

UNIVERSIDAD NACIONAL DE COLOMBIA

TRABAJO FINAL DE MAESTRÍA

---

# Interacción Suelo Estructura en Tablestacas

---

*Autor:*

Jorge Ernesto Rojas Correa

*Director:*

Profesor Félix Hernández  
Rodríguez

*Un trabajo presentado en cumplimiento de los requisitos  
para el grado de Magíster en Ingeniería - Geotecnia*

Departamento de Ingeniería Civil y Agrícola

6 de febrero de 2022



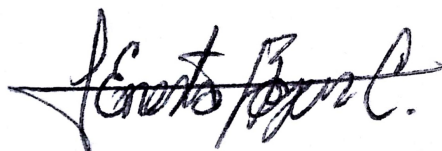
## Declaración de Autoría

Yo, Jorge Ernesto Rojas Correa, declaro que este trabajo titulado, «Interacción Suelo Estructura en Tablestacas» y todo su contenido son de mi autoría.

Así mismo, por medio de la presente confirmo que:

- Este trabajo se realizó mientras estaba en la candidatura para el título de maestría en esta Universidad.
- Cuando he consultado publicaciones de otros autores se ha atribuido claramente su trabajo.
- Donde he citado el trabajo de otros, la fuente se atribuye explícitamente. Con la excepción de tales citas, este documento es completamente propio.

Firma:



Fecha:

6 de febrero de 2022



*«There is nothing more vital to becoming educated, and there is nothing more vital than education to your future, and the future of those around you.»*

Jordan B. Peterson



UNIVERSIDAD NACIONAL DE COLOMBIA

# *Resumen*

Facultad de Ingeniería  
Departamento de Ingeniería Civil y Agrícola

Magíster en Ingeniería - Geotecnia

## **Interacción Suelo Estructura en Tablestacas**

Jorge Ernesto Rojas Correa

Este trabajo final de maestría propone un método incremental de interacción suelo estructura para el diseño de las tablestacas que incluye la evaluación de la capacidad última y el análisis de las deformaciones ocasionadas por la excavación del terreno. El análisis de falla, o de capacidad última, sirve para predimensionar la estructura; es decir para evaluar la profundidad de hincado y la rigidez a la flexión de la tablestaca. Por otro lado, el análisis de las deformaciones se hace mediante el procedimiento de interacción suelo estructura del profesor Zeevaert (1980) para cada una de las etapas de la excavación. El método que se propone tiene en cuenta que cada etapa de la excavación genera en el suelo que está al frente de la tablestaca una descarga, y con esto, una disminución tanto del esfuerzo de plastificación del suelo como de su rigidez. Este hecho conduce a que se obtengan mayores deformaciones y a que se requieran longitudes de anclaje superiores a las que comúnmente se utilizan en la construcción de estas estructuras.

Todos estos procedimientos se desarrollan por medio del diseño y la programación de un algoritmo y un aplicativo que permiten comparar fácilmente la respuesta de la tablestaca para cada una de las etapas de excavación por medio de las reacciones en el contacto suelo estructura, los valores de la fuerza cortante y del momento flector, y los desplazamientos horizontales de la estructura.

### **Palabras clave:**

- Interacción Suelo Estructura
- Tablestacas
- Rigidez del suelo
- Presiones activa y pasiva
- Presión lateral de tierras

# Abstract

## Soil-structure interaction in sheet piles

This master's thesis proposes an incremental soil-structure interaction method for the design of sheet piles that includes the evaluation of the ultimate capacity and the analysis of the deformations caused by soil excavation. The failure analysis, or ultimate capacity analysis, is used to pre-dimension the structure, in other words, to evaluate the driving depth and the bending stiffness of the sheet pile. On the other hand, the deformation analysis is carried out by means of the soil-structure interaction procedure of Professor Zeevaert (1980) for each stage of the excavation. The proposed method takes into consideration that each stage of excavation generates in the soil in front of the sheet pile an unloading, and with this, a decrease in both the plasticization stress of the soil and its stiffness. This fact leads to greater deformations and to the need for anchor lengths greater than those commonly used in the construction of these structures.

All these procedures are developed through the design and programming of an algorithm and an application that allow to easily compare the response of the sheet pile for each of the excavation stages by means of the reactions in the soil-structure contact, the values of the shear force and bending moment, and the horizontal displacements of the structure.

### Keywords:

- Soil Structure Interaction
- Sheet piles
- Soil stiffness
- Active and passive pressures
- Lateral earth pressure

## *Agradecimientos*

Agradezco enormemente a las personas más importantes en mi vida, mis padres y hermano, quienes por medio de su amor incondicional han hecho lo que soy hoy en día.

De igual manera, agradezco a mi director de trabajo final de maestría, el Ingeniero Félix Hernández, por su paciencia y apoyo, no solamente en el desarrollo de este trabajo, sino también en mi formación como profesional.

Finalmente, doy las gracias a la Universidad Nacional de Colombia por su compromiso en la formación de nuevas generaciones de profesionales y su incansable búsqueda de construir un mejor país para todos los colombianos.



# Índice general

<b>Declaración de Autoría</b>	<b>III</b>
<b>Resumen</b>	<b>VII</b>
<b>Agradecimientos</b>	<b>IX</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Diseño de tablestacas</b>	<b>5</b>
2.1. Generalidades . . . . .	5
2.1.1. Aplicaciones de las tablestacas . . . . .	6
2.1.2. Materiales de los tablestacados . . . . .	7
2.2. Análisis de tablestacas . . . . .	11
Coeficiente de presión de tierras en reposo . . . . .	13
2.2.1. Análisis de falla . . . . .	13
Coeficientes de presión de tierras en la falla . . . . .	14
Esfuerzos de falla . . . . .	15
2.2.2. Análisis de deformaciones . . . . .	17
2.3. Métodos convencionales de diseño de tablestacas . . . . .	18
<b>3. Análisis de interacción incremental suelo estructura en tablestacas</b>	<b>21</b>
3.1. Análisis de falla y predimensionamiento . . . . .	21
3.2. Análisis de deformaciones . . . . .	25
3.2.1. Métodos de Interacción Suelo Estructura . . . . .	25
Método de Zeevaert . . . . .	26
Método de relajación - deformación . . . . .	30
3.2.2. Discretización del dominio y condiciones iniciales del problema . . . . .	34
Condición inicial de esfuerzos . . . . .	34
Variación del módulo de reacción en profundidad y por la excavación . . . . .	36
3.2.3. Análisis de la tablestaca en el proceso de excavación . . . . .	38
Esfuerzos críticos . . . . .	38
Proceso de descarga de la tablestaca . . . . .	39
3.2.4. Cálculo de las reacciones nodales a partir de EMIH . . . . .	42
Condición $X_i = 0$ . . . . .	42
Condición $X_i = 1$ . . . . .	44
3.2.5. Análisis de plastificación . . . . .	48

3.2.6. Reducción de la rigidez del suelo en el frente de la tablestaca . . . . .	51
3.3. Algoritmo de diseño . . . . .	53
<b>4. Aplicación del algoritmo de diseño</b>	<b>57</b>
4.1. Parámetros de entrada . . . . .	57
4.2. Resultados . . . . .	59
<b>5. Conclusiones y Recomendaciones</b>	<b>65</b>
<b>A. Código del algoritmo de diseño</b>	<b>67</b>
A.1. SP_SEI.py . . . . .	67
A.2. appSPSEI_ui.py . . . . .	81
A.3. main.py . . . . .	86
<b>Bibliografía</b>	<b>97</b>

# Índice de figuras

2.1. Aplicaciones de las tablestacas (Modificado de Clayton y Milititsky, 1986) . .	8
2.2. Secciones transversales típicas para tablestacas de acero (Modificado de Das, 2012) . . . . .	9
2.3. Sección transversal típica para tablestacas de concreto prefabricado (Modificado de Das, 2012) . . . . .	10
2.4. Sección transversal típica para tablestacas de madera . . . . .	10
2.5. Condiciones de movimiento y estados de empujes del suelo (Modificado de Clayton y Milititsky, 1986) . . . . .	12
2.6. Trayectoria de esfuerzos para los estados activo y pasivo del suelo . . . . .	14
2.7. Esfuerzo activo y pasivo (Modificado de Das, 2012) . . . . .	16
2.8. Método del soporte fijo (Modificado de Clayton y Milititsky, 1986) . . . . .	19
2.9. Método del soporte fijo (Modificado de Clayton y Milititsky, 1986) . . . . .	19
2.10. Método de Padfield & Mair (1984) (Modificado de Conte, Troncone y Vena, 2017)	20
3.1. Esfuerzos asumidos para el predimensionamiento de la tablestaca . . . . .	22
3.2. Diagrama de flujo del pre-dimensionamiento . . . . .	24
3.3. Variación del error en el predimensionamiento de la tablestaca . . . . .	25
3.4. Definición de la ecuación matricial de interacción horizontal EMIH (Modificado de Zeevaert, 1980) . . . . .	28
3.5. Factores de influencia de esfuerzos unitarios horizontales (Modificado de Zeevaert, 1980) . . . . .	29
3.6. Articulaciones . . . . .	30
3.7. Momentos aplicados a un elemento tipo viga . . . . .	31
3.8. Análisis de cargas, momentos y desplazamientos en la metodología de relajación - deformación . . . . .	32
3.9. Esf nodales . . . . .	33
3.10. Geom Modelo . . . . .	35
3.11. Condición inicial de esfuerzos para un nodo . . . . .	35
3.12. Función hiperbólica para el módulo de reacción . . . . .	37
3.13. Distribución no lineal de esfuerzos en una tablestaca . . . . .	39
3.14. Corrección del esfuerzo activo . . . . .	40
3.15. Análisis de las etapas de descarga . . . . .	41
3.16. Plastificación del resorte del respaldo para condición activa . . . . .	48
3.17. Plastificación del resorte del frente para condición pasiva . . . . .	49
3.18. Plastificación del resorte del frente para condición pasiva y el resorte del respaldo en condición activa . . . . .	50

3.19. Plastificación del resorte del frente por condición activa . . . . .	50
3.20. Modificación de EMIH por la plastificación de nodos. . . . .	52
3.21. Condición corriente de esfuerzos para el análisis de pérdida de rigidez. . . . .	52
3.22. Diagrama de flujo del algoritmo de diseño . . . . .	55
4.1. Predimensionamiento - Interfaz gráfica de diseño . . . . .	58
4.2. Pestaña de diagramas - Interfaz gráfica de diseño . . . . .	59
4.3. Diagramas para la etapa de excavación No. 1 . . . . .	60
4.4. Diagramas para la etapa de excavación No. 2 . . . . .	60
4.5. Diagramas para la etapa de excavación No. 3 . . . . .	61
4.6. Diagramas para la etapa de excavación No. 4 . . . . .	61
4.7. Diagramas para la etapa de excavación No. 5 . . . . .	62
4.8. Diagramas para la etapa de excavación No. 6 . . . . .	62
4.9. Resortes plastificados en la última etapa de excavación . . . . .	64

# Lista de Abreviaciones

<b>EMAH</b>	Ecuación Matricial de Asentamientos o Desplazamientos <b>H</b> orizontales
<b>EMIH</b>	Ecuación Matricial de Interacción <b>H</b> orizontal
<b>ISE</b>	Interacción <b>S</b> uelo - <b>E</b> structura
<b>PFR</b>	Polímero de Fibra <b>R</b> eforzada
<b>PVC</b>	Polyvinyl Chloride
<b>RSC</b>	Relación de <b>S</b> obreconsolidación
<b>UV</b>	Ultra <b>V</b> ioleta



# Lista de Símbolos

$c'$	Cohesión efectiva del suelo	kN/m m
$E$	Módulo de elasticidad de la tablestaca	MPa
$k$	Rigidez unitaria del suelo	kN/m <sup>2</sup> m
$K_0$	Coefficiente de presión lateral de tierras en reposo	Adimensional
$K_a$	Coefficiente activo de presión lateral de tierras	Adimensional
$K_p$	Coefficiente pasivo de presión lateral de tierras	Adimensional
$q$	Esfuerzo de contacto o reacción	kN/m m
$P_a$	Fuerza activa	kN
$P_p$	Fuerza pasiva	kN
$Q$	Carga de reacción o fuerza de reacción	kN
$V$	Esfuerzo cortante	kN/m m
$M$	Momento flector	kN m
$X_i$	Reacción de la interfaz suelo estructura	kN
$z_c$	Profundidad de la grieta de tracción	m
$\delta$	Desplazamiento	m, mm
$(\delta_j)_0$	Desplazamiento inicial del nodo $j$	m, mm
$\kappa$	Rigidez concentrada o rigidez de cimentación	kN/m
$\lambda$	Espesor de la banda	m
$\gamma'$	Peso unitario sumergido del suelo	kN/m <sup>3</sup>
$\phi'$	Ángulo de fricción interna del suelo	°
$\sigma_a$	Esfuerzo activo	kN/m m
$\sigma_p$	Esfuerzo pasivo	kN/m m



# Capítulo 1

## Introducción

El objetivo principal de este trabajo final de maestría consiste en representar el comportamiento carga-desplazamiento de una tablestaca mediante el uso de una metodología de interacción suelo estructura. Los análisis realizados se enfocan en plantear un algoritmo que permita evaluar las condiciones de descarga presentes durante los procesos de excavación y así mismo, considerar los aumentos en la deformación de la tablestaca por efecto de los cambios de rigidez que sufre el suelo durante este proceso.

En general, debido a la necesidad de realizar excavaciones temporales o permanentes para la construcción de obras como redes de suministro, vías y muelles, se han utilizado algunas estructuras como las tablestacas puesto que brindan algunas ventajas por su facilidad de instalación y el manejo de diferentes niveles de excavación con o sin presencia de agua.

Así pues, al revisar el estado del arte respecto al diseño de tablestacas se puede encontrar que las primeras metodologías utilizadas están fundamentadas en evaluar un estado límite de falla mediante la suposición de distribuciones de esfuerzos idealizadas. Uno de los primeros planteamientos fue la propuesta de Blum (1931, citado en Conte, Troncone y Vena, 2017), quien definió una distribución uniforme de esfuerzos activos y pasivos que empujan la tablestaca haciéndola girar sobre un punto de rotación fijo. Otros autores como Padfield y Mair (1984) establecieron una distribución de esfuerzos que está compuesta por tres zonas de esfuerzos, para que al realizar un equilibrio de fuerzas y momentos se pueda determinar analíticamente el punto de rotación.

