this is the most expensive part, mostly due to the repetitive task done for $N = (N_{points} \cdot N_{disc}) \cdot (N_{elements} \cdot N_{piles})$ times of computing the integral involving complete elliptic functions in equations (20) and (22). Where N_{points} is the number of points discretising each tunnel advance; N_{disc} the number of discretisations along the tunnel length; $N_{elements}$ the number of elements in each pile; N_{piles} the number of piles.

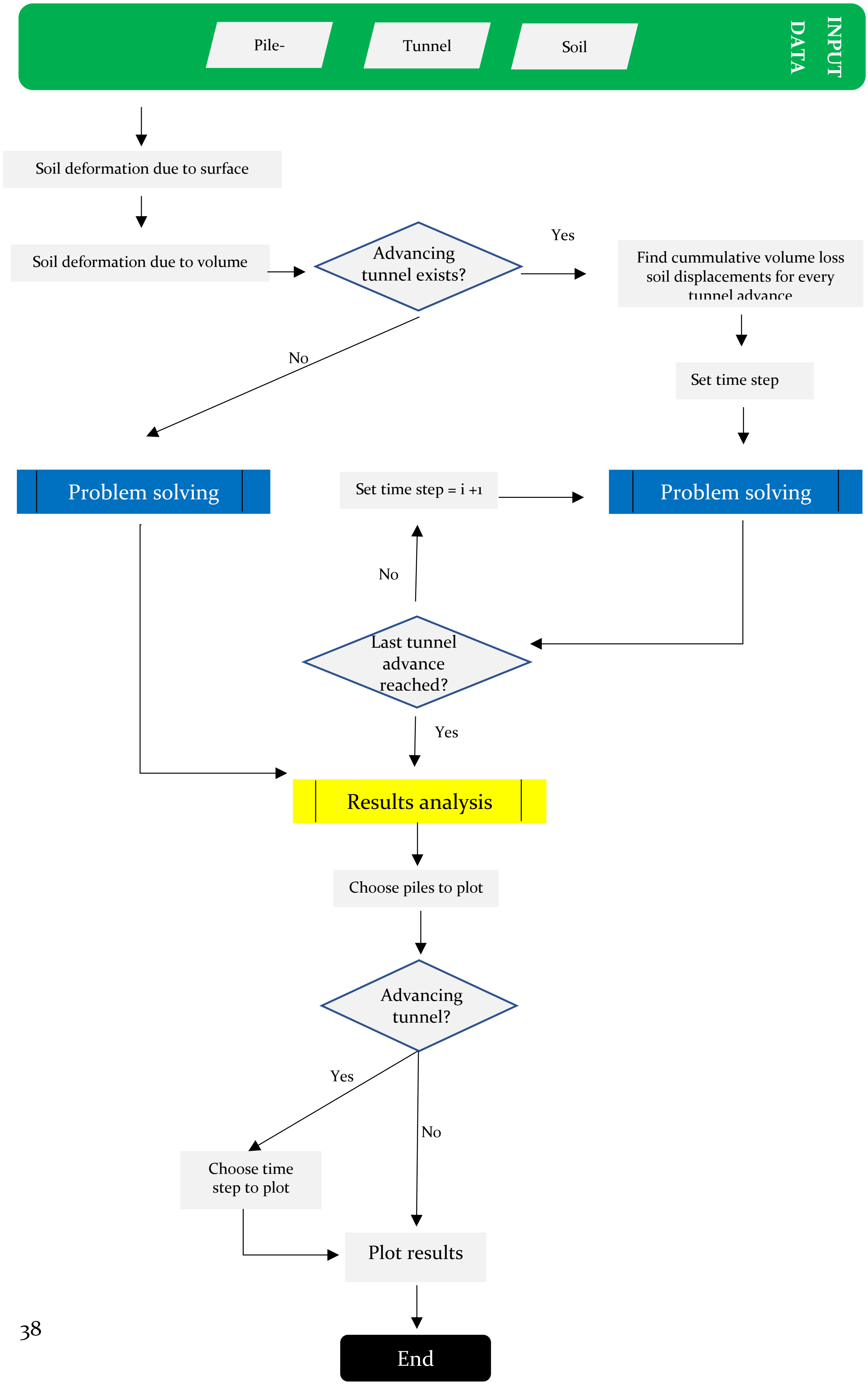
Computational time follows a linear relationship with an increase of elements (either in the tunnel or in the piles). This means that a 20 elements pile discretisation is half as fast as a 40 elements discretisation.

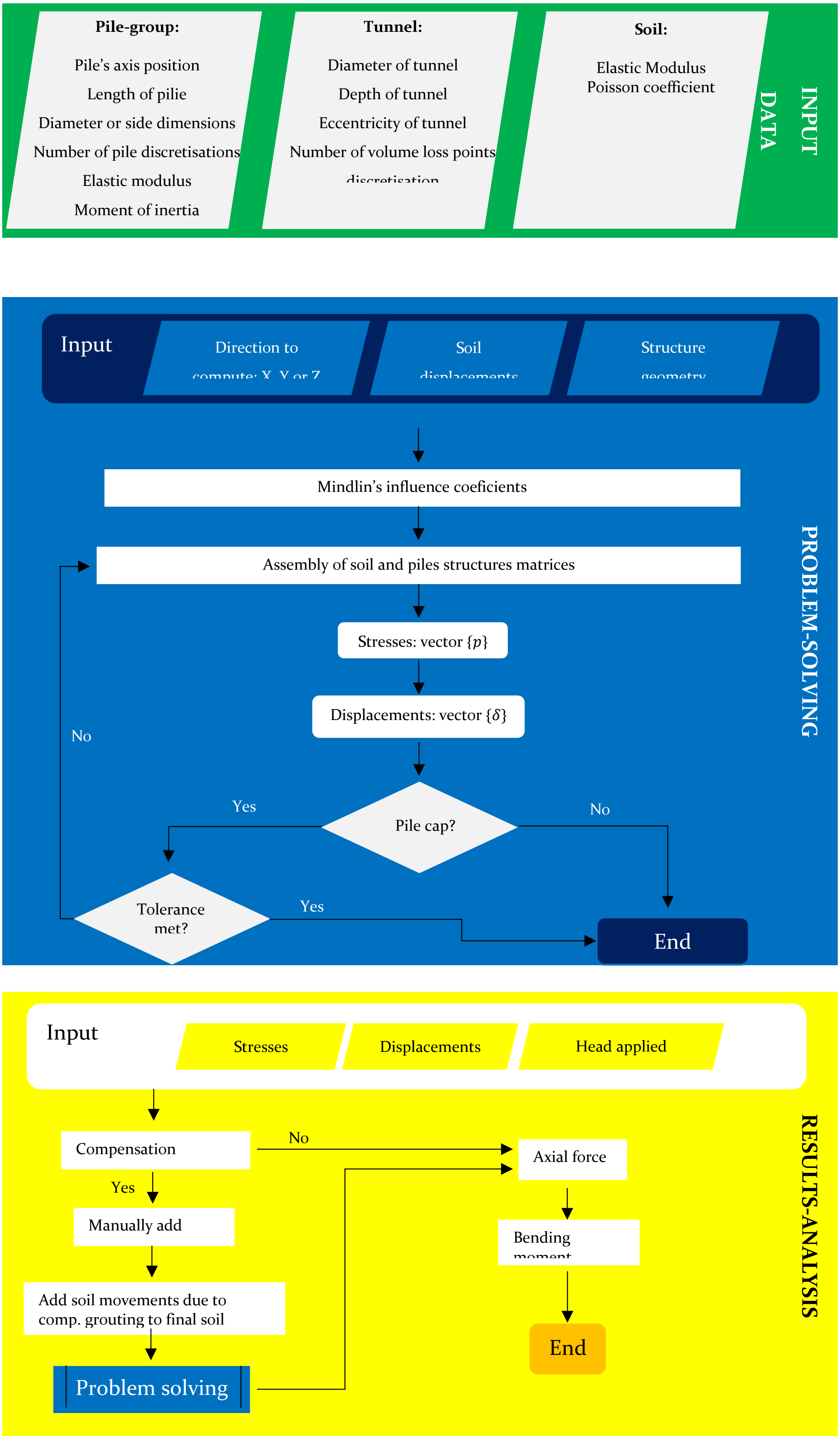
Once the vectors of soil displacements are known for each tunnel advance, they are sent to the horizontal and vertical analyses. In each analysis, the system of equations as described in sections 2.3 and earlier, are assembled. Working out the Mindlin's coefficients for the in-pile interaction absorbs most of this subprocess computational time, mainly due to the required integration (as developed in section 2.3) for both horizontal and vertical analyses. Nevertheless, it is far less important than the integral for the Sagaseta soil movements involving elliptic functions, moreover, assembling the rest of matrices and solving the actual system of equations is quickly done.

The results analysis (calculating bending moments, etc.) and the final plots entail no significant computational effort and this is done at the end of the code.

In the following section a schematic workflow diagram of the code is presented, two predefined subprocesses are further explained afterwards to show insight in what is called Problem-solving and Results-analysis. The required input data is also specified.

3.1 Code flow chart





Pile-group:
Pile's axis position
Length of pilie
Diameter or side dimensions
Number of pile discretisations
Elastic modulus
Moment of inertia
Tunnel:
Diameter of tunnel
Depth of tunnel
Eccentricity of tunnel
Number of volume loss points
Soil:
Elastic Modulus
Poisson coefficient
INPUT DATA
Input
Direction to
Soil
Structure
Mindlin's influence coeficients
Assembly of soil and piles structures matrices
Stresses: vector {p}
Displacements: vector {δ}
No
Yes
Pile cap?
No
Tolerance met?
Yes
End
PROBLEM-SOLVING
Input
Stresses
Displacements
Head applied
Compensation
No
Axial force
Yes
Manually add
Bending
Add soil movements due to comp. grouting to final soil
End
Problem solving
RESULTS-ANALYSIS

4 CODE VERIFICATION

4.1 STANDARDISED SOLUTIONS

Mattes and Poulos (1969), and David and Poulos (1974) presented the following standardised solutions. The results show the good response of the developed code.

Parameters found in subsequent figures are: ν_S soil Poisson ratio; H is the applied horizontal force at the pile head, L and d pile length and diameter, respectively, and p the horizontal or vertical pressure acting along the pile shaft.

4.1.1 Vertical

In this case, different soil stiffnesses are compared. Poisson's ratio has a small effect as it is implied from the next figure. For the test, the soil's elastic modulus is related to the pile's modulus as:

$$E_S = \frac{E_p}{K}$$

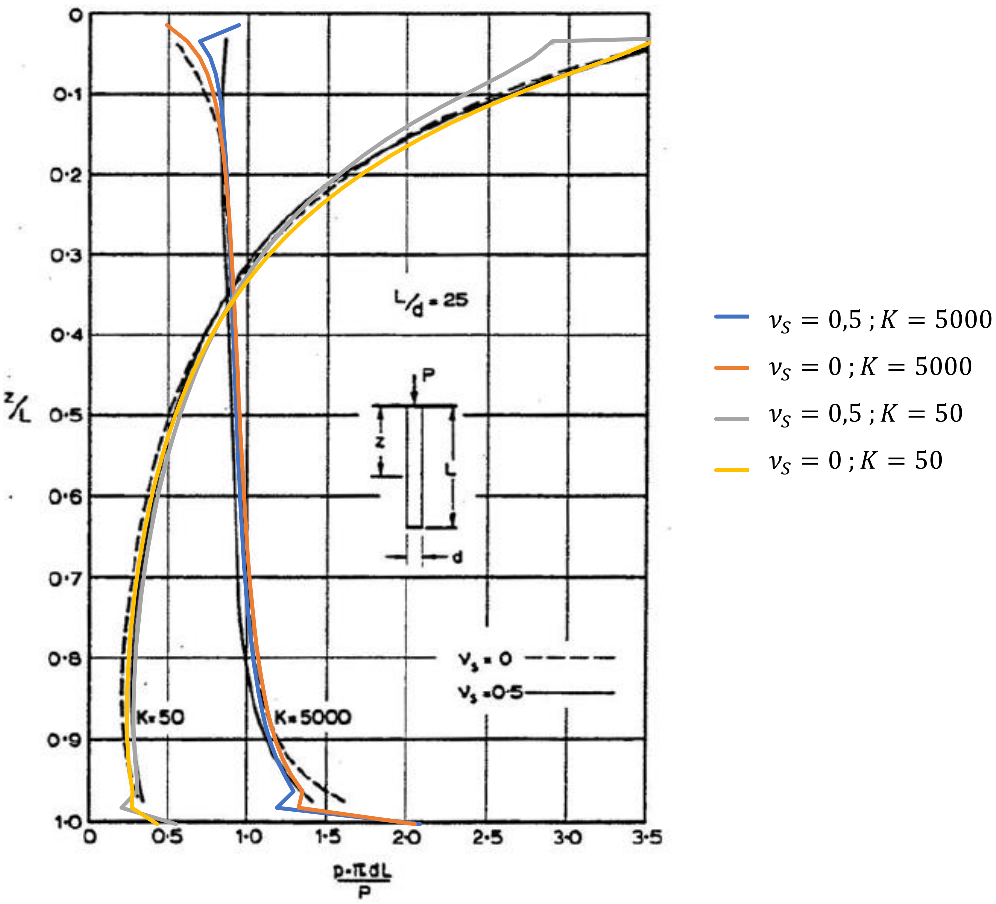


Figure 16. Adapted from Mattes and Poulos (1969), in colour present project results.

4.1.2 Horizontal load free head pile

For subsequent figures regarding horizontal behaviour, the soil elastic modulus is defined as:

$$E_S = \frac{E_p I_p}{L^4\ K_R}$$

Where K_R relates pile and soil stiffnesses.

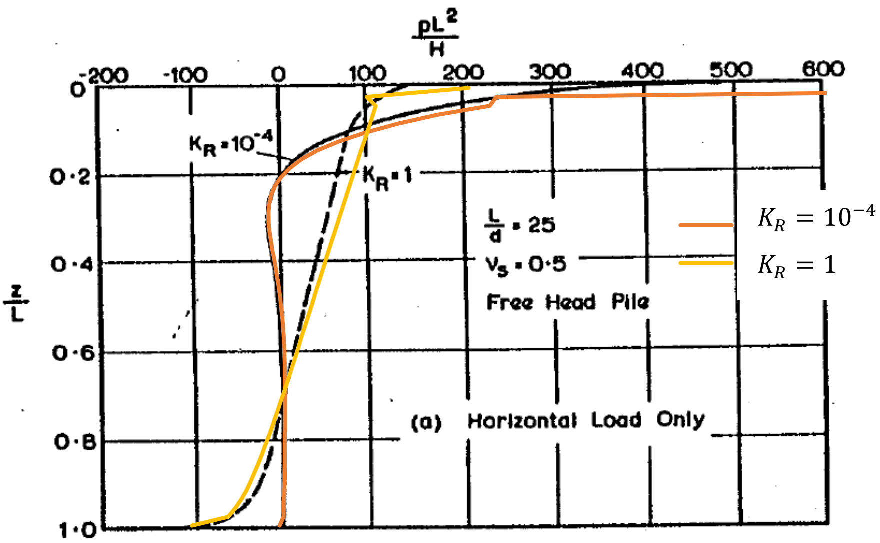


Figure 17. Adapted from Davis and Poulos (1974), in colour present project results.

The resulting bending moment for a pile subjected to a horizontal force H at its head.

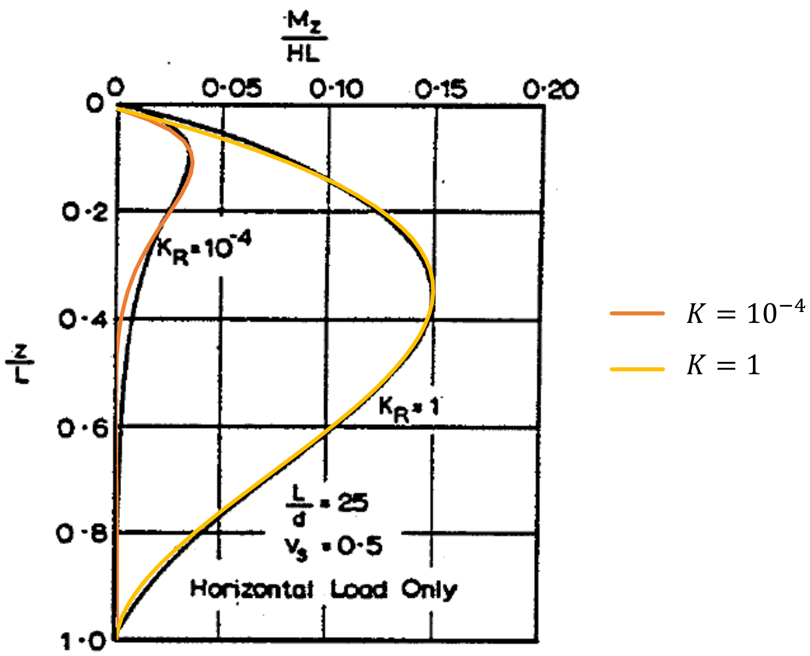


Figure 18. Adapted from Davis and Poulos (1974), in colour present project results.

4.1.3 Horizontal load fixed head pile

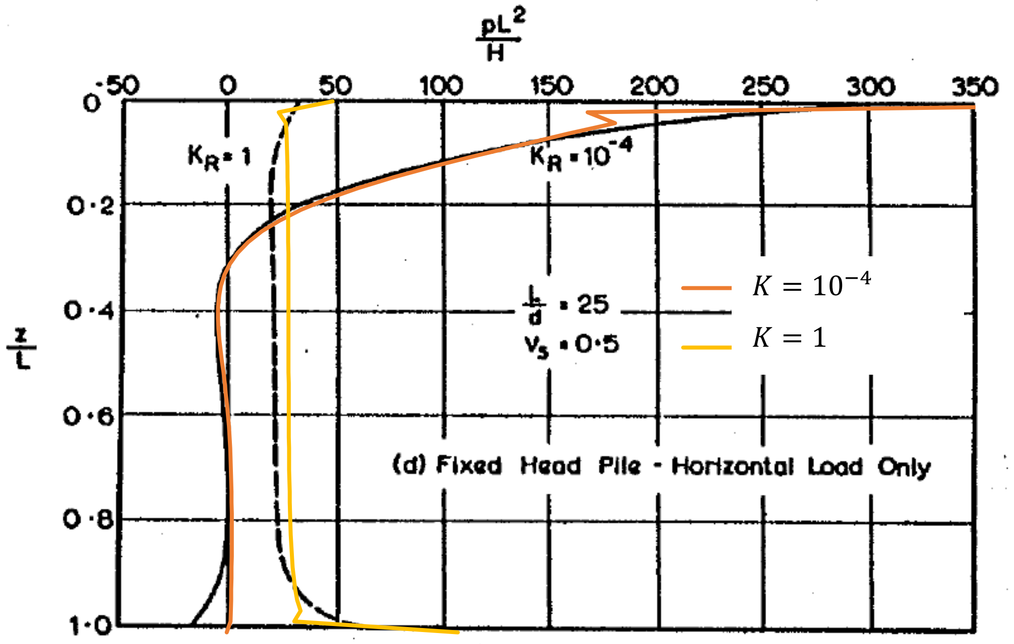


Figure 19. Adapted from Davis and Poulos (1974), in colour present project results.

The resulting bending moment:

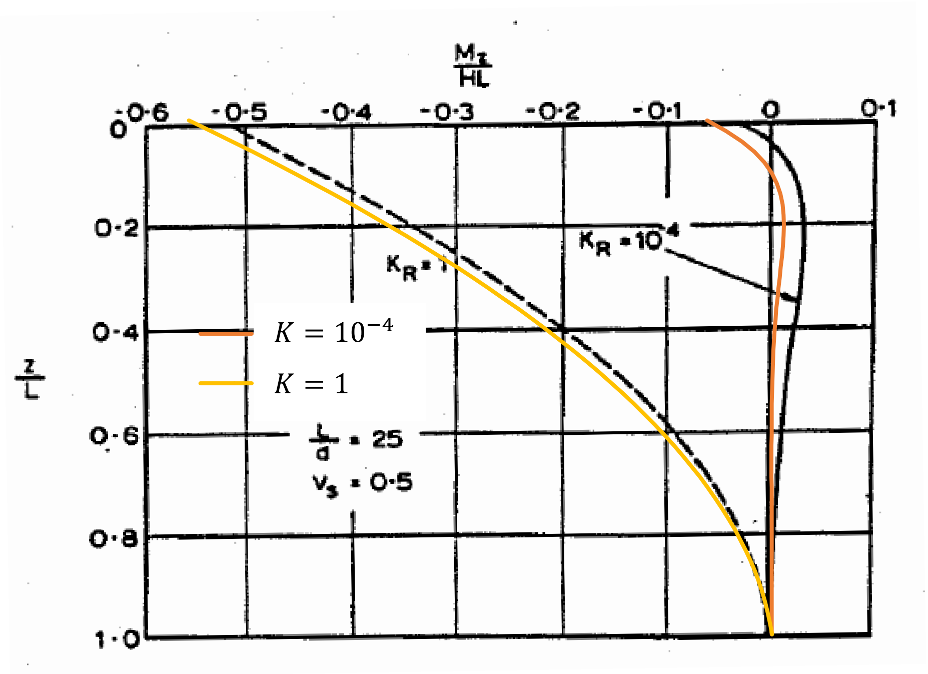


Figure 20. Adapted from Davis and Poulos (1974), in colour present project results.

4.2 PLAXIS GROUP PILE TEST

Continuing with the validation of this project code, a 9 pile group with applied external forces will be modelled in a commercial geotechnical software, PLAXIS. Two different analysis will be carried out, firstly, one with external horizontal loads applied at the top, and secondly, a vertical analysis with external vertical loads, so as to see the responses in each direction.

The most relevant information used in this section is gathered in Table 1.

Table 1. Used soil and piles parameters.

Piles	***Value***
Length (m)	20
Diameter (m)	1.65
Spacing (m)	4
Young modulus (kN/m²)	$3 \cdot 10^7$
Number of piles	9
Vertical load (kN/pile)	3000
Horizontal load (kN/pile)	3000
Soil	***Value***
Elastic modulus (kN/m²)	100000
Poisson ratio	0,5

To configure the problem in PLAXIS, a borehole is defined with the corresponding properties in Table 1. The size of the soil mesh is a cube that spans 100 m wide and 50 m in depth. The software requires to input some soil properties, as defined in Table 1. As for structures, the only ones are the embedded beams used to simulate the piles.

The code of this project will model the group of piles connected via a rigid pile cap, this means that it constraints horizontal displacements to be equal for every pile head. Moreover the vertical displacements at the pile heads must be coplanar. To account for this in PLAXIS, a squared surface is generated containing every pile cap, this can later be transformed into a plate with concrete properties similar to those of the piles. Finally, the piles must have the rigid cap option selected. It is important to highlight that PLAXIS does not calculate for any Poisson ratio of 0,5, and recommends inserting ratio values below 0,4, this will be relevant in later sections.

4.2.1 Horizontal analysis

Figure 22 shows piles displacements calculated by PLAXIS (dashed) and by the model developed for this project (continuous line). It can be easily understood from Figure 22 that both models well capture the effect of the pile cap. All the pile heads present, at the top, an exactly equal horizontal displacement, naturally differing downwards. There is good correspondence between the two models.

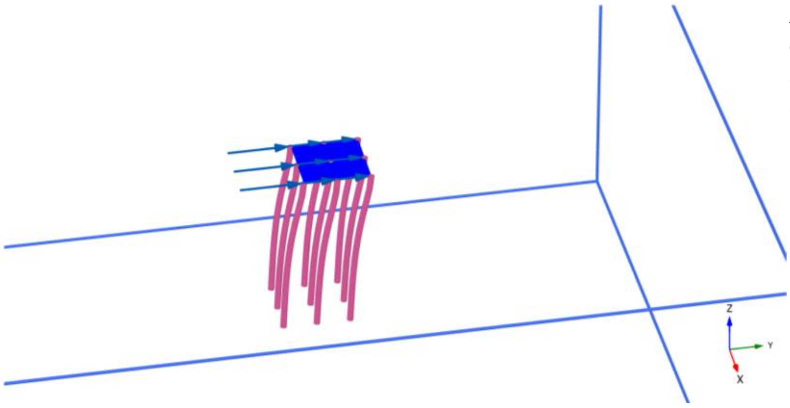


Figure 21. PLAXIS 3D representation of a piled group with applied horizontal load at the head (soil mesh not visible).

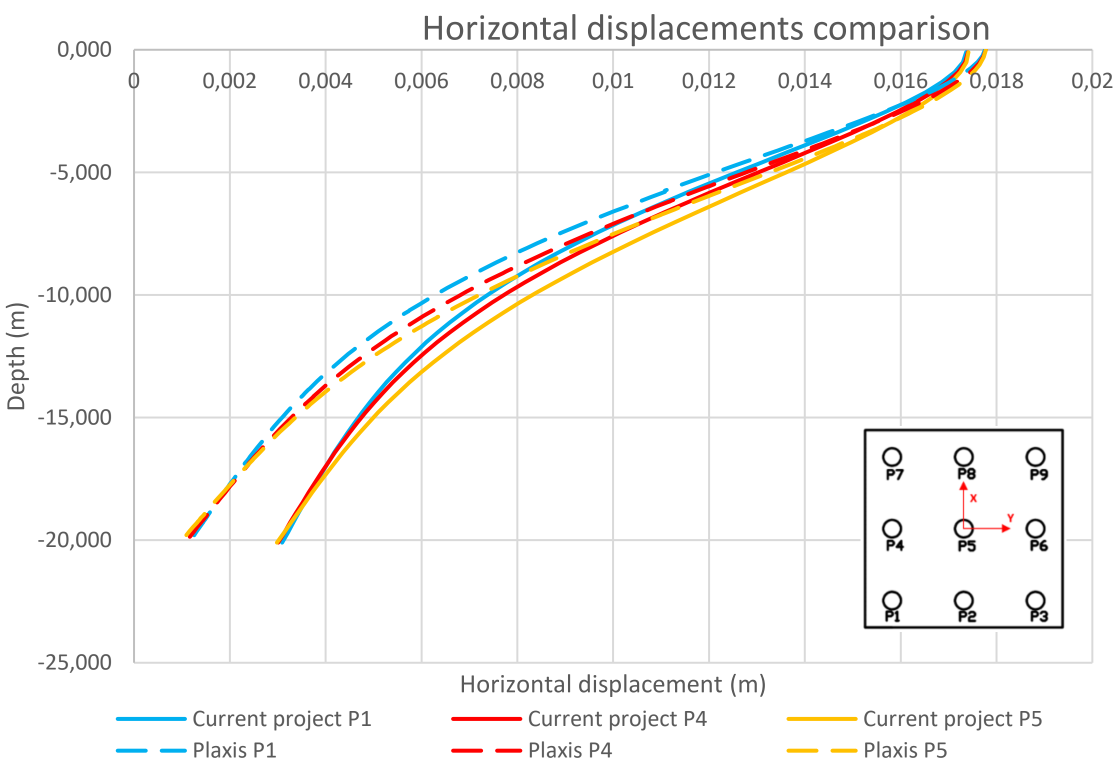


Figure 22. Results by PLAXIS (dashed) and this project (continuous) for horizontal displacement at different piles due to a 2000 kN per pile vertical load.

4.2.2 Vertical analysis

The analysis is done with the specified values above. Below, a 3D representation extracted from PLAXIS. Results shown in this section represent settlements as negative, unlike for the rest of the figures over the project, as stated in section 2.3. Likewise in the horizontal direction, both models capture the effect of the cap, in this case all the pile heads present the same settlement, hence inferring that they are in an horizontal plane. There is acceptable correspondence.

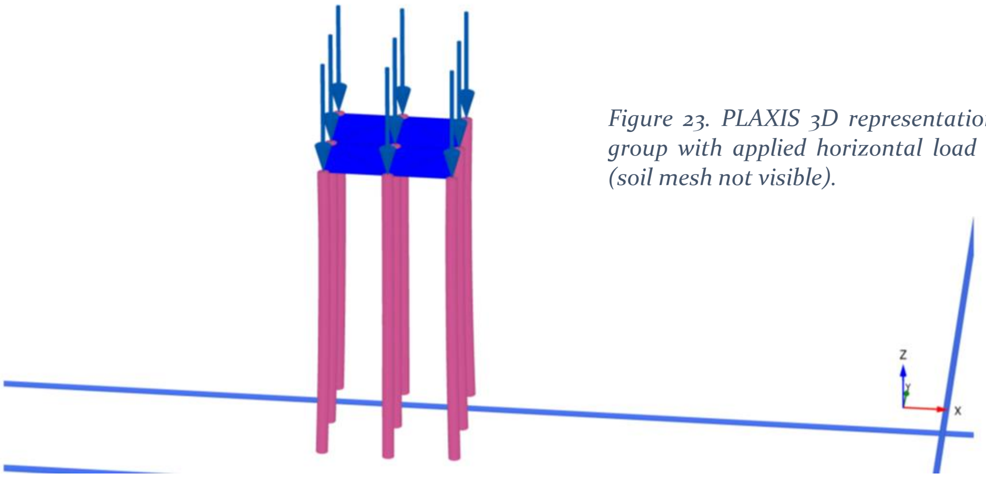


Figure 23. PLAXIS 3D representation of a piled group with applied horizontal load at the head (soil mesh not visible).

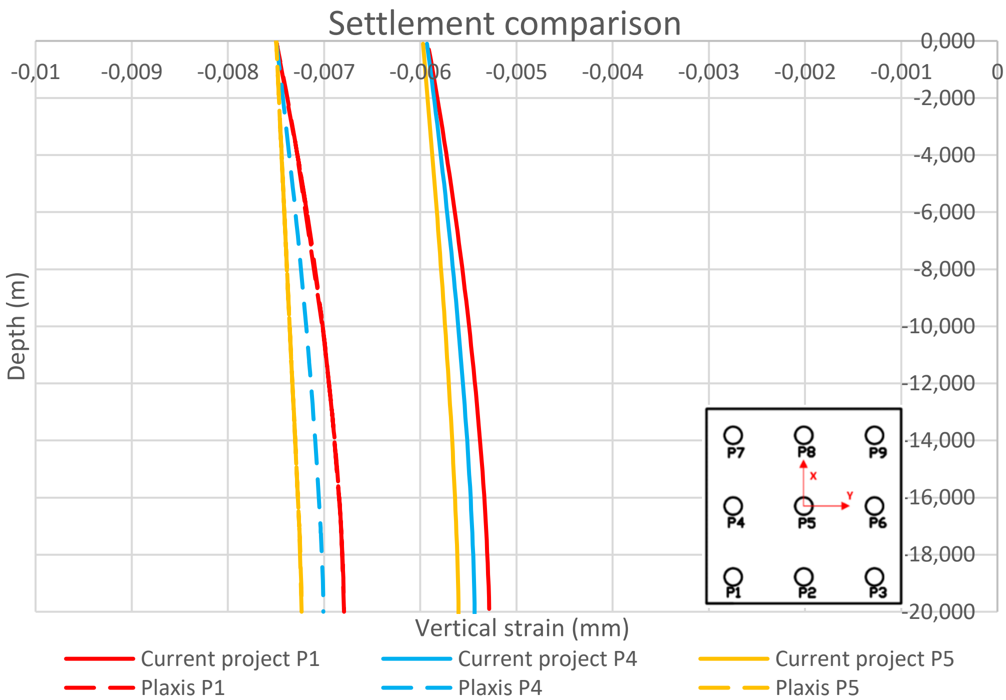


Figure 24. Results by PLAXIS (dashed) and this project (continuous) for vertical displacement at different piles due to a vertical 2000 kN per pile.

5 SENSITIVITY ANALYSIS

In the following section, once the code is verified, it is time to study problems of interest with soil volume losses. Firstly, a sensitivity analysis will be carried out for different pile-group configurations to see their behaviour upon a point volume loss. Secondly, an advancing tunnel will be simulated in order to study the tunnel-piles interaction as the tunnel progresses, and how it varies for different areal loss ratios.

5.1 PILE-GROUP RESPONSE TO A POINT SOIL VOLUME LOSS

5.1.1 Influence of pile length and diameter

For this section, various tests are carried out so as to compare a 9 pile group for varying diameters and lengths undergoing the effects of a nearby point volume loss, Figure 25 depicts the geometry of problem.

Combinations are for 15, 20, 25 and 30 m pile lengths; 0,5, 1 and 1,5 m diameters. For all the cases, the parameters gathered in the following table remain constant.

Table 2. Piles, soil and volume loss parameters used in this section.

Piles	***Value***
Number of piles	9
Pile cap	Rigid
Young modulus (kN/m^2)	$3 \cdot 10^7$
Spacing (m)	4
Vertical load (kN/pile)	0
Horizontal load (kN/pile)	0
Soil	***Value***
Elastic modulus (kN/m^2)	100000
Poisson ratio	0,5
Point volume loss	***Value***
Volume (m^3)	0,5

Since the only source of distress is the point volume loss, the nearest piles are likely to be the most affected, i.e. 1, 4 and 7 (for pile labelling see Figure 25 and thumbnails next to plots). For the sake of simplicity, figures in this section show results for piles 4 and 7. An assumption can be formulated beforehand, that the farther the piles, the less compromised they will be, therefore the plots show two of the closest piles, hence the most affected once. This can be corroborated with the following plots.

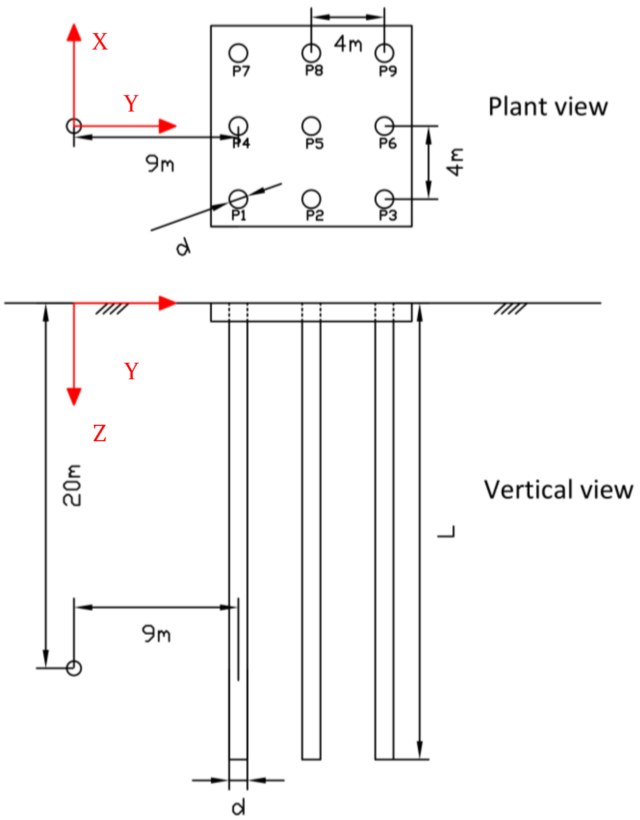


Figure 25. Plant view of a 9 piled-raft foundation affected by a nearby point volume loss.

X direction

Right to each horizontal displacements plot one can see the corresponding bending moment. As one could recognise beforehand, since the sink point is located at $x = 0$, horizontal X displacements for piles 4, 5 and 6 should be zero, piles 7, 8 and 9 should be slightly deflected towards X positive direction and 1, 2 and 3 in the opposite direction.

A remark is made in how piles shorter than the depth of the point loss present the slightest of the displacements, whereas the 20 m length pile (which coincides with the depth of the sink point) is the most affected.

For piles larger than 20 m, the broadest the diameters, the least deflection they take due to its higher inertia, at the expenses of a larger bending moment.

Notice how the only visible deflected profiles correspond to pile 7, pile 4 behaves as expected and the results are no movement at all, hence no bending moment, since it is horizontally aligned with the sink point.