Sin embargo, debido a que estas propuestas contemplan únicamente la evaluación de un estado límite de falla que asume unos empujes críticos simplificados, no pueden ser utilizados como diseños definitivos debido a que no evalúan los esfuerzos y las deformaciones reales desde una perspectiva de funcionalidad. Por lo tanto, estas deben ser utilizadas únicamente como herramientas para establecer el dimensionamiento inicial de la tablestaca.

Además de estas metodologías, existen otros tipos de análisis que establecen el comportamiento carga-desplazamiento de diversas estructuras denominados métodos de «Interacción suelo-estructura» o «ISE». Estos involucran de manera conjunta la respuesta del suelo y de la tablestaca al utilizar ecuaciones que compatibilizan los esfuerzos y deformaciones a lo largo

de toda la estructura.

Por otro lado, los avances realizados en estas metodologías desde los años sesenta han permitido el desarrollo de múltiples propuestas que parten en su mayoría de la rama de la Ingeniería Estructural y que toman como fundamento los métodos de las flexibilidades y las rigideces utilizados para la solución de vigas indeterminadas al analizar la tablestaca como una viga elástica. Dos de las metodologías de ISE que se estudiarán con profundidad en este trabajo corresponden a las propuestas de Holloway (1991), nombrada como «Relajación - deformación», y la de Zeevaert (1980) debido a su fácil implementación en la programación de algoritmos en computador.

De acuerdo con lo anterior, en este trabajo final de maestría se propone una metodología para establecer el comportamiento carga-desplazamiento de una tablestaca sometida a un proceso de excavación hacia uno de sus costados mediante etapas sucesivas de descarga. Adicionalmente, con el fin de evaluar el efecto de cambio de rigidez del suelo en el frente de la estructura producido por la excavación, se plantea un modelo hiperbólico simple que muestra la importancia de evaluar este efecto dentro de los diseños de estas estructuras.

Para evaluar el problema mediante ISE, se analiza a la tablestaca como una viga a la cual se le aplican unas fuerzas de descarga en el frente de excavación que son equivalentes a los empujes finales de la etapa anterior. Así mismo, la compatibilidad de esfuerzos y deformaciones se evalúa con la incorporación de resortes unidos a la tablestaca que simulan la rigidez del suelo por medio de unos módulos de reacción conocidos que varían en función de la profundidad y del proceso de excavación.

De esta manera, una vez definidos los elementos principales del suelo y de la tablestaca, se procede a solucionar la viga estáticamente indeterminada por medio del método de Zeevaert (1980) para calcular las reacciones nodales y así mismo, establecer las deformaciones generadas por el proceso de descarga y por la plastificación del suelo.

Debido a que la metodología se plantea como un proceso incremental en el cual los resultados de la etapas se deben sumar acumulativamente y además se requiere de un proceso de verificación continuo e iterativo, se realizó la programación de un algoritmo en el lenguaje Python 3 y una interfaz gráfica de fácil manejo con la biblioteca PyQt5 con el fin de implementar una herramienta de análisis que pueda utilizarse para realizar diseños básicos y que a futuro pueda complementarse con la inclusión de nuevas características desarrolladas en futuros trabajos de investigación.

Si bien la metodología planteada en este trabajo considera la implementación de algunas hipótesis para tener en cuenta en los diseños de tablestacas, se presentan a continuación algunas limitaciones que deben ser tenidas en cuenta para futuros trabajos.

1. El modelo se desarrolló para un suelo homogéneo sin presencia de nivel freático.

2. Se definieron unos módulos de reacción conocidos para el suelo en función de la profundidad. Esto significa que para el análisis de ISE solo se tuvo en cuenta la ecuación matricial de interacción horizontal EMIH.
3. Se considera que al instalar la tablestaca en el terreno no se perturbo el suelo y por tanto se parte de una condición de esfuerzos en reposo.
4. La reducción de la rigidez del suelo por efecto de la excavación se presenta únicamente hacia el costado excavado.
5. Se considera una variación lineal de los módulos de reacción del suelo en función de la profundidad.

Finalmente, la estructura del documento se organizó de forma secuencial con el propósito de orientar al lector en el desarrollo de la metodología de diseño y así mismo, el procedimiento de cálculo realizado para obtener el comportamiento carga-desplazamiento de la tablestaca. Por lo tanto, la estructura del documento es la siguiente:

*CAPÍTULO 2: ANÁLISIS DE TABLESTACAS.* Inicialmente, este capítulo plantea la definición de una tablestaca como estructura flexible, las aplicaciones en proyectos de ingeniería, y los materiales más comunes que las componen. Adicionalmente, se realiza un breve resumen de las metodologías de diseño convencionales y de interacción suelo estructura.

*CAPÍTULO 3: ANÁLISIS DE INTERACCIÓN INCREMENTAL SUELO ESTRUCTURA EN TABLESTACAS.* En este capítulo se presenta todo el conjunto de procedimientos que permiten realizar el análisis de interacción suelo estructura y el algoritmo de diseño desarrollado para aplicar estos procedimientos.

*CAPÍTULO 4: APLICACIÓN DEL ALGORITMO DE DISEÑO.* En este capítulo se presentarán algunos ejemplos de aplicación del algoritmo de diseño que se consideraron relevantes para evaluar el cambio de rigidez de los resortes por efecto de la excavación y la importancia de la cohesión en la resistencia del suelo.

*CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES.* En este capítulo se presentarán las conclusiones más relevantes respecto a la metodología desarrollada y los resultados obtenidos de acuerdo con los objetivos planteados en este trabajo final de maestría. Adicionalmente, se incluirán algunas recomendaciones para los estudios futuros que puedan desarrollarse posteriormente.

*APÉNDICES:* Finalmente, en este capítulo se presentará el código del algoritmo de diseño y de la interfaz gráfica desarrollada en el lenguaje Python 3.



## Capítulo 2

# Diseño de tablestacas

### 2.1. Generalidades

Las tablestacas son un tipo de estructura de contención flexible constituida por múltiples elementos generalmente prefabricados que son embebidos o hincados en el suelo para resistir empujes horizontales generados por excavaciones parciales hacia alguno de sus costados. Estas estructuras son comúnmente utilizadas por sus características de flexibilidad y bajo costo debido a que permiten realizar avances significativos en los frentes de trabajo sin requerir grandes demandas de materiales y mano de obra. Adicionalmente, pueden ser utilizadas como contención de suelos blandos o con consistencias medias a firmes y son especialmente útiles en suelos con altos niveles freáticos o en zonas donde se requieren excavaciones con frentes húmedos como en el caso de costas y bordes de ríos.

Así mismo, según Singh y Chatterjee (2020) las tablestacas son usadas como estructuras permanentes o temporales que pueden soportar una cantidad limitada de suelo de relleno (hasta 5 m dependiendo del tipo de suelo y del sistema de contención), esta limitación radica en que la estabilidad depende principalmente a los esfuerzos de contacto desarrollados bajo el nivel de excavación y sobre la profundidad de empotramiento. De esta manera, dependiendo de las necesidades del proyecto se pueden utilizar estructuras adicionales como anclajes para mejorar la estabilidad de las tablestacas.

Aunque las ventajas de este tipo de estructuras son ampliamente conocidas, también existen algunas desventajas que según Eskandari y Kalantari (2011) dependen en su mayoría del método constructivo y de las condiciones ambientales presentes en la zona de estudio. Algunas de estas desventajas se resumen a continuación:

- Las secciones o perfiles de tablestaca raramente pueden ser usados como parte de la estructura permanente.
- La instalación de las tablestacas es difícil en suelos con gravas y cantos rodados. En estos casos es posible que no se alcancen las profundidades de hincado requeridas en los diseños.
- Las formas de la excavación están definidas por la geometría de la sección de tablestaca seleccionada, por lo tanto, para dar el acabado final de excavación se debe efectuar mediante trabajos manuales con el fin de evitar daños en la estructura.

- En caso que la instalación se realice mediante hincado, se pueden presentar alteraciones del suelo circundante debido a las vibraciones que se transmiten al suelo.

### 2.1.1. Aplicaciones de las tablestacas

Dadas las características mencionadas anteriormente, las tablestacas suelen ser usadas en diferentes proyectos de infraestructura en los cuales se involucran procesos de excavación y manejo de altos niveles freáticos como es el caso de puertos, conducción de ríos y construcciones *Offshore*. Por tanto, de acuerdo con los requerimientos de cada proyecto se debe establecer la conveniencia de aplicar diferentes métodos constructivos. Así pues, la instalación de la tablestaca puede hacerse antes de iniciar la excavación para contener el terreno natural, o se puede excavar hasta el nivel deseado para luego colocar la tablestaca y rellenar uno de sus costados. En la siguientes secciones se presentan algunas configuraciones típicas que son utilizadas para la construcción de estas estructuras.

#### Tablestacas en voladizo

Las tablestacas en voladizo se comportan como vigas en voladizo que son empotradas en su parte inferior en suelos de mayor rigidez, por lo que la misma estructura es la encargada de soportar los empujes de tierra del suelo contenido. Estas son generalmente usadas como muros de inundación o como muros de contención de tierras por su facilidad de instalación y economía, sin embargo, suelen presentar problemas cuando los niveles de excavación superan los 4 metros de altura debido a las altas deformaciones laterales que pueden producirse. La Figura 2.1a presenta un esquema general de la instalación de las tablestacas en voladizo en el que se indican los diferentes niveles del terreno y agua a los cuales pueden estar sometidas.

Este tipo de estructuras se encuentran generalmente en proyectos viales en los cuales se necesitan realizar excavaciones temporales o permanentes como zanjas para redes de suministro, vías y en la construcción de muelles pequeños.

#### Tablestacas ancladas

La característica principal de las tablestacas ancladas es que su estabilidad depende de la combinación del empotramiento de la estructura en el suelo y un sistema de anclajes localizado en la parte superior como se muestra en la Figura 2.1b. Esta configuración produce una mayor restricción el movimiento de la estructura, reduce las deformaciones laterales y momentos comparado con una tablestaca en voladizo. Por estos motivos, Clayton y Milititsky (1986) aconsejan utilizarlas para contener excavaciones de hasta 20 m de altura mediante la instalación de varias filas de anclajes.

Al igual que en el caso anterior, este tipo de tablestacas se usan comunmente en proyectos viales y construcción de muelles. Sin embargo, gracias a la mayor resistencia que ofrecen

pueden utilizarse en grandes excavaciones como en la construcción de sótanos y muelles de mayor magnitud.

### Entibados

Los entibados son un tipo de estructura de contención que consiste en la instalación de dos tablestacados, uno frente al otro, sobre los que se excavará el suelo para generar una trinchera. Debido a que se presentan deformaciones de ambos tablestacados hacia la excavación de manera convergente, se instalan una serie de puntales y largueros que soportan las cargas al conformar un arreglo de elementos verticales y horizontales como lo muestra la Figura 2.1c. Este tipo de estructura puede tener uno o más niveles de soporte en la parte externa y son diseñados generalmente para confinar excavaciones de gran profundidad.

Este tipo de estructuras son utilizadas generalmente en la construcción de grandes redes de acueducto o alcantarillado que pueden alcanzar grandes profundidades, además, son de gran utilidad para la contención temporal del terreno en proyectos viales con tramos deprimidos y accesos bajo el terreno natural.

### Ataguías

En el caso de las ataguías y otras estructuras de cerramiento como *Cofferdams*, se componen en general por tablestacas en voladizo o con anclajes que son diseñadas para mantener el agua por fuera del frente de obra de manera que se pueda construir sobre superficies secas como se muestra en la Figura 2.1d. De esta manera, al igual que las tablestacas ancladas, su estabilidad depende del empotramiento de la estructura y del tipo de anclaje utilizado en su instalación.

Debido a las múltiples aplicaciones que tienen este tipo de estructuras, es muy común encontrar diferentes arreglos geométricos como dobles tablestacados y arcos conectados que facilitan la ejecución de trabajos de desviación de cauces en las riberas de los ríos para la construcción de pilas de puentes y la instalación de grandes pilas en alta mar para proyectos petroleros que se encuentran lejos de la costa.

#### 2.1.2. Materiales de los tablestacados

En general, el uso de diferentes materiales en este tipo de estructuras dependen fundamentalmente de las cargas a soportar, la vida útil del proyecto y las condiciones de saturación del terreno. Aunque la instalación de tablestacas es mucho más difícil que las pilas o pilotes, si el diseño de estas estructuras es óptimo puede llegar a proveer una larga vida de servicio. A partir de lo anterior, a continuación se especifican los materiales más comunes que son utilizados para este tipo de estructura.