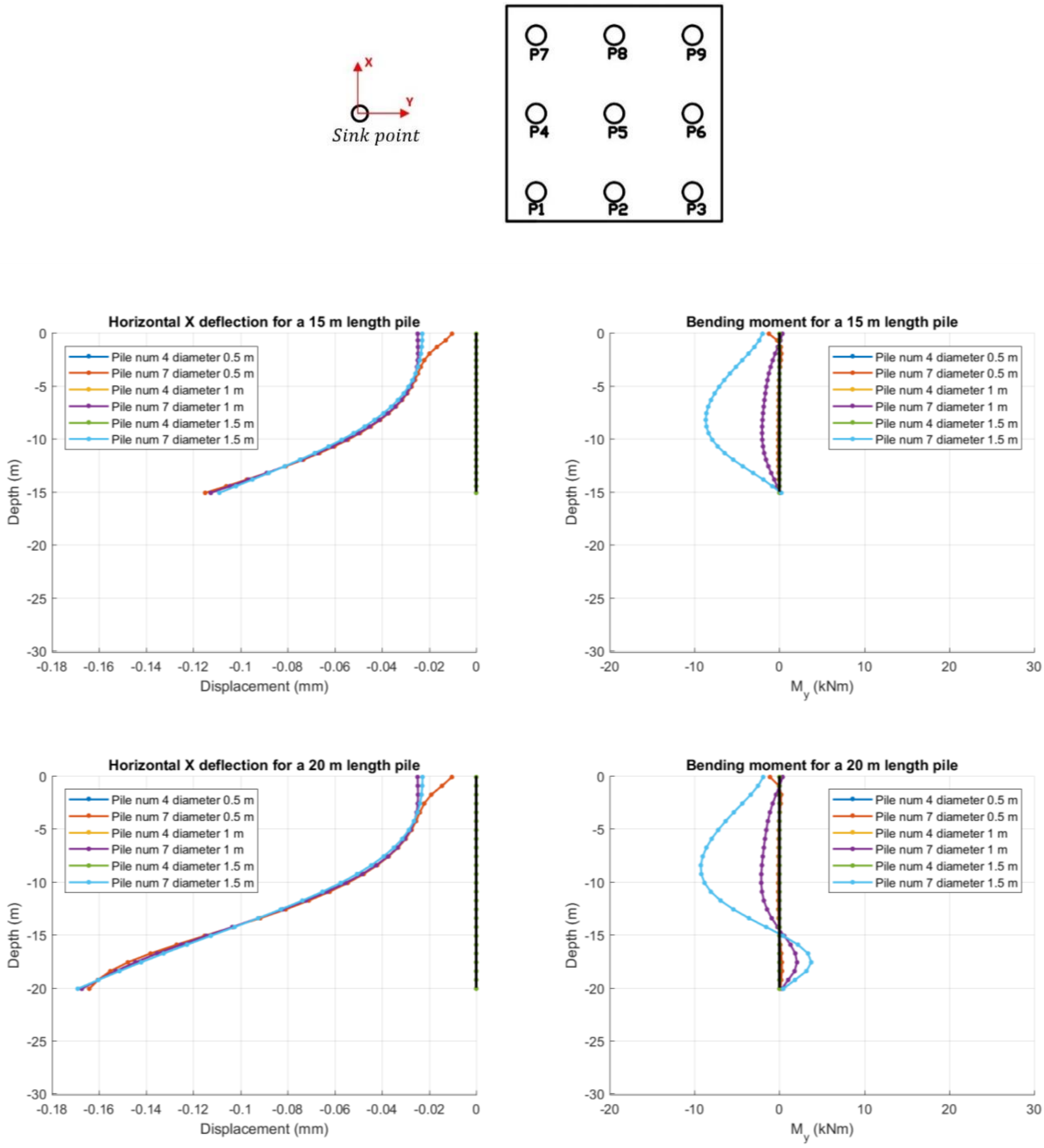


Figure 26. Deflection in X coordinate (left plots) along corresponding bending moment (right plots) for different pile lengths and diameters, calculated with the MATLAB code developed in this project.

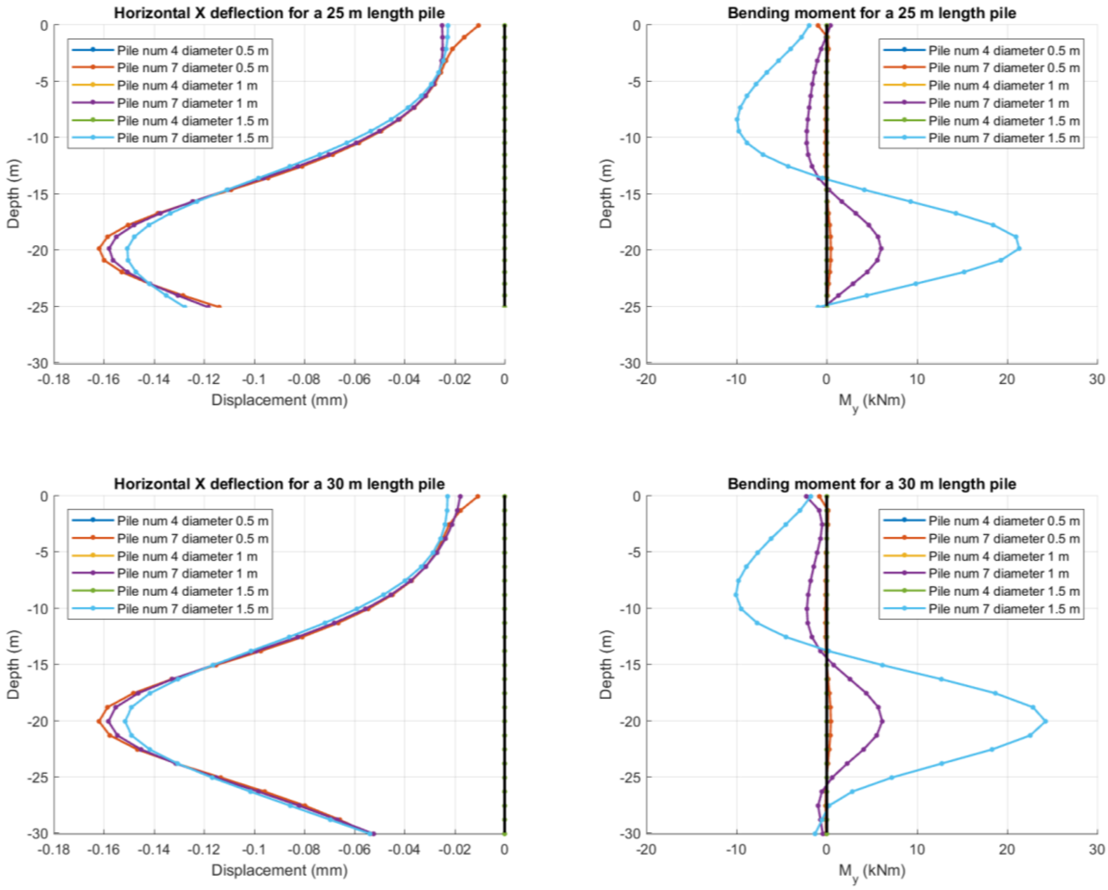


Figure 27. Deflection in X coordinate (left plots) along corresponding bending moment (right plots) for different pile lengths and diameters, calculated with the MATLAB code developed in this project.

Y direction

Again, piles 4 and 7 are depicted. In this direction, the most compromised pile is 4, the reason being that pile 7 is located farther off. The rest of the piles are deformed similarly, albeit less intensely owing to a farther location. Observe that the difference between both piles grows as the length increases.

Notice that displacements are similar for increasing diameters, however, the bending moments do vary significantly. Likewise in the X direction, the higher the dimeter, the higher the inertia, thus bigger bending moment. So, upon the choice, a smaller diameter would be the least expensive option regarding the small bending moment it is subjected to, provided an hypothetical case where the only area of interest was this particular lateral response.

The maximum bending moment occurs owing to the deflection of the pile at the sink point depth. Observe the presence of a bending moment at the head of the piles ($z = 0$), this is proof of the fixing moment due to the pile cap. As it as well happened in the X direction, the moments at the tip of the piles are zero, corroborating the expected behaviour imposed in the formulation of the model.

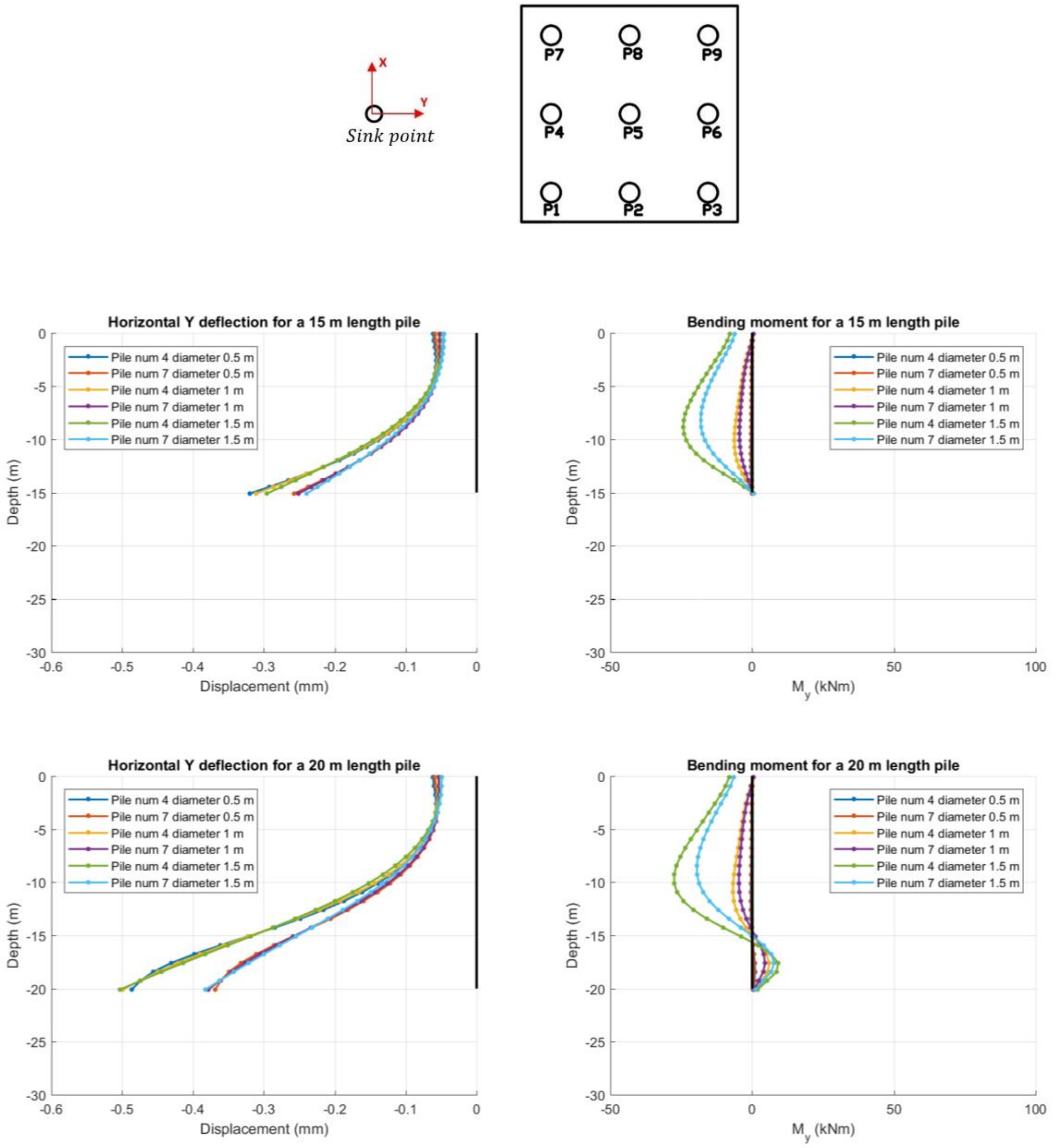


Figure 28. Deflection in Y coordinate (left plots) along corresponding bending moment (right plots) for different pile lengths and diameters, calculated with the MATLAB code developed in this project.

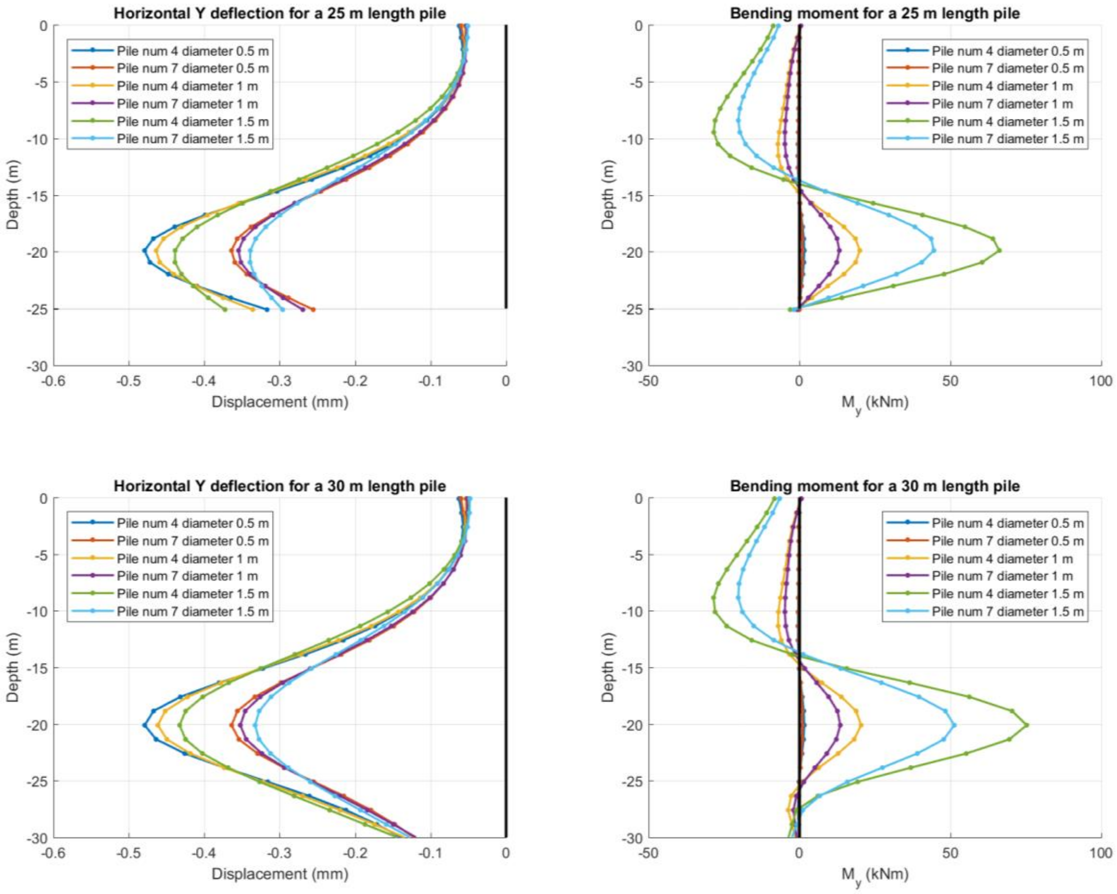


Figure 29. Deflection in Y coordinate (left plots) along corresponding bending moment (right plots) for different pile lengths and diameters, calculated with the MATLAB code developed in this project.

Z direction

For this comparison piles 4 and 5 will be used. Since coordinate Z is positive downwards, settlement has positive values, as can be seen in subsequent figures. As stated in section 2.3, axial compression is represented as positive forces.

A remark is made in how pile 4, closest to the point volume loss, behaves: for lengths that fall above the depth of the point sink, the vertical displacement increases with z, meaning that pile 4 is elongating, hence the tension (negative axial force) seen in the corresponding plot. However, pile 5, farther off in the second row, is compressed, and it appears that the rest of the pile-group is supporting the first row, as if these were hanging from the pile cap. Provided all the piles are located above the point, all experience an exclusive downward displacements. The difference between the intensity of the displacements creates a differential settlement that results in this situation, also being coherent with the fact that there is a rigid pile cap and that there is an interaction between them.

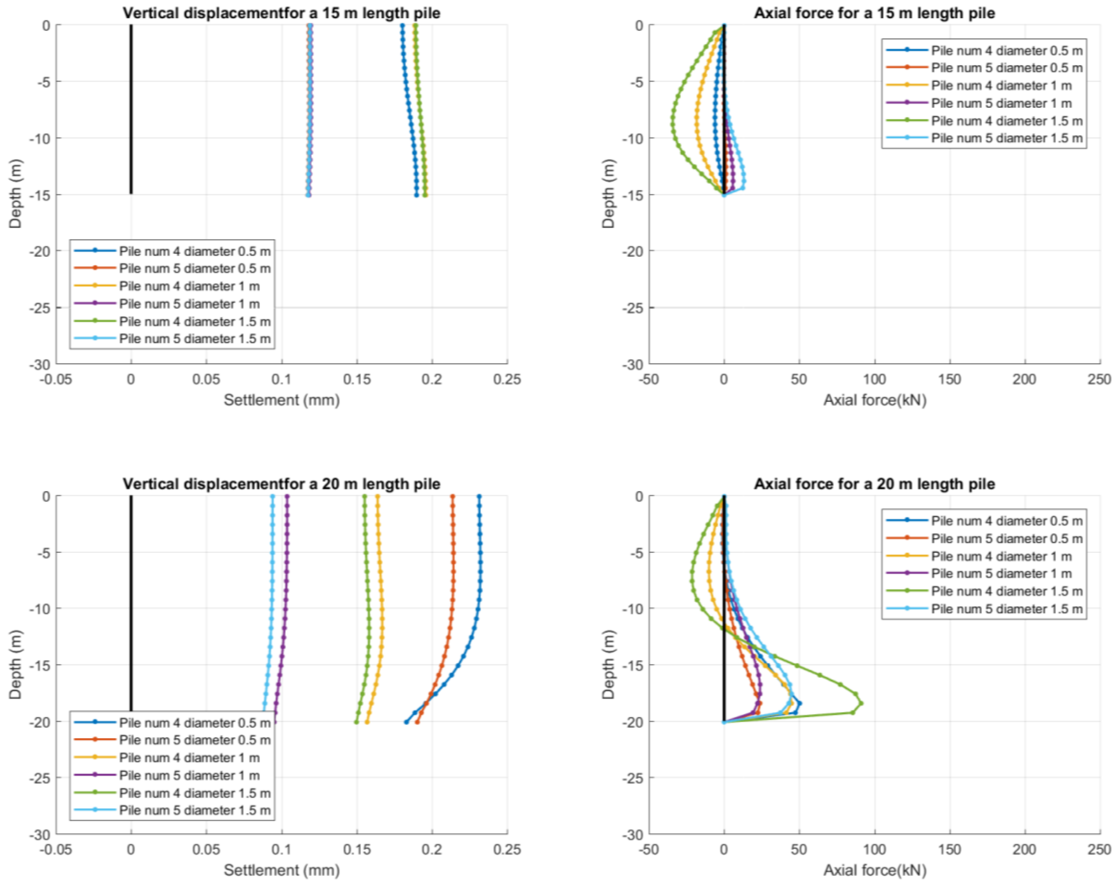


Figure 30. Displacements in Z coordinate (left plots) along corresponding axial force (right plots) for different pile lengths and diameters, calculated with the MATLAB code developed in this project.

For longer piles, one can see that around the depth where the sink point is located, axial force increases significantly, this is due to the fact that a sink point (as described by Sagaseta 1987) produces downwards soil movements above the point and upwards movements for the soil below that point, this is clearly visible for piles of 30 m length.

Also in terms of the piles length, it is clear that for longer piles the maximum axial force increases. Perhaps owing to a larger area where the soil introduces shear stresses.

On the other hand, boarder diameters translate into smaller settlements, as expected, owing to an increase of skin and base area, hence more subsurface load taken into account within the Mindlin’s factors.

To summarise, this section has shown that closer piles (to the source of soil loss) tend to be more affected and that there is a direct proportionality, and how the soil moves due to a point volume loss.

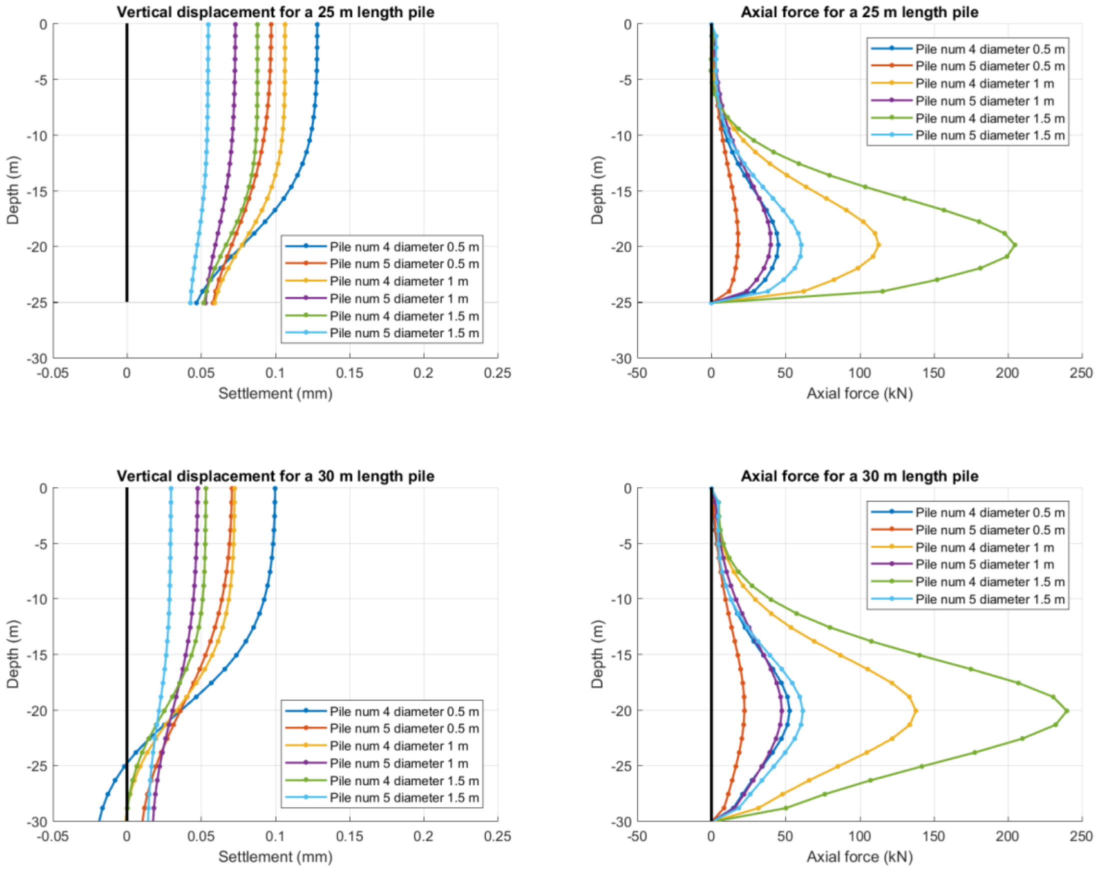


Figure 31. Displacements in Z coordinate (left plots) along corresponding axial force (right plots) for different pile lengths and diameters, calculated with the MATLAB code developed in this project.

5.2 ADVANCING TUNNEL

The aim of the current section is to simulate an advancing tunnel and to study the response of a 9 pile group for two different tunnel areal loss ratios, the geometry in both simulations remains the same. A sketch of the problem geometry shows that there is a 4 m gap between the outermost part of the tunnel and the closest pile axis. The group of piles are connected via a rigid pile cap.

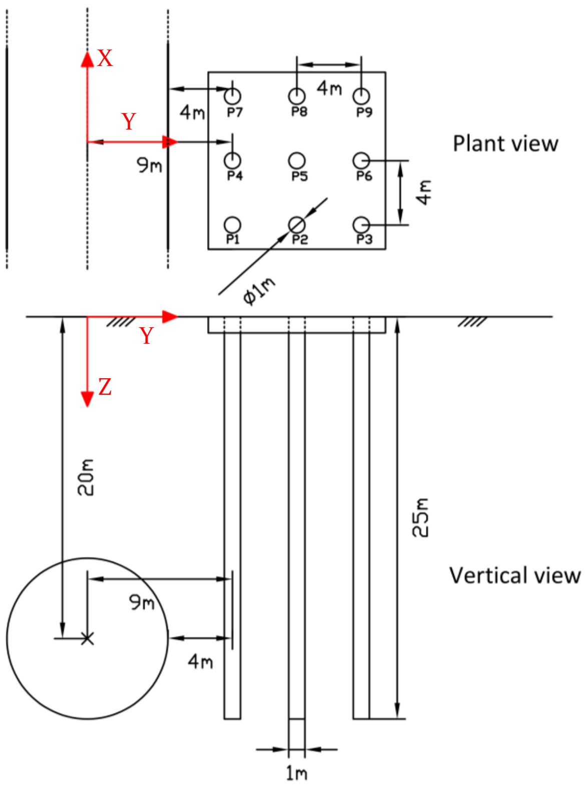


Figure 32. Plant and vertical views of the adjacent tunnel and the 9 piled-raft foundation. The tunnel starts at -50 and ends at 50 m along X direction.

Table 3 and

Table 4 gather the values that would need to be input in this project code.

Table 3. Piles, soil and tunnel relevant parameters used for the current simulation.

Piles	***Value***
Length (m)	25
Diameter (m)	1
Young modulus (kN/m²)	$3 \cdot 10^7$
Number of piles	9

Soil	*Value*
Elastic modulus (kN/m²)	50000
Poisson modulus	0,3
Tunnel	***Value***
Tunnel areal loss (%)	1
Tunnel initial diameter (m)	10
Tunnel centreline depth (m)	20

Table 4. Piles location of the 9 piled-raft for the problem in this section.

Group	***X***	***Y***
Pile 1	-4	9
Pile 2	-4	13
Pile 3	-4	17
Pile 4	0	9
Pile 5	0	13
Pile 6	0	17
Pile 7	4	9
Pile 8	4	13
Pile 9	4	17

Tunnel-wise, it will be discretised into 100 segments, its trajectory goes from -100 to 100 m in the X direction, spanning 200 m. This choice is a compromise between accuracy and computational time. As it is supported by subsequent figures and argued later on, a start 100 m away from the piles is accurate enough, provided that the farther it is, the less influence on the foundation it has and the more insignificant the contribution grows. The tunnel will be discretised into 100 parts (2 m long each one), to illustrate this, Figure 33 shows the meaning of what time steps mean in successive graphs. For instance, a plot that represents results for time step 25/100, shows the behaviour when the tunnel has advanced 50 m, from -100 to -50 m in the X direction. Time step 85/100 infers that the tunnel has advanced 170 m, from -100 m to +70 m in the X direction, and so forth.

For this particular problem, the associated volume loss will be distributed among 15 points for each tunnel segment. In terms of accuracy, more tunnel slices or points are found to give similar results at the expenses of an increase in computational time.

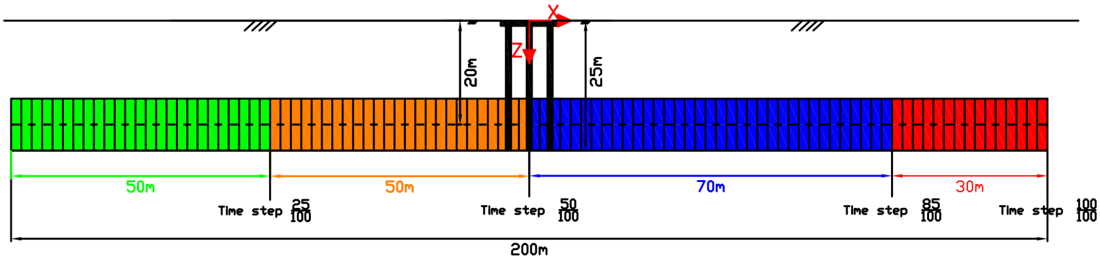


Figure 33. Lateral view of the current problem with its corresponding tunnel discretisations. Samples are 25/100, 50/100, 80/100 and 100/100, showing the interpretation of time steps.

5.2.1 Effect of tunnel areal loss

Getting down to the problem itself, it consists of a comparison between the effects of a 0.5% and 1% tunnel volume loss ratio. Parallelly, a first glance at the general response of piles due to a tunnel is cast. Subsequent graphs will show displacements, bending moments and axial forces plotted along the vertical profile of the piles for different time steps, and a second typology of plot illustrates the varying behaviour of the maximum value of some variables of interest (maximum bending moments, maximum axial forces…).

Lateral response

Concerning the X direction, while the tunnel is approaching the group of piles these are deformed towards the source of the volume loss, in this case, towards negative X direction. When then tunnel moves away, piles slowly return to its original position by being deformed in the opposite direction, i.e at time step 100/100 the final displacements are zero with null bending moment (Figure 34). This is due to the tunnel's symmetry in the YZ plane: for an infinitely long tunnel the final displacements should, theoretically, be zero. At time step 1/100, the displacements are also zero (negligible), as a consequence, endorsing the hypothesis that the tunnel exerts a very small influence already 100 m away from the foundations, and the modelling approximation is sufficiently representative of an infinitely long tunnel.

There is quite a contrast between the maximum deflection of piles 4 and 5. The shielding effect of the group of piles, or rather the interaction considered by the Mindlin problem, makes pile number 5 (in the centre of the group) to be less affected, whereas outer piles seem to receive a greater influence from the tunnel.

In terms of the areal loss, although similar in shape, it is patented the existing correlation between greater displacements and greater ratios. Notice, for instance, the head displacements in the X direction for 1% that double those at 0,5%.

Figure 35, depicts the varying maximum X displacement and bending moment M_Y as the tunnel approaches and moves away from the piles. It must be said that these figures represent the maximum numbers in absolute value.

The behaviour in the X direction fulfils the presumed intuitive ideas. The maximum displacement in this direction occurs when the tunnel is exactly located horizontally with the pile in question, thus the recognisable peaks in Figure 35. In a similar way, the maximum bending moment also tops when the tunnel is at the X location of the corresponding pile. It appears that in every displacement curve there is a change of slope about 5 m before the maximum, which is interesting to analyse. In relation to that, a remark is made in how there is a lag between the largest maximum displacement and bending moment, the latter occurring more or less 8 m before the former.

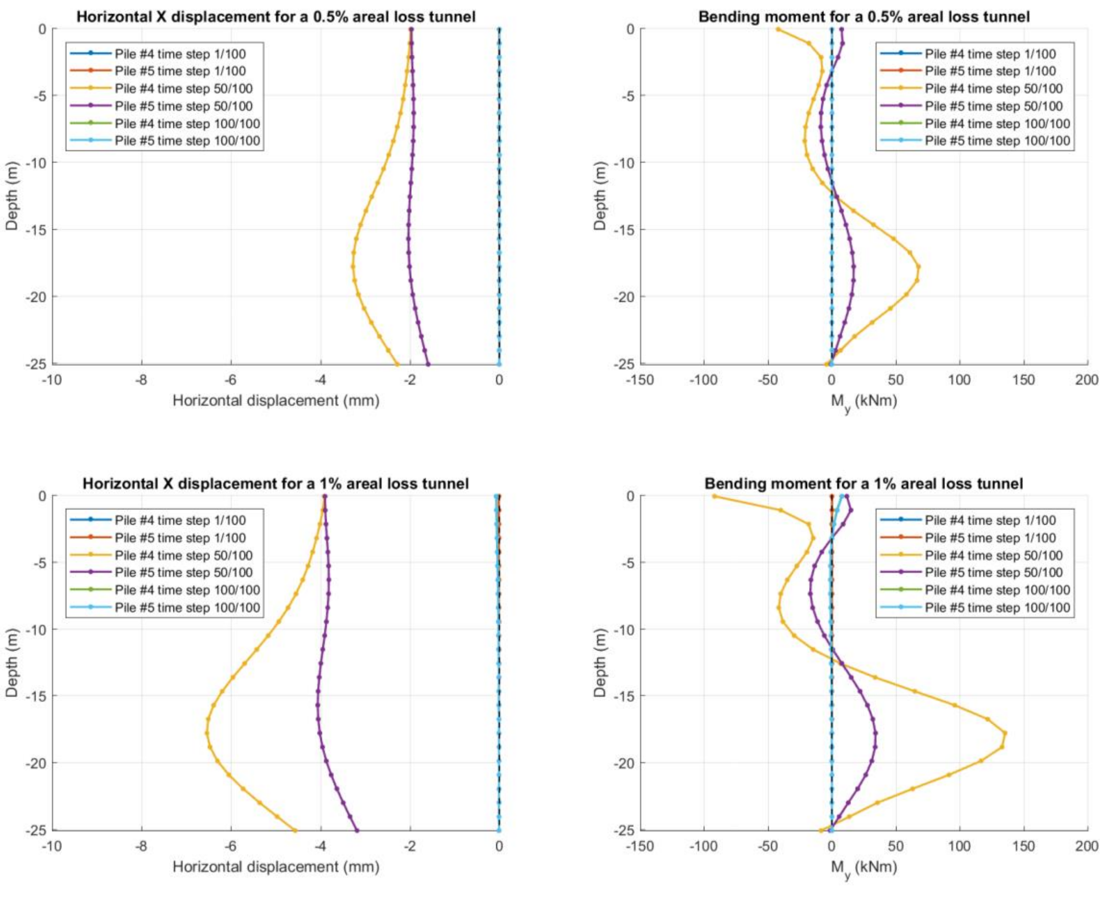
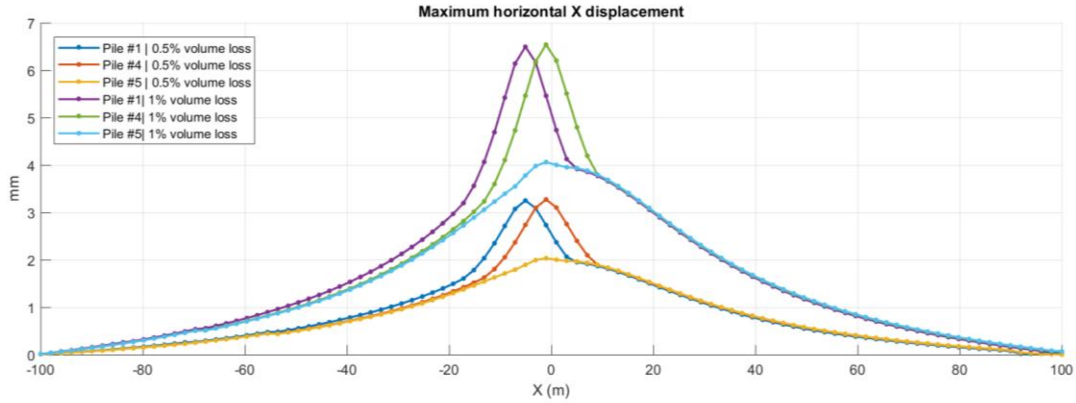


Figure 34. Comparison of displacements in the X coordinate (left plots) along corresponding bending moment (right plots) generated by 0,1% and 0,5% areal loss tunnels, calculated with the MATLAB code developed in this project.



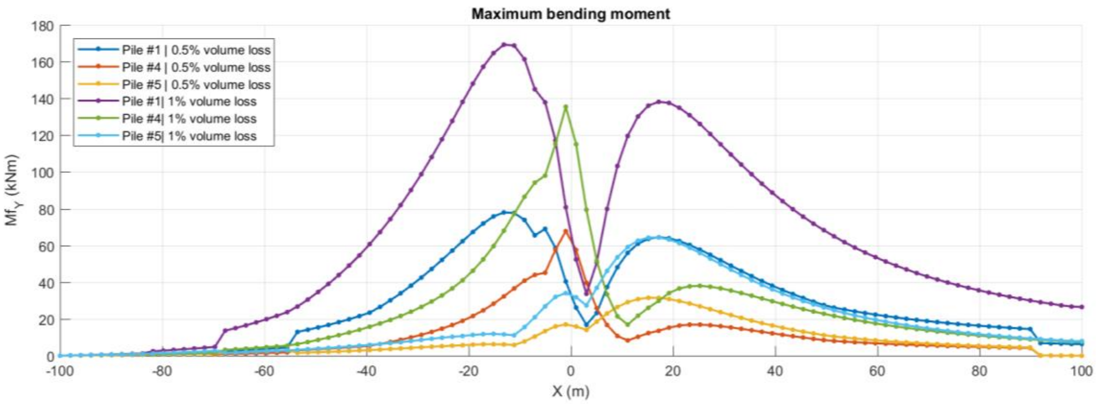


Figure 35. Variation of the maximum lateral X deflection and corresponding bending moment as a tunnel is advancing (both in absolute value). Comparison for 0,5 and 1% areal loss ratio, calculated with the MATLAB code developed in this project.