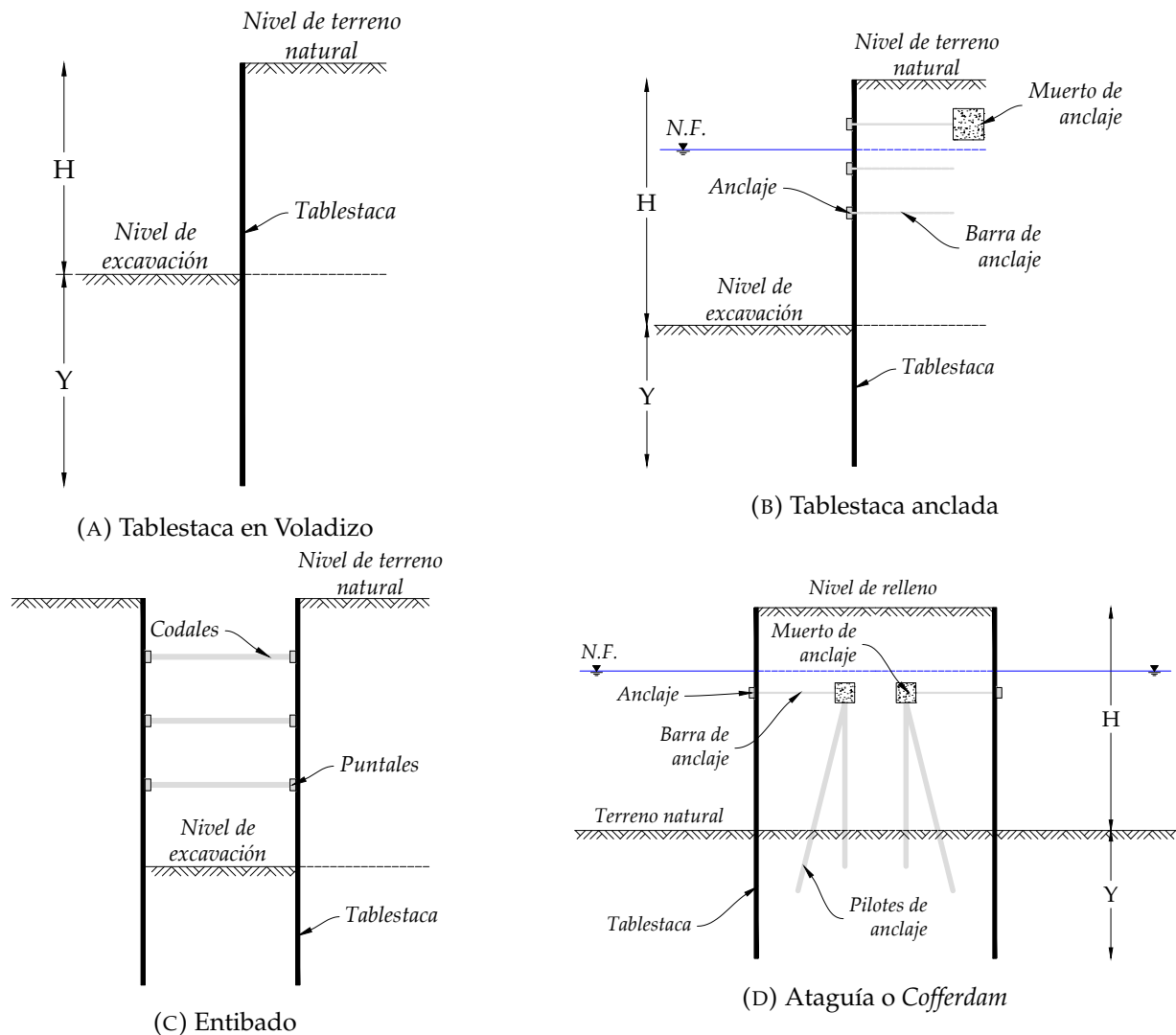


FIGURA 2.1: Aplicaciones de las tablestacas (Modificado de Clayton y Milititsky, 1986)

### Acero de bajo calibre

El acero de bajo calibre es el más común en la construcción de muros de tablestacas con frentes húmedos debido a su alta resistencia frente a las presiones de hincado, fácil instalación por su bajo peso y larga vida de servicio. Dentro de sus desventajas se destacan los problemas de corrosión que se generan por la exposición a condiciones ambientales muy fuertes, sin embargo, se pueden utilizar recubrimientos como pinturas epóxicas para prevenir estos inconvenientes.

A partir de los estudios realizados por Eskandari y Kalantari (2011), las tablestacas de acero de bajo calibre tienen espesores entre 10 y 15 mm y pueden llegar a soportar esfuerzos entre 170 MPa y 210 MPa. Además, existen diferentes secciones transversales, como

las presentadas en la Figura 2.2, que son utilizadas para la construcción de estructuras con diferentes geometrías y dar un mejor acabado a las excavaciones.

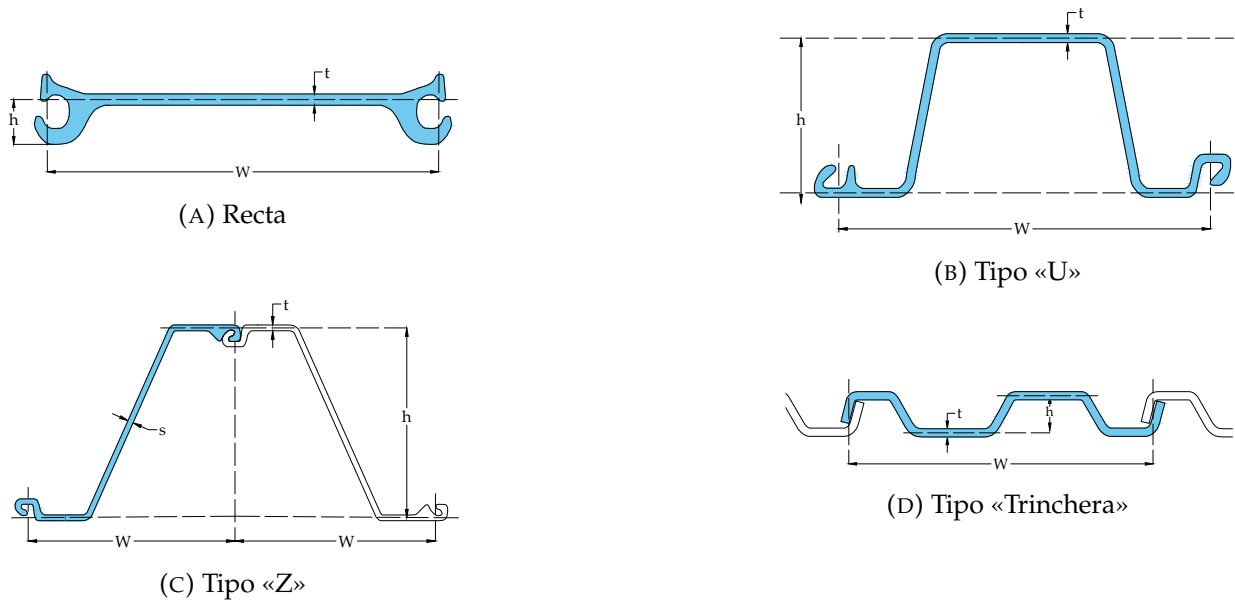


FIGURA 2.2: Secciones transversales típicas para tablestacas de acero (Modificado de Das, 2012)

### Concreto prefabricado

Las tablestacas de concreto prefabricado son mucho más estéticas pero producen mayores perturbaciones en los suelos al instalarse. Generalmente se utilizan como estructuras permanentes para canales y puertos en las que se tienen en cuenta su capacidad de soportar ambientes corrosivos por el recubrimiento de concreto sobre el acero de refuerzo.

Las secciones transversales para este material presentan formas rectas que son conectadas mediante articulaciones machihembradas y en la mayoría de los casos están recortadas diagonalmente en la parte inferior para facilitar el hincado como se muestra en la Figura 2.3.

### Madera

La madera como material para la construcción de tablestacados, se usa principalmente como una estructura de contención temporal de excavaciones poco profundas debido a su fácil instalación y economía. Sin embargo, todas las maderas se degradan gradualmente al exponerse a diferentes agentes ambientales como el agua o temperaturas extremas, las cuales permiten la proliferación de hongos y bacterias que aceleran su descomposición. Aún con estas desventajas, Eskandari y Kalantari (2011) sugieren que las tablestacas de madera se usen junto con algunos tratamientos químicos con cromo o aluminio para mejorar su durabilidad.

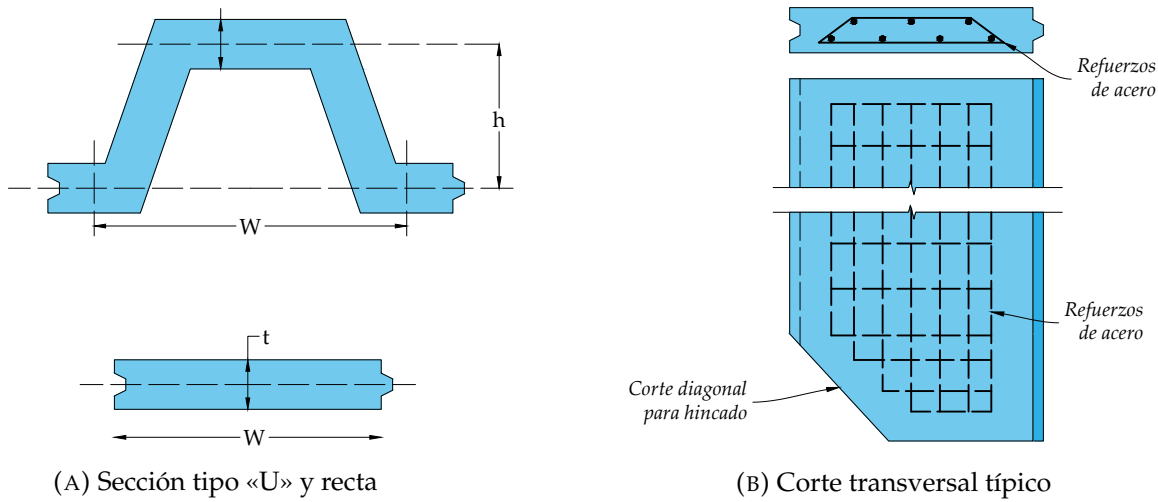


FIGURA 2.3: Sección transversal típica para tablestacas de concreto prefabricado (Modificado de Das, 2012)

La instalación de las tablestacas de madera se realiza en su mayoría mediante el hincado de secciones machihembradas reforzadas con elementos horizontales llamados «largeros» que conforman una estructura básica como la mostrada en la Figura 2.4a. Adicionalmente, en la Figura 2.4b se presenta otro tipo de instalación conformada por secciones hincadas de mayores dimensiones que soportan los empujes laterales y otras horizontales conocidas como «tablones» o «placas» que contienen el suelo.

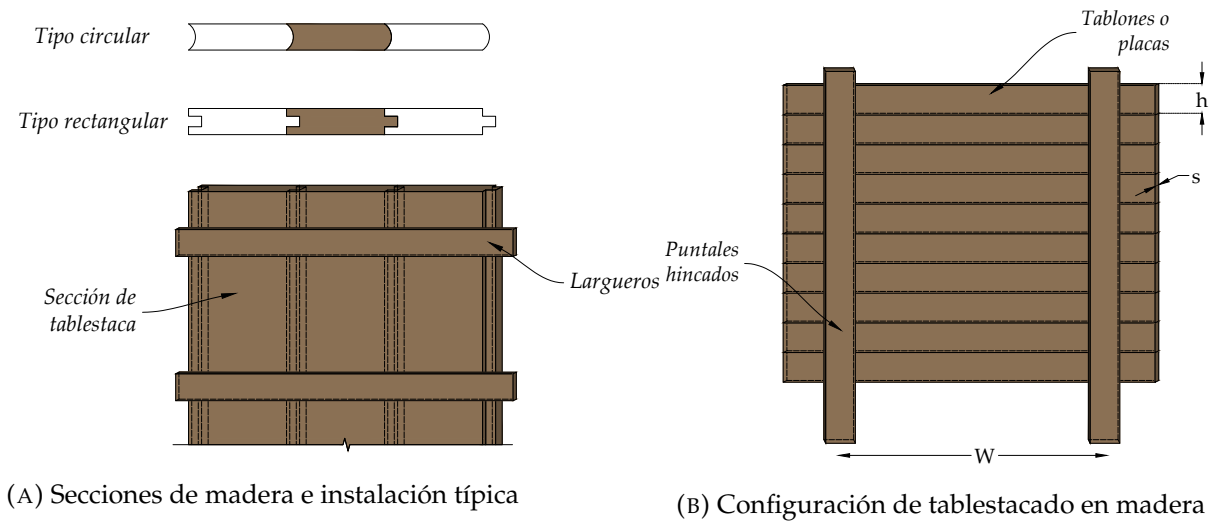


FIGURA 2.4: Sección transversal típica para tablestacas de madera

## Otros materiales

Debido a que los materiales convencionales como el concreto prefabricado, el acero y la madera tienen un componente de degradación importante al estar expuestos a condiciones ambientales muy fuertes, existen otros materiales como el aluminio, el PVC y los polímeros de fibras reforzadas PFR que son resistentes a la corrosión y presentan una alta durabilidad. De esta manera, en la Tabla 2.1 se presentan las características más destacadas para estos tres materiales.

Material	Características
Aluminio	<ul style="list-style-type: none"> <li>▪ Se construyen mediante perfiles de aluminio de aleación 6061.</li> <li>▪ Tienen alta resistencia a la corrosión, buena ductilidad y maleabilidad.</li> <li>▪ Suelen ser mucho más costosos que otro tipo de materiales.</li> <li>▪ Fácil transporte e instalación por su bajo peso.</li> </ul>
PVC	<ul style="list-style-type: none"> <li>▪ Son resistentes a los ambientes corrosivos, a los rayos UV y a productos químicos.</li> <li>▪ No requieren mantenimiento.</li> <li>▪ Son más ligeros que otros materiales, lo que facilita su instalación y transporte.</li> <li>▪ Son de bajo costo debido a que son plásticos y de fácil manufactura.</li> </ul>
Polímeros de fibras reforzadas	<ul style="list-style-type: none"> <li>▪ Son resistentes a los ambientes corrosivos.</li> <li>▪ Las tablestacas que son reforzadas con fibra de vidrio presentan una integridad estructural similar a la del acero.</li> <li>▪ Son muy ligeros y de alta resistencia, lo que facilita su instalación y transporte.</li> </ul>

TABLA 2.1: Características de materiales no convencionales de tablestacas

En general, las secciones transversales de estos materiales presentan las mismas geometrías utilizadas en las tablestacas de acero y pueden ser instaladas mediante los métodos tradicionales como el hincado, la vibración y el prensado estático.

## 2.2. Análisis de tablestacas

Debido a que las tablestacas son un tipo de estructura de contención, están sujetas a diferentes tipos de empujes laterales de tierra que dependen de la condición de movimiento que se ejerce sobre el suelo. Esta condición de movimiento puede ser analizada mediante un sistema idealizado compuesto por una estructura rígida que sostiene un suelo y que presenta un punto de giro en la base de la excavación.