The reason of this marked slope variation is due to the change in the location where this takes place within the pile. The idea is that in the first stages of the tunnel excavation, the maximum bending moment takes place at the head of the pile (also the maximum displacement), when the tunnel gets closer there is more influence around the excavation depth (for this particular case $z = 17{,}8\ m$) creating a sort of belly-shaped deflected beam (see Figure 36). At that point, the maximum bending moment also takes place at $z = 17{,}8\ m$.

Take for instance pile number 1 for 1% tunnel volume loss ratio (Figure 35 in purple). There is a maximum at around $x = -14\ m$ (time step 45/100 in Figure 36) happening at the head. When the tunnel keeps approaching the pile (time step 50/100 in Figure 36), the maximum deflection does indeed increase in contrast with the decreasing maximum bending moment. When this occurs, the head gains in verticality, reducing the bending moment at the cap while the maximum now occurs at the tunnel depth (time step 50/100 Figure 36), much smaller than before.

Having seen this and continuing with pile 1 as an example, it is easy to understand the dip in bending moment when the tunnel is around $x = 0\ m$ which is due to the verticality the pile presents at that point. Afterwards, the maximum bending moment increases again until around $x = 18\ m$, once more owing to a maximum bending moment taking place at the head (time step 59/100 in cyan, Figure 36).

In conclusion, throughout this process there are two critical parts in every pile, the head and at the depth halfway through the tunnel crown and centreline axis, in this case $z = 17{,}8\ m$. Let us also remark that in general, the maximum bending moment ever at that depth ($z = 17{,}8\ m$) is much smaller than that at the head.

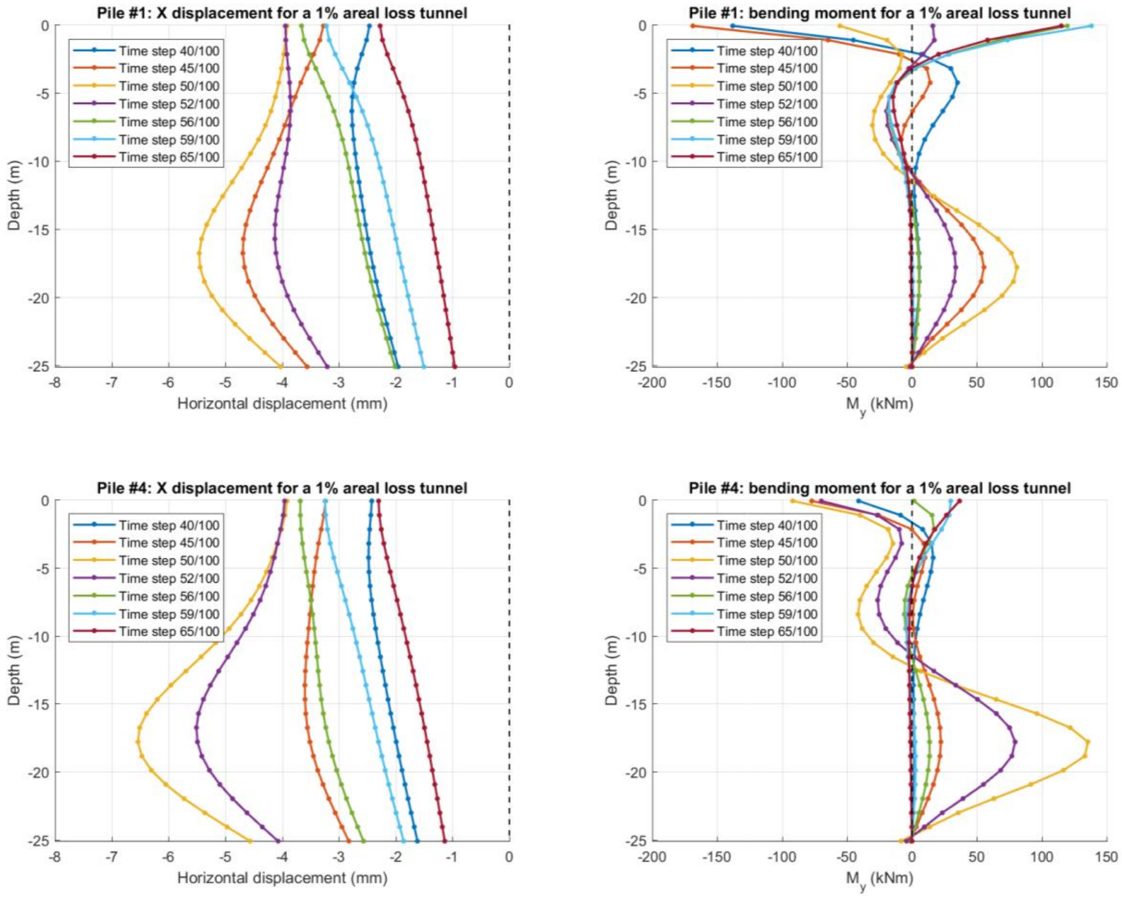


Figure 36. Displacements and corresponding bending moment for relevant time steps, calculated with the MATLAB code developed in this project.

Coming to the response in the Y direction, the differences between the X direction analysis emerge towards the last half of the tunnel excavation. In this case, piles are always influenced towards negative Y in contrast with the changing direction as for the X response.

It is clever to realise again how pile 5 is less affected than the rest. In general, piles are to remain unchanged once the tunnel has passed, then the critical parts would be the head and at $z = 17{,}8\ m$, where both present the largest bending moments. Therefore, it is crucial to analyse these results against the resistance of the piles, because they will remain like this for the rest of their lifespan.

In Figure 38 it is clearly seen that most of the displacement is gained when the tunnel is within 20 m from the piles.

If one takes a close look, there is a small dip in M_X just before the tunnel meets the pile. Again, this owes to a change in the location of the most affected part. Similar to what happens in the X direction, if the maximum bending moment at the 100/100 takes place at 17,8 m of depth, before that dip, it occurs at the head of the piles, check Figure 37.

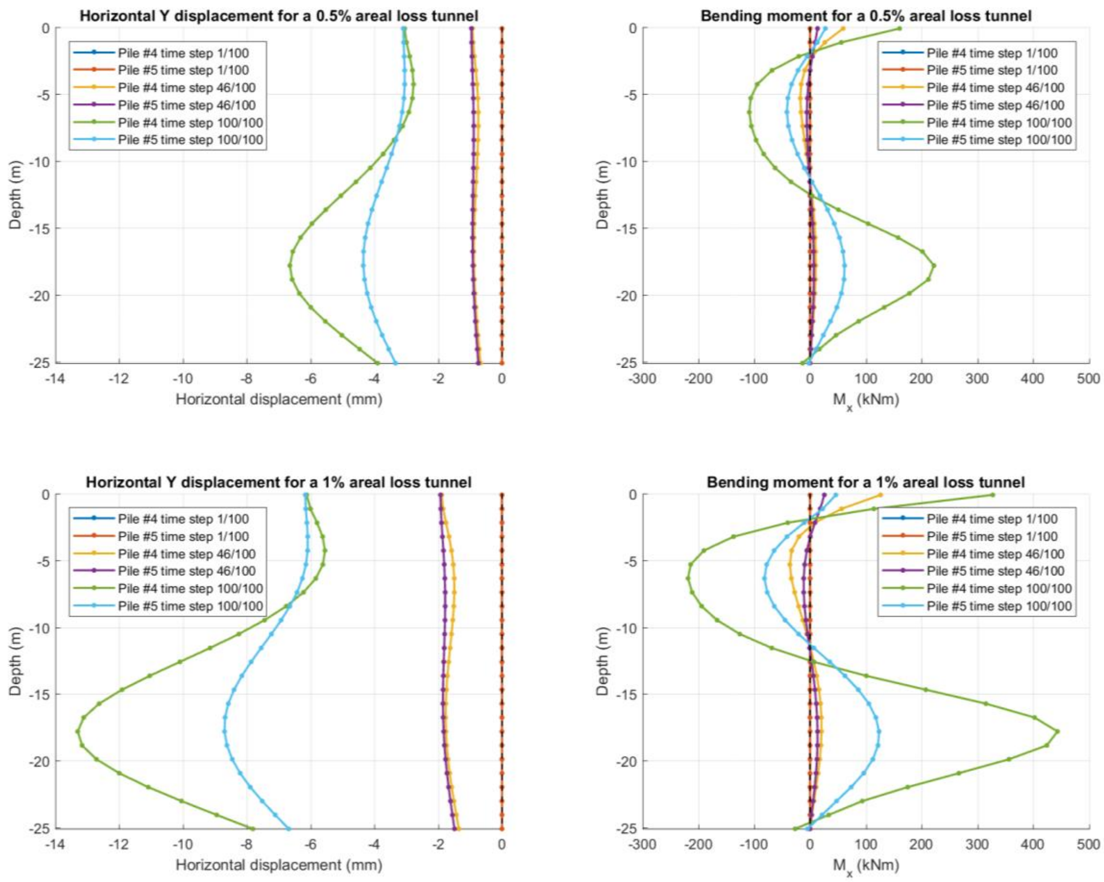
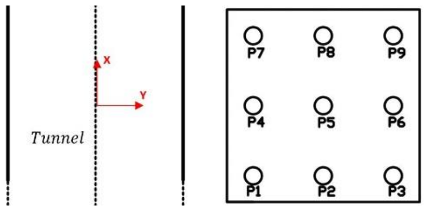
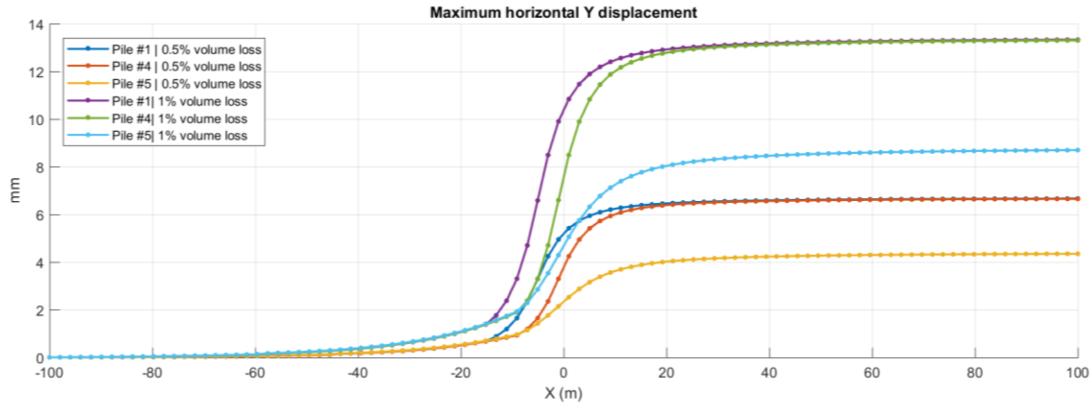


Figure 37.Comparison of displacements in the Y coordinate (left plots) along corresponding bending moment (right plots) generated by 0,1% and 0,5% areal loss tunnel, calculated with the MATLAB code developed in this project.





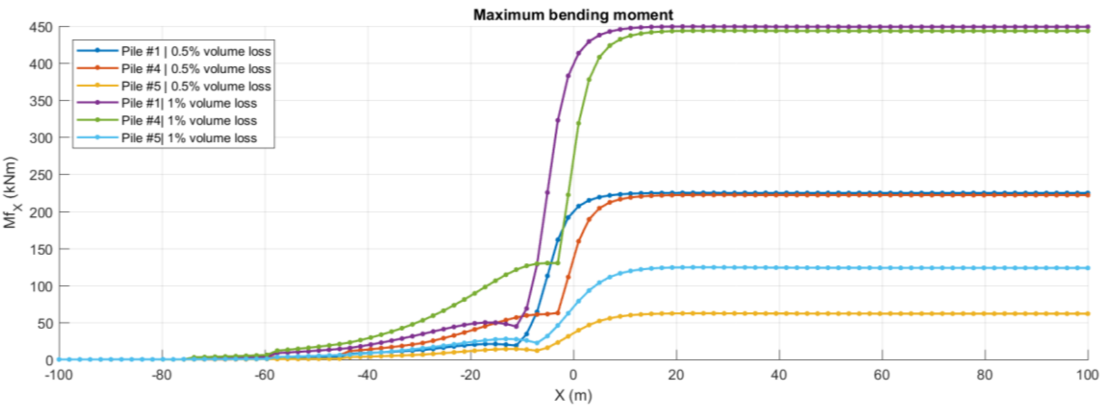


Figure 38. Variation of the maximum lateral X deflection and corresponding bending moment as a tunnel is advancing. Comparison for 0,5 and 1% areal loss ratio, calculated with the MATLAB code developed in this project.

Vertical response

Likewise, the vertical displacements reflect the differences between the 0,5% and 1% ratios as both horizontal analysis have demonstrated. The greater the ratio, the greater the settlement and consequent axial forces undergone.

In the following figure (Figure 39), piles 4 and 5 are represented showing that due to the interaction between the group, number 5 being at the centre surrounded by 8 other piles it presents half the maximum axial force.

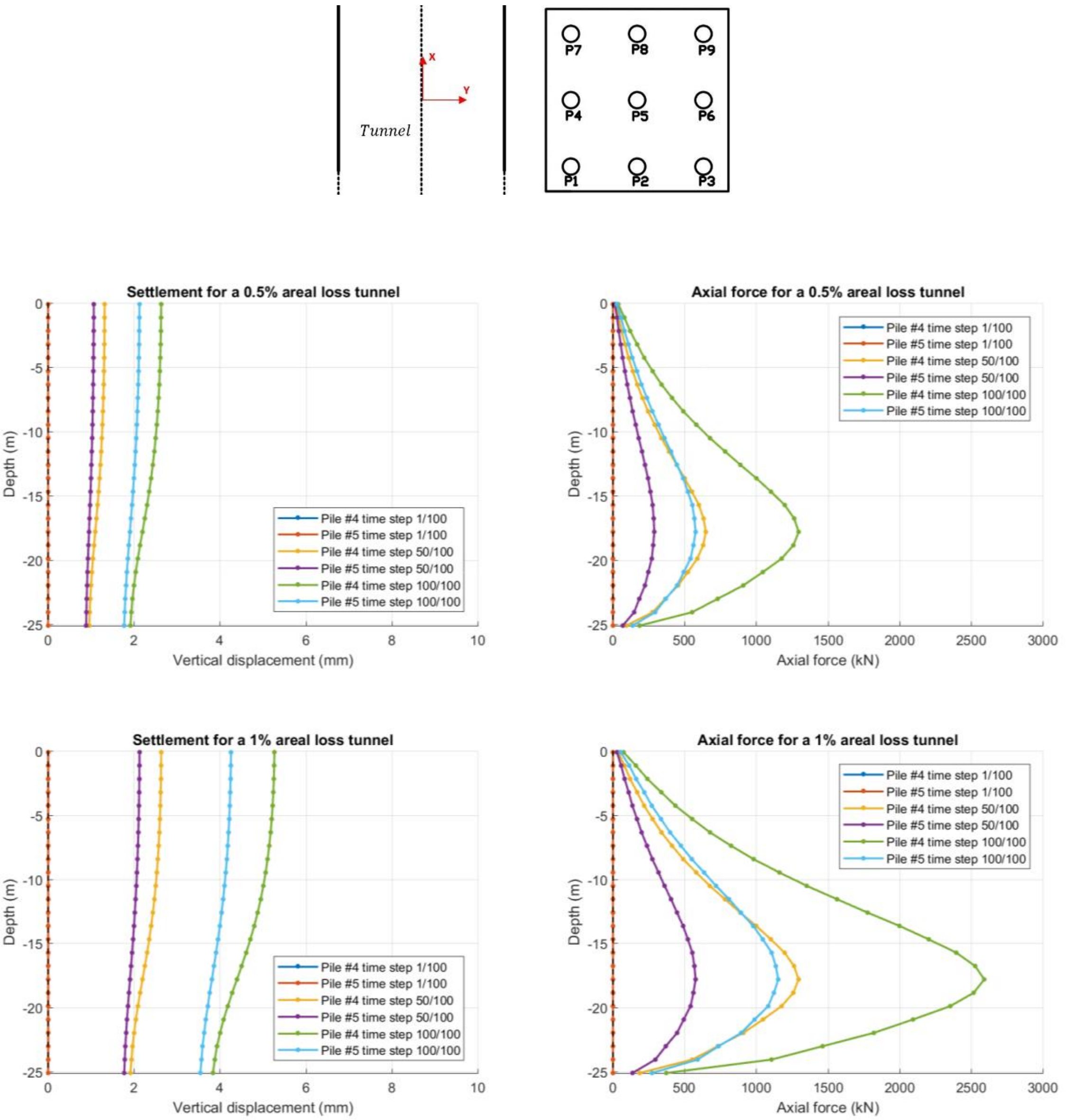


Figure 39. Comparison of displacements in the Z coordinate (left plots) along corresponding bending moment (right plots) generated by 0,1% and 0,5% areal loss tunnels, calculated with the MATLAB code developed in this project.

Timestep-wise, the settlement and axial force, gradually build up, especially when the tunnel is close to the piles. The maximum axial force soon stabilises: 20 m after the tunnel has passed ($x = 10\ m$ in Figure 40), showing no further changes. As it was commented earlier, pile 5 compared to the rest appears to have the least settlement. Moreover, pile 4 compared to pile 1 also has smaller maximum axial force. This will remain a constant for any squared group of piles (9, 16…), further analysis in section 6.1.

In Figure 39 note how the maximum compression takes place at about $z = 17{,}8\ m$. This is not coincidental with what happened in the horizontal directions. In terms of the vertical stresses, Figure 41 confirms what could be a first assumption and was argued in section 5.1: the soil movements induced by the tunnel create downwards shear stresses from the top to $z = 17{,}8\,m$ and upward lift below that point, in the plot negative and positive respectively.

In Figure 41, a drastic increase at the tip of the pile stands out, this represents the load at the base, which obviously is completely different from the shaft stress. The chosen time steps in Figure 41 is irrelevant, insofar the most important fact to grasp in this lines is the general behaviour of the said loads. Besides, the regularity derived from Figure 39 and Figure 40 implies that the vertical shear stress the pile is subjected to, would only increase with time maintaining the same shape of the load distribution law.

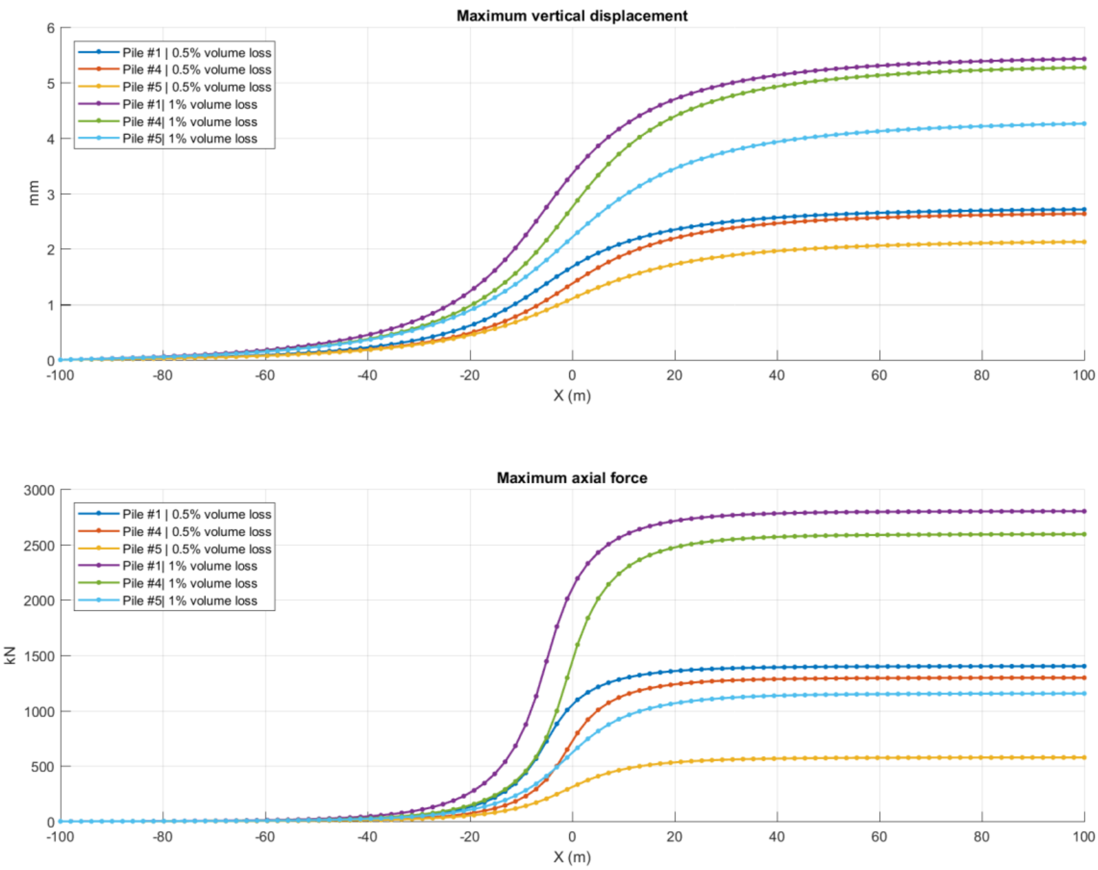
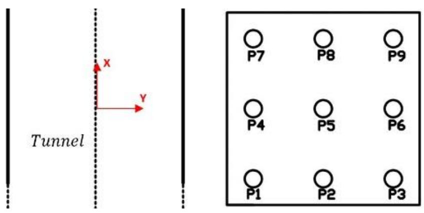


Figure 40. Variation of the maximum lateral X deflection and corresponding bending moment as a tunnel is advancing. Comparison for 0,5 and 1% areal loss ratio, calculated with the MATLAB code developed in this project.



Finally, to justify the choice of the length of the advancing tunnel (200 m), it has been made clear from figures 35, 38 and 40 that the maximum responses are already stabilised when the tunnel crosses the 100 m mark. Even if a longer excavation is preferred, the results would not significantly change.

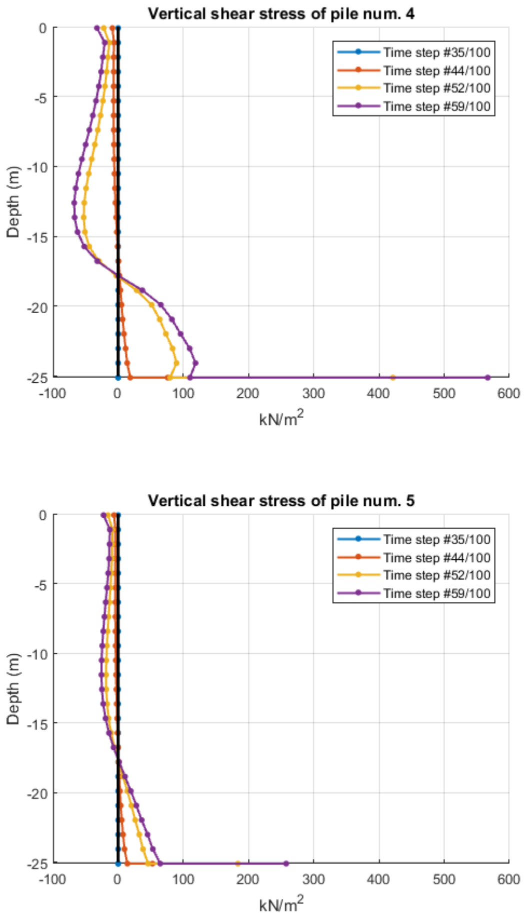


Figure 41. Vertical shear stress along the piles of interest, for key some steps and 1% tunnel areal loss ratio, calculated with the MATLAB code developed in this project.

6 PROBLEMS OF INTEREST

After studying the tunnel-pile interaction, this chapter will go through two different case studies so as to showcase the applicability of the developed code. In the first place, an advancing tunnel is again calculated, then compared with the commercial software PLAXIS and finally the effect of applying compensation grouting is simulated. In the second place, based on the work by Ledesma and Alonso (2015), the Sagrada Família pile wall, which was executed on occasion of the approaching Spanish high velocity railway tunnel, will be studied.

6.1 ADVANCING TUNNEL

Similar to what has been simulated in section 5.2, the following problem varies in that there is an external vertical load, which could well represent the weight of a building. In fact, the geometry is the same as the previous problem. Below, the specifications are recalled in the form of Figure 42, Table 5 and Table 6. It is important to highlight that, as depicted below, the group of piles are connected via a rigid pile cap (a pile cap as defined in section 2.4).

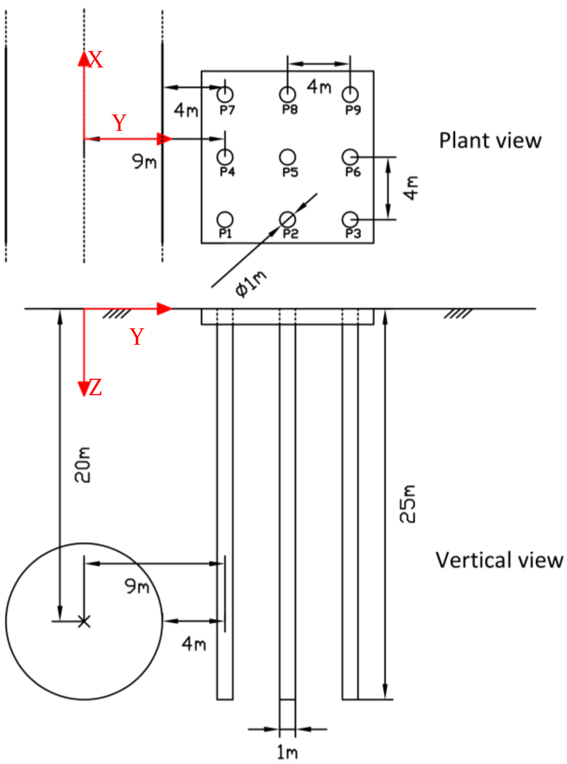


Figure 42. Plant and vertical views of the adjacent tunnel and the 9 piled-raft foundation. The tunnel starts at -50 and ends at 50 m along X direction.

Table 5. Piles, soil and tunnel relevant parameters used for the current simulation.

Piles	***Value***
Length (m)	25
Diameter (m)	1
Young modulus (kN/m²)	$3 \cdot 10^7$
Number of piles	9
Vertical load per pile (kN)	2000
Soil	***Value***
Elastic modulus (kN/m²)	50000
Poisson modulus	0,5
Tunnel	***Value***
Tunnel areal loss (%)	1
Tunnel initial diameter (m)	10
Tunnel centreline depth (m)	20

Table 6. Piles location at the 9 piled-raft.

Group	***X***	***Y***
Pile 1	-4	9
Pile 2	-4	13
Pile 3	-4	17
Pile 4	0	9
Pile 5	0	13
Pile 6	0	17
Pile 7	4	9
Pile 8	4	13
Pile 9	4	17

Upon this occasion, the pile group is bearing a total 18000 kN of vertical force, hence 2000 kN per pile.

Again, as in section 5.2, the tunnel is discretised into 100 segments, its trajectory goes from -100 to 100 m in the X direction, spanning 200 m. Figure 43 has been coloured accordingly with the plots in the following pages.

The associated volume loss will be distributed among 15 points for each tunnel segment.

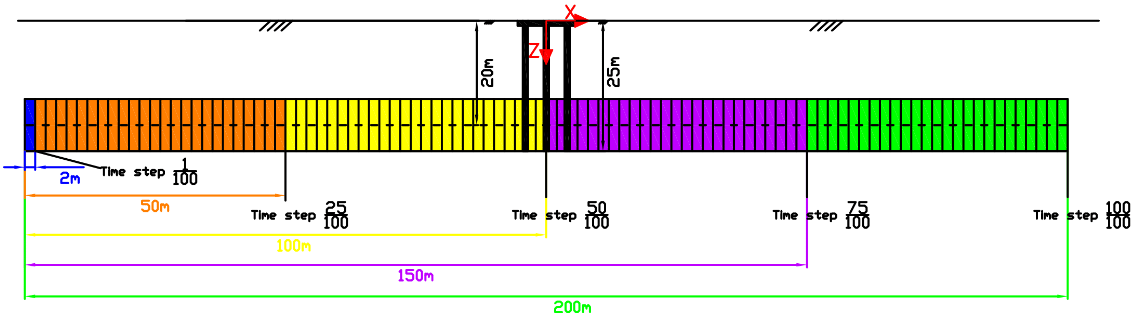
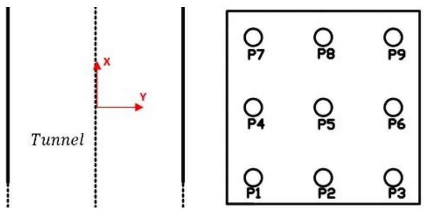


Figure 43. Lateral view of the tunnel and the pile group, depicted in blue, orange, yellow, magenta and green the represented time steps in subsequent plots.

Lateral response

Given that the structure of the problem is exactly the same as the 1% tunnel areal loss ratio problem, described in the sensitivity analysis (section 5.2.1), one can expect beforehand that these results must be identical. Bear in mind that the analysis in each direction is uncoupled, therefore, the fact that there is vertical load should not alter the lateral behaviour. The following figures prove the robustness of the code by supporting this previous assumption.



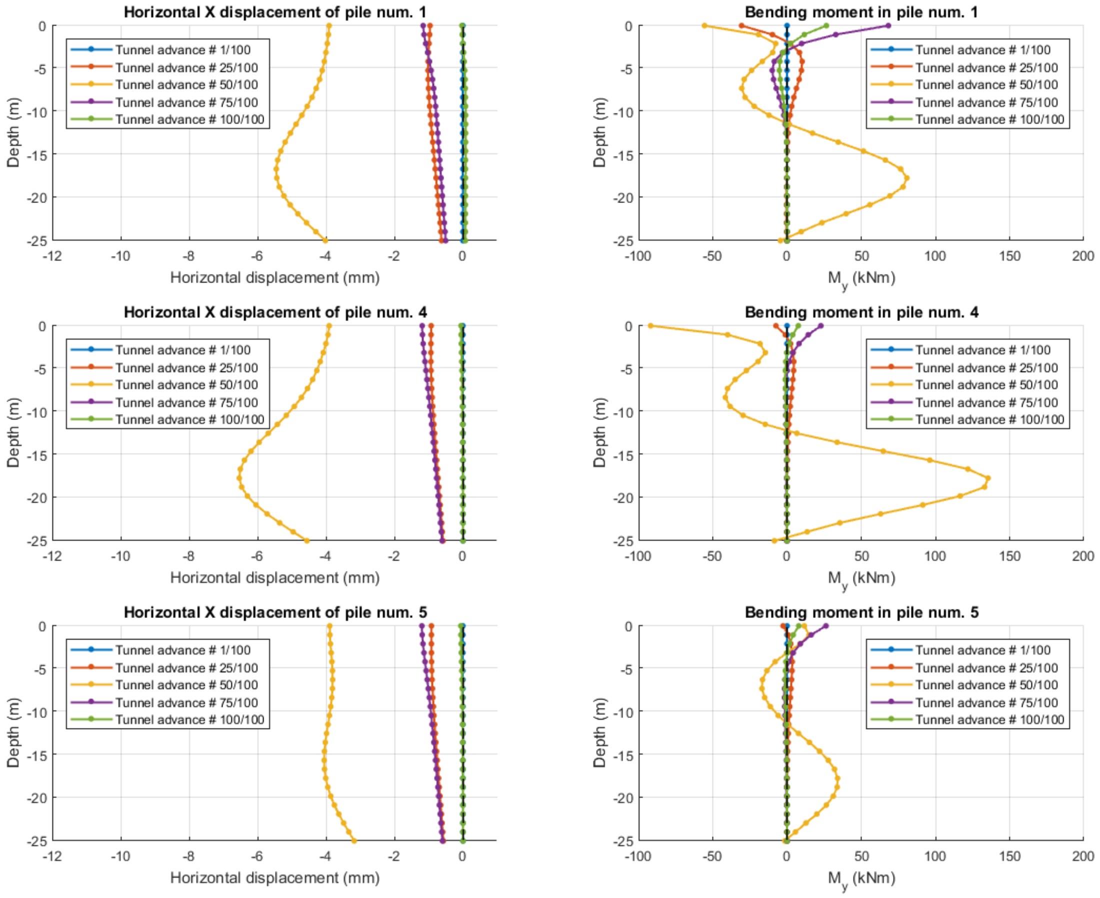


Figure 44. Displacements in X coordinate (left plots) along corresponding bending moment (right plots) generated by a tunnel excavation represented for different time steps, calculated with the MATLAB code developed in this project.

As it is expected from previous knowledge, there are two critical parts of a pile where maximum bending moment occurs, the head and that at the tunnel's depth, again 17,8 m. It is paramount to supervise the maximum bending moment occurring at the head, since it puts at risk the stability of the structure, given the hypothetical scenario of head detachment. At that point the pile would become worthless, by not contributing to the transmission of the superstructure load. This is represented below. Note how M_Y varies from negative to positive, corresponding to before and after the tunnel overcomes the pile in question. From Figure 46 it can be concluded that for closer piles (to the tunnel) M_X is far more relevant at the end of the tunnel excavation than M_Y, although the latter must not be dismissed. On the contrary, for farther rows of piles, for instance piles 5 and 6, the temporal nature of M_X (it diminishes when the tunnel is completed) must not be overlooked, since it represents the maximum bending moment these piles experience, M_Y remains small.

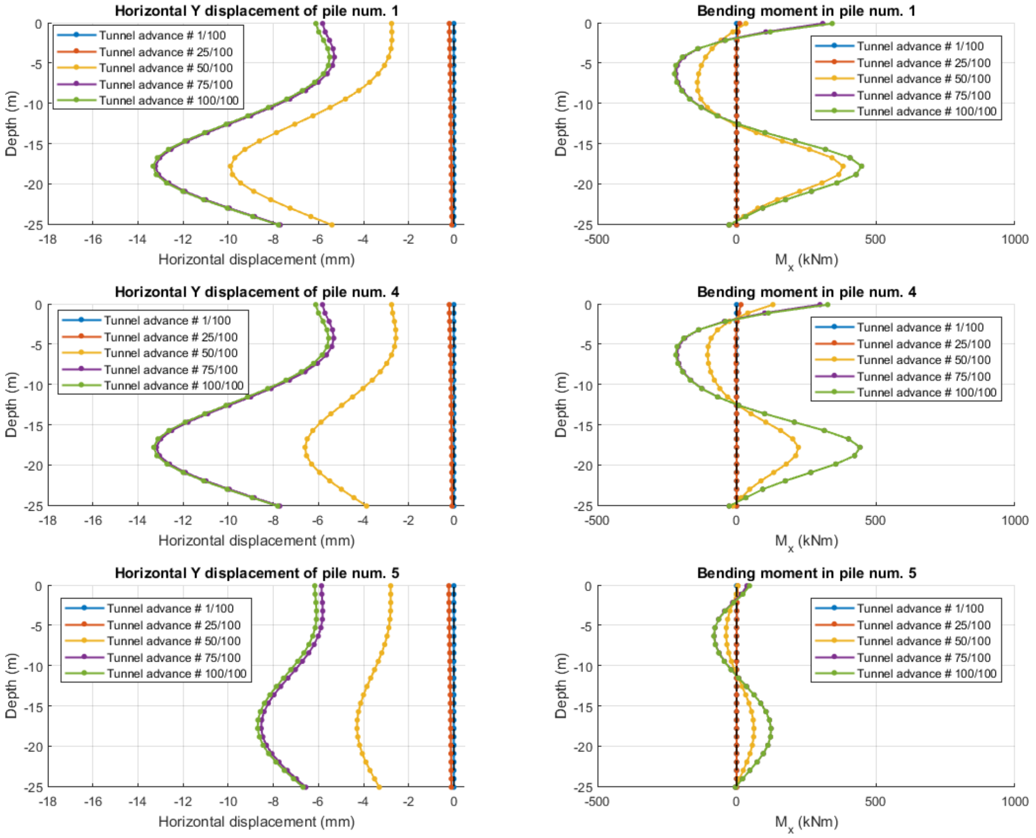


Figure 45. Displacements in Y coordinate (left plots) along corresponding bending moment (right plots) generated by a tunnel excavation represented for different time steps, calculated with the MATLAB code developed in this project.

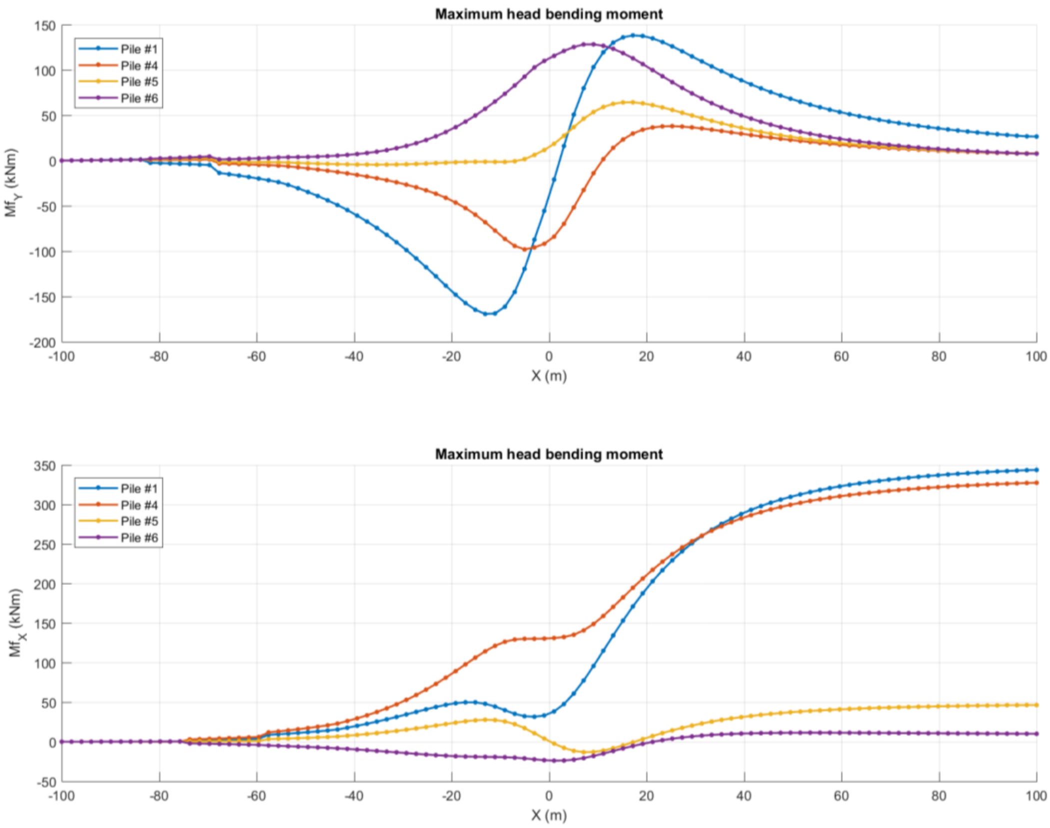


Figure 46. Varying M_Y (above) and M_X (below) at the piles head as a tunnel excavation passes by, calculated with the MATLAB code developed in this project.

Vertical response

The vertical response is similar to that in the previous section, albeit not identical since there are external forces. This fact is useful to provide more insight about the behaviour of a group of piles. These are now loaded, hence producing greater settlement. Comparing vertical results from section 5.2 (Figure 39) and this section (Figure 47) the maximum settlement after the tunnel is gone, at time step 100/100, increases from about 5 to 15 mm. Consequently, the contribution of the loading translates in an additional 10 mm of settlement.

Focusing in the axial force diagrams, the results for the head (where the loading is applied) are around 2000 kN. This is coherent insofar as the applied load was exactly 2000 kN per pile. However, pile 5 presents a far smaller value around 1500 kN, owing to the presence of a pile cap and the shielding effect of neighbouring piles.

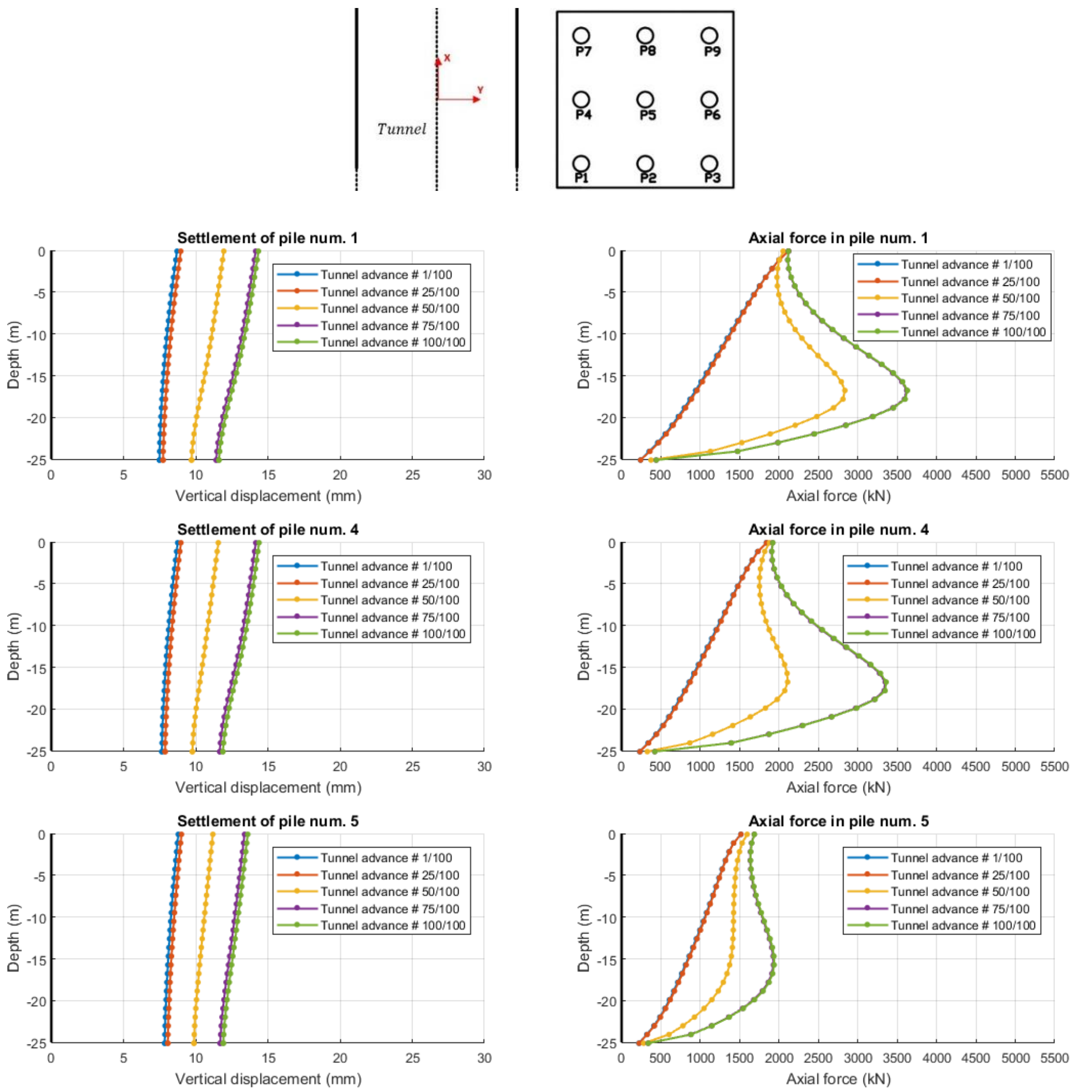


Figure 47. Displacements in Z coordinate (left plots) along corresponding axial force (right plots) generated by a tunnel excavation represented for different time steps, calculated with the MATLAB code developed in this project.

To show more insight in how the total load is distributed among piles, Figure 48 illustrates the axial force both at the head (above) and where the maximum occurs, at $z = 17{,}8\ m$ (below).

There is a general rule of thumb for a loaded pile group. Imagine the said group without the presence of a nearby tunnel, the beginning ($x = -100\ m$) of Figure 48 well represents the said situation, at that point the tunnel influence is imperceptible. Keeping in mind how the piles are labelled above, generally, the corner piles 1, 3, 7 and 9 absorb the greatest amount of external load coming from the superstructure. The side piles 2, 4, 6 and 8 are the next in order, whereas pile 5, which is completely surrounded by the rest of the piles presents the smallest value. This is due to both the interaction between piles, namely the shielding effect, and most importantly the presence of a rigid pile cap. The general lines of this rule of thumb is maintained throughout the whole tunnel excavation, with some nuances. For instance, note how pile 5 increases its head axial force (Figure 48), the reason it gains in compression is due to a growing shaft shear stress at a certain depth, provoked by the soil movements and taken into account by means of the Mindlin factors. This is purely due to a direct drag force that does not go through the pile cap, therefore it loses the effect of redistribution among the rest of the piles.

In the lower graph in Figure 48, the aforementioned rule of thumb is clearly altered. At $x = 100\ m$: the closest piles 1, 4 and 7 are subjected to the greatest compression; piles 2, 5 and 8 (second closest row) form the second most affected group, although pile 5 is less influenced; the third group is a bit below the second, formed by piles 3, 6 and 9 (third row). Observe how they are in order of closest rows. The reason of this new results is obviously due to the presence of a tunnel, whose soil movements are greater for closer locations. Therefore, the basic idea changes so far as the relevant factor becomes the distance to the tunnel.

In a nutshell, it is important to comprehend and distinguish the two sources of the difference in maximum compression. On the one hand, the proximity to the tunnel, on the other hand the interaction between piles.

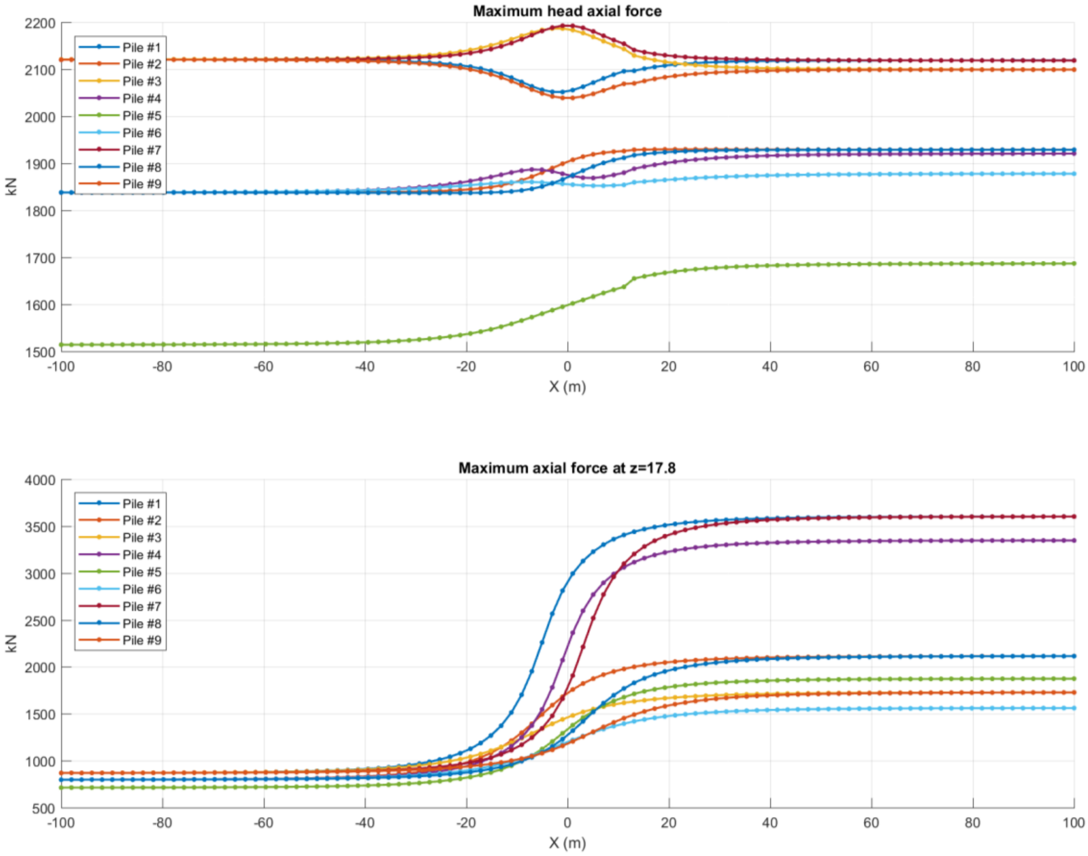


Figure 48. Varying head axial force (above) and axial force at z=17,8 m (below) as a tunnel excavation passes by calculated with the MATLAB code developed in this project. Notice, they are always subjected to compression (positive axial force).

6.1.1 PLAXIS comparison

The same problem is modelled and analysed in PLAXIS. For the sake of simplicity, only the last step is compared, i.e. when the tunnel is already 100 m away.

The tunnel has been modelled following the next steps:

1. In the structures screen, click create tunnel.
2. In the cross section box define the tunnel cross section.
3. In the properties box select the perimetric line defining the tunnel cross section and add a plate.
4. Do as step 3 but add a negative interface.
5. Do as step 3 but add a surface contraction and insert $C_{ref} = 0{,}99\%$

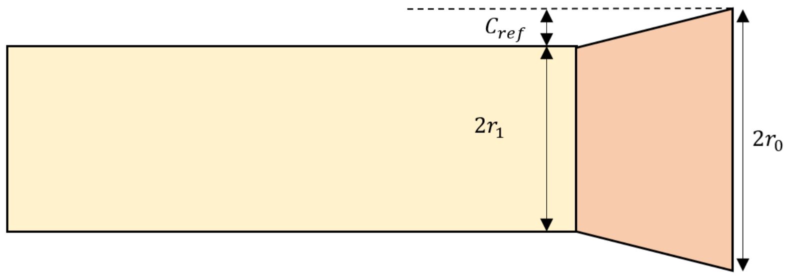


Figure 49. Sketch of PLAXIS interpretation of a volume loss due to a tunnel.

Figure 49 shows how PLAXIS interprets volume loss generated by a tunnel. Using notation from section 2.7, the new parameter C_{ref} is a percentage that relates initial and final areas. Knowing that in this project:

$$\varepsilon_0 = \frac{A_{initial} - A_{final}}{A_{final}} = \frac{\pi r_0^2 - \pi r_1^2}{\pi r_1^2} = \frac{r_0^2}{r_1^2} - 1$$

$$r_0 = \left(r_1^2 \cdot (\varepsilon_0 + 1)\right)^{1/2}$$

For PLAXIS:

$$\pi \cdot r_0^2 \cdot C_{ref} = \pi r_1^2$$

$$C_{ref} = \frac{r_1^2}{r_0^2}$$

$$C_{ref} = \frac{r_1^2}{\left(r_1^2 \cdot (\varepsilon_0 + 1)\right)} = \frac{1}{\varepsilon_0 + 1}$$

In the particular case of this simulation:

$$C_{ref} = \frac{1}{\frac{1}{100} + 1} = 0{,}99\ \%$$

Which confirms what PLAXIS states in its Manuals that for small values $\varepsilon_0 \cong C_{ref}$

The soil model is a cube 200 m of side and 100 m of depth. Moreover, like it was introduced in the MATLAB code developed in this project, the tunnel spans 200 m (from $x = -100\ m$ to $x = 100\ m$).

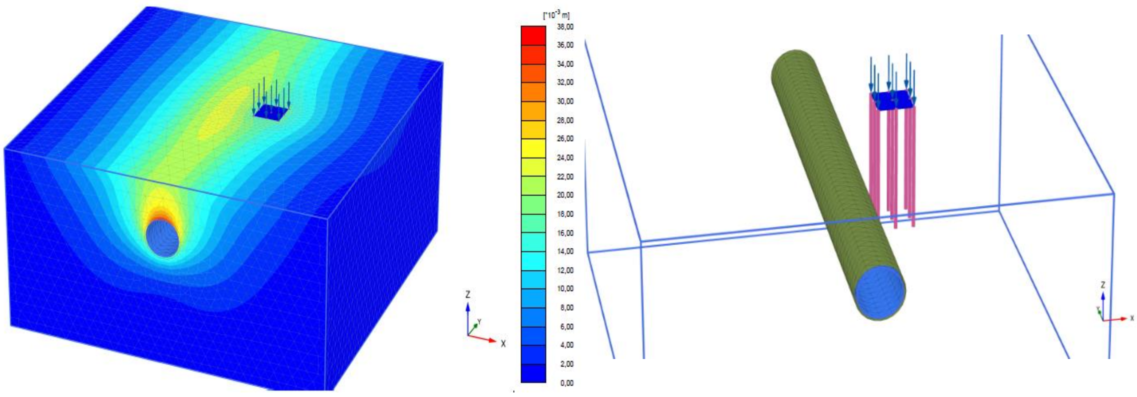


Figure 50. 3D representation of the PLAXIS model. On the left hand side the deformed mesh, on the right hand side a view of the structural elements: tunnel and group of piles. Extracted from the OUTPUT PLAXIS mode.

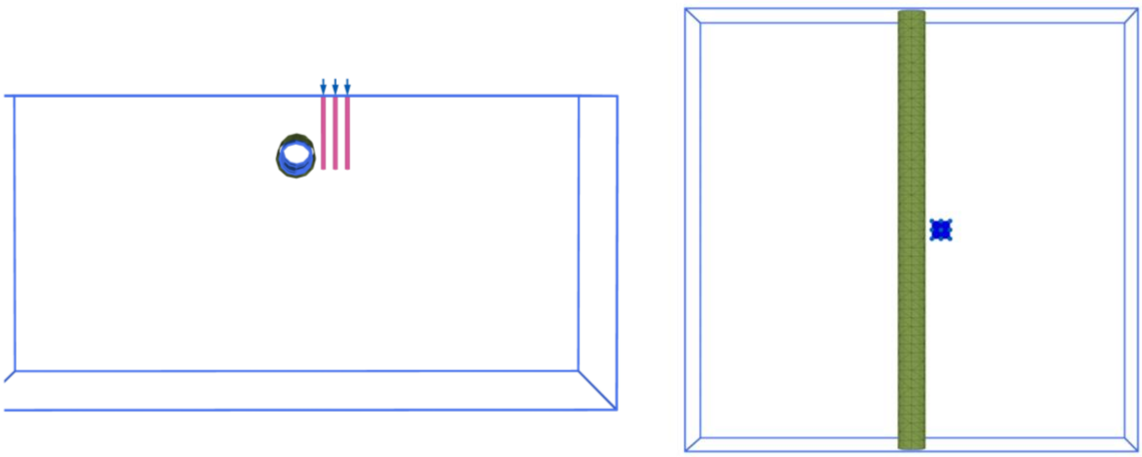


Figure 51. Front (left) and plant (right) view of the PLAXIS model extracted from the OUTPUT PLAXIS mode.

Horizontal response

As for the horizontal X displacements, these are zero at the last step as it has been extensively proven earlier, the following figure shows the comparison for Y displacements. These results present an acceptable correspondence.

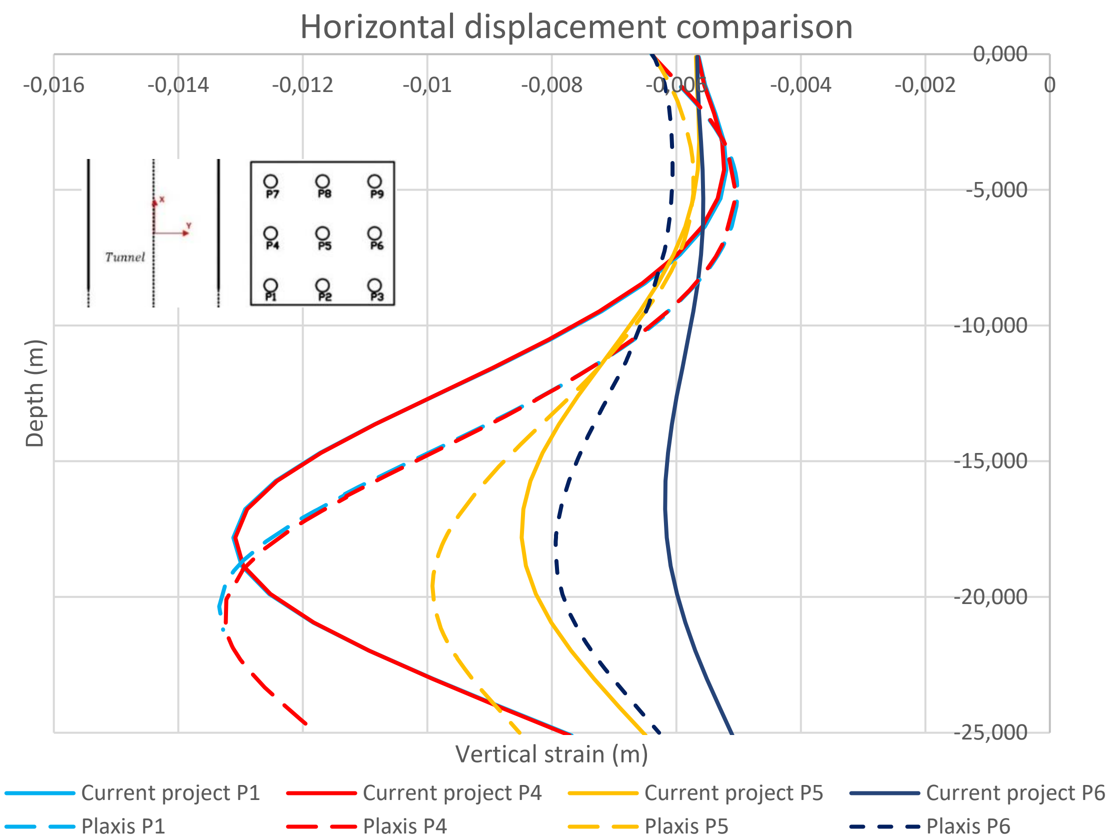


Figure 52. Horizontal displacements at time step 100/100 calculated with the MATLAB code developed in this project (continuous lines) and PLAXIS (dashed lines).

Vertical response

In this comparison, it is clear that piles do not present the same settlement at the cap, this is due to the fact that they are forming an inclined plane, not a horizontal one. Results provided by this project code: piles 1, 4 and 7 present the same settlement, 14,5 mm; piles 2, 5 and 8 also present the same settlement, 13,8 mm; so do piles 3, 6 and 9 with 13 mm. However, the plane formed by both PLAXIS and this project is not the same, the former being a bit more tilted. Figure 54 may help in further comprehending the previous results.

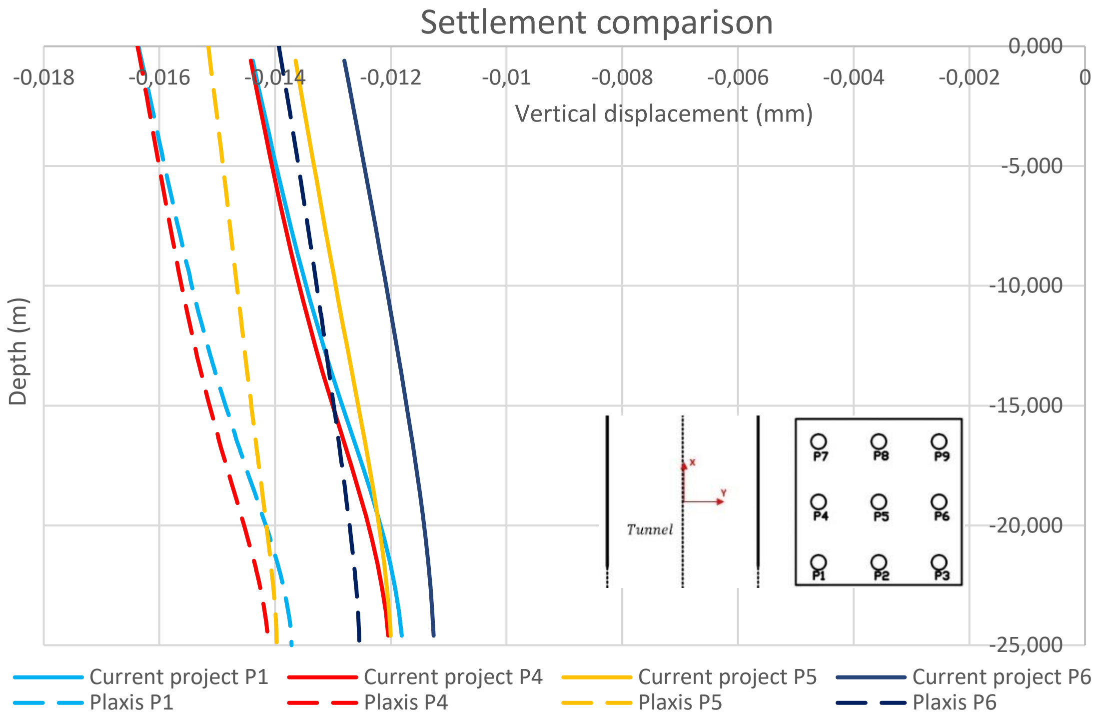


Figure 53. Vertical displacements at time step 100/100 calculated with the MATLAB code developed in this project (continuous lines) and PLAXIS (dashed lines).

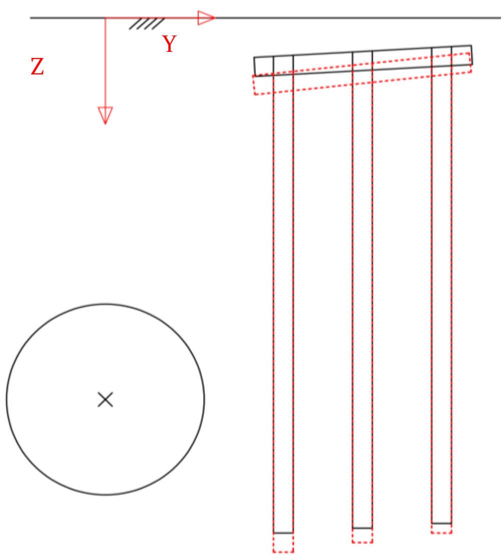


Figure 54. Sketch of settlement between PLAXIS results (dashed red) and this project results (continuous black), not to scale.

It is important to bear in mind throughout this project, that the elemental theory for soil movements from Sagaseta is constraint to incompressible soils. Whereas PLAXIS uses a FEM model only being able to accept a maximum 0,4 Poisson ratio as input. Moreover, the sheer nature of the FEM model taking into account the physical presence of piles and a tunnel is in contrast with the uncoupled approach of the formulae used in this project.

In view of the following comparison, the settlement curves resemble, although at a certain depth, the vertical displacements at the location of the piles ($x\epsilon[9,17]$) is larger for PLAXIS than for this project.

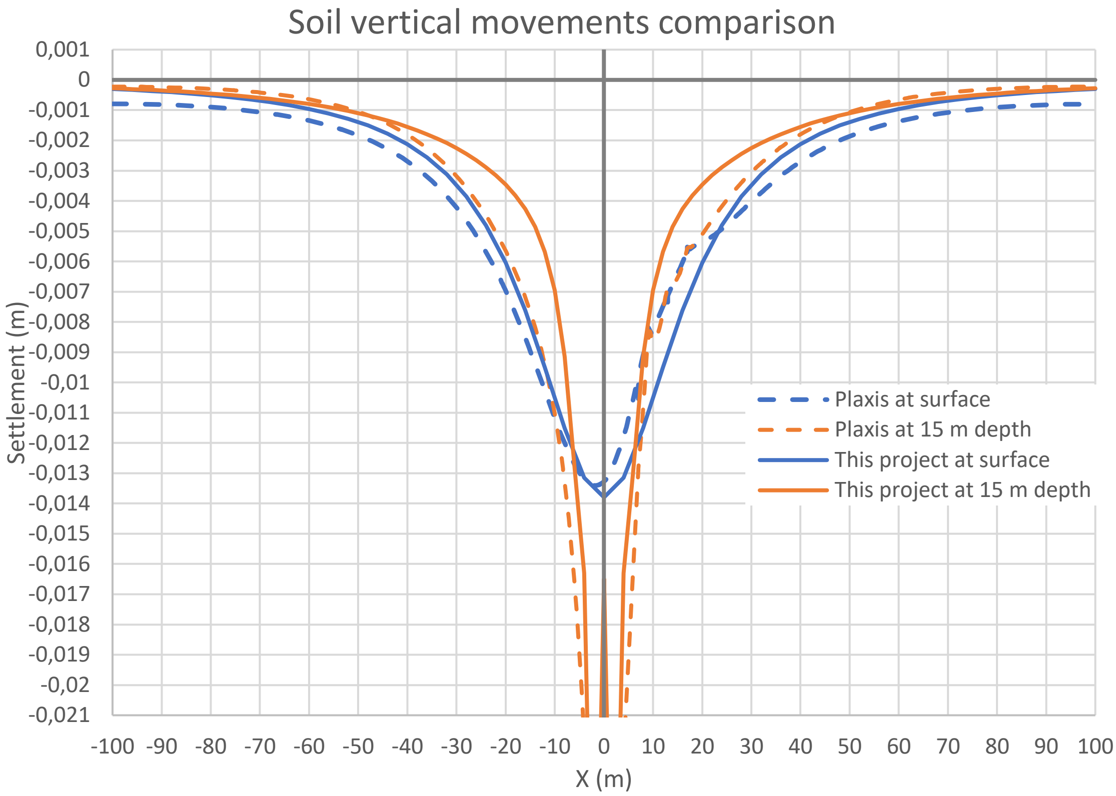


Figure 55. Vertical displacements profiles at surface (blue) and 15 m of depth (orange) calculated with the MATLAB code developed in this project (continuous lines) and PLAXIS (dashed lines).

As for the horizontal displacements, the boundary rigidity effect in PLAXIS comes into action with more relevance, at coordinates $x = -100\ m$ and $x = 100\ m$ the displacements are made zero. Nevertheless, both models provide the same results for the central region (between -50 and 50).

At the group of piles location, just after $x = 13\ m$, PLAXIS tends to present more lateral displacements, also influenced by its taking into account the physical presence of the piles. Which is coherent with results in Figure 53 were the PLAXIS results at $z = 15\ m$ tend to be larger than those for this dissertation code. On the contrary, the situation for the first row of piles is reversed, since at $x = 9\ m$ the MATLAB code results are bigger. The remarkable peaks at $z = 15\ m$ that take place close to the centre ($x = 0\ m$), are due to the proximity to the tunnel, whose crown is actually at those coordinates. However, on the whole, both models do present symmetric horizontal displacements, i.e. as much positive as negative movements.

Therefore, it is important to realise the increase in discrepancy between PLAXIS and the MATLAB code, is attributable to a different calculation approach. It might as well be due to the sum of several differences: the taking into account the presence of the tunnel in the

PLAXIS FEM model, the way of dealing with the interaction between piles, possibly the rigidity effect that a mesh implies within close piles and perhaps approximation errors or assumptions in the developed MATLAB code that may explain the differences between both models in Figure 52 and Figure 53.

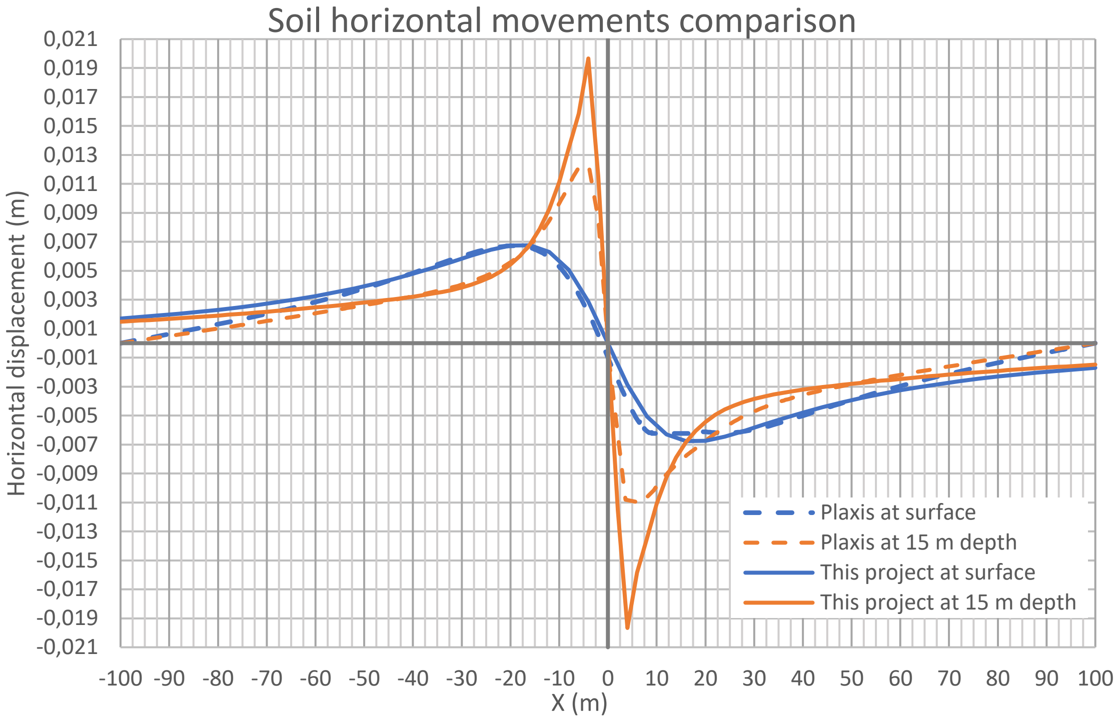


Figure 56. Horizontal displacements profiles at surface (blue) and 15 m of depth (orange) calculated with the MATLAB code developed in this project (continuous lines) and PLAXIS (dashed lines).

6.1.2 Compensation grouting

One of the applications for what the code could be used for, is to figure out the utility of compensation grouting or to work out an efficient injection. Continuing with the problem of section 6.2, a compensation grouting injection is added between the tunnel and the closest piles (there is a 2 meters gap at each side) as seen in Figure 57. It has been modelled as a strip of expanding points, moving parallel to the tunnel, with the following characteristics:

Table 7. Values used for compensation grouting in this section.

Piles	*Value*
Start X coordinate (m)	-20
End X coordinate (m)	20
Y coordinate (m)	7
Depth, Z coordinate (m)	17,8
Total volume (m3)	5

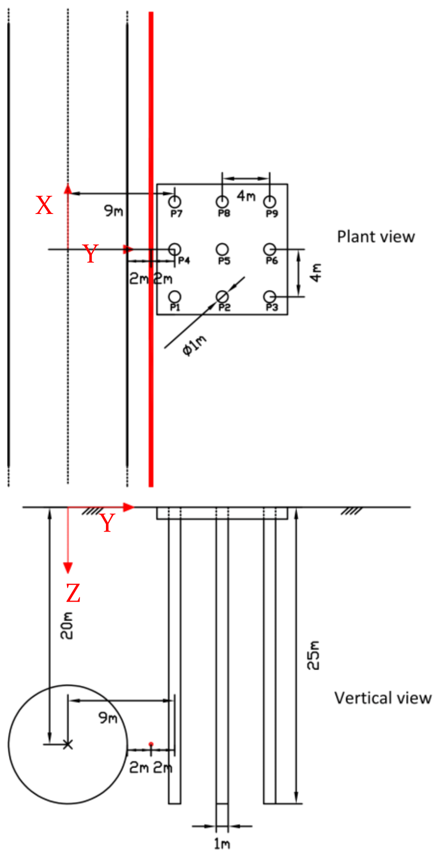


Figure 57. Vertical and plant view of the problem including a strip of compensation grouting points. In red the compensation grouting injection.

In fact, there is no apparent previously known good solution, the choice in this section is genuinely personal. It all boils down to what the user inputs in the MATLAB code developed for this project. With that being said, the results below are just an example of the capabilities of the code. In a hypothetical real case scenario, it would all depend on what the engineer would be able to refine until a certain configuration would be deemed as acceptable so as to ameliorate the at risk foundations.

Lateral Response

Below, the final results (time step 100/100) before and after applying the compensation grouting injection are represented. In fact, one could expect a priori that these results should be zero, which is corroborated in Figure 58.

Since the worse situation takes place when the tunnel approaches the pile in question, it would be wise to see what happens then, approximately at $x = 0$. Especially for piles in the first row, provided previous knowledge says these are the more in distress. The maximum bending moment at $z = 17{,}8\ m$ diminishes, however, those at the head remain more or less the same (Figure 59).