Cuando la estructura permanece rígida en su sitio sin ningún movimiento y puede ser colocada sin perturbar el suelo como se muestra en la Figura 2.5a, se establece una condición de «Reposo» que corresponde a la condición inicial del suelo en su estado natural antes de realizar cualquier intervención.

Para el caso en el cual la estructura rota alejándose del suelo como se muestra en la Figura 2.5b, y la potencial falla del suelo involucra una pequeña porción del volumen contenido que es soportado en parte por los esfuerzos cortantes  $\tau$  en el plano de falla, se denomina como un caso «Activo». Por ende, en caso contrario en el que la estructura rota hacia el suelo contenido como en la Figura 2.5c, se puede generar una falla que involucra una gran masa de suelo con un esfuerzo cortante  $\tau$  que actúa contra el muro y que es denominado como un caso «Pasivo».

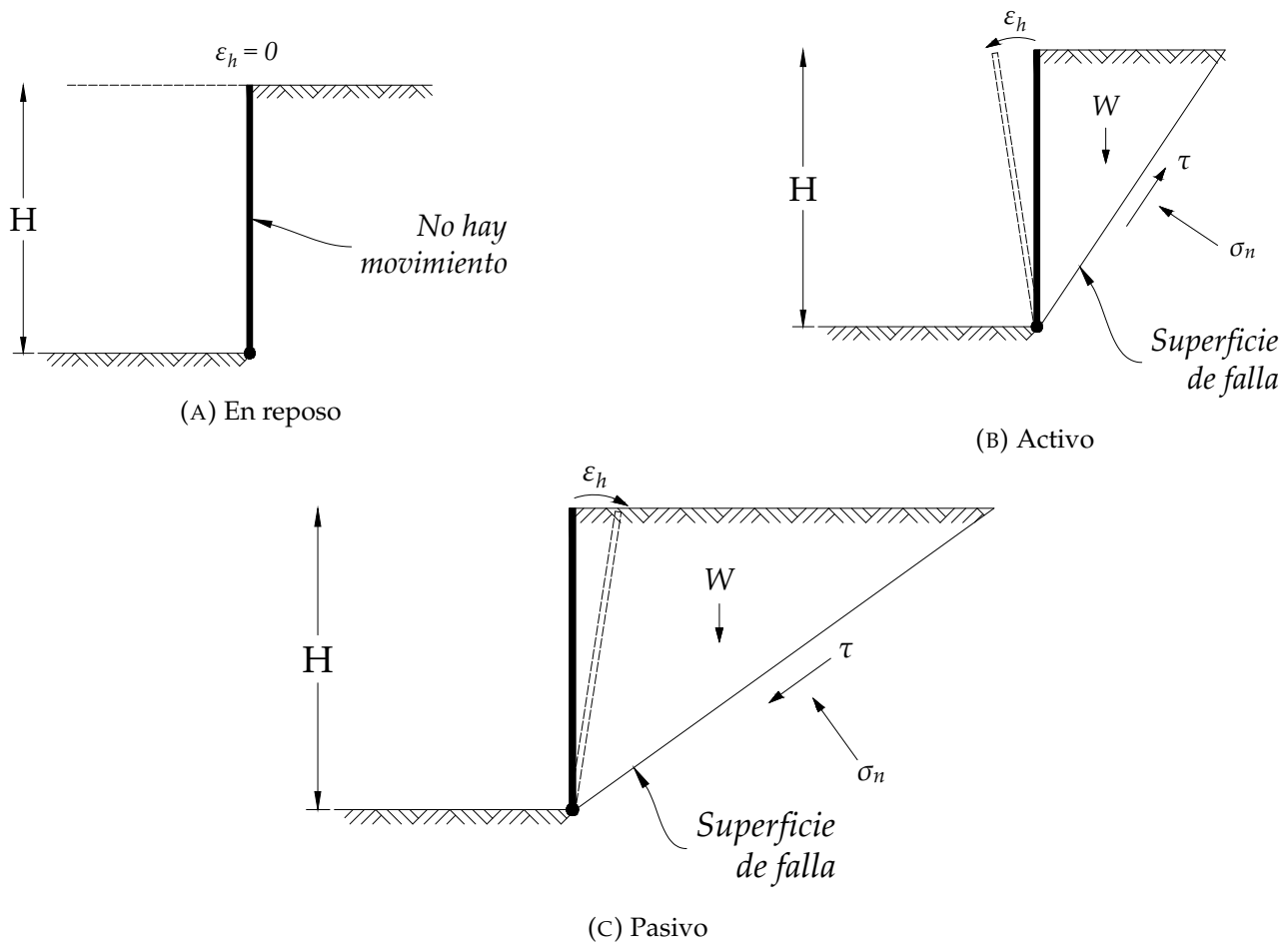


FIGURA 2.5: Condiciones de movimiento y estados de empujes del suelo (Modificado de Clayton y Milititsky, 1986)

A partir de diferentes estudios realizados por Terzaghi entre los años 1920 y 1930, en los cuales estudió el efecto de las presiones laterales entre una estructura y el suelo, definió el

concepto básico del coeficiente de presión lateral de tierras  $K$  que considera una relación entre el esfuerzo horizontal y vertical efectivos en cualquier profundidad medida desde la superficie.

$$K = \frac{\sigma_h}{\sigma_v} \quad (2.1)$$

Donde,

$\sigma_h$ : Esfuerzo horizontal efectivo

$\sigma_v$ : Esfuerzo vertical efectivo

### Coeficiente de presión de tierras en reposo

Teniendo en cuenta que el valor del coeficiente de presión de tierras en reposo  $K_0$  es un parámetro intrínseco de cada suelo, existen algunas aproximaciones empíricas utilizadas para calcularlo. Una de ellas es la propuesta de Jaky (1944) que fue planteada para suelos normalmente consolidados y luego fue complementada por Mayne y Kulhawy (1982) para suelos sobreconsolidados como se muestra en las ecuaciones 2.2 y 2.3 respectivamente.

$$K_0 \approx 1 - \sin \phi' \quad (2.2)$$

$$K_0 = (1 - \sin \phi') RSC^{\sin \phi'} \quad (2.3)$$

Donde,

$K_0$ : Coeficiente de presión lateral de tierras en reposo

$\phi'$ : Ángulo de fricción interna

$RSC$ : Relación de sobreconsolidación

#### 2.2.1. Análisis de falla

Al evaluar un punto  $(s_i, t_i)$  en un diagrama de trayectorias  $s - t$  como el presentado en la Figura 2.6, se observa que la trayectoria de esfuerzos parte de una condición inicial en la línea  $K_0$  y se mueve hacia las diferentes condiciones de falla dependiendo del proceso de carga o descarga al que se somete el suelo. Por lo tanto, la condición de falla activa se presenta cuando el suelo se descarga lateralmente como en un ensayo de compresión triaxial por extensión lateral y la falla pasiva cuando el suelo se carga lateralmente como en un ensayo de extensión triaxial por compresión lateral. Como consecuencia de lo anterior, se evidencia que el esfuerzo necesario para llegar a la condición de falla activa es mucho menor que la condición de falla pasiva y por consiguiente, el coeficiente  $K_a$  debe ser menor que el  $K_p$ .

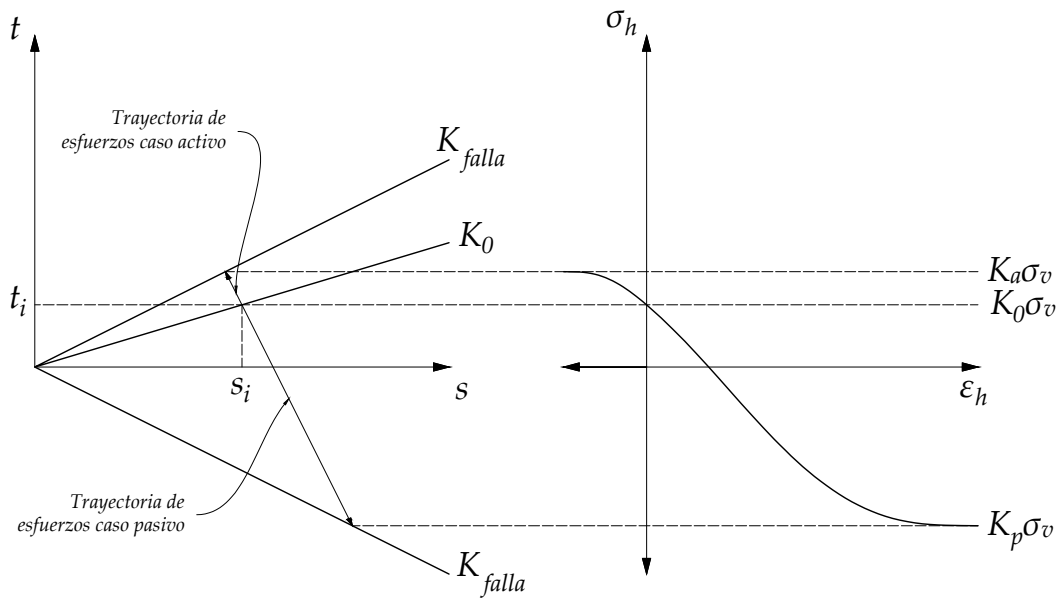


FIGURA 2.6: Trayectoria de esfuerzos para los estados activo y pasivo del suelo

### Coefficientes de presión de tierras en la falla

Por otro lado, para la definición de los coeficientes de presión lateral de tierras activo y pasivo, se han desarrollado una serie de metodologías que aplican las ecuaciones de equilibrio, compatibilidad de deformaciones, la aplicación de una ley constitutiva y la evaluación de las condiciones de frontera del problema para establecer el comportamiento de la estructura en la falla.

Una de estas metodologías corresponde al análisis de equilibrio límite, la cual busca encontrar una condición en la cual todas las fuerzas desestabilizadoras son iguales a las fuerzas resistentes mediante el uso de las ecuaciones de equilibrio horizontal, vertical y momentos que deben ser igual a cero para todo punto analizado.

Así mismo, otros métodos como el análisis límite evalúan los empujes laterales a partir de dos teoremas llamados límite inferior y límite superior, que se fundamentan en el comportamiento plástico de los materiales para solucionar el problema. Según Bojorque Iñiguez (2011), el teorema del límite inferior busca una distribución de esfuerzos que cumpla con todas las condiciones de equilibrio sin exceder el criterio de falla establecido mientras se maximiza la carga de falla. Por otra parte, el teorema del límite superior, busca una distribución de velocidad que cumpla con las ecuaciones de compatibilidad de deformaciones y el criterio de falla mientras se minimiza la energía interna de disipación.

Por lo tanto, mediante la aplicación de estas metodologías para estructuras de contención en condiciones planas de deformación, se pueden obtener los coeficientes de presión lateral

de tierras por Rankine (1857), para estructuras sin fricción y Coulomb (1776) para estructuras con fricción.

- Para el caso Rankine

$$K_a = \frac{1 - \sin \phi'}{1 + \sin \phi'} = \tan^2 \left( \frac{\pi}{4} - \frac{\phi'}{2} \right) \quad (2.4)$$

$$K_p = \frac{1 + \sin \phi'}{1 - \sin \phi'} = \tan^2 \left( \frac{\pi}{4} + \frac{\phi'}{2} \right) \quad (2.5)$$

- Para el caso Coulomb

$$K_a = \frac{1 - \sin \phi' \cos \psi}{1 + \sin \phi' \cos \psi} = \tan^2 \left( \frac{\pi}{4} - \frac{\phi' - \psi}{2} \right) \quad (2.6)$$

$$K_p = \frac{1 + \sin \phi' \cos \psi}{1 - \sin \phi' \cos \psi} = \tan^2 \left( \frac{\pi}{4} + \frac{\phi' + \psi}{2} \right) \quad (2.7)$$

Donde,

$\phi'$  : Ángulo de fricción interna

$\psi$  : Ángulo de fricción suelo - muro

$K_a$ : Coeficiente activo de presión lateral de tierras

$K_p$ : Coeficiente pasivo de presión lateral de tierras

### Esfuerzos de falla

Si se toma como referencia la metodología descrita por Rankine (1857), al relacionar los esfuerzos principales en un círculo de Mohr que toca la envolvente de falla Mohr-Coulomb para un material  $c' - \phi'$  se pueden establecer las ecuaciones 2.8 y 2.9 que corresponden a los esfuerzos activos y pasivos que actúan sobre la estructura de contención.

$$\sigma_a = \sigma'_0 K_a - 2c' \sqrt{K_a} \quad (2.8)$$

$$\sigma_p = \sigma'_0 K_p + 2c' \sqrt{K_p} \quad (2.9)$$

Donde,

$\sigma'_0 = \gamma z$ : Esfuerzo vertical efectivo

$K_a$ : Coeficiente activo de presión lateral de tierras

$K_p$ : Coeficiente pasivo de presión lateral de tierras

$c'$ : Cohesión efectiva del suelo

De acuerdo con lo anterior, los esfuerzos activos, pasivos y en reposo estarían relacionados así:

$$\sigma_a < (\sigma_h)_0 < \sigma_p \quad (2.10)$$

De esta manera, de las ecuaciones anteriores se puede observar que tanto el esfuerzo activo como el pasivo presentan un primer término relacionado con el ángulo de fricción y otro

con la cohesión del suelo. En la Figura 2.7 se puede observar una representación esquemática de estos esfuerzos en los cuales se aprecia una distribución negativa para el caso activo que disminuye hasta una profundidad  $z = z_c$  que es comúnmente llamada «Profundidad de la grieta de tracción», debido a que el esfuerzo negativo de tensión del suelo genera una grieta entre el suelo y el muro. Esta profundidad  $z_c$  puede ser calculada fácilmente al reemplazar  $\sigma_a = 0$  y despejar  $z_c$  como lo muestra la Ecuación 2.11.

$$\begin{aligned}\sigma_a &= \sigma'_0 K_a - 2c' \sqrt{K_a} \\ 0 &= \gamma z_c K_a - 2c' \sqrt{K_a} \\ z_c &= \frac{2c'}{\gamma \sqrt{K_a}}\end{aligned}\quad (2.11)$$