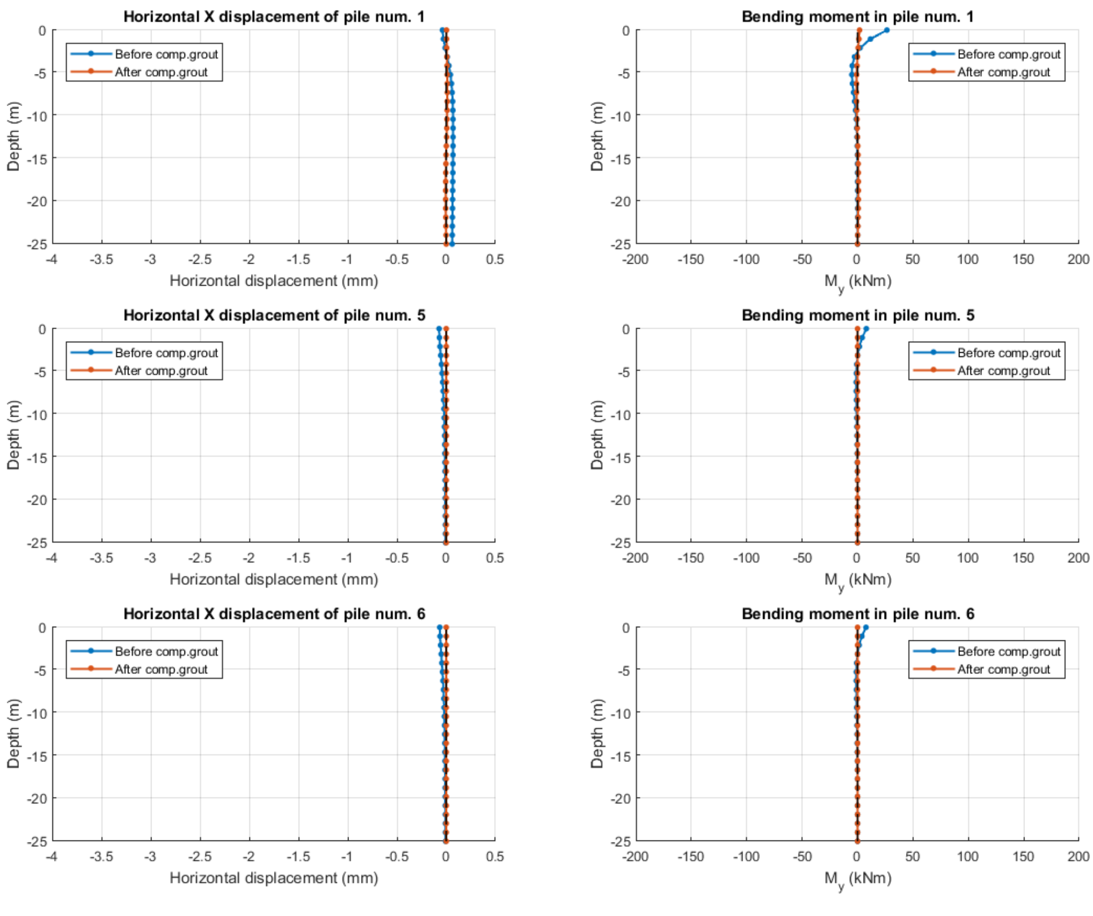


Figure 58. Compensation grouting effects X direction for time step 100/100, calculated with the MATLAB code developed in this project.

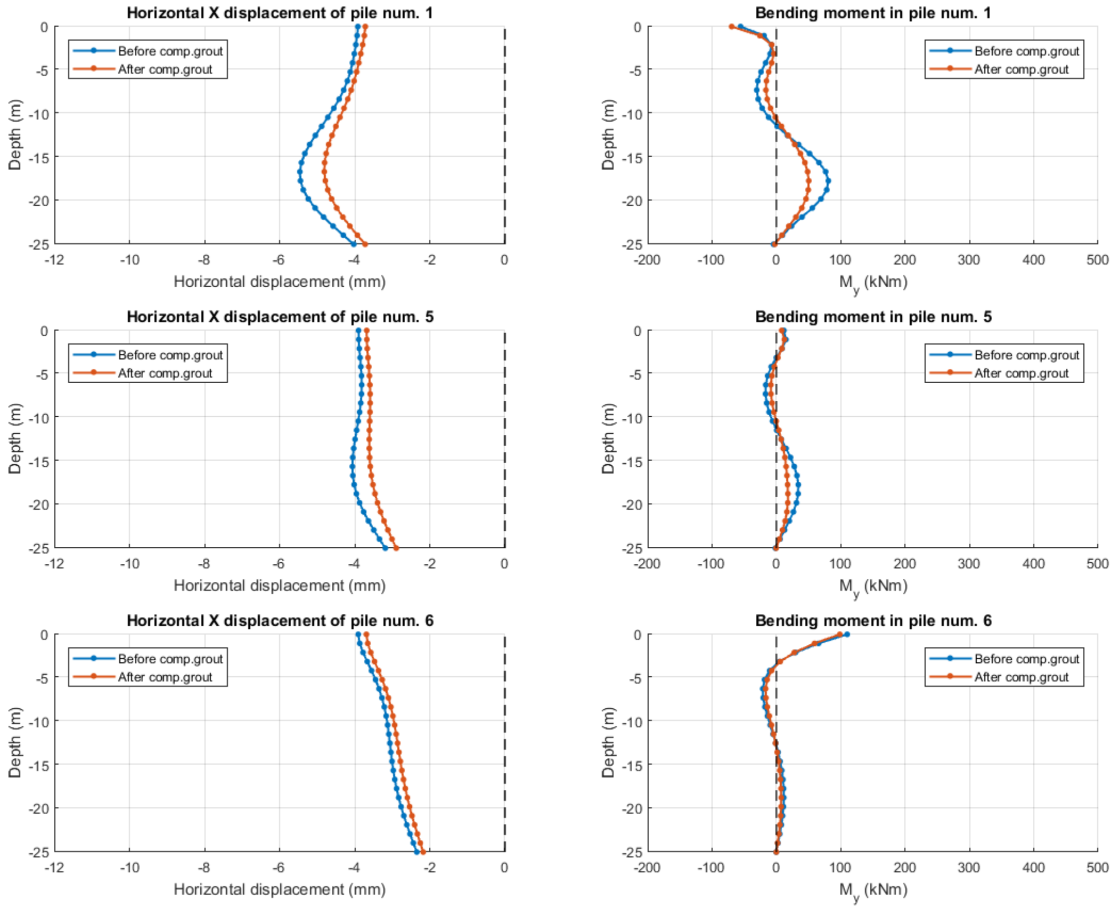


Figure 59. Compensation grouting effects X direction at time step 50/100, calculated with the MATLAB code developed in this project.

It is interesting to observe the effects in the Y direction. In an orange tone, piles displacements at some depth are reduced. At the same time, the bending moment laws also ameliorate, especially at its maximum where pile 1 (representing the first row) reduces from a maximum of 450 kNm to 3 local maximums of around 200-250 kNm. However, it is worth noticing how the fixing moment at the pile head remains the same. On the other hand, piles 5 and 6 are less benefited from the compensation grouting.

It was found that a higher volume injection would result in excessive bending, thus turning the other way around by creating an even larger bending moment. Here lies the importance of finding a reasonable compensation grouting configuration, somehow by means of the trial and error method.

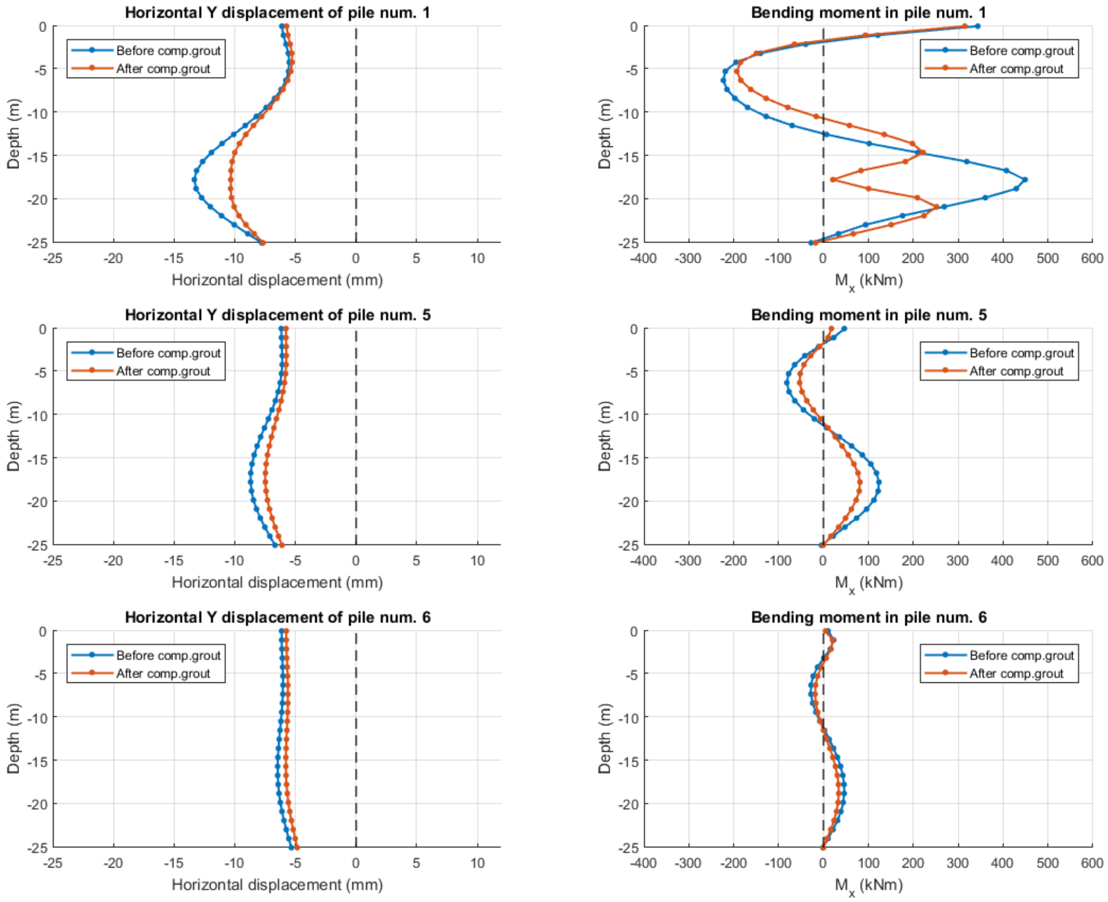


Figure 60. Compensation grouting effects Y direction for time step 100/100, calculated with the MATLAB code developed in this project.

Vertical Response

Concerning the vertical response, there is slightly less settlement, the reason being that the vertical load still induces a great portion of the settlement.

It is quite interesting to see how the maximum axial force indeed decreases. If the aim of the compensation grouting was to reduce the settlement, as if to comply with an allowable settlement requirement, this configuration would not be optimum.

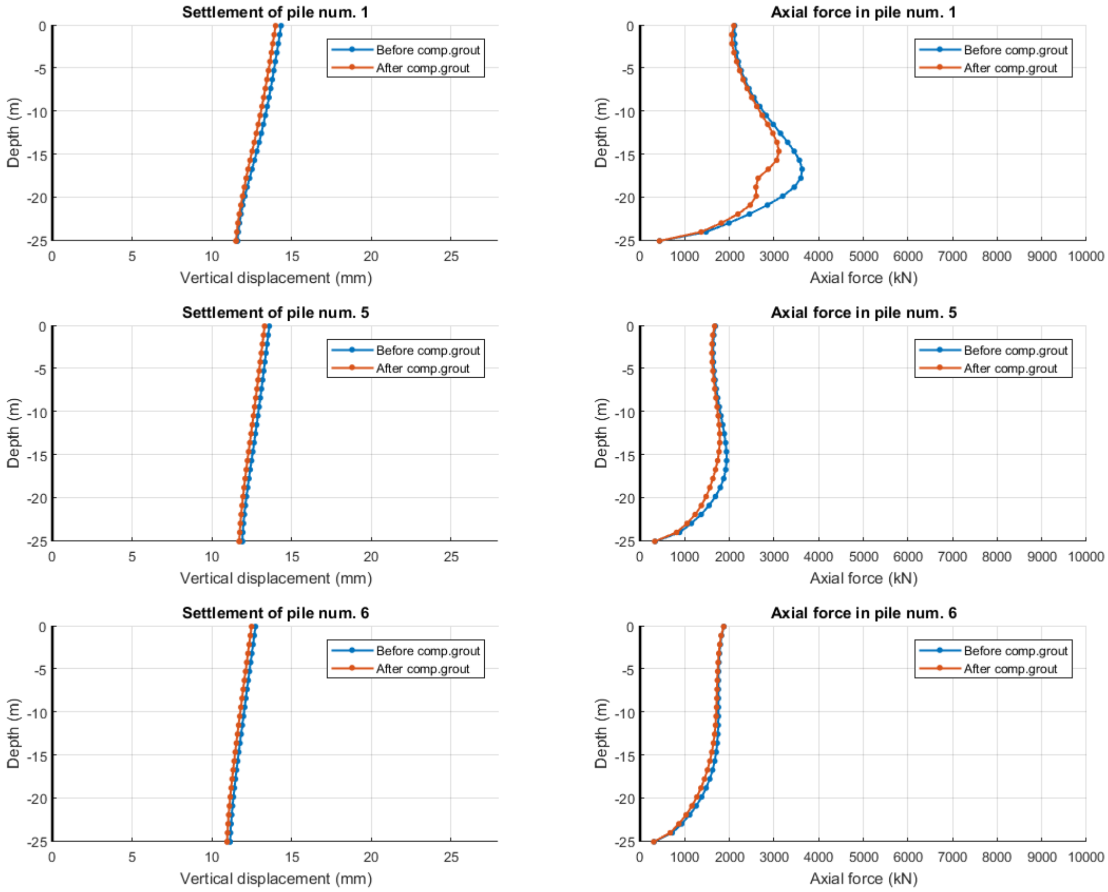


Figure 61. Compensation grouting effects Z direction for time step 100/100, calculated with the MATLAB code developed in this project.

Alternative injection

If the main objective was to reduce the settlement, one straightforward idea could be that of injecting grout right below the group of piles. The current section shows these alternative results so as to prove the previous hypothesis and show the capabilities of the developed MATLAB code, in spite of the fact that providing a realistic compensation grouting is not the main goal of this section.

Table 8. Alternative compensation grouting characteristics.

Piles	*Value*
Start X coordinate (m)	-20
End X coordinate (m)	20
Y coordinate (m)	13
Depth, Z coordinate (m)	26
Total volume (m3)	10

The settlement gets indeed reduced considerably more than 5 mm, at the expenses of a great compression for the piles right above the strip: 2, 5 and 8. Pile 5 serves as an example since it nearly reaches 9000 kN.

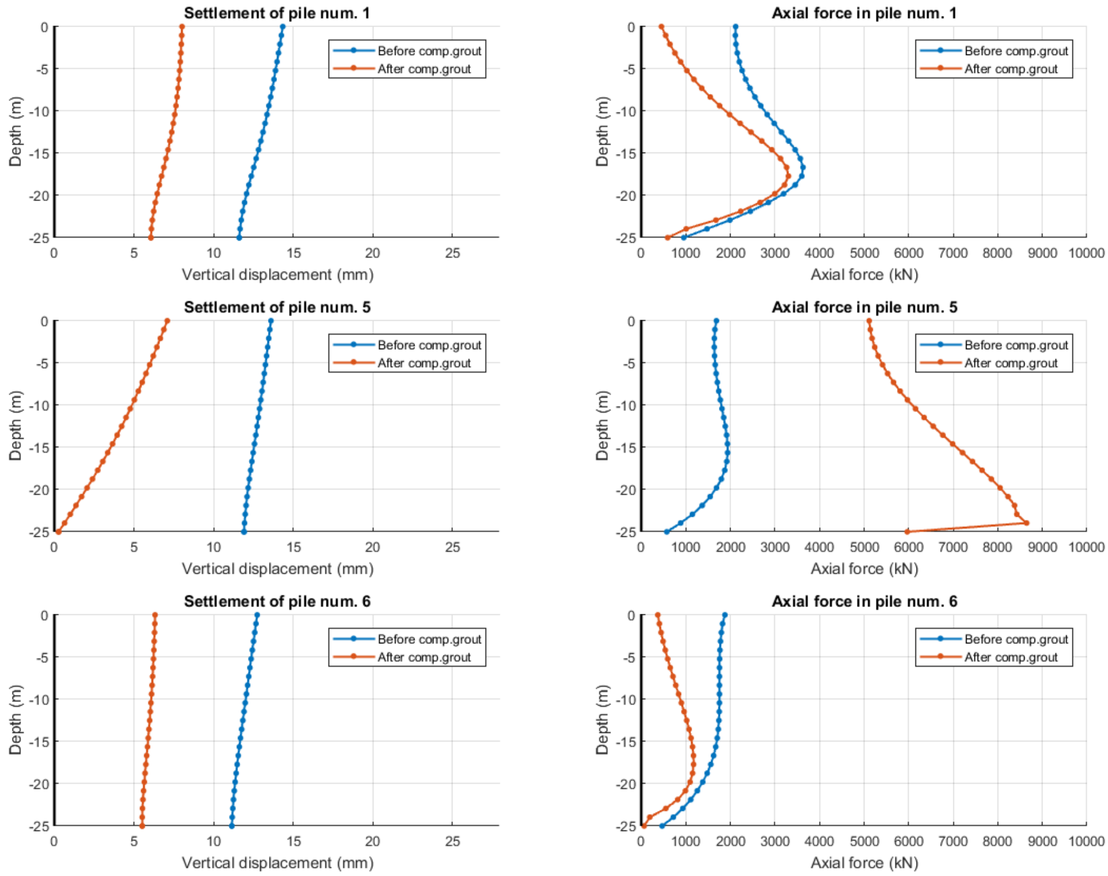


Figure 62. Compensation grouting effects Z direction for time step 100/100, calculated with the MATLAB code developed in this project.

Given the main objective was to reduce the settlement, it should not be surprising to find out there is an increase in the bending moment for some piles. M_Y gets worse for both critical stages: 50/100 and 100/100. There is an extreme maximum bending moment especially for piles in the middle row (represented by pile 5) of up to 2000 kNm.

As for the Y direction, it is clear how piles in the first and last row are really influenced by the compensation grouting strip especially at the tip. In contrast with the "before" scenario, pile 1 moves towards the negative Y and pile 6 moves towards the positive Y. Although pile 1 situation is surprisingly enhanced, pile 6 bending moment increases significantly, from 50 to a maximum of 500 kNm.

As a conclusion, this would not be a realistic compensation grouting injection provided their drawbacks, but has demonstrated to achieve the expected goal of drastically reducing the settlement. Therefore, this exposes even more the need to find a compromise between improved and worsened piles.

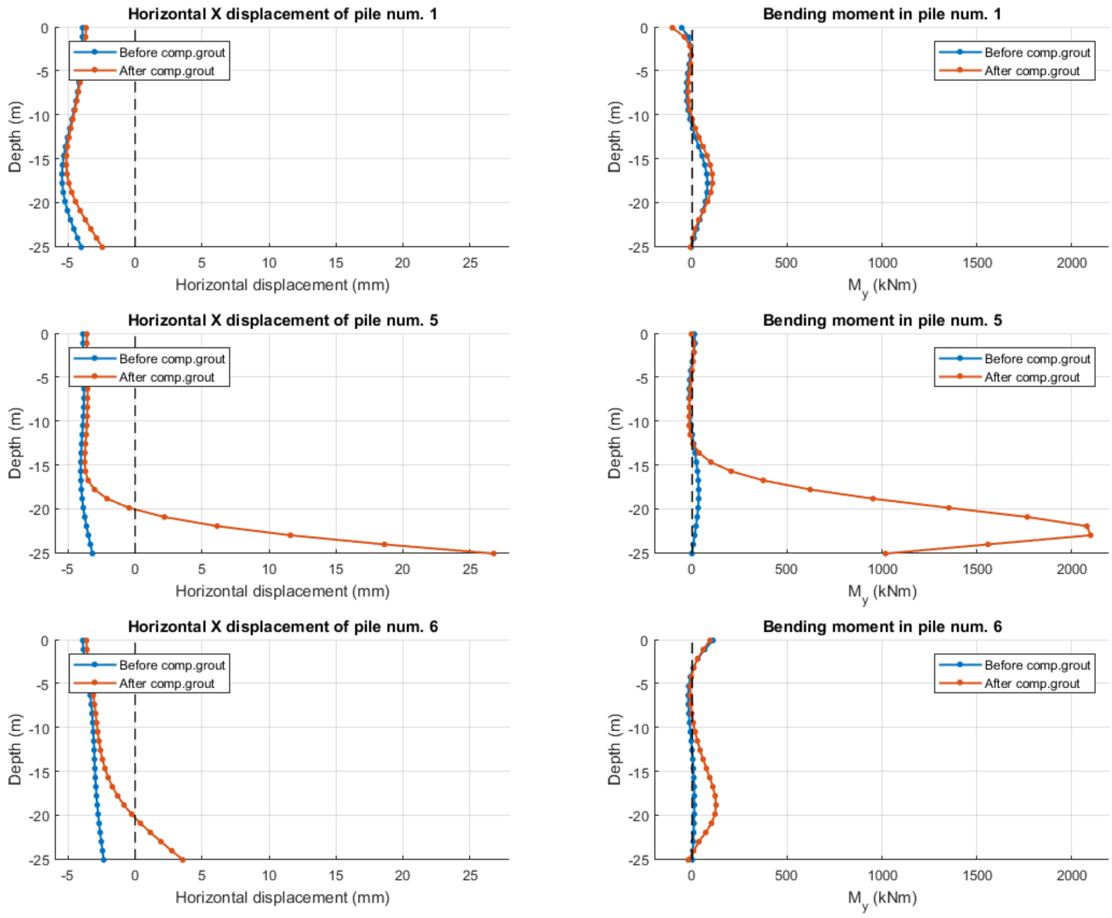


Figure 63. Compensation grouting effects X direction at time step 50/100, calculated with the MATLAB code developed in this project.

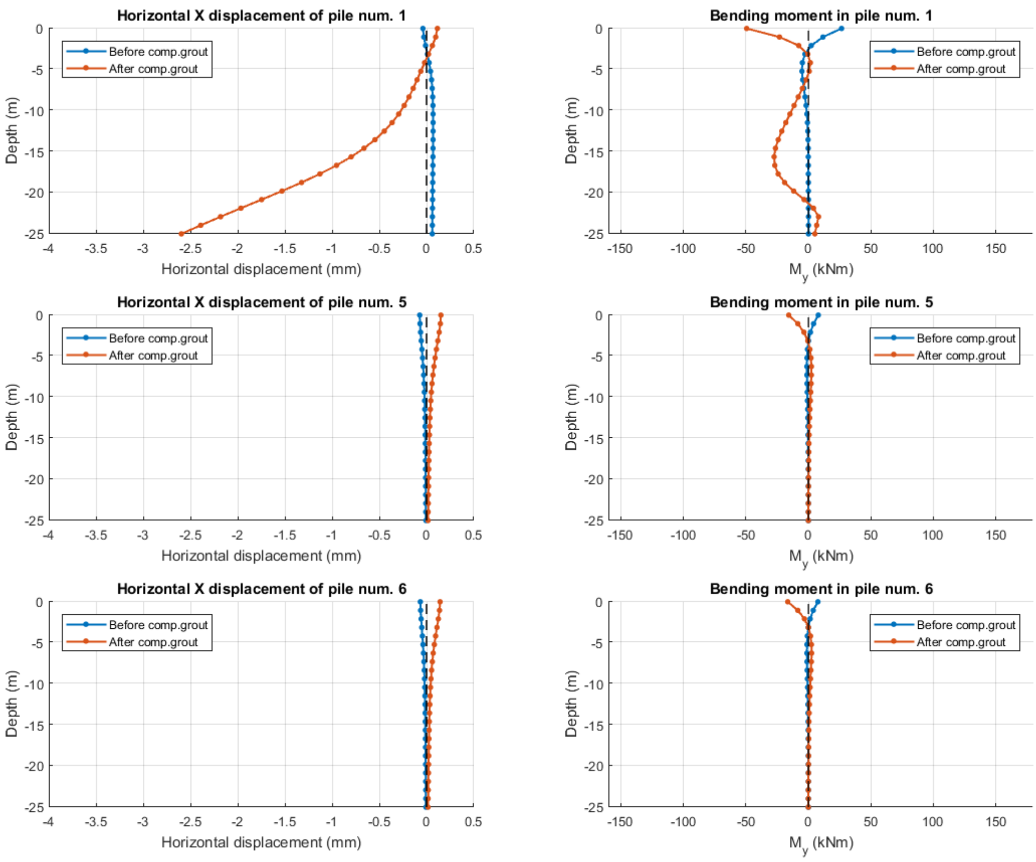


Figure 64. Compensation grouting effects X direction at time step 100/100, calculated with the MATLAB code developed in this project.

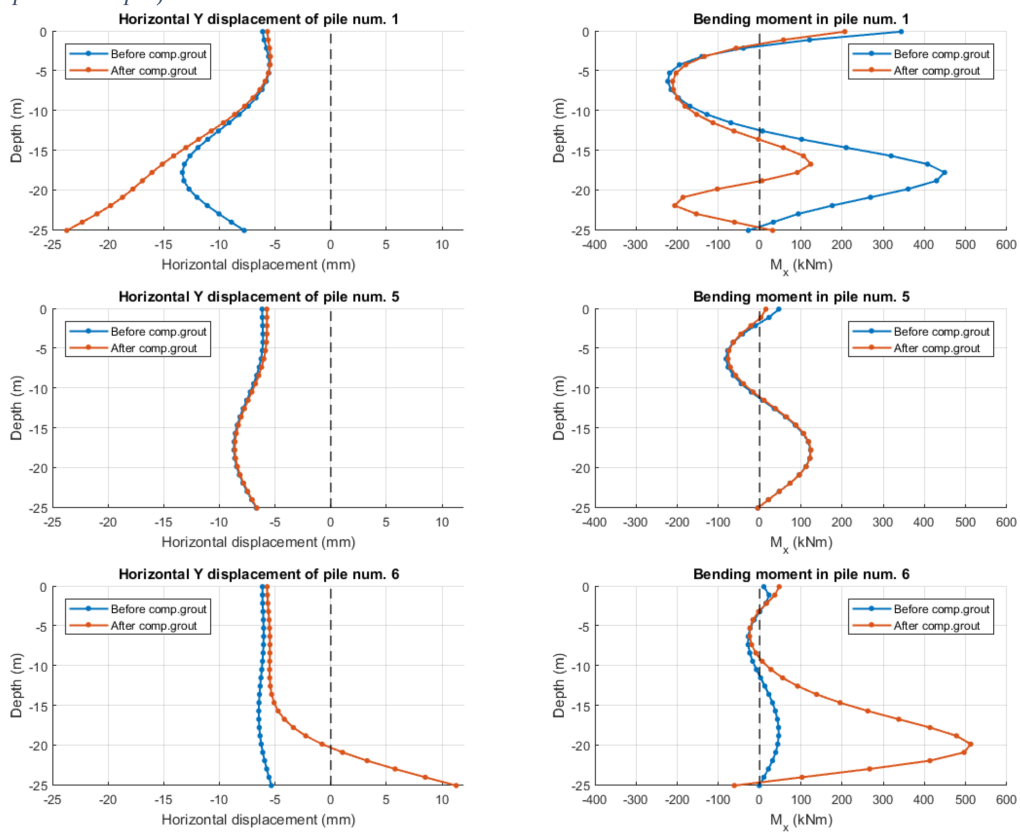


Figure 65. Compensation grouting effects Y direction for time step 100/100, calculated with the MATLAB code developed in this project.

6.2 SAGRADA FAMÍLIA PILE WALL

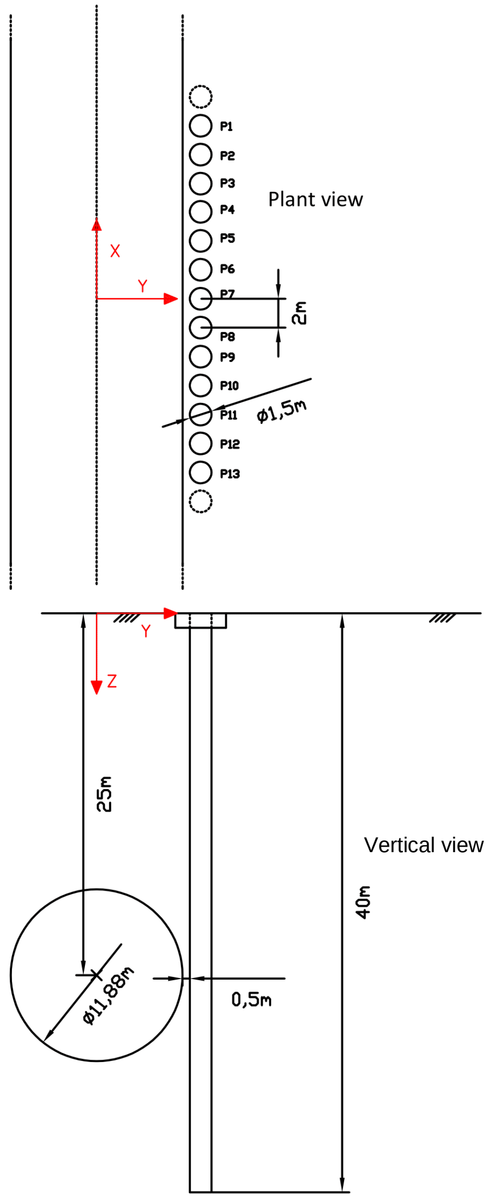


Figure 66. Sagrada Família pile wall problem.

Another interesting application of the developed code is that of analysing the behaviour of the piling screen constructed to shield the Sagrada Família. Measurements confirm the safety of the cathedral by only observing a maximum 2 mm vertical displacement, which can be calibrated to a 0,04% tunnel volume loss (Ledesma and Alonso, 2015), clearly meaning a great execution job of the excavation.

This 2 mm maximum settlement trough has been fitted, within this project code, to a value of 0,14% volume loss tunnel by an iterative approach. We have to consider the differences between the two models, to understand the variation in these results. The calculation for this project was done for a greenfield scenario as plotted in Figure 67 which can be compared to the measurements in magenta and blue. The effect of the pile wall upon the soil cannot be taken into account due to the way the developed MATLAB code was originally created. Unlike in the Ledesma & Alonso article, as can be seen in Figure 67, where this interaction is indeed considered, note how computed results in black do take into account the retaining effect of the wall a drop in settlement is recognisable.

To simulate the Sagrada Família pile wall problem, a hypothesis is made in order to simplify the more than 100 piles executed along 230 m. Only 13 piles are simulated aiming to analyse the middle one (number 7), which would be representative of the majority of the wall, except for those piles at the ends (which are represented by the rest of the piles:

1,2,3…). The reason of this simplification is to dismiss farther piles, that become less influent with distance, hence redundant. In this case, a reasonable computational time was the limiting factor that arose the formulation of this hypothesis. In order for it to be true, one should be able to find convergence in the results. By convergence meaning that the final responses (last time step) for the middle piles tend to be equal, either for lateral or vertical results.

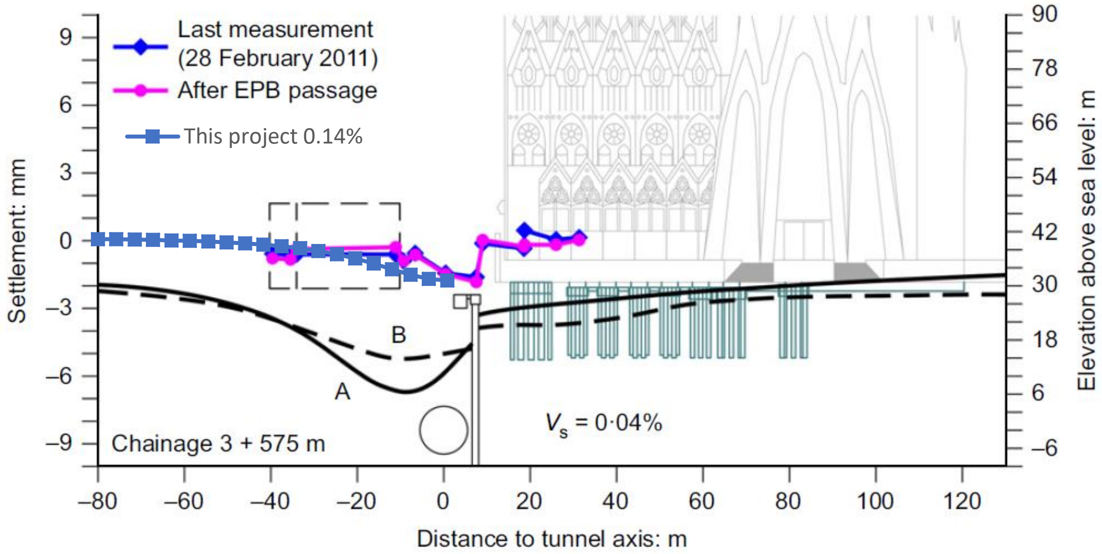


Figure 67. Adapted from Ledesma & Alonso (2015). Superposed in blue are the results calculated with this project code.

The nature of the set-up is a row or piles whereas it was a squared group for the previous cases (sections 6.1 for example), however, the learnt knowledge can similarly be applied with some nuances. The lateral deflection behaves as expected. From the variation of the maximum bending moments M_Y tops the 300 kNm (Figure 70), whereas M_X reaches the 400 kNm, slightly more at the extremes (pile 1). From Figure 68 and Figure 69 the fixing moment at the head is relatively small, less than 100 kNm in any time step for both M_Y and M_X.

Again the abrupt change in slope around 5 m before the tunnel meets the pile (Figure 70), defines the region where the maximum displacement occurs at the head versus that occurring at around the tunnel's depth, in this case 23 m (see Figure 69).

In what concerns deflection in the Y direction, it is interesting to see how the maximum final displacement of pile 1 is slightly bigger than for piles 3, 5 and 7. What is more important is the convergence of piles 3, 5 and 7 after the tunnel has passed. Especially piles

5 and 7 that present the exact same results for both displacements and bending moment. Consequently, it can be inferred that the interaction behaviour between piles is already well taken into account after pile 3. This fact supports the simplification that was assumed before, in simulating the whole pile wall with 13 piles.

On this matter, as for the X direction, the same reasoning applies, despite the fact that the last time step is less important given all the piles return to its original position and null bending moment for a sufficiently long tunnel. However, what is relevant is that all the piles present a maximum bending moment M_Y around 300 kNm when the tunnel meets them and a bit over 2 mm of maximum displacement (Figure 70).

As for the settlement analysis, it provides more insight where one can see how the maximum final settlement of pile 3 is slightly above piles 5 and 7, yet these two do converge (Figure 73). These last piles vary in no more than 0,015 mm for the settlement analysis.

Therefore, a minimally realistic pile setup model would be that of at least 9 piles, provided that pile number 5 would be representative of the piles in the middle of the wall and the rest would correspond to those at the extremes.

As it would be expected from the experience of tunnel-pile interactions (for example section 5.2), the maximum compression takes place a bit above the tunnel centreline, 23 m down in this case. Observe how the said maximum axial force is bigger at the end piles (pile 1) than at pile 7, which is benefited from being surrounding by some piles, 1050 kN versus 700 kN, respectively. Finally, the maximum settlement takes place at the head: 0,47 mm for pile 1 and 0,33 mm for pile 7. Corroborating the article's conclusion, that the boring machine induced so little movements that were almost imperceptible; at the accuracy degree of the measuring instruments.