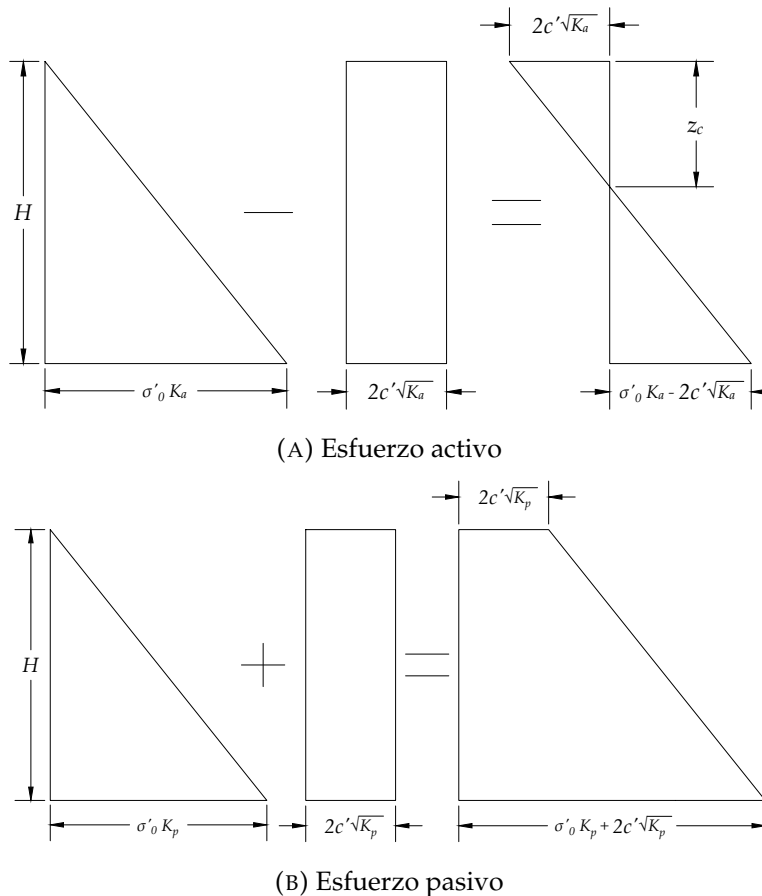


FIGURA 2.7: Esfuerzo activo y pasivo (Modificado de Das, 2012)

Adicionalmente, las fuerzas activas y pasivas totales por longitud unitaria de tablestaca se determinan a partir del área de los diagramas de esfuerzos o empujes laterales y corresponden a las ecuaciones 2.12 y 2.13 respectivamente.

$$P_a = \frac{1}{2}\gamma H^2 K_a - 2c'H\sqrt{K_a} \quad (2.12)$$

$$P_p = \frac{1}{2}\gamma H^2 K_p + 2c'H\sqrt{K_p} \quad (2.13)$$

Donde,

$H$ : Profundidad de análisis.

$P_a$ : Fuerza activa a una profundidad  $H$ .

$P_p$ : Fuerza pasiva a una profundidad  $H$ .

$\gamma$ : Peso unitario del suelo

Sin embargo, una vez aparece la grieta de tracción la fuerza activa se generará únicamente por la distribución de profundidades cuando  $z_c < z < H$ , así pues, la fuerza activa se puede expresar de la siguiente manera:

$$P_a = \frac{1}{2}(H - z_c)(\gamma H K_a - 2c'\sqrt{K_a})$$

$$P_a = \frac{1}{2}\left(H - \frac{2c'}{\gamma\sqrt{K_a}}\right)(\gamma H K_a - 2c'\sqrt{K_a}) \quad (2.14)$$

### 2.2.2. Análisis de deformaciones

Los análisis de deformaciones para las tablestacas se fundamentan principalmente en asumir su comportamiento como el de una viga elástica que se carga con los empujes calculados en el análisis de falla. Por consiguiente, al conocer los empujes, la longitud de la tablestaca y la rigidez a la flexión de la viga ( $EI$ ), se pueden calcular los diagramas de cortante, momento, giro y deflexión al realizar la integración de las ecuaciones 2.15, 2.16, 2.17 y 2.18.

$$V = \int \frac{-qdx}{EI} = EI \frac{d^3v}{dx^3} = EIv''' \quad (2.15)$$

$$M = \int \frac{Vdx}{EI} = EI \frac{d^2v}{dx^2} = EIv'' \quad (2.16)$$

$$\theta = \int \frac{-Mdx}{EI} = EI \frac{dv}{dx} = EIv' \quad (2.17)$$

$$v = \int \frac{\theta dx}{EI} \quad (2.18)$$

Donde,

$q$ : Distribución de carga en la estructura

$V$ : Esfuerzos cortantes

$M$ : Momentos flectores

$\theta$ : Giros

$\nu$ : Deflexiones

$E$ : Módulo de elasticidad

$I$ : Inercia de la tablestaca

Debido a que los diseños convencionales de tablestacas aplican el análisis de falla y de deformaciones por separado, los resultados obtenidos no representan el comportamiento real que sufre la estructura dado que estas características se producen de manera simultánea en un proceso de carga o descarga. Por este motivo, es necesario involucrar un análisis acoplado que permita evaluar el efecto combinado de los esfuerzos y deformaciones con el fin de mejorar y optimizar los diseños de estas estructuras.

Como consecuencia de las limitaciones que plantean estos procedimientos, se han desarrollado otras metodologías como las de Interacción Suelo-Estructura que garantizan la compatibilidad de esfuerzos y deformaciones en el contacto entre el suelo y la tablestaca. Algunos de los planteamientos utilizados para estos análisis se presentan en la siguiente sección.

### 2.3. Métodos convencionales de diseño de tablestacas

Los métodos convencionales se pueden definir como el conjunto de procedimientos basados en el método de soporte libre *Free Earth Support Method* y el método de soporte fijo *Fixed Earth Support Method* que fueron desarrollados y complementados a partir de los trabajos de diferentes autores como Rowe (1952), Terzaghi (1954) y Blum (1931, citado en Conte, Troncone y Vena, 2017) para tablestacas ancladas y en voladizo.

El método de soporte libre es utilizado para el análisis de tablestacas ancladas y en voladizo que asume la estructura como un elemento rígido que rota sobre un punto  $A$  donde se encuentra localizado un anclaje que no permite el movimiento como se muestra en la Figura 2.8. Así pues, la profundidad de empotramiento necesaria y la fuerza que resiste el anclaje son calculados al realizar un equilibrio de momentos en el punto  $A$  y equilibrio de fuerzas horizontales respectivamente.

Por otro lado, el método de soporte fijo considera la tablestaca como un elemento flexible que debe hincarse hasta una profundidad específica que garantice su empotramiento. Independientemente si la tablestaca es anclada o en voladizo, los esfuerzos entre los puntos  $C$  y  $D$  son reemplazados por una fuerza equivalente  $F_c$  como se muestra en la Figura 2.9a y 2.9b. De esta manera, la profundidad de empotramiento para tablestacas ancladas se calcula iterativamente hasta que el desplazamiento del anclaje en relación con el punto de fijación o empotramiento sea cero, y para tablestacas en voladizo al haber equilibrio de momentos respecto el punto  $C$  donde se aplica la fuerza equivalente  $F_c$ .

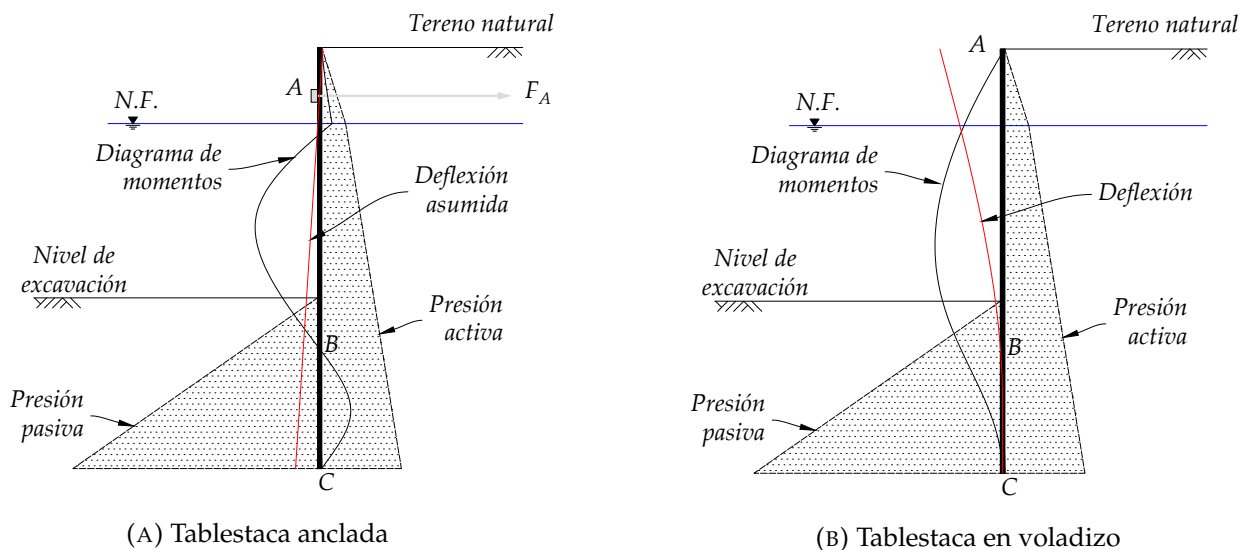


FIGURA 2.8: Método del soporte fijo (Modificado de Clayton y Milititsky, 1986)

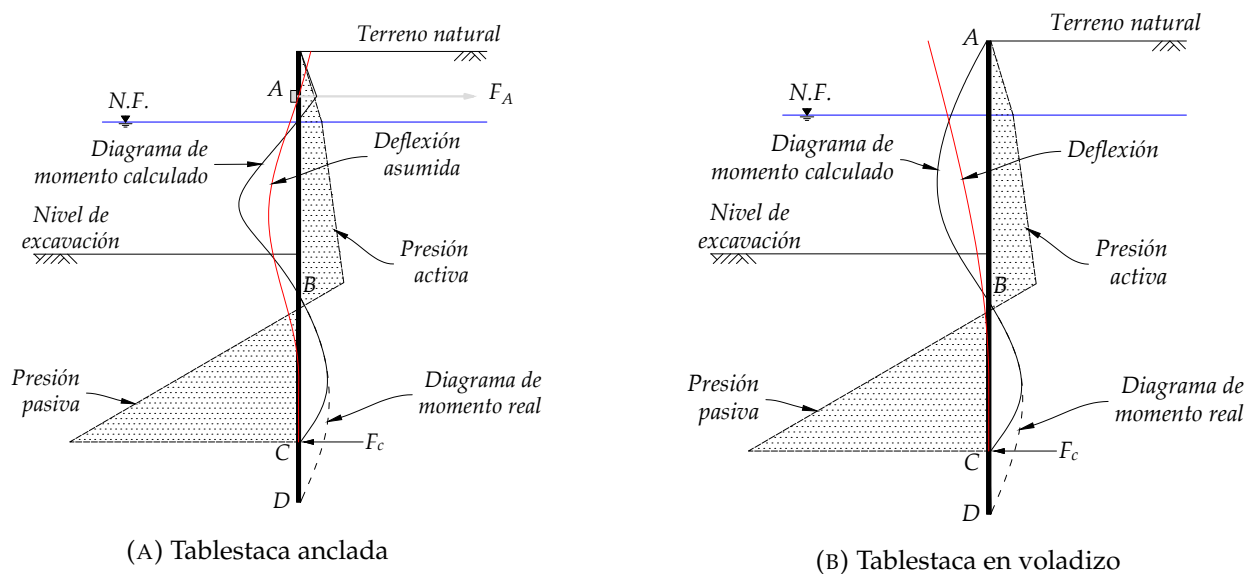


FIGURA 2.9: Método del soporte fijo (Modificado de Clayton y Milititsky, 1986)

Gracias a estos planteamientos, otros autores como Padfield y Mair (1984) desarrollaron una metodología para suelos cohesivos que involucra una distribución de esfuerzos a lo largo de una tablestaca en voladizo y gira sobre un punto definido bajo el nivel de excavación como se muestra en la Figura 2.10. Así mismo, los empujes están definidos de manera similar al método del soporte fijo por lo que se debe realizar equilibrio de fuerzas y momentos para determinar el punto de rotación y la profundidad de empotramiento de la estructura.

Además de estas metodologías, existen diferentes consideraciones de tipo experimental

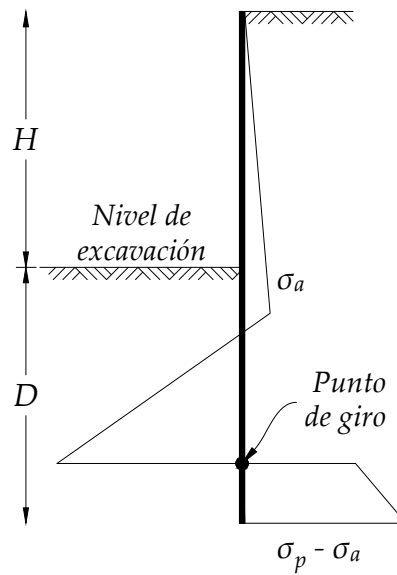


FIGURA 2.10: Método de Padfield & Mair (1984) (Modificado de Conte, Troncone y Vena, 2017)

que han permitido evaluar otras condiciones de empujes para suelos cohesivos y granulares. Sin embargo, dado que estas metodologías se fundamentan en un análisis de falla y no consideran las deformaciones tanto del suelo como de la tablestaca, solo deben aplicarse en diseños conceptuales o en casos donde se necesite realizar un predimensionamiento de la estructura.

## Capítulo 3

# Análisis de interacción incremental suelo estructura en tablestacas

La metodología propuesta permite evaluar el comportamiento de una tablestaca a medida que se excava en etapas sucesivas y se evalúa el cambio de rigidez de los módulos de reacción del suelo a partir de un análisis de interacción incremental suelo estructura que evalúa los esfuerzos y deformaciones que se producen en cada etapa de descarga. De esta manera, a continuación se presentan los procedimientos utilizados en el desarrollo de esta metodología.