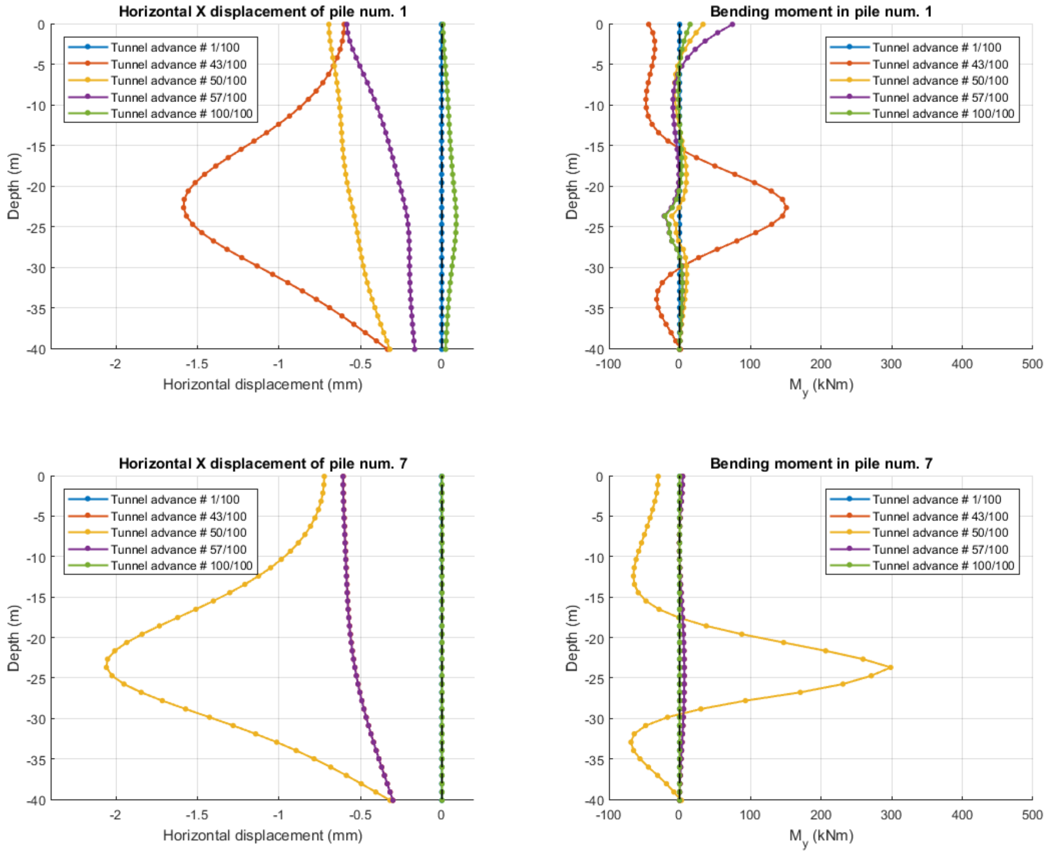


Figure 68. Displacements in the X coordinate (left plots) along corresponding bending moment (right plots) generated by a tunnel excavation represented for different time steps, calculated with the MATLAB code developed in this project.

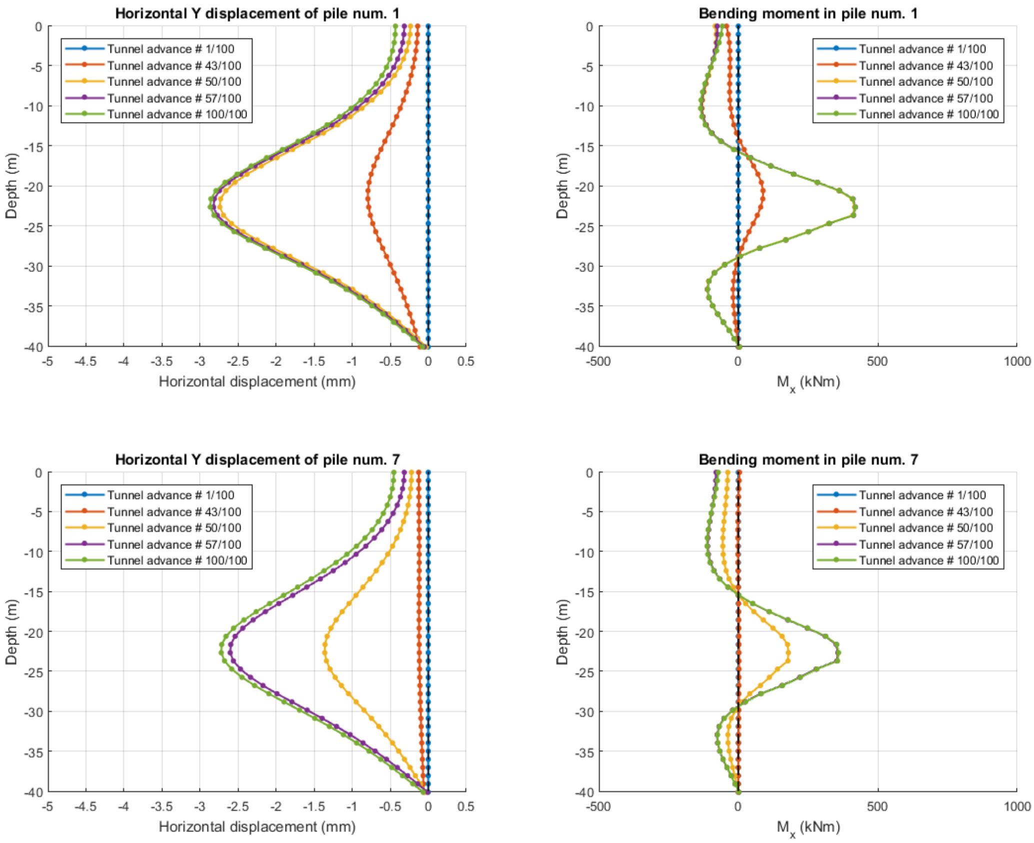


Figure 69. Displacements in the Y coordinate (left plots) along corresponding bending moment (right plots) generated by a tunnel excavation represented for different time steps, calculated with the MATLAB code developed in this project.

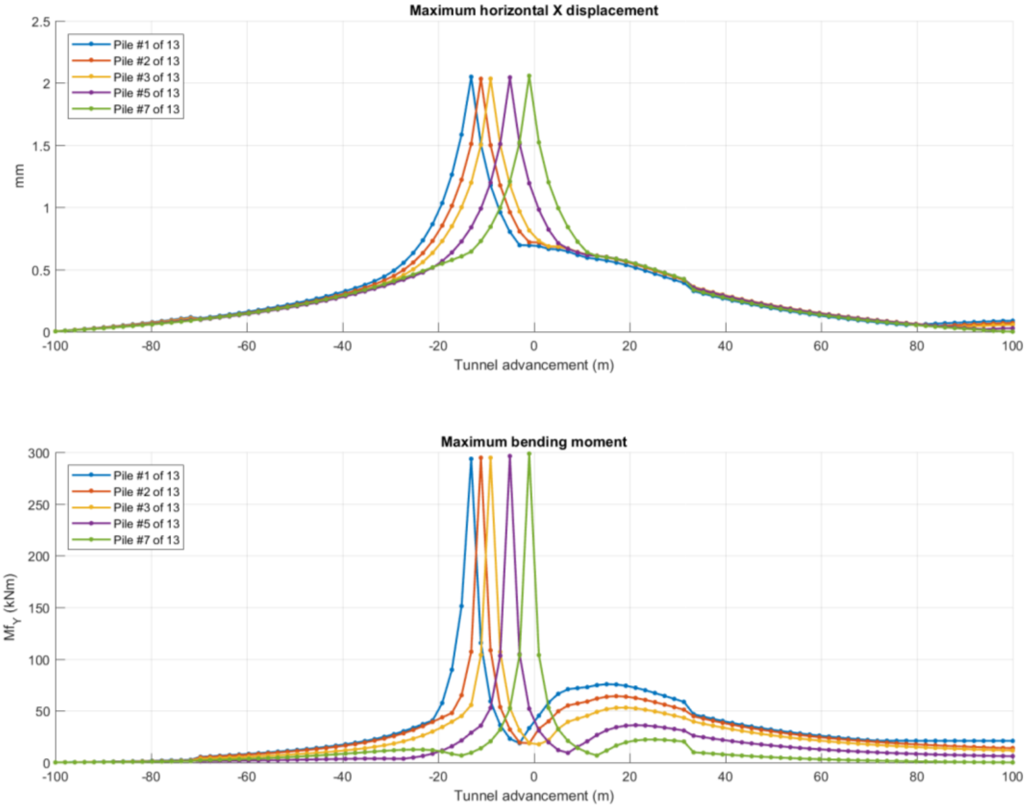


Figure 70. Variation of the maximum lateral X deflection and corresponding bending moment for the Sagrada Família analysis, calculated with the MATLAB code developed in this project.

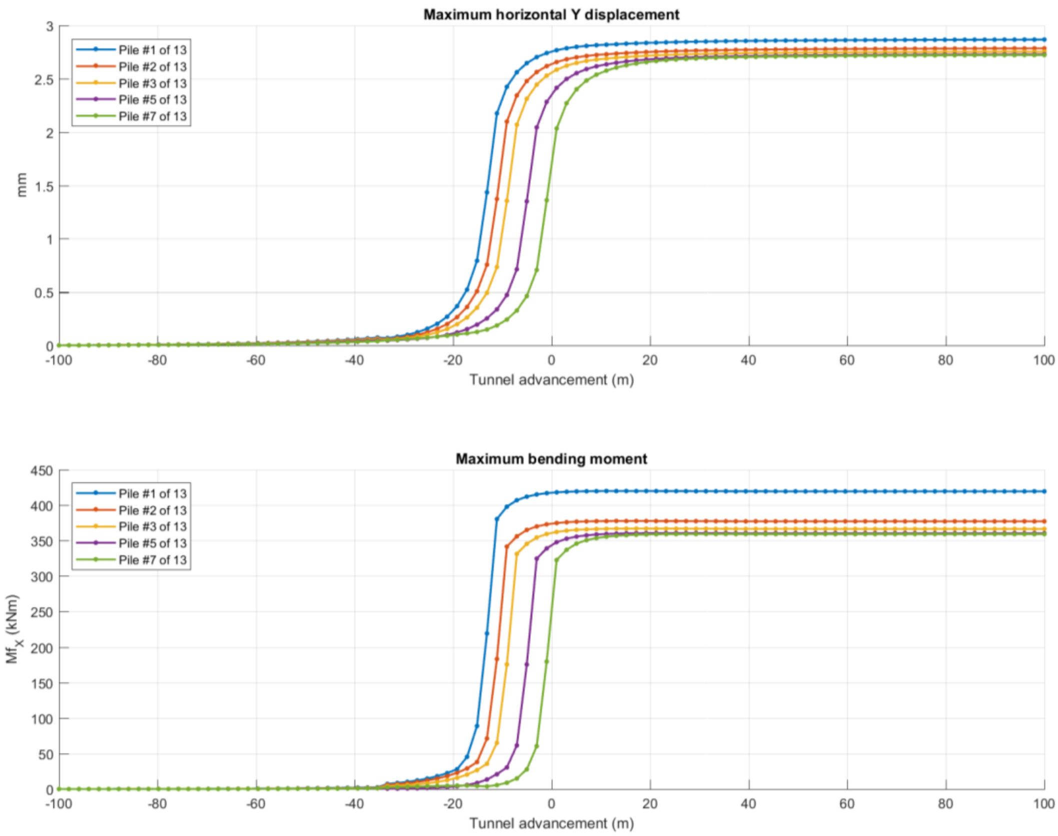


Figure 71. Variation of the maximum lateral Y deflection and corresponding bending moment for the Sagrada Família analysis, calculated with the MATLAB code developed in this project.

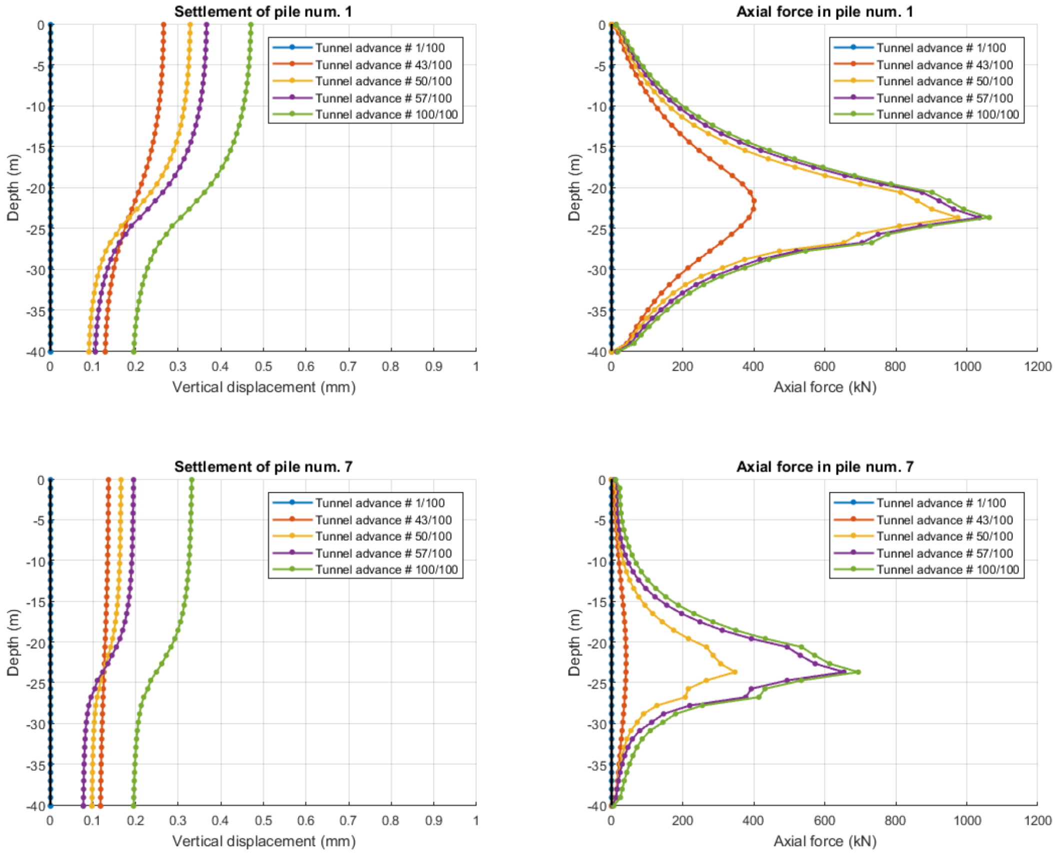


Figure 72. Displacements in the Z coordinate (left plots) along corresponding bending moment (right plots) generated by a tunnel excavation represented for different time steps, calculated with the MATLAB code developed in this project.

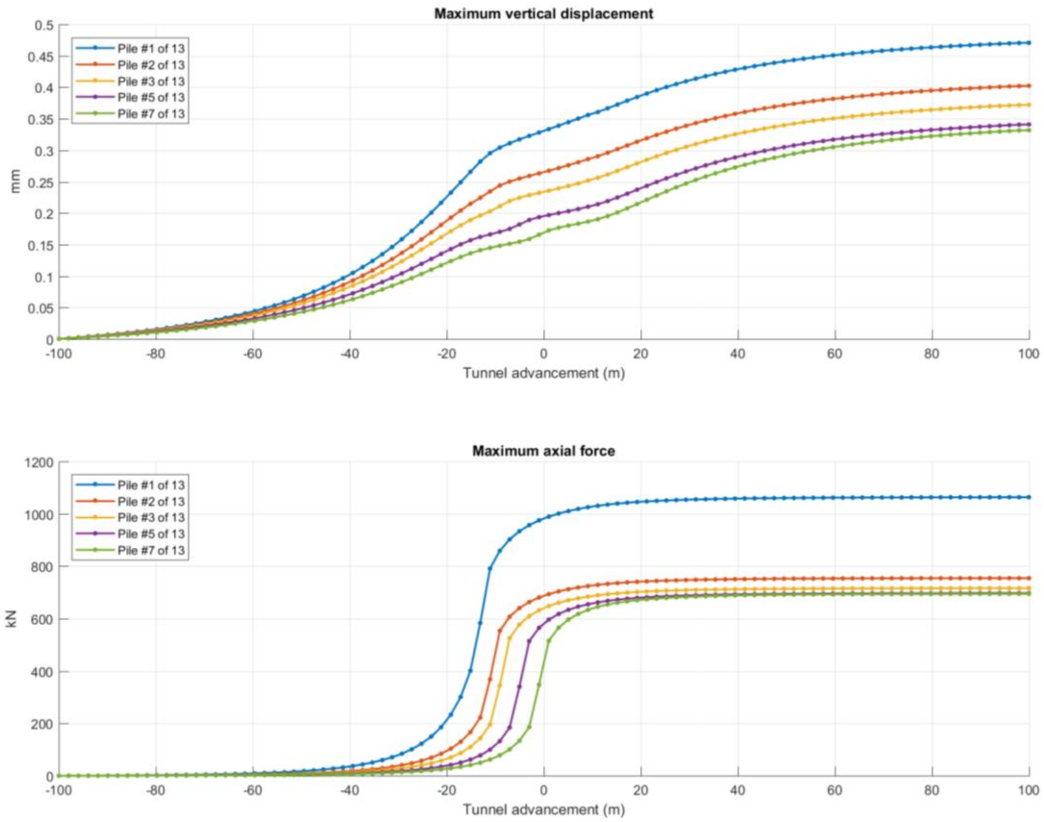


Figure 73. Variation of the maximum settlement and corresponding axial force for the Sagrada Família analysis, calculated with the MATLAB code developed in this project.

7 CONCLUSIONS

An operational code has been developed to analyse piled foundations response due to a nearby tunnel excavation. Conclusions may be formulated over three areas: firstly in terms of the author's learning process as this dissertation seen as a milestone upon completion of his civil engineering studies; secondly, in relation to the project objectives; and thirdly, regarding further work.

Throughout the dissertation, the process of creating a functional code has been the core element. Not only in view of the computational engineering-skills side of it, but most importantly in relation to the soil mechanics and structural knowledge. The MATLAB language has served both as a means and as an end, in learning terms. Good comprehension of the presented formulae is evidenced when it is, indeed, manipulated and fitted into an algorithm, especially when it has to properly work along with other formulation.

The performance of the developed code is deemed as satisfactory, as it can function for a number of different case studies. For instance, greenfield surface displacements, squared or asymmetrical piled-groups, pile walls and for a swelling injection of compensation grouting. Even more, it could be used to simulate an expanding layer of soil.

As for the actual tunnel-pile interaction, the basic mechanisms have been thoroughly described, such as it is a directly proportional function to the proximity and intensity of the ground loss source. The rigidity effect of the pile cap strongly affects the distribution of external loads coming from the superstructure, and finally, the shielding effect of a group of piles just boils down to the subsurface soil movements due to subsurface forces, as in the Mindlin problem (1937). Results have been computed for an advancing tunnel, and it has been interesting to analyse the piles behaviour along the tunnel advancement. From a safety point of view, conclusions point towards two critical parts in any pile: the head and a region around the tunnel centreline axis depth. The most affected parts change location, as the tunnel moves in the vicinity of the group of piles.

The results provided by the MATLAB code are quite good when verified against existing software. However, there are some intrinsic limitations in the fundamental elastic theory used. For instance, the Sagaseta point ground volume loss formulation is constraint to an elastic, undrained, homogeneous, isotropic and incompressible soil. In this case, let us remark that the calculations are independent of the soil elastic modulus. The fact that it was developed for an incompressible soil translates that the Poisson ratio must be 0,5, which limits the applicability in real case scenarios. In terms of further work and enhancement of the MATLAB code, Sagaseta itself discusses (1998) a new formulation published in 1996 by A. Verruijt & J.R. Booker under the name of *Surface settlements due to deformation of a tunnel in an elastic half plane* where the Poisson ratio is a variable to introduce. Therefore, it would be reasonable to explore and incorporate a 3D formulation applicable to the existing MATLAB code.

In a nutshell, the tunnel-pile interaction study has been proven to be reliably calculated with the developed code, that could as well be used as a real first approach tool in some geotechnical applications, making clear the effectiveness of simplified semi-analytical procedures.

7.1 CONCLUSIONS ON FURTHER WORK

Aiming beyond the objectives of the current dissertation, the code is open to further work, which may be channelled, for example, towards the refinement of the soil model. For instance, to account for different soil strata, thus different elastic modulus. Or just a linearly increasing soil modulus, nevertheless, in this regard, the sole fact that the Mindlin subsurface force equation is only applicable to a soil with constant modulus, would still make the solution an approximation.

Another area where to enhance the performance of the code, would be in including an option to analyse end-bearing piles or even battered piles, Poulos & Davis (1980) provide basic theory in relation to that. This dissertation has carried out an analysis for floating piles.

Tunnel-wise, the assumptions taken can be improved by following the precise formulation developed by C. González & C. Sagaseta (2001), that accounts for tunnel ovalisation. Along the same lines, the tunnel modelling may be expanded by including changes in direction or pitch, i.e. a tunnel with trajectory.

In view of the analysis of the Sagrada Família pile wall, it would be interesting to adapt the code so that for any soil line profile, the movements take into account the effect of a pile or piles.

With that being said, the essence of a simplified semi-analytical procedure is to provide a first realistic calculation for any problem in question. Therefore, in engineering terms, it would be more interesting to expand the applicability of the developed code rather than unbalancing the compromise between computational-costs-results, only for the benefit of small refinements in the results. On the other hand, it is indeed relevant to consider the limitations of the Sagaseta fundamental theory, as described in the conclusions section, and deepen into the improvement of the formulation for any Poisson ratio soil.

8 BIBLIOGRAPHY

Alonso, E., Sauter, S., & Ramon, A. (2015). Pile groups under deep expansion: a case history. *Canadian geotechnical journal*, 1111 - 1121.

Brinkgrever, R., Kumarswamy, S., Swolfs, W., & Foria, F. (2017). Plaxis 2017 Manual. Netherlands: PLAXIS.

Broere, W., & Brinkgreve, R. (2002). Phased simulation of a tunnel boring proces in soft soil. *Numerical Methods in Geotechnical Engineering*, 529-536.

González, C., & Sagaseta, C. (2001). Patterns of soil deformations around tunnels. Application to the extension of Madrid Metro. *Computers and Geotechnics*, 445-468.

Huang, M., Zhang, C., & Li, Z. (2008). A simplified analysis method for the influence of tunneling on grouped piles. *Tunnelling and Underground Space Technology*, 410-422.

Ledesma, A., & Alonso, E. E. (2015). Protecting sensitive constructions from tunnelling: the case of World Heritage buildings in Barcelona. *Géotechnique*, 914-925.

Lee, K., Rowe, K., & Lo, K. (1992). Subsidence owing to tunnelling. Estimating the gap parameter. *Canadian Geotechnique*, 929-940.

Poulos, H., & Davis, E. (1991). *Elasic Solutions For Soil And Rock Mechanics.* Sydney, Australia: Center for Geotechnical Research, The University Of Sidney.

Poulos., H., & Davis, E. (1980). *Pile Foundation Analysis and Design.* Sydney: The University of Sydney.

Sagaseta, C. (1987). Analysis of Undarined Soil Deformation Due To Ground Loss. *Géotechnique*, 301-320.

Sauter, S. (2012). Analysis of Pile Group Foundations Using Elastic Solutions. Barcelona: ETSECCPB.

Verruijt, A., & Booker, J. (1996). Surface settlements due to deformation of a tunnel in an elastic half plane. *Géotechnique*, 753-756.

ANNEX: MATLAB CODE

Below, the developed routines and subroutines, the different list levels indicate the nested degrees of each function.

1. Main.m
    1.1. Assemble_piles.m
    1.2. V_loss_points.m
        1.2.1. Tunnel_global_coords.m
        1.2.2. Tunnel_local_coord.m
    1.3. Boussinesq.m
    1.4. Sagaseta_3D.m
        1.4.1. Sagaseta_3D_integral.m
    1.5. Horizontal_analysis.m
        1.5.1. Horizontal_assemble_pile_structure.m
        1.5.2. Horionztal_assemble_mindlin.m
            1.5.2.1. Horizontal_mindlin_inpile.m
            1.5.2.2. Horizontal_mindlin_pile2pile.m
    1.6. Vertical_analysis.m
        1.6.1. Vertical_assemble_mindlin.m
            1.6.1.1. Vertical_mindlin_base.m
            1.6.1.2. Vertical_mindlin_shaft.m
            1.6.1.3. Vertical_mindlin_pile2pile.m
        1.6.2. Vertical_pile_cap.m
    1.7. Results_analysis.m
    1.8. Results_plots.m
2. Compensation_grouting.m

```
clear all; clc; close all;
% File name: Main.m
% Author: Genis Majoral Oller
% Date: 16/7/2018
% For Civil Engineering bachelor's degree dissertation
% Computation of group of piles' displacements and stresses due to a tunnel
% excavation using Boussinesq, Mindlin and Sagaseta theory

                              tic
%% %-----------------------------SOIL------------------------------%%%
% ---------------------------- INPUT DATA
soil_E=100000; %kN/m2
soil_v=0.5;
soil=[soil_E,soil_v];

%% %-----------------------------PILES------------------------------%%%
% ---------------------------- INPUT DATA

piles_num  = 9;                %number of piles
piles_head_type = 1;           % "0" for free, "1" for casted
piles_length = 20;             %m
piles_dia=1;                   %m
piles_E= 3*10^7;               %kN/m2
piles_I= pi/4*(piles_dia/2)^4;
num_elem = 50;                 % Number of elements per pile: minimum 10
find_surface=0;                % Find surface settlement defined by
                               % surface_pts below, no piles. 1-> YES. 0-> NO

surface_pts=[zeros(1,num_elem); %1st row is X coordinate 2nd Y coord Z depth
            linspace(-100,100,num_elem);
            15*ones(1,num_elem)];

piles_input = [-4 9 0.1;      % Head coords [x,y,z(@ cap)] (m) 1row per pile
              -4 13 0.1;
              -4 17 0.1;
              0 9 0.1;
              0 13 0.1;
              0 17 0.1;
              4 9 0.1;
              4 13 0.1;
              4 17 0.1;];


    piles_forces_top = [0 0 0 0 0;
                        0 0 0 0 0;
                        0 0 0 0 0;
                        0 0 0 0 0;
                        0 0 0 0 0;
                        0 0 0 0 0;
                        0 0 0 0 0;
                        0 0 0 0 0;
                        0 0 0 0 0;];


% units: piles_froces_top=[Fx, Fy, Fz, Mx, My] (in kN)

surface_forces=[0 0 0 0 0 0 0 0];

% units: surface_forces=[Fx Fy Fz x1 x2 y1 y2 0/1/2]
% 0 there are no surface forces
% 1 there is surface force
% 2 thre is regtangular surface load defined by corners x1 x2 y1 y2
%-----------------------------------------------------------------------

```

```
piles=assemble_piles(piles_input,piles_forces_top,piles_num,num_elem,...
                        piles_length,find_surface,surface_pts);

%% %-----------------------------TUNNEL-------------------------------%%%
% ----------------------------- INPUT DATA

depth=20;                    %meters - depth of tunnel axis
tun_coord=[0,0,depth];          % Coordinates it must be at x=0; y=0;
time_step=1;                 % number of tunnel discretizations
dia_0=10;                    % Tunnel diameter
V_loss = 1/100;               % Percentage of areal volume loss: e0
n_points=10;                  % Points in every XS
v=linspace(-50,50,time_step);   % [coord X, coord X,#discretisations]
                        % tunnel length of action

[V_loss_pts]=V_loss_pts(depth,tun_coord,time_step,dia_0,V_loss,n_points,v);

                     time_geom=toc;
                     fprintf('Geometry created --> ok!\n')
                     fprintf('    Elapsed time: %.1f s \n',time_geom);


%% %---------------BOUSSINESQ AND SAGASETA DISPLACEMENTS----------------%%%
% ---------------------------------------------------------------------- %
tic

soil_bous=boussinesq(piles, num_elem,soil, surface_forces,time_step,piles_num,V_loss_pts);

if V_loss==0

    soil_sagaseta=zeros(num_elem,4,piles_num,time_step);
else
[soil_sagaseta]=sagaseta_3D(piles,V_loss_pts,...
                     time_step); %total soil displacement
end
soil_strain=soil_bous+soil_sagaseta;
% soil_strain: [num_elem,(x,y,z),piles_num,time_step]
                     time_soil=toc;
                     fprintf('Soil displacements computed --> ok!\n')
                     fprintf('    Elapsed time: %.1f s \n',time_soil);


%% %----------------------PILE-GROUP HORIZONTAL "X" AND "Y" PROBLEM----------------------%%%
% ---------------------------------------------------------------------- %
[a,b]=size(piles_forces_top);
ppp=zeros(a,b,length(soil_strain(1,1,1,:)));

coord=1; %x direction
tic
[Is_global_x,BC_global_x,Ap_global_x,D_global_x,p_global_x,...
    w_global_x,w_final_x,w_head_x,BC_change_x,ppp,w_dif_x2,...
    soil_strain_auxx] = horizontal_analysis( piles,soil,piles_dia,...
            piles_forces_top,piles_E,piles_I,piles_head_type,piles_num,...
                        soil_strain,coord,num_elem,time_step,ppp);

                 time_x=toc;
                 fprintf('Horizontal X analysis computed --> ok!\n')
                 fprintf('    Elapsed time: %.1f s \n',time_x);


coord=2; %y direction
tic
[Is_global_y,BC_global_y,Ap_global_y,D_global_y,p_global_y,...
    w_global_y,w_final_y,w_head_y,BC_change_y,ppp,w_dif_y2,...
```

```
    soil_strain_auxy] = horizontal_analysis( piles,soil,piles_dia,...
            piles_forces_top,piles_E,piles_I,piles_head_type,piles_num,...
                        soil_strain,coord,num_elem,time_step,ppp);

                time_y=toc;
                fprintf('Horizontal Y analysis computed --> ok!\n')
                fprintf('    Elapsed time: %.1f s \n\n',time_y);

%% %----------------------PILE-GROUP VERTICAL "Z" PROBLEM----------------------%%%
% ------------------------------------------------------------------------ %
tic;
[p_global_z,p_global_z2,v_global,v_global2,v_global3,v_final,...
    D_global_z,Ap_global_z,Is_global_z,BC_z,BC_change_z,u_dif_u,u_dif_o,...
    ppp,u_head_z,u_mean,Z_head,u_dif_z,soil_strain_auxz]=...
    vertical_analysis(piles,piles_num,piles_dia,piles_head_type,...
    piles_length,soil,piles_E,soil_strain,num_elem,piles_forces_top,ppp,find_surface);
                time_z=toc;
                fprintf('Vertical Z analysis computed --> ok!\n');
                fprintf('   Elapsed time: %.1f s \n',time_z);


%% %-------------------------RESULTS' ANALYSIS--------------------------%%%
% ------------------------------------------------------------------------ %

[Mf_final_x,Mf_final_y,Q_global_x,axial_final]=results_analysis(...
    piles_num,piles_length,piles_dia,piles_I,piles_E,piles_forces_top,...
    num_elem,w_global_x,w_global_y,v_global3,p_global_z2,time_step,...
    piles_head_type,ppp,find_surface);

%% %-----------------------------PLOTTINGS------------------------------%%%
% ------------------------------------------------------------------------ %

results_plots(num_elem,piles,w_final_x,w_final_y,v_final,...
    Mf_final_x,Mf_final_y,axial_final,V_loss_pts,dia_0,find_surface,piles_length,soil_strain_auxz,soil_strain_auxy);

%% %----------------------------SAVE WORKSPACE VARIABLES

 s_name="sample_name";
save(s_name)
```

```
% File name: assemble_piles.m
% Author: Genis Majoral Oller
% Date: 1/3/2018
% For Civil Engineering bachelor's degree dissertation
% Enambling the "piles" matrix containing the elements of each pile
% OUTPUT: piles=size[num_elem,4,piles_num]
function piles=assemble_piles(piles_input,piles_forces_top,piles_num,num_elem,piles_length,find_surface,surface_pts);

if find_surface==0

        [rows,waste]=size(piles_input);
        [rows1, waste1]=size(piles_forces_top);

        if piles_num==rows && rows==rows1

        else
            error('Please check number of piles and piles´ position');

        end

        clear rows waste rows1 waste1;

        piles=zeros(num_elem,3,piles_num); %Prelocating pile global tensor

        % ------------------------------------------------------------ %
        %Generating a global pile-group matrix
        tic

        for k=1:piles_num

    v=linspace(piles_input(k,3),piles_length+piles_input(k,3),num_elem)';
    vv=linspace(piles_input(k,3)+piles_length/num_elem/2,...
    piles_length-piles_length/num_elem/2+piles_input(k,3),num_elem)';
    piles(:,3,k)=v;
    piles(:,4,k)=vv;
    piles(:,1,k)=piles_input(k,1);
    piles(:,2,k)=piles_input(k,2);

        end
        clear v;
else

    piles=zeros(num_elem,3); %Prelocating pile's global tensor

        % ------------------------------------------------------------ %
        %Generating a global pile-group matrix
        tic

        for k=1:num_elem

            piles(:,3)=surface_pts(3,k)*ones(num_elem,1);
            piles(:,4)=surface_pts(3,k)*ones(num_elem,1);

            piles(k,1)=surface_pts(1,k);
            piles(k,2)=surface_pts(2,k);

        end
        clear v;
end

end
```

```
% File name: V_loss_pts.m
% Author: Genis Majoral Oller
% Date final version: 29/03/2018
% For Civil Engineering bachelor's degree dissertation
% [OUTPUT]: Matrix of Vloss points discretising the tunnel

function [ V_loss_pts ] = V_loss_pts( depth,tun_coord,time_step,dia_0,...
                                      V_loss,n_points,v )



tunnel_adv_coord = zeros(time_step,3); % tunnel advance coordinates


for i=1:time_step
    if time_step==1
        tunnel_adv_coord(i,1:3) = [tun_coord(1:3)];
    else
        tunnel_adv_coord(i,1:3) = [v(i),tun_coord(2:3)] ;
    end
end

clear v i

% ---------------------------------------------------------------------- %
% Generating the global points' coords containing volume loss (in a matrix)


[V_loss_pts]=tunnel_glob_coord(V_loss,dia_0,n_points,tunnel_adv_coord,...
                                time_step,depth);
% [V_loss_pts]=["X", "Y", "Z", "dV"]


end