### 3.1. Análisis de falla y predimensionamiento

El primer paso en la metodología propuesta consiste en dimensionar la tablestaca para tener un punto de partida respecto a los valores de geometría y resistencia mínimos que deben utilizarse en el análisis de ISE.

El procedimiento utilizado para este predimensionamiento se fundamenta en el método de soporte libre, el cual evalúa una condición de falla en el cual se presentan esfuerzos de compresión y extensión lateral que definen los estados pasivos y activos respectivamente. Al realizar un análisis cinemático de la estructura se encuentra que esta gira sobre un punto fijo como se muestra en la Figura 3.1. De esta manera, para garantizar el equilibrio de fuerzas y momentos a lo largo de la estructura, se calcula la longitud de empotramiento mínima necesaria para mantener la estabilidad.

Teniendo en cuenta la configuración de empujes mostrada en la Figura 3.1 se pueden definir seis zonas de análisis en las que se calculan los esfuerzos  $\sigma$ , las cargas resultantes  $P$  y los brazos de momento  $Y$  utilizados para el equilibrio de fuerzas y momentos.

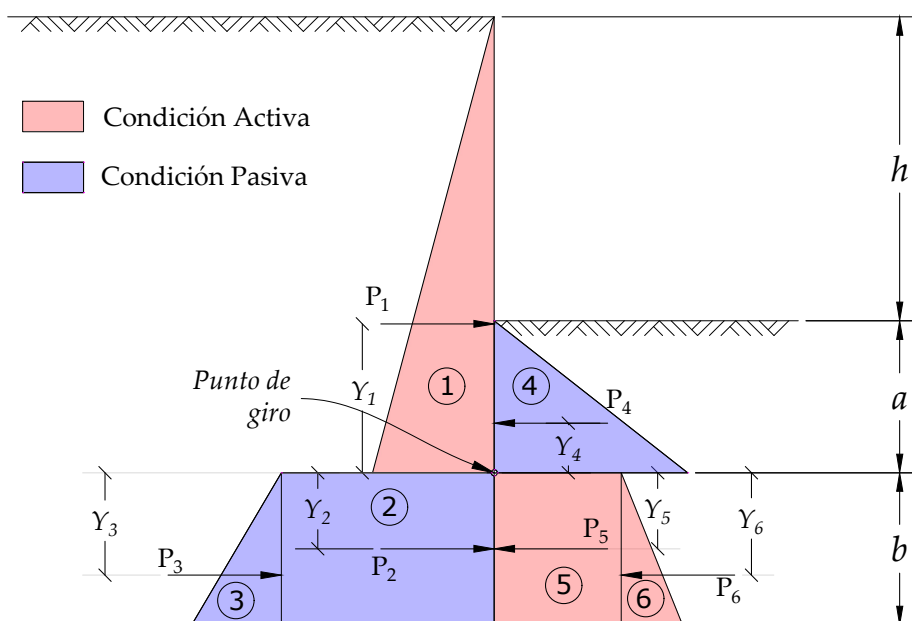


FIGURA 3.1: Esfuerzos asumidos para el predimensionamiento de la tablestaca

Así mismo, en la Tabla 3.1 se presenta el desarrollo numérico de cada una de las zonas de empujes.

Zona	$\sigma$	$P$	$Y$
1	$\gamma(h+a)K_a - 2c'\sqrt{K_a}$	$\frac{1}{2}(h+a) [\gamma(h+a)K_a - 2c'\sqrt{K_a}]$	$\frac{1}{3}(h+a)$
2	$\gamma(h+a)K_p + 2c'\sqrt{K_p}$	$b [\gamma(h+a)K_p + 2c'\sqrt{K_p}]$	$\frac{1}{2}b$
3	$b\gamma K_p$	$\frac{1}{2}b^2\gamma K_p$	$\frac{2}{3}b$
4	$a\gamma K_p + 2c'\sqrt{K_p}$	$\frac{1}{2}a [a\gamma K_p + 2c'\sqrt{K_p}]$	$\frac{1}{3}a$
5	$a\gamma K_a - 2c'\sqrt{K_a}$	$b [a\gamma K_a - 2c'\sqrt{K_a}]$	$\frac{1}{2}b$
6	$b\gamma K_a$	$\frac{1}{2}\gamma b^2 K_a$	$\frac{2}{3}b$

TABLA 3.1: Resumen de ecuaciones del predimensionamiento

Realizando la sumatoria de fuerzas se obtiene:

$$\begin{aligned}
\Sigma F &= 0 \\
&= \frac{1}{2}(h+a) \left[ \gamma(h+a)K_a - 2c'\sqrt{K_a} \right] + \left[ \gamma b(h+a)K_p + 2bc'\sqrt{K_p} \right] + \left[ \frac{1}{2}b^2\gamma K_p \right] - \\
&\quad \left[ \frac{1}{2}a(\gamma a K_p + 2c'\sqrt{K_p}) \right] - \left[ \gamma ab K_a - 2c'b\sqrt{K_a} \right] - \left[ \frac{1}{2}\gamma b^2 K_a \right] \\
&= \frac{1}{2}\gamma K_a \left[ (h+a)^2 - 2ab - b^2 \right] + c'\sqrt{K_a} [2b - (h+a)] + \\
&\quad \frac{1}{2}\gamma K_p \left[ 2b(h+a) + b^2 - a^2 \right] + c'\sqrt{K_p} [2b - a] \quad (3.1)
\end{aligned}$$

Y al realizar la sumatoria de momentos con respecto al punto de giro, se obtiene:

$$\begin{aligned}
\Sigma M &= 0 \\
&= -\frac{1}{6}(h+a)^2 \left[ \gamma(h+a)K_a - 2c'\sqrt{K_a} \right] + \frac{1}{2}b^2 \left[ \gamma(h+a)K_p + 2c'\sqrt{K_p} \right] + \frac{1}{3}b^3\gamma K_p + \\
&\quad \frac{1}{6}a^2 \left[ \gamma a K_p + 2c'\sqrt{K_p} \right] - \frac{1}{2}b^2 \left[ \gamma a K_a - 2c'\sqrt{K_a} \right] - \frac{1}{3}b^3\gamma K_a \\
&= -\frac{1}{6}\gamma K_a \left[ (h+a)^3 + 3ab^2 + 2b^3 \right] + \frac{1}{3}c'\sqrt{K_a} \left[ (h+a)^2 + 3b^2 \right] + \\
&\quad \frac{1}{6}\gamma K_p \left[ 3b^2(h+a) + 2b^3 + a^3 \right] + \frac{1}{3}c'\sqrt{K_p} \left[ 3b^2 + a^2 \right] \quad (3.2)
\end{aligned}$$

Una vez obtenidas las ecuaciones 3.1 y 3.2 se realiza el proceso iterativo mostrado en la Figura 3.2, el cual permite calcular las dimensiones de la tablestaca.

De esta manera, en el procedimiento anterior se define inicialmente un valor supuesto de  $a_s$  con el cual se calculan valores de  $b_1$  y  $b_2$  para la ecuación de equilibrio de fuerzas y momentos respectivamente. Luego, se define una función de error ( $D = b_2 - b_1$ ) la cual debe ser menor a 0.001 m (valor con el cual se obtiene una muy buena precisión) con el fin de evaluar si  $b_1$  y  $b_2$  son iguales, en el caso que no sean iguales se vuelve a suponer un valor de  $a_s$  hasta cumplir la condición del error admisible. Una vez que se conoce el valor de la dimensión  $a_{final}$  que cumple con las ecuaciones 3.1 y 3.2, se calcula el valor de  $b_{final}$ . Por consiguiente, el valor de la profundidad de empotramiento final es la suma de los valores  $a_{final}$  y  $b_{final}$ .

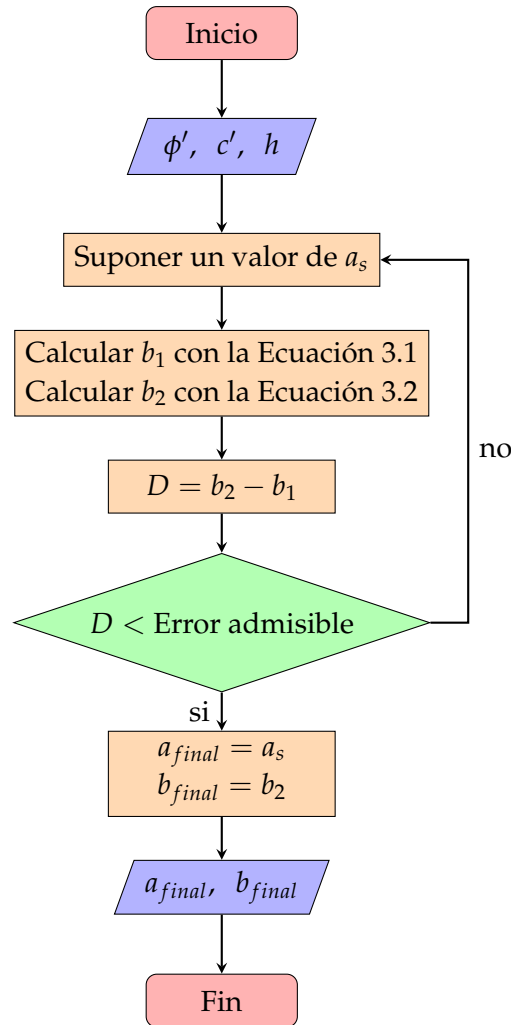


FIGURA 3.2: Diagrama de flujo del pre-dimensionamiento

En la Figura 3.3 se presenta un ejemplo de la variación de  $D$  respecto a los valores supuestos de  $a$ . Nótese que el valor de  $a_{final}$  corresponde a la intersección de la curva con el eje  $a$  del gráfico.

Una vez determinada la profundidad de empotramiento se utilizan los empujes finales para obtener los diagramas de cortante y momento de la estructura a partir de las ecuaciones de la viga elástica definidas en la sección 2.2.2.

Debido a que el predimensionamiento de la tablestaca se realiza mediante un análisis de falla, los valores máximos de los diagramas anteriores corresponden a los cortantes y momentos de fluencia que pueden utilizarse para calcular los valores mínimos del momento de inercia que debe tener la tablestaca para iniciar con el análisis de interacción suelo-estructura.

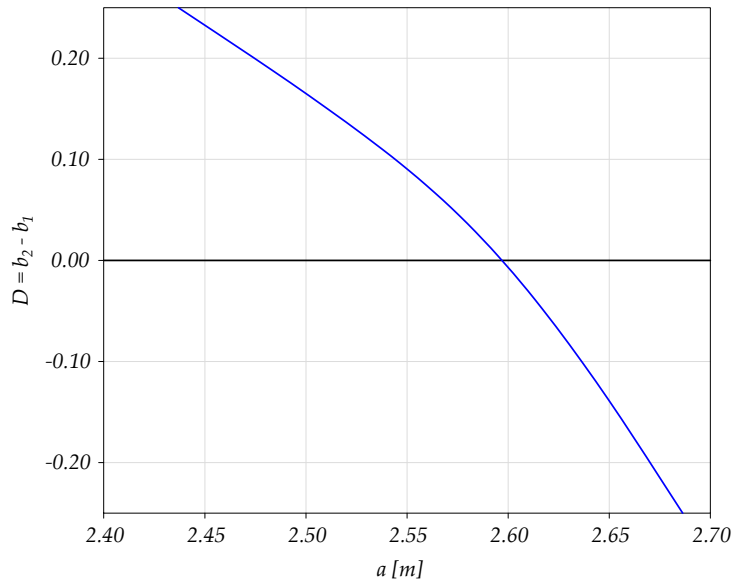


FIGURA 3.3: Variación del error en el predimensionamiento de la tablestaca

## 3.2. Análisis de deformaciones

Para realizar el análisis de deformaciones de la tablestaca se presenta a continuación un resumen de los métodos de interacción suelo estructura junto con sus aplicaciones dentro de la metodología propuesta.

### 3.2.1. Métodos de Interacción Suelo Estructura

Debido a la necesidad de involucrar análisis que garanticen la compatibilidad de esfuerzos y deformaciones en el contacto entre el suelo y la estructura, se han desarrollado los Métodos de Interacción Suelo - Estructura o ISE que presentan una diferencia importante respecto a los diseños convencionales. Estos métodos involucran el análisis del suelo mediante el uso de resortes elásticos y la discretización de la estructura en elementos como nodos o bandas sobre las que actúan las fuerzas externas y las reacciones del suelo. Según lo planteado por FEMA-P-2091 (2020), este tipo de modelo es denominado comúnmente como un «enfoque de subestructura», en donde los resortes se orientan siguiendo las rotaciones de la estructura y permitiendo los desplazamientos horizontales generados por las reacciones internas y externas aplicadas en los nodos mencionados.

La definición de los resortes elásticos que simulan el comportamiento del suelo se fundamenta en la propuesta de Winkler (1867), la cual relaciona la presión  $p$  que se ejerce sobre el suelo y su respectiva deformación  $\rho$  a partir de un módulo de reacción  $k_h$  como se muestra en la Ecuación 3.3.

$$p = k_h * \rho \quad (3.3)$$

De esta manera, para realizar los análisis de ISE se asumen generalmente valores del módulo de reacción del suelo mediante ensayos de placa de carga, pruebas de carga de tablestacas a escala real y relaciones empíricas con otras propiedades del suelo. La definición de este parámetro se presenta detalladamente en la sección 3.2.2.

Aunque existen diferentes métodos de ISE, para el desarrollo de este documento se describirán los planteamientos de Zeevaert (1980) y Holloway (1991), denominado como «Relación - deformación», por su fácil aplicación en la programación de algoritmos en computador.

### Método de Zeevaert

El método de Zeevaert (1980) aplicado a estructuras sujetas a empujes laterales como las tablestacas consiste en encontrar las reacciones del suelo y la estructura que generan una misma respuesta en términos de desplazamientos diferenciales para ambos elementos. Debido a que las tablestacas se analizan como vigas elásticas apoyadas en resortes que simulan el comportamiento del suelo, se utiliza el método de las rigideces para solucionar la viga indeterminada y calcular las reacciones.