```

```
%File name: tunnel_glob_coord
%Author: Genis Majoral Oller
%Creation date: 29/03/2018
%INPUT: [V_loss, dia_0,control,n_point] - [known loss volume, original diameter,
                              % - depth of tunnel's axis, control variable]
%OUTPUT: [V_loss_pts] - [coordinates of points that contain a loss of volume]

function [V_loss_pts]=tunnel_glob_coord(V_loss, dia_0,n_points,...
                              tunnel_adv_coord,num_tun,depth)

% Creating the local coordinates "Y" and "Z" of the loss points in the
% tunnel crossection

V_loss_aux=tunnel_local_coord(V_loss, dia_0,n_points);
%V_loss_aux=[y,z,dA]

% Prelocating the final V_loss_pts matrix

V_loss_pts = zeros(n_points,4,num_tun);

    for i=1:num_tun; %for all different tunnel advancing crossections

        x=tunnel_adv_coord(i,1)*ones(n_points,1); % X coordinate XS.
        z=(depth-V_loss_aux(:,2));

        if i==1 || i==num_tun

            if num_tun==1
                delta_x=1; %For m3/m
            else
                if i==1
                    delta_x=(tunnel_adv_coord(i+1)-tunnel_adv_coord(i))/2;
                elseif i==num_tun
                    delta_x=(tunnel_adv_coord(i)-tunnel_adv_coord(i-1))/2;
                end
            end

        else
            delta_x=((tunnel_adv_coord(i+1)-tunnel_adv_coord(i)))/2+...
                    ((tunnel_adv_coord(i)-tunnel_adv_coord(i-1)))/2;
        end

                    dV=V_loss_aux(:,3)*(delta_x);
                    V_loss_pts(:,:,i)=[x, V_loss_aux(:,1), z, dV];

    end

end
```

```
%File name: tunnel_local_coord
%Author: Genis Majoral Oller
%Creation date: 29/03/2018

%INPUT: [V_loss, dia_0,control,n_point] - [known loss volume, original diameter,
                               % - depth of tunnel's axis, control variable]
%OUTPUT: [V_loss_aux] - [coordinates of points that contain a loss of volume]

function [V_loss_aux]=tunnel_local_coord(V_loss, dia_0,n_points)

if n_points==1
    V_loss_aux=[0,0,V_loss*pi*(dia_0/2)^2];
else

tram=linspace(0,2*pi,n_points+1);
    r_ini=sqrt((dia_0/2)^2*(V_loss+1));
    r_final=dia_0/2;
    r0=(r_ini-r_final);
    fi=-pi/2; % Where the second circle is located (angle)
    rad=r_ini-r0; %final radius
    rmed=r0/2;
    radmed=r_ini-rmed; %Medium radius to colocate sink points

    c1=@(r0,theta,fi,rad) (r0*cos(theta-fi)+sqrt(r_ini^2-r0^2*sin(theta-fi).^2));
    cmed=@(rmed,radmed,theta,fi,rad) (rmed*cos(theta-fi)+sqrt(radmed^2-rmed^2*sin(theta-fi).^2));
    cint=@(r0,theta,fi,rad) 1/2*(0*r0*cos(theta-fi)+sqrt(rad^2-0*r0^2*sin(theta-fi).^2)).^2;

        for i=1:length(tram);
            r2(i)=c1(r0,tram(i),fi,rad);
        end

        for i=1:length(tram)-1;
            dA(i)=((pi*(r_ini)^2)/n_points)-integral(@(theta)cint(r0,theta,fi,rad),tram(i),tram(i+1));

        end
        A_loss=sum(dA);
        for i=1:length(tram)-1;
            dV(i)=dA(i);
        end

        % Coordinates of Vloss points
        tram2=tram+(tram(2)-tram(1))/2;
        points_theta=tram2(1:length(tram2)-1);

        for i=1:length(points_theta);
            r_points(i)=cmed(rmed,radmed,points_theta(i),fi,rad);
            V_loss_y(i)=r_points(i)*cos(points_theta(i));
            V_loss_z(i)=r_points(i)*sin(points_theta(i));
        end

        V_loss_aux=zeros(length(V_loss_y),3);

        V_loss_aux(:,1)=V_loss_y(:)'; % 1st column y coord of Vloss point
        V_loss_aux(:,2)=V_loss_z(:)'; % 2nd column z coord of Vloss point

        V_loss_aux(:,3)=dV(:); %only area


end

end
```

```matlab
% File name: boussinesq.m
% Author: Genis Majoral Oller
% Date: 1/8/2018
% For Civil Engineering bachelor's degree dissertation
% Boussinesq displacements due to surface load

function soil_bous=boussinesq(piles, num_elem, soil, surface_forces,time_step,piles_num,V_loss_pts);

%% PRELOCATING VARIABLES
[aa,bb,c]=size(V_loss_pts);
[a,b,k]=size(piles);

soil_bous=zeros(a,b,k,c);
soil_bous_aux=zeros(a,b,k);

%%
Es=soil(1);
vs=soil(2);
form=surface_forces(8);
%local coordinates

num_forces=length(surface_forces(:,1));% number of applied loads/forces




%%
for i_pile=1:piles_num
for i_element=1:num_elem
        for i_forces=1:num_forces

        %% Actual Boussinesq formulation
        z=piles(i_element,4,i_pile);
        R=@(x,y)sqrt(x.^2+y.^2+z^2);



        Ibous_x=@(x,y)((1+vs)./(2*pi*R(x,y))).*...
                    (...
                    1+x.^2./(R(x,y).^2)+...
                    (1-2*vs)*(R(x,y)./(R(x,y)+z)-...
                    x.^2./(R(x,y)+z).^2)...
                        );
         Ibous_y=@(x,y)((1+vs)./(2*pi*R(x,y))).*...
                    (...
                    1+y.^2./(R(x,y).^2)+...
                    (1-2*vs)*(R(x,y)./(R(x,y)+z)-...
                    y.^2./(R(x,y)+z).^2)...
                        );
         Ibous_z=@(x,y)(1+vs)./(2*pi*R(x,y)).*...
                    (2*(1-vs)+z^2./R(x,y).^2);

        %% Depending on the type of force/load
        if form==0
            w_bous_x=0;
            w_bous_y=0;
            w_bous_z=0;

        elseif form==1
                x=piles(i_element,1,i_pile)-surface_forces(i_forces,4);
                y=piles(i_element,2,i_pile)-surface_forces(i_forces,6);


            w_bous_x=surface_forces(i_forces,1)/Es*Ibous_x(x,y);
```

```
        w_bous_y=surface_forces(i_forces,2)/Es*Ibous_y(x,y);
        w_bous_z=surface_forces(i_forces,3)/Es*Ibous_z(x,y);
    elseif form==2
            x1=piles(i_element,1,i_pile)-surface_forces(i_forces,4);
            x2=piles(i_element,1,i_pile)-surface_forces(i_forces,5);
            y1=piles(i_element,2,i_pile)-surface_forces(i_forces,6);
            y2=piles(i_element,2,i_pile)-surface_forces(i_forces,7);


        w_bous_x=surface_forces(i_forces,1)/Es*integral2(@(x,y)Ibous_x(x,y),min(x1,x2),max(x1,x2),min(y1,y2),max(y1,y2));
        w_bous_y=surface_forces(i_forces,2)/Es*integral2(@(x,y)Ibous_y(x,y),min(x1,x2),max(x1,x2),min(y1,y2),max(y1,y2));
        w_bous_z=surface_forces(i_forces,3)/Es*integral2(@(x,y)Ibous_z(x,y),x1,x2,y1,y2);
    else
    end


    soil_bous_aux(i_element,1,i_pile)=soil_bous_aux(i_element,1,i_pile)+w_bous_x;
    soil_bous_aux(i_element,2,i_pile)=soil_bous_aux(i_element,2,i_pile)+w_bous_y;
    soil_bous_aux(i_element,3,i_pile)=soil_bous_aux(i_element,3,i_pile)+w_bous_z;

    clear w_bous_x w_bous_y w_bous_z
    end
end
end



%% Finally

for i_time=1:time_step
soil_bous(:,:,:,i_time)=soil_bous_aux;
end


end
```

```matlab
%File name: sagaseta_3D
%Author: Genis Majoral Oller
%Creation date: 1/4/2018
%Description: This routine calculates the displacement in 3D for a pile or
%            group of piles for a given point or points, this includes a
%            tunnel as long as it is discretised in several points.

%INPUT: [piles, V_loss_pts]->Pile-group matrix
%                        [x,y,z] dim=(N_elem x 3 x N_piles)

%                        dV points matrix [x,y,z,dV]
%OUTPUT: [piles_sagaseta]->[x_new,y_new,z_new] dim=(N_elem x 3 x N_piles)

function [soil_sagaseta_strain]=sagaseta_3D(piles,V_loss_pts,time_step)

%% PRELOCATING VARIABLES
[aa,bb,c]=size(V_loss_pts);
[a,b,k]=size(piles);

%piles_sagaseta=piles;
soil_sagaseta_strain=zeros(a,b,k,c);


%% ADVANCING TUNNEL

fprintf('Soil strain calculations start here\n');

    for i_time=1:time_step % number of tunnel advances
        piles_sagaseta=zeros(a,b,k);
        for i_pile=1:length(piles(1,1,:)) % number of piles
            tic
            for i_element=1:length(piles(:,1,1)) % elements on piles

                x_0=piles(i_element,1,i_pile);
                y_0=piles(i_element,2,i_pile);
                z_0=piles(i_element,3,i_pile);
                z_1=piles(i_element,4,i_pile);
                piles_sagaseta_x=zeros(1,length(V_loss_pts(:,1,1)));
                piles_sagaseta_y=zeros(1,length(V_loss_pts(:,1,1)));
                piles_sagaseta_z=zeros(1,length(V_loss_pts(:,1,1)));

                parfor i_point=1:length(V_loss_pts(:,1,1))

                    % Pile element to volume-loss-point distances
                    Delta_x=x_0-V_loss_pts(i_point,1,i_time);
                    Delta_y=y_0-V_loss_pts(i_point,2,i_time);
                    Delta_z=z_0-V_loss_pts(i_point,3,i_time);

                    % Point
                    h= V_loss_pts(i_point,3,i_time);% Coordinate "Z" of the loss-vol-point

                    rad=sign(V_loss_pts(i_point,4,i_time))*...
                      (abs(V_loss_pts(i_point,4,i_time))*3/4/pi)^(1/3); %sphere

                    %% COORDINATE "X" --------------------------------------------
                    control_coord=1;% To assign sagaseta_3D_integral a calculation for coord "X"

                    [Sx_paved,Sx_free,Sy_paved,Sy_free]=sagaseta_3D_integral...
                            (Delta_x,Delta_y,Delta_z,rad,h,control_coord);

                    piles_sagaseta_x(i_point)=Sx_paved + Sx_free;


                    %% COORDINATE "Y" ---------------------------------------------
```

```
                    piles_sagaseta_y(i_point)=Sy_paved+ Sy_free;

                    %% COORDINATE "Z" -------------------------------------------------------
                    control_coord=3;% To assign sagaseta_3D_integral a calculation for coord "Z"
                    Delta_z=z_1-V_loss_pts(i_point,3,i_time);%Because for the vertical analysis pile nodes are different
                    [Sz_paved,Sz_free,S_paved2,S_free2]=sagaseta_3D_integral...
                            (Delta_x,Delta_y,Delta_z,rad,h,control_coord);

                    %Final "Z" displacement
                    piles_sagaseta_z(i_point)=Sz_paved + Sz_free;


                end
                piles_sagaseta(i_element,1,i_pile)=sum(piles_sagaseta_x);
                piles_sagaseta(i_element,2,i_pile)=sum(piles_sagaseta_y);
                piles_sagaseta(i_element,3,i_pile)=sum(piles_sagaseta_z);
                clear piles_sagaseta_x piles_sagaseta_y piles_sagaseta_z
            end
            temps=toc;
            fprintf('        Soil strain of pile %i/%i at time...%i/%i calculated in %.1f s\n',i_pile,k,i_time,c,temps);
        end

                soil_sagaseta_strain(:,:,:,i_time)=piles_sagaseta;
                clear piles_sagaseta;
    end


end
```

```
%File name: sagaseta_3D_integral
%Author: Genis Majoral Oller
%Date final version: 1/8/2018
%Description: This routine implements the Sagaseta solution for a sink point
%            to a particular point in space in 3D
%            group of piles for a given point or points.
%INPUT: [Delta_x,Delta_y,Delta_z,rad,h,control_coord]
%
%       rad --> radius of vol-loss sphere
%       h   --> depth of vol-loss point
%       control_coord --> internal variable to decide what coord
%       "X,Y,Z" to calculate
%
%OUTPUT: [S_paved,S_free]->(in meters)
function [S_paved,S_free,S_paved2,S_free2]=sagaseta_3D_integral(Delta_x,Delta_y,Delta_z,rad,h,control_coord)
%% Common data
          z=Delta_z+h;%Returning z to being the original coordinate of the element afected by Vloss
          x=sqrt(Delta_x^2+Delta_y^2);
          r1=sqrt(x^2+(z-h)^2);
          r2=sqrt(x^2+(z+h)^2);
          r=sqrt(Delta_x^2+Delta_y^2+Delta_z^2);
 if control_coord == 1 || control_coord == 2
%% For COORDINATES "X" AND "Y"
          ra=@(a) ((a-x).^2+z^2).^0.5;
          rb=@(a) ((a+x).^2+z^2).^0.5;
          k=@(a) (( 1-(ra(a).^2)./(rb(a).^2) ).^0.5).^2;
          E=@(a) ellipticE(k(a));% complete elliptic integral of 2nd kind
          F=@(a) ellipticK(k(a)); % complete elliptic integral of 1st kind
          IE=@(a)  1+1/2*z^2*( 1./(ra(a).^2)  +  1./(rb(a).^2) ) ;
          IF=@(a)  -1./(rb(a).^2).*(a.^2+x^2+2*z^2);
          f_integral = @(a) rb(a).*a.*1./( (h^2+a.^2).^(5/2) ).*( IE(a).*E(a)+IF(a).*F(a) );
          Sx_free=2/pi*rad^3*h/x*(integral(@(a)f_integral(a),0,inf));
          S_paved=-rad^3/3*(Delta_x/r1^3-Delta_x/r2^3);
          S_free=Sx_free*Delta_x/x;
          S_paved2=-rad^3/3*(Delta_y/r1^3-Delta_y/r2^3);
          S_free2=Sx_free*Delta_y/x;
elseif control_coord ==3
%% For COORDINATE "Z"
          ra=@(a) ((a-x).^2+z^2).^0.5;
          rb=@(a) ((a+x).^2+z^2).^0.5;
          k=@(a) (( 1-(ra(a).^2)./(rb(a).^2) ).^0.5).^2;
          E=@(a) ellipticE(k(a));
          F=@(a) ellipticK(k(a));
          JE=@(a) -1 + 2*(a.*(a-x)).*1./(ra(a).^2);
          fz_integral = @(a) 1./(rb(a).^2).*a.*1./((h^2+a.^2).^(5/2)).*...
```

```
            (JE(a).*E(a)+F(a)) ;


        Sz_free=2/pi*rad^3*h*z*(integral(@(a)fz_integral(a),0,inf));


        S_paved=-rad^3/3*((z-h)/r1^3-(z+h)/r2^3);
        S_free=Sz_free;
        S_paved2=0;
        S_free2=0;
 else
    fprintf('Error while calculating sagaseta displacements. Check control_coordinates')
end

end
```

```
%File name: horizontal_analysis
%Author: Genis Majoral Oller
%Date final version: 25/4/2018
%Description: Horizontal analysis of a group of piles, either free headed
%             or with pile cap
function [Is_global_x,BC_global_x,Ap_global_x,D_global_x,p_global_x,...
    w_global_x,w_final,w_head_x,BC_change_x,ppp,w_dif_x2,...
    soil_strain_aux] = horizontal_analysis( piles,soil,piles_dia,...
    piles_forces_top,piles_E,piles_I,piles_head_type,piles_num,...
     soil_strain,coord,num_elem,time_step,ppp)

        [Is_global_x]=horizontal_assemble_mindlin...
                                        (piles,soil,piles_dia,coord);


       if sum(piles_forces_top(:,coord))==0
           ppp(:,coord,1)=piles_forces_top(:,coord);
       else
           for i=1:length(soil_strain(1,1,1,:))
            ppp(:,coord,i)=piles_forces_top(:,coord);
           end
       end


        for i_pile=1:length(piles(1,1,:))

            iii_0=(i_pile-1)*(num_elem)+1;    %
            iii=iii_0+(num_elem)-1;

            Is_global_x_aux=Is_global_x(iii_0:iii,iii_0:iii);
            % Assembling the global system of equations for one pile - pile-structure

            [BC_x, Ap_x, D_x]=horizontal_assemble_pile_structure...
                (piles(:,:,i_pile),piles_forces_top(i_pile,:),piles_dia,...
                piles_E,piles_I,piles_head_type,soil);

            ii_0=(i_pile-1)*(length(Ap_x(:,1)))+1;
            ii=ii_0+(length(Ap_x(:,1)))-1;

            D_global_x(ii_0:ii,iii_0:iii)=D_x;
            Ap_global_x(ii_0:ii,iii_0:iii)= Ap_x;
            BC_global_x(ii_0:ii,1)=BC_x;

        end
            clear row_end col col_end iii iii_0 ii ii_0

        N=num_elem;
soil_strain_aux=zeros(N*piles_num,length(soil_strain(1,1,1,:)));
w_head_x=zeros(piles_num,length(soil_strain(1,1,1,:)));

for i_time=1:length(soil_strain(1,1,1,:))


    for i_pile=1:length(soil_strain(1,1,:,1))
        iii_0=(i_pile-1)*(num_elem)+1;
        iii=iii_0+(num_elem)-1;
        for j=1:i_time
        soil_strain_aux(iii_0:iii,i_time)=...
            soil_strain_aux(iii_0:iii,i_time)+soil_strain(:,coord,i_pile,j);
        %coordinate x or y defined by control variable "coord"
        end
    end

    p_global_x_aux=(Ap_global_x-D_global_x*Is_global_x)\...
```

```
    (D_global_x*soil_strain_aux(:,i_time)+BC_global_x);


w_global_x(:,i_time)=Is_global_x*p_global_x_aux+...
                                    soil_strain_aux(:,i_time);
p_global_x(:,i_time)=p_global_x_aux;



     %% iteration x-dir %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

piles_forces_top=ppp(:,:,i_time);
dir=coord;
i_step=1;

for i=1:piles_num
    w_head_x(i,i_time)=w_global_x((i-1)*num_elem+1,i_time);
end

if piles_head_type==0
    BC_change_x='No change';%no action
    w_dif_x2=0;
else
    w_dif_x=max(w_head_x(:,i_time))-min(w_head_x(:,i_time));
    w_dif_x2(:,i_time)=w_dif_x;
    w_mean=mean(w_head_x(:,i_time));
    F_change=zeros(piles_num,1);

    if w_dif_x < -0.00005 || w_dif_x > 0.00005
    while w_dif_x < -0.00005 || w_dif_x > 0.00005
        BC_change_x=zeros(piles_num*num_elem,1);
        for it=1:piles_num
        if i_step==1
            if sum(piles_forces_top(:,dir))==0
                Fc=(w_mean-w_head_x(it,i_time))*100000;
             else
                Fc=piles_forces_top(it,dir)+(w_mean-w_head_x(it,i_time))*...
                    abs(sum(piles_forces_top(:,dir)))/piles_num*100;
             end
         else
             condition=sum(piles_forces_top(:,dir));
             if condition>-0.01 && condition<0.01
                Fc=piles_forces_top(it,dir)+...
                                  (w_mean-w_head_x(it,i_time))*100000;
             else
                Fc=piles_forces_top(it,dir)+...
                        (w_mean-w_head_x(it,i_time))*...
                    abs(sum(piles_forces_top(:,dir)))/piles_num*100;
             end
         end
             %B-matrix of pile displacement
             N=num_elem;
             Bp=zeros(N,1);
             Bp(1,1)=Fc;
             BC_change_x((it-1)*(N)+1:it*(N),1)=Bp;
             F_change(it,1)=Fc;
         end %end changing force
             piles_forces_top(:,dir)=F_change;

         %new shear/displacement
         p_global_x(:,i_time)=(Ap_global_x-D_global_x*Is_global_x)\...
                (D_global_x*soil_strain_aux(:,i_time)+BC_change_x);

          w_global_x(:,i_time)=Is_global_x*p_global_x(:,i_time)+...
```

```
                                    soil_strain_aux(:,i_time);

        %new condition
        for i=1:piles_num
            w_head_x(i,i_time)=w_global_x((i-1)*num_elem+1,i_time);
        end
        w_dif_x=max(w_head_x(:,i_time))-min(w_head_x(:,i_time));
        w_dif_x2(:,i_time)=w_dif_x;
        w_mean=mean(w_head_x(i,i_time));
        i_step=i_step+1;
        dir_iter=[dir i_step];
    end
    else
        BC_change_x='No change';

    end

end

  ppp(:,coord,i_time)=piles_forces_top(:,coord);

clear dir_i dir_iter piles_forces_top a Fc
        for i_pile=1:piles_num

            kk_0=(N)*(i_pile-1)+1;
            kk=kk_0+(N-1);

            w_final(:,i_pile,i_time)=w_global_x(kk_0:kk,i_time);

        end

end


end
```

```
%File name: horizontal_assemble_pile_structure.m
% Author: Genis Majoral Oller
%For Civil Engineering bachelor's degree dissertation

%INPUT: pile,          piles_forces_top,   dia
%       [x,y,z](100x3)  [Fx,Fy,Fz,Mx]       diameter or width


function [BC_x, Ap_x, D_x]=horizontal_assemble_pile_structure(pile,...
        piles_forces_top,piles_dia,piles_E,piles_I,piles_head_type,...
                                          soil)

    %Preparing the external horizontal load  and Moment in "x" direction
    %[kNm]

    H=piles_forces_top(1);   %[kN]
    M=piles_forces_top(5);   %[kNm]


    %pile position and properties

    Ep=piles_E;            %[kN/m2]
    N=length(pile(:,1));  %[-]
    L=pile(N,3);          %[m]


    %soil properties
    Es=soil(1);            %[kN/m2]
    vs=soil(2);            %[-]

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Ip - moment of inertia
    Ip=piles_I; %b of a square is equal to b=(3*pi)^0.25/2*d
    h=(3*pi)^0.25/2*piles_dia; %square,
    d=h;

    %% PILE PROBLEM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Bending equation finite difference matrix
    D_x=zeros(N,N);
    EI=Ep*Ip;

 if piles_head_type==0



            for r=1:N
                if r==1
                    D_x(r,:)=0;
                elseif r==2
                    D_x(r,1:4)=[-2, 5, -4, 1];
                elseif r==N-1
                    D_x(r,N-3:N)=[1, -4, 5, -2];
                elseif r==N
                    D_x(r,:)=0;
                else
                    D_x(r,r-2:r+2)=[1, -4, 6, -4, 1];
                end
            end


        %B-matrix of pile BC
        BC_x=zeros(N,1);
            for r=1:N
                if r==1
```

```
                BC_x(r,1)=H;
            elseif r==2
                BC_x(r,1)=M*L^2/(N-1)^2/EI;
            elseif r==3
                BC_x(r,1)=0;
            elseif r==4
                BC_x(r,1)=0;
            elseif r==N
                BC_x(r,1)=-M;
            else
            end
        end


elseif piles_head_type==1 %head-casted



        for r=1:N %fill except two last rows, where BC apply
            if r==1
                D_x(r,:)=0;
            elseif r==2
                D_x(r,1:4)=[-4, 7, -4, 1];
            elseif r==N-1
                D_x(r,N-3:N)=[1, -4, 5, -2];
            elseif r==N
                D_x(r,1:4)=-[2, -5, 4, -1]*EI/( L^2/(N-1)^2 );
            else
                D_x(r,r-2:r+2)=[1, -4, 6, -4, 1];
            end
        end


      %B-matrix of pile BC
      BC_x=zeros(N,1);
        for r=1:N
            if r==1
                BC_x(r,1)=H;
            elseif r==2
                BC_x(r,1)=0;
            elseif r==N
                BC_x(r,1)=-M;
            else
            end
        end
else
end


 %Distribution of shear
 Ap_x=zeros(N,N);
 coef=-d/(EI)*((L/(N-1))^4);

for r=1:N
    if r==1
        Ap_x(r,1)=1*0.5*(L)/(N-1)*d;
        Ap_x(r,2:N-1)=1*L/(N-1)*d;
        Ap_x(r,N)=1*0.5*(L)/(N-1)*d;
    elseif r==N
        if piles_head_type==0
              Ap_x(r,1)=0.25*L^2/(N-1)^2*d;
              for j=2:N-1
              Ap_x(r,j)=(j-1)*L^2/(N-1)^2*d;
              end
```

```
            Ap_x(r,N)=(0.5*(N-1)-0.125)*L^2/(N-1)^2*d;
        elseif piles_head_type==1
            Ap_x(r,1)=0.5*d*L/(N-1)*( 0.5*0.5*L/(N-1) );
            Ap_x(r,2:N-1)=d*L/(N-1)*( pile(2:N-1,3)-pile(1,3));
            Ap_x(r,N)=0.5*d*L/(N-1)*( pile(N,3)-pile(1,3)-0.5*0.5*L/(N-1));


        else
        end
    else
        Ap_x(r,r)=1*coef;
    end
end


end

```

```
%File name: horizontal_assemble_mindlin.m
%Author: Genis Majoral Oller
%For Civil Engineering bachelor's degree dissertation
%Date: 15/05/2018
%OUTPUT: Is_global_x
%        A matrix (Num_elements*Num_piles)x(Num_elements*Num_piles)

function [Is_global_x]=horizontal_assemble_mindlin(piles,...
      soil,piles_dia,coord)
%% Preparing input data for mindlin problem

    N_piles = length(piles(1,1,:));

    %Soil properties
    Es=soil(1);             %[kN/m2]
    vs=soil(2);             %[-]

    %Ip - moment of inertia
    %Ip=pi/4*(dia/2)^4; %b of a square is equal to b=(3*pi)^0.25/2*d
    h=(3*pi)^0.25/2*piles_dia; %square,
    d=h;

    pile_acting=piles(:,:,1);
    pile_afected=piles(:,:,1);
    N=length(pile_afected(:,1));  %[num]
    L=piles(N,3);


[Is_temporal_inpile]=horizontal_mindlin_in_pile...
                                (pile_acting,pile_afected,d,L,N,vs,Es);

for i_acting=1:N_piles;

    %Acting pile
          pile_acting=piles(:,:,i_acting);

    for i_afected=1:N_piles;

       %Afected pile

       pile_afected=piles(:,:,i_afected);



       N=length(pile_afected(:,1));  %[num]
       L=piles(N,3);          %[m]



       %% SOIL PROBLEM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
       %soil displacement due to shear force

       row=(i_acting-1)*N+1;
       row_end=row+N-1;
       col=(i_afected-1)*N+1;
       col_end=col+N-1;

       %Is it pile2pile or pile within itself?

       if i_acting==i_afected %diagonal terms


        Is_global_x(row:row_end,col:col_end)=1/Es*Is_temporal_inpile;
       else
```

```
        [Is_temporal]=horizontal_mindlin_pile2pile...
                                (pile_acting,pile_afected,vs,coord);

        Is_global_x(row:row_end,col:col_end)=d*L/(N-1)/Es*Is_temporal;

        end


    end
end
end

```

```
%Author: Genís Majoral Oller
%File name: horizontal_mindlin_in_pile
%Date: 25/05/2018
%Mindlin coeficients to find displacement due to a
%horizontal load acting upon a rectangular area (i.e. discretisation of the
%pile)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Input
function [Is_hor]=horizontal_mindlin_in_pile(Pos_force,Pos_pile,d,L,N,vs,Es)

% Pos_force - pile(x,y,z) "N" rows
% Pos_pile - pile(x,y,z)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
z_afected=Pos_pile(:,3); % column vector
z_acting=Pos_force(:,3); % column vector
Is_hor=zeros(length(z_afected),length(z_acting));
x=0;
    for i_afected=1:length(z_afected)
        z=z_afected(i_afected);

        for i_force=1:length(z_acting)

            if i_force==1 %for 1st and last elements half length

                cc=z_acting(i_force);

                y_inf=0;
                y_sup=d/2;
                c_inf=cc;
                c_sup=cc+( (L)/(N-1)*0.5 );

            elseif i_force==length(z_acting)

                cc=z_acting(i_force);

                y_inf=0;
                y_sup=d/2;
                c_inf=cc-( (L)/(N-1)*0.5 );
                c_sup=cc;

            else
                cc=z_acting(i_force);

                y_inf=0;
                y_sup=d/2;
                c_inf=cc-( (L)/(N-1)*0.5 );
                c_sup=cc+( (L)/(N-1)*0.5 );
            end

            %%


                R1=@(y,c) sqrt(x^2+y.^2+(z-c).^2);
                R2=@(y,c) sqrt(x^2+y.^2+(z+c).^2);

                Is_aux=@(y,c) (1+vs)/(8*pi*(1-vs))*((3-4*vs)./R1(y,c)+...
                    1./R2(y,c)+...
                    x^2./R1(y,c).^3+...
                    (3-4*vs)*x^2./R2(y,c).^3+...
```

```
                    2*c.*z./R2(y,c).^3.*(1-3*x^2./R2(y,c).^2)+...
                    4*(1-vs)*(1-2*vs)./(R2(y,c)+z+c).*...
                    ( 1 -x^2./( R2(y,c).*(R2(y,c)+z+c) )  )   );


                Is_hor(i_afected,i_force)=2*integral2...
                    (@(y,c)Is_aux(y,c),y_inf,y_sup,c_inf,c_sup);



        end
    end



end
```

```
%Author: Genís Majoral Oller
%Date: 25/5/2018
%File name: horizontal_mindlin_pile2pile


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Input
function [Is_hor]=horizontal_mindlin_pile2pile(Pos_force,Pos_pile,vs,direction)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%direction (1=x,2=y)
Is_hor=zeros(length(Pos_pile(:,3)),length(Pos_force(:,3)));

for i_afected=1:length(Pos_pile(:,3))
    for i_force=1:length(Pos_force(:,3))
        z=Pos_pile(i_afected,3);
        c=Pos_force(i_force,3);
        x=Pos_pile(i_afected,1)-Pos_force(i_force,1);
        y=Pos_pile(i_afected,2)-Pos_force(i_force,2);
        R1=sqrt(x^2+y^2+(z-c)^2);
        R2=sqrt(x^2+y^2+(z+c)^2);
        if direction==1
            if i_force==1 || i_force==length(Pos_force(:,3))
            Is_hor(i_afected,i_force)=(0.5)*(1+vs)/(8*pi*(1-vs))*...
                ((3-4*vs)/R1+1/R2+x^2/...
                R1^3+(3-4*vs)*x^2/R2^3+2*c*z/R2^3*(1-3*x^2/R2^2)+...
                4*(1-vs)*(1-2*vs)/(R2+z+c)*(1-x^2/(R2*(R2+z+c))));
            %0.5 accounts for half the discretisation, it will be
            %multiplied later by d*L/(N-1)
            else
                Is_hor(i_afected,i_force)=(1+vs)/(8*pi*(1-vs))*...
                    ((3-4*vs)/R1+1/R2+x^2/...
                R1^3+(3-4*vs)*x^2/R2^3+2*c*z/R2^3*(1-3*x^2/R2^2)+...
                4*(1-vs)*(1-2*vs)/(R2+z+c)*(1-x^2/(R2*(R2+z+c))));
            end
        elseif direction==2
                Is_hor(i_afected,i_force)=(1+vs)/(8*pi*(1-vs))*...
                    ((3-4*vs)/R1+1/R2+y^...
                    2/R1^3+(3-4*vs)*y^2/R2^3+2*c*z/R2^3*(1-3*y^2/R2^2)+...
                    4*(1-vs)*(1-2*vs)/(R2+z+c)*(1-y^2/(R2*(R2+z+c))));
         else
                %no action
        end
    end
end
```

```
%Author: Genís Majoral Oller
%Date: 12/6/2018
%File name: vertical_analysis

function [p_global_z,p_global_z2,v_global,v_global2,v_global3,v_final,D_global_z,Ap_global_z,...
    Is_global_z,BC_z,BC_change_z,u_dif_u,u_dif_o, ppp,u_head_z,u_mean,Z_head,u_dif_z,...
    soil_strain_aux]=vertical_analysis(piles,...
    piles_num,piles_dia,piles_head_type,piles_length,soil,piles_E,soil_strain,num_elem,piles_forces_top,ppp,find_surface)
%% Pile structure
if find_surface==0
        Ep=piles_E;
        Es=soil(1);
        L=piles_length;
        d=piles_dia;
        N=num_elem;
        Ra=1;
        coef1=Ra*d*Ep/(4*( (L)/(N) )^2);
        [a,b]=size(piles_forces_top);

        if sum(piles_forces_top(:,3))==0
            ppp(:,3,1)=piles_forces_top(:,3);
        else
            for i=1:length(soil_strain(1,1,1,:))
             ppp(:,3,i)=piles_forces_top(:,3);
            end
        end

        D_global_z_aux=zeros(N+1,N+1);

        for r=1:N+1
            if r==1
                 D_global_z_aux(r,1:2)=[-2 2];
            elseif r==N+1
                 D_global_z_aux(r,:)=0;
            elseif r==N
                D_global_z_aux(r,N:N+1)=[1,-1];
            else
                 D_global_z_aux(r,r-1:r+1)=[1, -2, 1];
            end
        end

        D_global_z_aux=coef1*D_global_z_aux;


        Ap_global_z_aux=zeros(N+1,N+1);

         for i=1:N+1
             if i==N+1
                 Ap_global_z_aux(i,:)=pi*d*(L/(N));
                 Ap_global_z_aux(i,N+1)=pi*(d/2)^2;
             elseif i==N
                 Ap_global_z_aux(i,N)=(0.5*L/(N))^2*pi*d/Ep/(pi*d^2/4);
                 Ap_global_z_aux(i,N+1)=1/2*L/(N)/Ep;
             elseif i==1
                 Ap_global_z_aux(i,1)=1+coef1*(L/(N))^2*pi*d/Ep/(pi*d^2/4);
             else
                 Ap_global_z_aux(i,i)=1;
             end

         end

         for i_pile=1:piles_num
                 BC_z_aux=zeros(N+1,1);
                 P=piles_forces_top(i_pile,3);
```

```
            BC_z_aux(1,1)=coef1*P/(pi*d^2/4)*2*L/(N)/Ep;
            BC_z_aux(N+1,1)=P;

            BC_z_aux2=zeros(N+1,1);
            BC_z_aux2(1,1)=0;
            BC_z_aux2(N+1,1)=0;

            ii_0=(i_pile-1)*(num_elem+1)+1;
            ii=ii_0+(num_elem+1)-1;
            BC_z(ii_0:ii,1)=BC_z_aux;
            BC_z2(ii_0:ii,1)=BC_z_aux2;

            D_global_z(ii_0:ii,ii_0:ii)=D_global_z_aux;
            Ap_global_z(ii_0:ii,ii_0:ii)=Ap_global_z_aux;

        end


%% Soil Structure

%Mindlin matrix

        for i_pile=1:piles_num %pile afected

            Pos_pile=piles(:,:,i_pile);

            for j_pile=1:length(piles(1,1,:))%pile acting

                Pos_force=piles(:,:,j_pile);

                    %------------------------------------

                    col_0=(num_elem+1)*(j_pile-1)+1;
                    col=col_0+(num_elem+1)-1;
                    row_0=(num_elem+1)*(i_pile-1)+1;
                     row=row_0+(num_elem+1)-1;



                if j_pile==1 && j_pile==i_pile

                    control=1;%inpile coefficients
                    [Is_z_diagonal]=vertical_assemble_mindlin(Pos_pile,Pos_force,...
                    soil,piles_dia,piles_length,num_elem,control);
                    Is_global_z(row_0:row,col_0:col)=Is_z_diagonal;

                elseif i_pile==j_pile && j_pile>1 %so that Is_Z_diagonal is not computed unnecessary times

                    Is_global_z(row_0:row,col_0:col)=Is_z_diagonal;


                else
                    control=2;%pile to pile coefficients
                    [Is_z]=vertical_assemble_mindlin(Pos_pile,Pos_force,...
                    soil,piles_dia,piles_length,num_elem,control);
                    Is_global_z(row_0:row,col_0:col)=Is_z;
                end



            end
        end

        Is_global_z=1/Es*Is_global_z;
```

```
%% Finding p

p_global_z=zeros((num_elem+1)*(piles_num),length(soil_strain(1,1,1,:)));
v_global=zeros((num_elem+1)*(piles_num),length(soil_strain(1,1,1,:)));
soil_strain_aux=zeros((N+1)*piles_num,length(soil_strain(1,1,1,:)));

 for i_time=1:length(soil_strain(1,1,1,:))

   for i_pile=1:piles_num
     row_0=(num_elem+1)*(i_pile-1)+1;
     row=row_0+(num_elem+1)-1;
     for j=1:i_time
     soil_strain_aux(row_0:row,i_time)=soil_strain_aux(row_0:row,i_time)+[soil_strain(:,3,i_pile,j);...
        soil_strain(N,3,i_pile,j)];
     end
   end

   if i_time==1
     p_global_z(:,i_time)=(Ap_global_z-(D_global_z*Is_global_z))\(BC_z+D_global_z*soil_strain_aux(:,i_time));
     v_global(:,i_time)=Is_global_z*p_global_z(:,i_time)+soil_strain_aux(:,i_time);

   else

       p_global_z(:,i_time)=(Ap_global_z-(D_global_z*Is_global_z))\(BC_z+D_global_z*soil_strain_aux(:,i_time));
       v_global(:,i_time)=Is_global_z*p_global_z(:,i_time)+soil_strain_aux(:,i_time);


   end
 end

 %%
 v_global2=v_global;
 v_global3=zeros(size(v_global));


for i_time=1:length(soil_strain(1,1,1,:))



  [ppp,p_global_z2(:,i_time),v_global3(:,i_time),BC_change_z,u_dif_u(i_time),...
     u_dif_o(i_time),u_head_z(:,i_time),u_mean(:,i_time),Z_head,u_dif_z(:,i_time)]=...
     vertical_pile_cap(p_global_z,v_global2,...
                 i_time,Ap_global_z,D_global_z,Is_global_z,...
                 soil_strain_aux,N,piles_num,coef1,d,L,Ep,ppp,...
                 piles_head_type,piles,num_elem);




  for i_pile=1:piles_num

     kk_0=(N+1)*(i_pile-1)+1;
     kk=kk_0+(N+1-1);

             v_final(:,i_pile,i_time)=v_global3(kk_0:kk,i_time);

  end
end

else
                N=num_elem;
                soil_strain_aux=zeros((N+1)*piles_num,length(soil_strain(1,1,1,:)));