#### Ecuación matricial de interacción horizontal EMIH

De esta manera, se utiliza el método de superposición para analizar dos condiciones de carga que al sumarse permiten determinar estas reacciones horizontales. La primera condición denominada  $X_i = 0$  se define al dejar únicamente dos apoyos para obtener una viga determinada y al aplicar las cargas o descargas externas  $P$  que generan deformaciones por flexión  $\Delta_{i0}^P$  y por la compresión o extensión de los resortes  $\Delta_{i0}^A$  como se muestra en la Figura 3.4a.

Para el caso de una tablestaca libre de girar en sus extremos, los desplazamientos horizontales totales  $\Delta_{i0}$  de la condición  $X_i = 0$  son calculados mediante la Ecuación 3.4.

$$\Delta_{i0} = \Delta_{i0}^A - \Delta_{i0}^P \quad (3.4)$$

Donde,

$\Delta_{i0}^P$  : Deformaciones por flexión de la tablestaca apoyada en  $a$  y  $b$ , debido a la descarga horizontal  $P$ .

$\Delta_{i0}^A$  : Desplazamiento horizontal de la tablestaca como elemento rígido al ceder los apoyos  $a$  y  $b$  por efecto de la deformación del suelo. Estos valores son función del modulo de cimentación  $k_a$  y  $k_b$  respectivamente.

Por otro lado, para calcular el efecto de las cargas o descargas externas  $P$  en los demás resortes se establece una segunda condición denominada  $X_i = 1$  en la cual se aplica una carga unitaria en cada nodo  $i$  para calcular los desplazamientos unitarios  $\bar{S}_{ji}$  mostrados en la Figura 3.4b.

Estos desplazamientos unitarios para la condición  $X_i = 1$  se obtienen al realizar la suma de todos los efectos combinados de los desplazamientos que sufre la tablestaca, los cuales se pueden definir de la siguiente manera:

Para el punto  $i$ ,

$$\bar{S}_{ii} = \bar{S}'_{ii} + \bar{S}''_{ii} + \bar{S}'''_{ii} \quad (3.5)$$

Y para el punto  $j$ ,

$$\bar{S}_{ji} = \bar{S}'_{ji} + \bar{S}''_{ji} \quad (3.6)$$

Donde,

$\bar{S}'_{ii}$  : Desplazamiento por flexión en el punto  $i$  por la carga unitaria aplicada en  $i$ .

$\bar{S}'_{ji}$  : Desplazamiento por flexión en el punto  $j$  por la carga unitaria aplicada en  $i$ .

$\bar{S}''_{ii}$  : Desplazamiento horizontal en el punto  $i$  como elemento rígido por deformación de los apoyos  $a$  y  $b$  con carga unitaria aplicada en  $i$ .

$\bar{S}''_{ji}$  : Desplazamiento horizontal en el punto  $j$  como elemento rígido por deformación de los apoyos  $a$  y  $b$  con carga unitaria aplicada en  $i$ .

$\bar{S}'''_{ii} = \frac{1}{k_i}$  : Desplazamiento del suelo por la carga unitaria aplicada en el punto  $i$ .

Una vez se conocen los desplazamientos  $\Delta_{i0}$  de la condición  $X_i = 0$  y los desplazamientos unitarios de la condición  $X_i = +1$ , se define la compatibilidad de deformaciones para cada una de los puntos donde se aplicaron las reacciones desconocidas como lo muestra la Ecuación 3.7 .

$$\begin{aligned} \bar{S}_{11}X_1 + \bar{S}_{12}X_2 + \bar{S}_{13}X_3 + \bar{S}_{14}X_4 &= \Delta_{10} \\ \bar{S}_{21}X_1 + \bar{S}_{22}X_2 + \bar{S}_{23}X_3 + \bar{S}_{24}X_4 &= \Delta_{20} \\ \bar{S}_{31}X_1 + \bar{S}_{32}X_2 + \bar{S}_{33}X_3 + \bar{S}_{34}X_4 &= \Delta_{30} \\ \bar{S}_{41}X_1 + \bar{S}_{42}X_2 + \bar{S}_{43}X_3 + \bar{S}_{44}X_4 &= \Delta_{40} \end{aligned}$$

De esta manera, al expresar matricialmente las ecuaciones anteriores se obtiene la «Ecuación matricial de interacción horizontal» EMIH que permite calcular las reacciones  $X_i$  para toda la tablestaca.

$$[\bar{S}_{ji}] \{X_i\} = \{\Delta_{i0}\} \quad (3.7)$$

#### **Ecuación matricial de asentamientos o desplazamientos horizontales EMAH**

Para calcular los desplazamientos horizontales del suelo que está en contacto con la tablestaca se divide el suelo en elementos horizontales denominados «Bandas» con longitudes

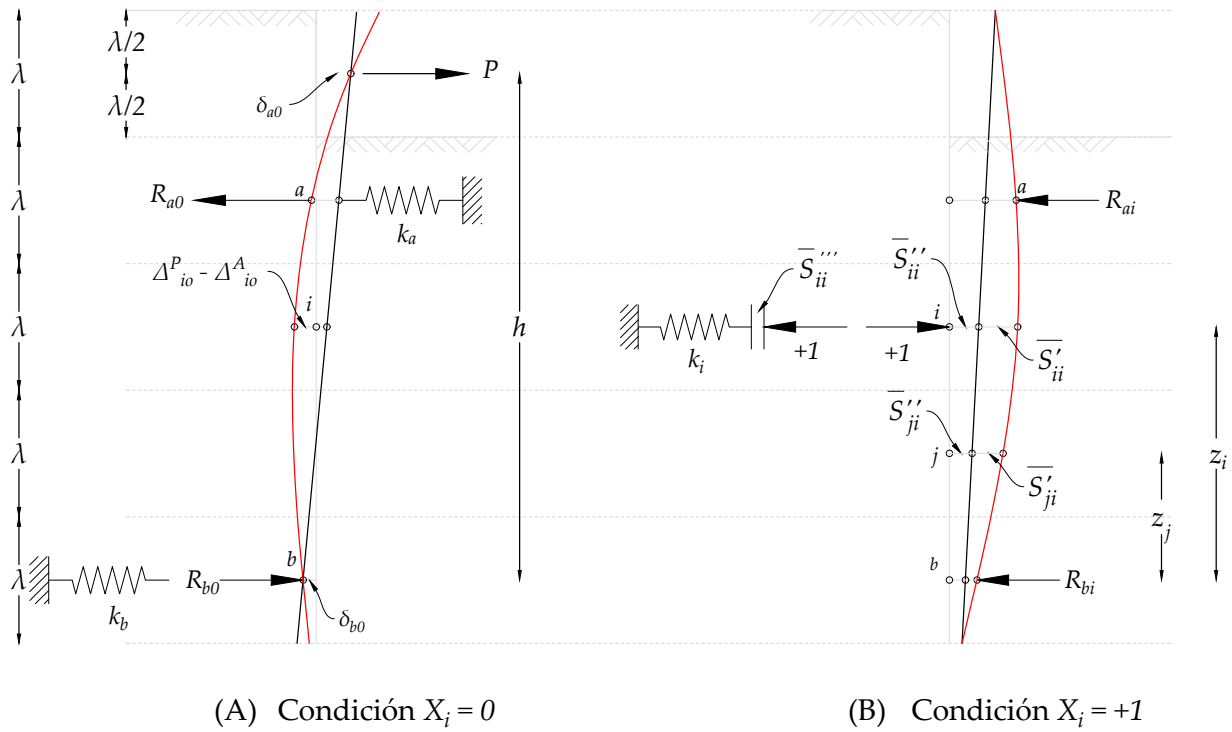


FIGURA 3.4: Definición de la ecuación matricial de interacción horizontal EMIH (Modificado de Zeevaert, 1980)

$d_i$ , en donde se aplica un esfuerzo unitario  $q_i = \frac{1}{\bar{a}}$  que está en función de un área tributaria  $\bar{a}$  y de la sección transversal de la tablestaca como se muestra en la Figura 3.5. Este esfuerzo permite determinar un factor de influencia  $I_{ji}^N$  para cada banda en el que define los desplazamientos horizontales unitarios de la siguiente manera:

$$\bar{\delta}_{ii} = (I'_{ii}M_i\Delta_x + I''_{ii}M_i\Delta_x + I'''_{ii}M_i\Delta_x + \dots)$$

$$\bar{\delta}_{ii} = (M_i\Delta_x) \sum_1^n I''_{ii}$$

Para la definición de estos factores de influencia en sentido horizontal se puede aplicar la solución de Mindlin (1936, Citado en Zeevaert, 1980) para una carga horizontal concentrada en un punto  $i$  bajo la superficie, en un medio semi-infinito, isótropo y homogéneo.

Si se analizan estos desplazamientos unitarios para cada punto  $i$  de forma matricial, al aplicar las reacciones  $X_1, X_2, X_3, \dots, X_i$  se obtienen los desplazamientos totales como se muestra en la Ecuación 3.8, a la cual se denomina «Ecuación matricial de asentamientos o desplazamientos horizontales» (EMAH).

$$\{\delta_i\} = [\bar{\delta}_{ji}] \left\{ \frac{X_i}{\bar{a}} \right\} \quad (3.8)$$

Estrato	$d_i$	$a$	$\delta_{ji}$	$I$ $\Delta x$	$II$ $\Delta x$	$III$ $\Delta x$	$IV$ $\Delta x$	$M_i$
A	$d_1$	1	$\delta_{1i}$	$I'_{1i}$	$I''_{1i}$	$I'''_{1i}$	$I''''_{1i}$	$M_1$
B	$d_2$	2	$\delta_{2i}$	$I'_{2i}$	$I''_{2i}$	$I'''_{2i}$	$I''''_{2i}$	$M_2$
C	$d_i$	$i$	$\delta_{ii}$	$I'_{ii}$	$I''_{ii}$	$I'''_{ii}$	$I''''_{ii}$	$M_i$
D	$d_4$	4	$\delta_{4i}$	$I'_{3i}$	$I''_{3i}$	$I'''_{3i}$	$I''''_{3i}$	$M_4$
E	$d_n$	$n$	$\delta_{ni}$	$I'_{4i}$	$I''_{4i}$	$I'''_{4i}$	$I''''_{4i}$	$M_5$

FIGURA 3.5: Factores de influencia de esfuerzos unitarios horizontales (Modificado de Zeevaert, 1980)

De esta manera, el cálculo de interacción suelo tablestaca se realiza aplicando las ecuaciones EMAH y EMIH planteadas anteriormente mediante un cálculo iterativo que involucra el módulo de reacción para encontrar la compatibilidad de deformación del sistema. Este proceso iterativo se puede resumir así:

1. Se suponen valores iniciales de los  $k_i$  de los resortes para encontrar las reacciones  $X_i$  por medio de EMIH.
2. Con estas reacciones se calculan los desplazamientos horizontales del suelo  $\delta_i$ , por medio de EMAH para determinar nuevos valores de  $k_i = \frac{X_i}{\delta_i}$ .
3. Con los nuevos valores de  $k_i$  se realiza nuevamente el ciclo de cálculo por EMAH y luego por EMIH hasta que los valores de las reacciones iniciales no cambien sensiblemente respecto a las calculadas.
4. Una vez terminado el ciclo se calculan los diagramas de cortantes y momentos de la tablestaca.

Debido a que el proceso de descarga planteado en este trabajo se realiza en etapas sucesivas hasta alcanzar la profundidad final de excavación, se define que el apoyo  $a$  para el análisis de EMIH se localiza en la banda inmediatamente inferior al nivel de excavación y el apoyo  $b$  se encuentra siempre en el nodo inferior de la tablestaca. Esta variación del apoyo superior se realiza con el fin de estimar con mayor precisión las reacciones finales de la

estructura y para facilitar el análisis de plastificación definido en la sección Sec: Plast. Así mismo, como limitación de la metodología de ISE planteada, se tomaron unos módulos de reacción del suelo conocidos por lo que no se utilizará EMAH para el cálculo de las reacciones.

### Método de relajación - deformación

Otra de las propuestas para realizar un análisis de Interacción Suelo - Estructura es la propuesta de Holloway (1991), que consiste en discretizar la tablestaca en nodos donde se colocan resortes en el frente y respaldo de la tablestaca que simulan la rigidez del suelo como se muestra en la Figura 3.6. Así mismo, cada uno de los nodos se toma como una articulación con el fin de evaluar las deformaciones generadas por las reacciones de los resortes. Debido a que las articulaciones producen una discontinuidad en el giro de la estructura, se aplican momentos en cada una de ellas para garantizar la continuidad.

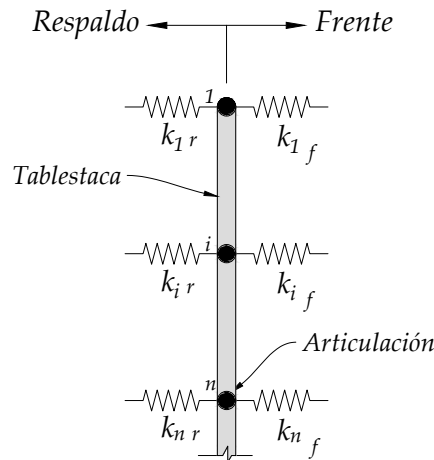


FIGURA 3.6: Articulaciones supuestas para el método de relajación-deformación

Al tomar un tramo de la estructura entre dos articulaciones y analizarlo como una viga elástica se pueden definir las ecuaciones de giro  $\theta_A$  y  $\theta_B$  para cada uno de los apoyos según lo mostrado en la Figura 3.7.

$$\theta_A = -\frac{L}{6EI}(2M_a + M_b) \quad (3.9)$$

$$\theta_B = \frac{L}{6EI}(M_a + 2M_b) \quad (3.10)$$

Así mismo, se analizan dos tramos de la estructura en el cual se aplica una carga  $P_i$  que corresponde a la reacción del suelo sobre el nodo  $i$  como se muestra en la Figura 3.8, se puede encontrar la relación entre esfuerzos, cargas, momentos y deformaciones.