```

```
            for i_time=1:length(soil_strain(1,1,1,:))

              for i_pile=1:piles_num
                row_0=(num_elem+1)*(i_pile-1)+1;
                row=row_0+(num_elem+1)-1;
                for j=1:i_time
                soil_strain_aux(row_0:row,i_time)=soil_strain_aux(row_0:row,i_time)+[soil_strain(:,3,i_pile,j);...
                    soil_strain(N,3,i_pile,j)];
                end
              end


            end
            p_global_z=0;
            p_global_z2=0;
            v_global=0;
            v_global2=0;
            v_global3=0;
            v_final=0;
            D_global_z=0;
            Ap_global_z=0;
            Is_global_z=0;
            BC_z=0;
            BC_change_z=0;
            u_dif_u=0;
            u_dif_o=0;
            ppp=0;
            u_head_z=0;
            u_mean=0;
            Z_head=0;
            u_dif_z=0;
end
end
```

```
%File name: vertical_assemble_mindlin.m
%Author: Genis Majoral Oller
%For Civil Engineering bachelor's degree dissertation
%Date: 5/06/2018

function [Is_z]=vertical_assemble_mindlin(Pos_pile,Pos_force,soil,...
        piles_dia,piles_length,num_elem,control)
%% Preparing input data for mindlin problem

    Is_z=zeros(length(Pos_pile(:,4)),length(Pos_force(:,4)));
    d=piles_dia;
    v=soil(2);
    L=piles_length;
    N=num_elem;
if control==1 %for inpile influence coefficients

    for i_afected=1:length(Pos_pile(:,4))+1 %this is the afected discretisation
                                            % of the afected pile

        for i_force=1:length(Pos_force(:,4))+1 %this is the acting discretisation
                                               % of the force due to the acting pile



            if i_force==length(Pos_pile(:,3))+1 && i_afected==length(Pos_pile(:,3))+1
                cas=1;    % base against base

                z_afected=Pos_pile(i_afected-1,3);
                z_force=Pos_force(i_force-1,3);
                x_afected=Pos_pile(i_afected-1,1);
                x_force=Pos_force(i_force-1,1);
                [Is_z_base_col, Is_z_base_base]=vertical_mindlin_base(L,d,v,z_afected,z_force,x_afected,x_force,cas);
                Is_z(i_afected,i_force)=Is_z_base_base;
            elseif i_force==length(Pos_pile(:,3))+1
                cas=2; %base against shaft

                    z_afected=Pos_pile(i_afected,4);
                    z_force=Pos_force(i_force-1,3);
                    x_afected=Pos_pile(i_afected,1);
                    x_force=Pos_force(i_force-1,1);

                [Is_z_base_col, Is_z_base_base]=vertical_mindlin_base(L,d,v,z_afected,z_force,x_afected,x_force,cas);
                Is_z(i_afected,i_force)=Is_z_base_col;

            else  %ij and bj cases

                    %Numerical integration over theta

                    [Is_z_aux]=vertical_mindlin_shaft(L,d,v,N,Pos_pile,Pos_force,i_afected,i_force);
                    Is_z(i_afected,i_force)=Is_z_aux;
            end



        end

    end
elseif control==2 %for pile to pile influence coefficients

        for i_afected=1:length(Pos_pile(:,4))+1 %this is the afected discretisation
                                                % of the afected pile

        for i_force=1:length(Pos_force(:,4))+1 %this is the acting discretisation
                                               % of the force due to the acting pile
```

```
        if i_force==length(Pos_pile(:,3))+1 && i_afected==length(Pos_pile(:,3))+1
            % base against base

            [Is_z_aux]=vertical_mindlin_pile2pile(L,d,v,N,Pos_pile,Pos_force,i_afected,i_force);
            Is_z(i_afected,i_force)=pi*d^2/4*Is_z_aux;
        elseif i_force==length(Pos_pile(:,3))+1
            %acting base affecting the rest of discretisations

            [Is_z_aux]=vertical_mindlin_pile2pile(L,d,v,N,Pos_pile,Pos_force,i_afected,i_force);
            Is_z(i_afected,i_force)=pi*d^2/4*Is_z_aux;

        else  %ij cases and bj cases


            [Is_z_aux]=vertical_mindlin_pile2pile(L,d,v,N,Pos_pile,Pos_force,i_afected,i_force);
             Is_z(i_afected,i_force)=0.5*pi*d*L/N*Is_z_aux;

        end



    end

    end

else
end

end

```

```
%File name: vertical_mindlin_base.m
%Author: Genis Majoral Oller
%For Civil Engineering bachelor's degree dissertation
%Date 11/06/2018
function [Is_z_base_col, Is_z_base_base]=vertical_mindlin_base(L,d,v,z_afected,z_force,x_afected,x_force,cas);
        if cas==2 %base upon shaft



         r=@(theta) sqrt((x_afected-x_force).^2+(d/2*sin(a(theta))).^2);
         c=z_force;
         h=z_afected;
         x= (x_afected-x_force);


         z=@(c) h+c;
         z1=@(c) h-c;
         R2=@(c,theta) (d^2/4+x^2-x*d.*cos(theta)+z(c).^2).^0.5;
         R1=@(c,theta) (d^2/4+x^2-x*d.*cos(theta)+z1(c).^2).^0.5;

         fun0=@(c,theta) z1(c).^2./R1(c,theta).^3;
         fun1=@(c,theta,v) (3-4*v)./R1(c,theta);
         fun2=@(c,theta,v) (5-12*v+8*v^2)./(R2(c,theta));
         fun3=@(c,theta,v) ((3-4*v)*z(c).^2-2*c.*z(c)+2*c.^2)./(R2(c,theta).^3);
         fun4=@(c,theta,v) (6*c.*z(c).^2.*(z(c)-c))./(R2(c,theta).^5);

         fun=@(r,c,theta,v) r.*( (1+v)/(8*pi*(1-v))*(fun0(c,theta)+fun1(c,theta,v)+...
           fun2(c,theta,v)+fun3(c,theta,v)+fun4(c,theta,v)) );


         Is_z_base_col=1/d*2*integral2(@(r,theta)fun(r,c,theta,v),0,d/2,0,pi); %compte
         Is_z_base_base=0;


         elseif cas==1 %base element upon base

         c=z_force;
         h=z_afected;
         x= @(r,theta)(r.*cos(theta));

         z=@(c) h+c;
         z1=@(c) h-c;
         R2=@(c,theta,r) (d^2/4+x(r,theta).^2-x(r,theta)*d.*cos(theta)+z(c).^2).^0.5;
         R1=@(c,theta,r) (d^2/4+x(r,theta).^2-x(r,theta)*d.*cos(theta)+z1(c).^2).^0.5;

         fun0=@(c,theta,r) z1(c).^2./R1(c,theta,r).^3;
         fun1=@(c,theta,v,r) (3-4*v)./R1(c,theta,r);
         fun2=@(c,theta,v,r) (5-12*v+8*v^2)./(R2(c,theta,r));
         fun3=@(c,theta,v,r) ((3-4*v)*z(c).^2-2*c.*z(c)+2*c.^2)./(R2(c,theta,r).^3);
         fun4=@(c,theta,v,r) (6*c.*z(c).^2.*(z(c)-c))./(R2(c,theta,r).^5);

         fun=@(r,theta,c,v) r.*( (1+v)/(8*pi*(1-v))*(fun0(c,theta,r)+fun1(c,theta,v,r)+...
           fun2(c,theta,v,r)+fun3(c,theta,v,r)+fun4(c,theta,v,r)) );


         Is_z_base_base=pi/4*2*integral2(@(r,theta)fun(r,theta,c,v),0,d/2,0,pi); %compte
         Is_z_base_col=0;


         else
         end
```

```

end

```

```
%File name: vertical_mindlin_shaft.m
%Author: Genis Majoral Oller
%For Civil Engineering bachelor's degree dissertation
%Date: 15/06/2018
function [Is_z]=vertical_mindlin_shaft(L,d,v,N,Pos_pile,Pos_force,i_afected,i_force);


    %% Preparing distances
    if i_afected==N+1
        z_afected=Pos_pile(i_afected-1,3);
        x_afected=Pos_pile(i_afected-1,1);
        x_force=Pos_force(i_force,1);
     else
        z_afected=Pos_pile(i_afected,4);
        x_afected=Pos_pile(i_afected,1);
        x_force=Pos_force(i_force,1);
     end

     %%
     if i_force==1
        z_force=Pos_force(i_force,4);

        lim_sup=z_force+L/(N)*0.5;
        lim_inf=z_force-L/(N)*0.5;
     elseif i_force==N

        z_force=Pos_force(i_force,4);

        lim_sup=z_force+L/(N)*0.5;
        lim_inf=z_force-L/(N)*0.5;

     else
        z_force=Pos_force(i_force,4);

        lim_sup=z_force+L/(N)*0.5;
        lim_inf=z_force-L/(N)*0.5;

     end


     %% Mindlin formula


      a=@(theta) 2*theta;
      r=d/2;

      h=z_afected;
      x= x_afected-x_force+d/2;
      y= @(theta)(d/2*sin(theta)+0.01);

      z=@(c) h+c;
      z1=@(c) h-c;
      R2=@(c,theta) (d^2/4+x^2-x*d*cos(2*theta)+z(c).^2).^0.5;
      R1=@(c,theta) (d^2/4+x^2-x*d*cos(2*theta)+z1(c).^2).^0.5;

      fun0=@(c,theta) z1(c).^2./R1(c,theta).^3;
      fun1=@(c,theta) (3-4*v)./R1(c,theta);
      fun2=@(c,theta) (5-12*v+8*v^2)./(R2(c,theta));
      fun3=@(c,theta) ((3-4*v)*z(c).^2-2*c.*z(c)+2*c.^2)./(R2(c,theta).^3);
      fun4=@(c,theta) (6*c.*z(c).^2.*(z(c)-c))./(R2(c,theta).^5);

      fun=@(c,theta) ( (1+v)/(8*pi*(1-v))*(fun0(c,theta)+fun1(c,theta)+...
          fun2(c,theta)+fun3(c,theta)+fun4(c,theta)) );

```

```
        Is_z=2*integral2(@(c,theta)fun(c,theta),lim_inf,lim_sup,0,pi/2);

end

```

```matlab
%File name: vertical_mindlin_pile2pile.m
%Author: Genis Majoral Oller
%For Civil Engineering bachelor's degree dissertation
%Date 10/06/2018
function [Is_z]=vertical_mindlin_pile2pile(L,d,v,N,Pos_pile,Pos_force,i_afected,i_force);


       %% Preparing distances
       if i_afected==N+1
           z_afected=Pos_pile(i_afected-1,3);
           x_afected=Pos_pile(i_afected-1,1);
           y_afected=Pos_pile(i_afected-1,2);

        else
           z_afected=Pos_pile(i_afected,4);
           x_afected=Pos_pile(i_afected,1);
           y_afected=Pos_pile(i_afected,2);

        end

        if i_force==N+1
           z_force=Pos_force(i_force-1,3);
           y_force=Pos_force(i_force-1,2);
           x_force=Pos_force(i_force-1,1);
        else
           z_force=Pos_force(i_force,4);
           y_force=Pos_force(i_force,2);
           x_force=Pos_force(i_force,1);
        end

        %% Funció en sí


         c=z_force;
         h=z_afected;
         x=(x_afected-x_force);
         y=(y_afected-y_force);


         z=h+c;
         z1=h-c;
         R1=(z1^2+x^2+y^2)^0.5;
         R2=(z^2+x^2+y^2)^0.5;

         fun0= z1^2/R1^3;
         fun1= (3-4*v)/R1;
         fun2= (5-12*v+8*v^2)/(R2);
         fun3= ((3-4*v)*z^2-2*c*z+2*c^2)/(R2^3);
         fun4= (6*c*z^2*(z-c))/(R2^5);

         Is_z=( (1+v)/(8*pi*(1-v))*(fun0+fun1+...
             fun2+fun3+fun4) );





end

```

```
%File name: vertical_pile_cap.m
%Original author: Simon Sauter (Pont de Candí, 2012)
%Original file name: -
%Adapted and modified by: Genis Majoral Oller
%For Civil Engineering bachelor's degree dissertation
%Date: 18/06/2018

function [ppp,p_global_z,v_global3,BC_change_z,u_dif_u,u_dif_o,u_head_z,u_mean,Z_head,u_dif_z]=vertical_pile_cap(p_global_z,v_global,...
                          i_time,Ap_global_z,D_global_z,Is_global_z,soil_strain_aux,...
                          N,piles_num,coef1,d,L,Ep,ppp,piles_head_type,piles,num_elem)

dir=3;
i_step=1;
u_head_z=zeros(piles_num,1);
p_global_z=zeros(size(p_global_z(:,1)));

piles_forces_top=ppp(:,:,i_time);


for i=1:piles_num
    u_head_z(i)=v_global((i-1)*(N+1)+1,i_time);
end

if piles_head_type==0
    %no action
    BC_change_z='No change';
    u_dif_u=0;
    u_dif_o=0;
    v_global3=v_global(:,i_time);
    u_mean=0;
    Z_head=0;
    u_dif_z=0;
else
    [u_mean Pos_head Z_head]=least_square(u_head_z,piles,piles_num);
    u_dif_z=zeros(piles_num,1);
    for i=1:piles_num
        u_dif_z(i)=u_head_z(i,1)-u_mean(i,1);
    end
    u_dif_mean=mean(u_dif_z);
    u_dif_u=min(u_dif_z);
    u_dif_o=max(u_dif_z);
    if (u_dif_u < -0.00005 || u_dif_o > 0.00005)&&(abs(u_dif_u-u_dif_o)>0.0001)
    while (u_dif_u < -0.00005 || u_dif_o > 0.00005)&&(abs(u_dif_u-u_dif_o)>0.0001)

        BC_change_z=zeros(piles_num*(N+1),1);
        for it=1:piles_num
            if i_step==1
                if sum(piles_forces_top(:,dir))==0
                    Fc=(u_dif_mean-u_dif_z(it))*10000;
                else
                    Fc=piles_forces_top(it,3)+(u_dif_mean-u_dif_z(it))*...
                        abs(sum(piles_forces_top(1:piles_num,3)))/piles_num*100;
                end
            else
                if sum(piles_forces_top(:,dir))>-0.01 && sum(piles_forces_top(:,dir))<0.01
                    Fc=piles_forces_top(it,3)+(u_dif_mean-u_dif_z(it))*10000;
                else
                    Fc=piles_forces_top(it,3)+(u_dif_mean-u_dif_z(it))*...
                        abs(sum(piles_forces_top(1:piles_num,3)))/piles_num*100;
                end
            end
            %B-matrix of pile displacement

```

```
            BC_change_z_aux=zeros(N+1,1);
            BC_change_z_aux(1,1)=coef1*Fc/(pi*d^2/4)*2*L/(N)/Ep;
            BC_change_z_aux(N+1,1)=Fc;

            ii_0=(it-1)*(num_elem+1)+1;
            ii=ii_0+(num_elem+1)-1;
            BC_change_z(ii_0:ii,1)=BC_change_z_aux;

            %
            piles_forces_top(it,dir)=Fc;

        end %end changing force

        %new stress/displacement

        p_global_z=(Ap_global_z-(D_global_z*Is_global_z))\(BC_change_z+D_global_z*soil_strain_aux(:,i_time));
        v_global3=Is_global_z*p_global_z+soil_strain_aux(:,i_time);
        %new condition
        for i=1:piles_num
            u_head_z(i)=v_global3((i-1)*(N+1)+1,1);
        end

        [u_mean Pos_head Z_head]=least_square(u_head_z,piles,piles_num);

        for i=1:piles_num
            u_dif_z(i)=u_head_z(i)-u_mean(i);
        end
        u_dif_mean=mean(u_dif_z);
        u_dif_u=min(u_dif_z);
        u_dif_o=max(u_dif_z);
        i_step=i_step+1;
        dir_iter=[dir i_step];

        if i_step>500
            dir_iter
            break
        else

        end

    end
    else
        BC_change_z='No change';
        v_global3=v_global(:,i_time);
    end

end
    I_step(3)=i_step;



  ppp(:,3,i_time)=piles_forces_top(:,3);

  clear dir_i dir_iter piles_forces_top a

end
```

```
%File name: results_analysis.m
%Author: Genis Majoral Oller
%Creation date: 20/4/2018
%Description: This subroutine calculates bending moments and axial forces
%              from displacements and lateral/vertical shaft loads

function [Mf_final_x,Mf_final_y,Q_global_x,axial_final]...
   =results_analysis(...
   piles_num,piles_length,piles_dia,piles_I,piles_E,piles_forces_top,...
    num_elem,w_global_x,w_global_y,v_global,p_global_z2,time_step,...
    piles_head_type,ppp,find_surface);

%% %-------------------------RESULTS' ANALYSIS--------------------------%%%
% ----------------------------------------------------------------------- %
if find_surface==0
N=num_elem;
for i_time=1:time_step

for i_pile=1:piles_num

    %------------------------------------------------Bending Moment
    ii_0=N*(i_pile-1)+1;
    ii=ii_0+(N-1);
    kk_0=(N+1)*(i_pile-1)+1;
    kk=kk_0+(N+1-1);

    w_aux_x=w_global_x(ii_0:ii,i_time);
    w_aux_y=w_global_y(ii_0:ii,i_time);
    v_aux=v_global(kk_0:kk,i_time);

    for r=1:length(w_aux_x)

    if r==1
       if piles_head_type==1
       Mf_aux(r,r:r+1)=[-2 2]*piles_E*piles_I/(piles_length/(N-1))^2;
       elseif piles_head_type==0
       Mf_aux(r,r:r+1)=[0 0]*piles_E*piles_I/(piles_length/(N-1))^2;
       else
       end

    elseif r==length(w_aux_x)
       Mf_aux(r,r-3:r)=[-1 4 -5 2]*piles_E*piles_I/(piles_length/(N-1))^2;
    else
      Mf_aux(r,r-1:r+1)=[1 -2 1]*piles_E*piles_I/(piles_length/(N-1))^2;
    end

    end

    Mf_final_x(:,1,i_pile,i_time)=Mf_aux*w_aux_x;
    Mf_final_y(:,1,i_pile,i_time)=Mf_aux*w_aux_y;
    %-----------------------------------------------------Shear Stress

    Q_aux=zeros(N,N);
    BC_V=zeros(N,1);
    BC_V(1,1)=-piles_forces_top(i_pile,1,1);

    for r=1:N

            if r==1
              Q_aux(r,r:r+1)=[-1,1]*2; %only in head-casted
            elseif r==2
                Q_aux(r,r-1:r+1)=[-1, 0, 1];
            elseif r==N
                Q_aux(r,r-3:r)=[-1 4 -5 2]*2;
```

```
        else
           Q_aux(r,r-1:r+1)=[-1, 0, 1];
        end

    end


    Q_global_x(:,1,i_pile,i_time)=-1/(2*piles_length/(N-1))*...
       Q_aux*Mf_final_x(:,1,i_pile,i_time);



    %------------------------------------------------------------ Axial
    auxiliar=zeros(num_elem,num_elem);
    L=piles_length;
    BC_axial=zeros(num_elem,1);
    BC_axial(1,1)=ppp(i_pile,3,i_time)-p_global_z2(kk_0,i_time)*...
                                    L/N*0.5*pi*piles_dia;
    BC_axial(num_elem,1)=p_global_z2(kk,i_time)*pi*piles_dia^2/4+...
                    p_global_z2(kk-1,i_time)*L/N*0.5*pi*piles_dia;

    for i=2:num_elem-1
        auxiliar(i,i-1:i+1)=-[-1,0,1];
    end

    axial_final(:,1,i_pile,i_time)=piles_E*pi*piles_dia^2/4/...
                            (2*L/N)*auxiliar*v_aux(1:N)+BC_axial;

end
end

else
    Mf_final_x=0;
    Mf_final_y=0;
    Q_global_x=0;
    axial_final=0;
end

end
```

```
%File name: results_plots.m
%Author: Genis Majoral Oller
%Creation date: 20/3/2018
%Description: bending moment plots. Piles to plot can be choosen in
%interesting_piles below, as well as timesteps at vector "last"

function results_plots(num_elem,piles,w_final_x,w_final_y,v_final,...
    Mf_final_x,Mf_final_y,axial_final,V_loss_pts,dia_0,find_surface,piles_length,soil_strain_auxz,soil_strain_auxy)
if find_surface==0
N=num_elem;
figure(99)
fig_aux=gcf;
fig_aux.Name='Tunnel';
hold on
grid minor

[X_S,Y_S]=meshgrid(-20:5:20,-50:5:50);
Z=X_S*0;
surf(X_S,Y_S,Z,'FaceAlpha',0.5,'EdgeColor','none');

for i_time=1:length(V_loss_pts(1,1,:))
plot3(V_loss_pts(:,2,i_time),V_loss_pts(:,1,i_time),...
    -V_loss_pts(:,3,i_time),'o','MarkerSize',8,'MarkerFaceColor',...
                                        'black','Markeredge','none');
end

for i=1:length(piles(1,1,:))
    plot3([piles(1,2,i),piles(N,2,i)],[piles(1,1,i),piles(N,1,i)],...
        -[piles(1,3,i),piles(N,3,i)],'o-','LineWidth',5,'MarkerSize',...
                                        6,'MarkerFaceColor','black')
end


 [X,Y]=meshgrid(-50:1:50,-dia_0/2:0.2:dia_0/2);
 Z=20-sqrt(dia_0^2/4-Y.^2);
 Z2=20+sqrt(dia_0^2/4-Y.^2);
 s=surf(Y,X,-Z);
 s.EdgeColor='none';
 s.FaceAlpha=0.75;
 s2=surf(Y,X,-Z2);
 s2.EdgeColor='none';
 s2.FaceAlpha=0.75;
 xlabel('HORIZONTAL "Y"')
 ylabel('HORIZONTAL "X"')
 zlabel('VERTICAL "Z"')
 title('Scheme: Surface + Tunnel + Piles')

 campos([495, -1562, 57])
 drawnow

hold off





%%

a=length(w_final_x(1,:,1));

switch a
    case 2
        interesting_piles=[1,2];
    case 4
```

```
        interesting_piles=[1,2,3,4];
    case 9
        interesting_piles=[1,4,5];
    case 13
        interesting_piles=[1,7];
    case 25
        interesting_piles=[1,2,3,4,8,13,25];
    otherwise
        interesting_piles=[1];
end

    final=length(w_final_x(1,1,:));
    if final ==1
        k=1;
    else
    k=4;
    end

    final=round(linspace(1,final,k));
%% X
for j=1:length(interesting_piles)
    figure(200)


    l=1;
    for i=final

        subplot(length(interesting_piles),2,2*(j-1)+1)
        hold on
        grid on
        plot(w_final_x(:,interesting_piles(j),i)*1000,-[piles(:,3,1)],...
                                '.-','LineWidth',1.5,'MarkerSize',12);
        st(l)='Tunnel advance # '+string(i)+'/'+...
                                    string(length(w_final_x(1,1,:)));
        hold off

        xlabel('Horizontal displacement (mm)')
        ylabel('Depth (m)')
        st_title(j)='Horizontal X displacement of pile num. '...
                                        +string(interesting_piles(j));


        subplot(length(interesting_piles),2,2*(j-1)+2)
        hold on
        grid on
        plot(Mf_final_x(:,1,interesting_piles(j),i),-[piles(:,3,1)],...
                                '.-','LineWidth',1.5,'MarkerSize',12);
        hold off

        xlabel('M_y (kNm)')
        ylabel('Depth (m)')
        st_title2(j)='Bending moment in pile num. '...
                                        +string(interesting_piles(j));
        l=l+1;
    end


end
    for j=1:length(interesting_piles)
    subplot(length(interesting_piles),2,2*(j-1)+1)
    hold on
    title(st_title(j))
    plot([0 0],[-30 0],'k--','LineWidth',1)
    axis([-100 0.5 -piles_length 0])
```

```
legend(st,'Location','northwest')

subplot(length(interesting_piles),2,2*(j-1)+2)
hold on
title(st_title2(j))
plot([0 0],[-30 0],'k--','LineWidth',1)
axis([-1600 1800 -piles_length 0])
legend(st)
end
set(gcf, 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);


%% Y Cumulative

for j=1:length(interesting_piles)
figure(250)


l=1;
for i=final

    subplot(length(interesting_piles),2,2*(j-1)+1)
    hold on
    grid on
    plot(w_final_y(:,interesting_piles(j),i)*1000,-[piles(:,3,1)],...
                              '.-','LineWidth',1.5,'MarkerSize',12);
    st(l)='Tunnel advance # ' + string(i)+'/'+...
                                   string(length(w_final_x(1,1,:)));
    hold off

    xlabel('Horizontal displacement (mm)')
    ylabel('Depth (m)')
    st_title(j)='Horizontal Y displacement of pile num. '...
                                     +string(interesting_piles(j));



    subplot(length(interesting_piles),2,2*(j-1)+2)
    hold on
    grid on
    plot(Mf_final_y(:,1,interesting_piles(j),i),-[piles(:,3,1)],...
                              '.-','LineWidth',1.5,'MarkerSize',12);
    hold off

    xlabel('M_x (kNm)')
    ylabel('Depth (m)')
    st_title2(j)='Bending moment in pile num. ' ...
                                     +string(interesting_piles(j));

    l=l+1;
end

end
for j=1:length(interesting_piles)
subplot(length(interesting_piles),2,2*(j-1)+1)
hold on
title(st_title(j))
plot([0 0],[-30 0],'k--','LineWidth',1)
axis([-10 0.5 -piles_length 0])
legend(st,'Location','northwest')

subplot(length(interesting_piles),2,2*(j-1)+2)
hold on
title(st_title2(j))
```

```
plot([0 0],[-30 0],'k--','LineWidth',1)
axis([-2000 2000 -piles_length 0])
legend(st)
end
set(gcf, 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);


%% Z Cumulative

for j=1:length(interesting_piles)
figure(300)

l=1;
for i=final

    subplot(length(interesting_piles),2,2*(j-1)+1)
    hold on
    grid on
    s=(interesting_piles(j)-1)*(N+1)+1;
    e=s+N;
    plot(v_final(:,interesting_piles(j),i)*1000,...
                -[piles(:,3,1);piles(N,3,1)],'.-','LineWidth',...
                                        1.5,'MarkerSize',12);
    st(l)='Tunnel advance # ' + string(i)+'/'+...
                                string(length(w_final_x(1,1,:)));
    hold off

    xlabel('Vertical displacement (mm)')
    ylabel('Depth (m)')
   st_title(j)='Settlement of pile num. '+string(interesting_piles(j));



    subplot(length(interesting_piles),2,2*(j-1)+2)
    hold on
    grid on
    plot(axial_final(1:N,1,interesting_piles(j),i),...
            -[piles(:,3,1)],'.-','LineWidth',1.5,'MarkerSize',12);
    hold off

    xlabel('Axial force (kN)')
    ylabel('Depth (m)')
    st_title2(j)='Axial force in pile num. '+...
                                 string(interesting_piles(j));


    l=l+1;
end

end

for j=1:length(interesting_piles)
subplot(length(interesting_piles),2,2*(j-1)+1)
hold on
title(st_title(j))
plot([0 0],[-30 0],'k-','LineWidth',2)
axis([-3 10 -piles_length 0])
legend(st,'Location','northeast')

subplot(length(interesting_piles),2,2*(j-1)+2)
hold on
title(st_title2(j))
plot([0 0],[-30 0],'k-','LineWidth',2)
axis([0 5500 -piles_length 0])
```

```
    legend(st)
    end

set(gcf, 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);







else
%%  Surface settlement plots
%%
N=num_elem;
a=length(w_final_x(1,:,1));

switch a
    case 2
        interesting_piles=[1,2];
    case 4
        interesting_piles=[1,2,3,4];
    case 9
        interesting_piles=[1,4,5];
    case 16
        interesting_piles=[1,2,3,6,7,16];
    case 25
        interesting_piles=[1,2,3,4,8,13,25];
    otherwise
        interesting_piles=[1];
end

    final=length(w_final_x(1,1,:));
    if final ==1
        k=1;
    else
    k=4;
    end

    final=round(linspace(1,final,k));
     %% Z Cumulative


    figure(300)
    plot(piles(:,2),-soil_strain_auxz(1:N,length(soil_strain_auxz(1,:))) );
    start='[X='+string(piles(1,1))+', Y='+string(piles(1,2))+'] ';
    final='[X='+string(piles(N,1))+', Y='+string(piles(N,2))+']'
    t='Surface settlement in XS: start '+start+' end '+final;
    title(t)
    ylabel('Settlement')
    xlabel('Horizontal XS')
    grid on
set(gcf, 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);

    %% Y Cumulative


    figure(301)
    plot(piles(:,2),-soil_strain_auxy(:,length(soil_strain_auxy(1,:))) );
    start='[X='+string(piles(1,1))+', Y='+string(piles(1,2))+'] ';
    final='[X='+string(piles(N,1))+', Y='+string(piles(N,2))+']'
    t='Y surface disp in XS: start '+start+' end '+final;
    title(t)
    ylabel('Disp mm')
    xlabel('Horizontal XS')
```

```
    grid on
set(gcf, 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);
end

end
```

```
% File name: compensation_grouting.m
% Author: Genis Majoral Oller
% Date: 16/7/2018
% For Civil Engineering bachelor's degree dissertation
% Computation of group of piles' displacements and stresses due to a tunnel
% excavation using Boussinesq, Mindlin and Sagaseta theory

clear all;

load('sample_filename.mat')%load existing problem to which comp. grout is applied


%% %---------------------COMPENSATION GROUTING
%% INPUT UP TO THE USER
% Define num of comp grout strips
num_strip=1;

%X of the strips

x1=linspace(-10,10,time_step);
x2=linspace(-20,20,time_step);
x3=linspace(-20,20,time_step);
%%% Depth of the strips
dp1=ones(length(x1),1)*(piles_length+1);
dp2=ones(length(x1),1)*(depth+0.5);
dp3=ones(length(x1),1)*(18+3);
dp4=10;

%%% Y of the strips
y=piles(1,2,2);
yy=piles(1,2,1);
yyy=0;
yyyy=2;

y1=ones(length(x1),1)*y;
y2=ones(length(x1),1)*yy;
y3=ones(length(x1),1)*yyy;
y4=ones(length(x1),1)*yyyy;


%%% Total volume of each strip
V1=-10/time_step*ones(size(x1));
V2=V1*3;
V3=V1/2;
V4=V1;

%%
V_comp_pts=zeros(num_strip,4,time_step);

for i=1:time_step
    if num_strip==1
    V_comp_pts(1,:,i)=[x1(1,i),y1(i),dp1(i),V1(1,i)];
    elseif num_strip==2
        V_comp_pts(1,:,i)=[x1(1,i),y1(i),dp1(i),V1(1,i)];
        V_comp_pts(2,:,i)=[x2(1,i),y2(i),dp2(i),V2(1,i)];
    elseif num_strip==3
         V_comp_pts(1,:,i)=[x1(1,i),y1(i),dp1(i),V1(1,i)];
        V_comp_pts(2,:,i)=[x2(1,i),y2(i),dp2(i),V2(1,i)];
         V_comp_pts(3,:,i)=[x3(1,i),y3(i),dp3(i),V3(1,i)];
    elseif num_strip==4
         V_comp_pts(1,:,i)=[x1(1,i),y1(i),dp1(i),V1(1,i)];
        V_comp_pts(2,:,i)=[x2(1,i),y2(i),dp2(i),V2(1,i)];
         V_comp_pts(3,:,i)=[x3(1,i),y3(i),dp3(i),V3(1,i)];
         V_comp_pts(4,:,i)=[x4(1,i),y4(i),dp4(i),V4(1,i)];
```

```
    end
end

%% %%%%%

tic

[soil_strain_comp]=sagaseta_3D(piles,V_comp_pts,time_step); %soil displacement around the piles


                        time_soil=toc;
                        fprintf('Soil displacements computed --> ok!\n')
                        fprintf('    Elapsed time: %.1f s \n',time_soil);

%% %%%%%%%%%%% SINK + COMPENSATION SOIL DISPLACEMENTS

soil_strain_final=soil_strain+soil_strain_comp;
%% %%%%%%%%%%%%%%%%%%%% HORIOZONTAL X ANALYSIS

[a,b]=size(piles_forces_top);
ppp_comp=zeros(a,b,length(soil_strain(1,1,1,:)));

coord=1; %x direction
tic
[cls_global_x,cBC_global_x,cAp_global_x,cD_global_x,p_global_x_comp,...
    w_global_x_comp,w_final_x_comp,w_head_x_comp,BC_change_x_comp,ppp_comp,w_dif_x2_comp,soil_strain_auxx_comp] = horizontal_analysis( piles,soil,piles_dia,...
    piles_forces_top,piles_E,piles_I,piles_head_type,piles_num,soil_strain_final,coord,num_elem,time_step,ppp_comp);

                        time_x=toc;
                        fprintf('Horizontal X analysis computed --> ok!\n')
                        fprintf('    Elapsed time: %.1f s \n',time_x);
clear cls_global_x cBC_global_x cAp_global_x cD_global_x

%% %%%%%%%%%%%%%%%%%%%% HORIZONTAL Y ANALYSIS

[a,b]=size(piles_forces_top);
ppp_comp=zeros(a,b,length(soil_strain(1,1,1,:)));

coord=2; %y direction
tic
[cls_global_x,cBC_global_x,cAp_global_x,cD_global_x,p_global_y_comp,...
    w_global_y_comp,w_final_y_comp,w_head_y_comp,BC_change_y_comp,ppp_comp,w_dif_y2_comp,soil_strain_auxy_comp] = horizontal_analysis( piles,soil,piles_dia,...
    piles_forces_top,piles_E,piles_I,piles_head_type,piles_num,soil_strain_final,coord,num_elem,time_step,ppp_comp);

                        time_y=toc;
                        fprintf('Horizontal Y analysis computed --> ok!\n')
                        fprintf('    Elapsed time: %.1f s \n',time_y);
clear cls_global_x cBC_global_x cAp_global_x cD_global_x

%% %%%%%%%%%%%%%%%%%%%% VERTICAL Z ANALYSIS

tic;
[p_global_z_comp,p_global_z2_comp,v_global_comp,v_global2_comp,v_global3_comp,v_final_comp,cD_global_z, cAp_global_z,...
    cls_global_z,cBC_z,cBC_change_z,u_dif_u_comp,u_dif_o_comp,ppp_comp,u_head_z_comp,u_mean_comp,Z_head_comp, u_dif_z_comp,soil_strain_auxz_comp]=...
    vertical_analysis(piles,piles_num,piles_dia,piles_head_type,...
    piles_length,soil,piles_E,soil_strain_final,num_elem,piles_forces_top,ppp_comp,find_surface);
                        time_z=toc;
                        fprintf('Vertical Z analysis computed --> ok!\n');
                        fprintf('   Elapsed time: %.1f s \n',time_z);

```

```
clear cIs_global_x cBC_global_x cAp_global_x cD_global_x




%% %-------------------------RESULTS' ANALYSIS--------------------------%%%
% ---------------------------------------------------------------------- %

[Mf_final_x_comp,Mf_final_y_comp,Q_global_x_comp,axial_final_comp]...
    =results_analysis(piles_num,...
    piles_length,piles_dia,piles_I,piles_E,piles_forces_top,num_elem,...
    w_global_x_comp,w_global_y_comp,v_global3_comp,p_global_z2_comp,time_step,piles_head_type,ppp_comp,find_surface);

%% %-----------------------------PLOTTINGS-----------------------------%%%
% ---------------------------------------------------------------------- %

results_plots_comp(w_final_x,w_final_y,v_final,...
    Mf_final_x,Mf_final_y,axial_final,num_elem,piles,w_final_x_comp,w_final_y_comp,v_final_comp,...
    Mf_final_x_comp,Mf_final_y_comp,axial_final_comp,V_loss_pts,dia_0,p_global_z2,p_global_z2_comp,p_global_x,p_global_x_comp,p_global_y,p_global_y_comp);

save('sample_filename');

```