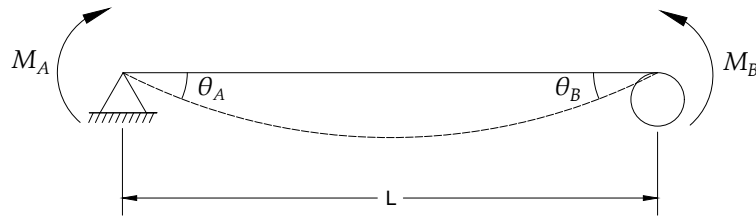


FIGURA 3.7: Momentos aplicados a un elemento tipo viga

Desde  $i - 1$ , se utiliza la Ecuación 3.10:

$$\frac{a}{6EI} (M_{i-1} + 2M_i) = \frac{\delta_{i-1} - \delta_i}{a} \quad (3.11)$$

Desde  $i + 1$ , se utiliza la Ecuación 3.9:

$$\frac{b}{6EI} (2M_i + M_{i+1}) = \frac{\delta_{i+1} - \delta_i}{b} \quad (3.12)$$

Luego, se deben sumar las ecuaciones 3.11 y 3.12,

$$\begin{aligned} \frac{1}{6EI} [M_{i-1}a + 2M_i(a+b) + M_{i+1}b] &= \frac{\delta_{i-1}}{a} - \delta_i \left( \frac{1}{a} + \frac{1}{b} \right) + \frac{\delta_{i+1}}{b} \\ \left[ M_{i-1} + 2M_i \left( 1 + \frac{b}{a} \right) + M_{i+1} \frac{b}{a} \right] &= \frac{6EI}{a^2} \left[ \delta_{i-1} - \delta_i \left( 1 + \frac{a}{b} \right) + \frac{a}{b} \delta_{i+1} \right] \end{aligned} \quad (3.13)$$

Según la teoría de resortes de Winkler (1867):

$$\begin{aligned} \delta_{i-1} &= \frac{q_{i-1}}{k_{r_{i-1}} + k_{f_{i-1}}} \\ \delta_i &= \frac{q_i}{k_{r_i} + k_{f_i}} \\ \delta_{i+1} &= \frac{q_{i+1}}{k_{r_{i+1}} + k_{f_{i+1}}} \end{aligned} \quad (3.14)$$

Al agrupar los términos que multiplican el lado derecho de la Ecuación 3.13, se obtiene la rigidez relativa suelo - estructura en el nodo  $i$ .

$$g_i = \frac{6EI}{a^2 k_i} \quad (3.15)$$

De esta manera, al reemplazar las ecuaciones 3.14 y 3.15 en la Ecuación 3.13 se obtiene lo siguiente:

$$M_{i-1} + 2M_i \left( 1 + \frac{b}{a} \right) + M_{i+1} \frac{b}{a} = g_i \left[ \frac{k_i}{k_{i-1}} q_{i-1} - q_i \left( 1 + \frac{a}{b} \right) + \frac{k_i}{k_{i+1}} \frac{a}{b} q_{i+1} \right]$$

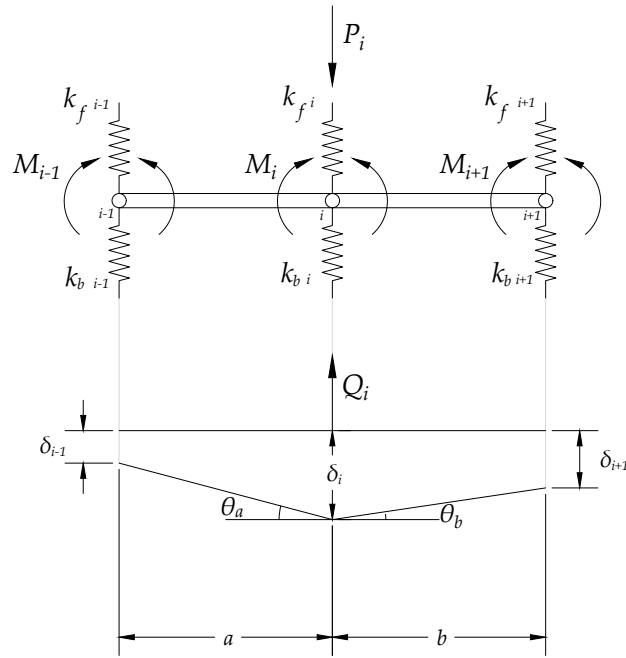


FIGURA 3.8: Análisis de cargas, momentos y desplazamientos en la metodología de relajación - deformación

Expresado lo anterior en forma matricial, se obtiene la ecuación de compatibilidad angular en el nodo  $i$ .

$$\left[ -g_i \frac{k_i}{k_{i-1}}, 1, \left(1 + \frac{a}{b}\right) g_i, 2 \left(1 + \frac{b}{a}\right), -g_i \frac{k_i}{k_{i+1}} \frac{a}{b}, \frac{b}{a} \right] \begin{Bmatrix} q_{i-1} \\ M_{i-1} \\ q_i \\ M_i \\ q_{i+1} \\ M_{i+1} \end{Bmatrix} = 0 \quad (3.16)$$

Una vez se ha realizado el análisis de la compatibilidad angular, se plantea un equilibrio de fuerzas para cada nodo. En el caso del nodo  $i$ , al evaluar las cargas y momentos que actúan se obtiene que:

Desde  $a$ ,

$$q_{i_a} = \frac{M_{i-1}}{a} - \frac{M_i}{a} \quad (3.17)$$

Desde  $b$ ,

$$q_{i_b} = \frac{M_{i+1}}{b} - \frac{M_i}{b} \quad (3.18)$$

Se realiza el equilibrio de fuerzas para el nodo  $i$  y luego se reemplazan los valores de  $q_{i_a}$  y  $q_{i_b}$  de las ecuaciones 3.17 y 3.18 se obtiene la ecuación de equilibrio en el nodo  $i$ .

$$\begin{aligned} \frac{M_{i-1}}{a} - M_i \left( \frac{1}{a} + \frac{1}{b} \right) + \frac{M_{i+1}}{b} + Q_i &= P_i \\ M_{i-1} - M_i \left( 1 + \frac{a}{b} \right) + M_{i+1} \frac{a}{b} &= a(P_i - Q_i) \end{aligned} \quad (3.19)$$

Adicionalmente, debido a que se trabaja cada punto con una relación de cargas y esfuerzos es necesario establecer una ecuación que permita hacer la conversión. En la Figura 3.9 se presentan tres elementos nodales sobre los cuales se aplican esfuerzos distribuidos linealmente.

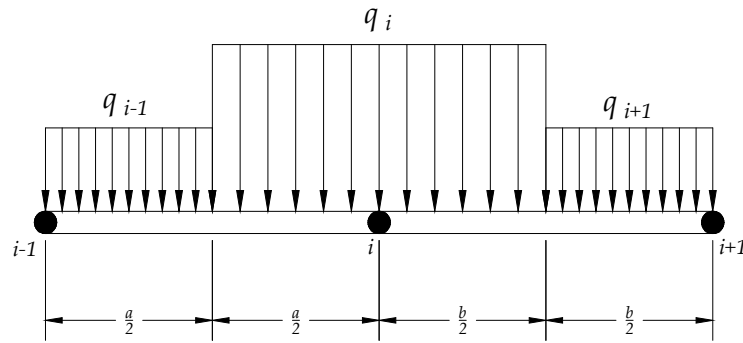


FIGURA 3.9: Relación entre cargas y esfuerzos nodales

De esta manera, al analizar el elemento desde  $a$  y toman los momentos respecto a  $i - 1$  se obtiene:

$$\begin{aligned} Q_{i_a} a &= q_{i-1} \frac{a}{2} \cdot \frac{a}{4} + q_i \frac{a}{2} \cdot \frac{3a}{4} \\ Q_{i_a} &= \frac{1}{8} (q_{i-1} a + 3q_i a) \end{aligned} \quad (3.20)$$

Desde  $b$  y se toman los momentos respecto a  $i + 1$ :

$$\begin{aligned} Q_{i_b} b &= q_{i-1} \frac{b}{2} \cdot \frac{b}{4} + q_i \frac{b}{2} \cdot \frac{3b}{4} \\ Q_{i_b} &= \frac{1}{8} (q_{i-1} b + 3q_i b) \end{aligned} \quad (3.21)$$

Se suman las ecuaciones 3.20 y 3.21 se obtiene la relación entre cargas y esfuerzos nodales para cualquier articulación o nodo intermedio.

$$Q_i = \frac{a}{8} \left[ q_{i-1} + 3q_i \left( 1 + \frac{b}{a} \right) + q_{i+1} \frac{b}{a} \right] \quad (3.22)$$

En el caso de un nodo inicial,

$$Q_i = \frac{b}{8} (3q_i + q_{i+1}) \quad (3.23)$$

En el caso de un nodo final,

$$Q_i = \frac{a}{8} (q_{i-1} + 3q_i) \quad (3.24)$$

En conclusión, la solución de este método consiste en aplicar los momentos necesarios en cada articulación para reestablecer la continuidad en los giros y de esta manera determinar los valores de las reacciones en cada resorte. Para este caso, es evidente que debido a la configuración del problema no se tienen cargas estructurales por lo que se dificulta el entendimiento del problema, en general las cargas aplicadas corresponden a un análisis idealizado de los esfuerzos que presentaría la tablestaca que supone cargas estructurales y resuelve el sistema como una viga.

### 3.2.2. Discretización del dominio y condiciones iniciales del problema

Para realizar los análisis de ISE se utilizó el método de Zeevaert (1980) debido a la facilidad de involucrar cargas estructurales dentro de la programación del algoritmo y la localización de los nodos en el centro de las bandas que permiten definir una mejor condición de frontera en los niveles de excavación. De esta manera, siguiendo esta metodología se caracteriza gráficamente el problema por medio de elementos geométricos discretos denominados como «bandas» que representan las capas de suelo y que son utilizadas para controlar las etapas de excavación en el frente de la estructura por medio de su longitud  $\lambda$ .

Así mismo, cada banda tiene asociado un nodo central sobre el cual se colocan dos resortes que definen el comportamiento reológico del suelo a partir de unos módulos de reacción  $k_f$  y  $k_b$  para los resortes del frente y respaldo respectivamente, que sirven como puntos de control para el cálculo de esfuerzos, fuerzas y deformaciones. Esta discretización del dominio se puede apreciar de manera gráfica en la Figura 3.10, donde las bandas se nombran alfabéticamente y los nodos numéricamente.

Además de los parámetros geométricos de las bandas y los módulos de reacción del suelo, también debe definirse la geometría de la tablestaca por medio del módulo de elasticidad y el módulo de Inercia que dependen del material y la sección de la tablestaca respectivamente. Estos últimos parámetros corresponden a los definidos en el predimensionamiento de la estructura y no deben variar hasta terminar el análisis de ISE donde se revisará si cumplen con los cortantes y momentos de fluencia.

#### Condición inicial de esfuerzos

Para los análisis de ISE desarrollados en esta metodología se considera que los esfuerzos iniciales en ambos lados de la estructura se encuentran en una condición de reposo que se calcula a partir del coeficiente de presión lateral de tierras propuesto por Jaky (1944). Según el modelo de interacción suelo - estructura planteado, para esta condición inicial de esfuerzos, se define que una unidad de tablestaca está entre dos resortes de igual módulo de rigidez  $k_j$  los cuales están comprimidos una cantidad  $(\delta_j)_0$  desde su longitud inicial  $l$  incluso antes

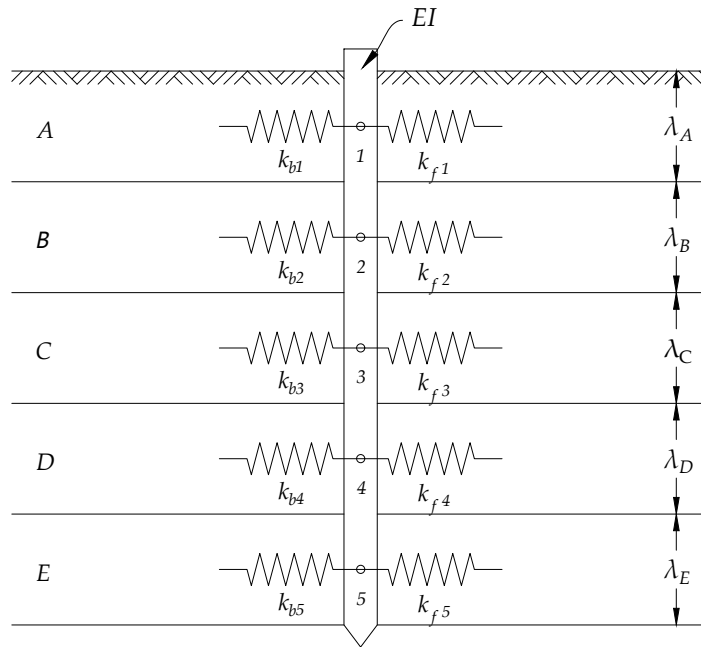


FIGURA 3.10: Discretización del dominio

de empezar con las etapas de excavación como se muestra en la Figura 3.11.

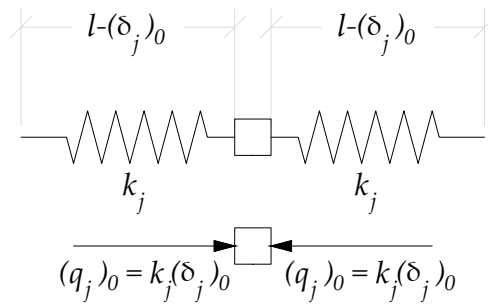


FIGURA 3.11: Condición inicial de esfuerzos para un nodo

De esta manera, ya que los esfuerzos de contacto  $q_j$  son iguales para ambos resortes, y al conocer el esfuerzo horizontal en reposo para el nodo  $j$ , se define que el desplazamiento en este nodo para una condición inicial de esfuerzos es el siguiente:

$$\begin{aligned}
 k_j(\delta_j)_0 &= \gamma' z_j K_0 \\
 (\delta_j)_0 &= \frac{\gamma' z_j K_0}{k_j}
 \end{aligned}
 \tag{3.25}$$

Donde,

$z_j$ : Profundidad del nodo  $j$  medido desde el nivel de referencia (terreno natural).

$K_0$ : Coeficiente de presión de tierras en reposo.

$k_j$ : Rigidez unitaria del suelo en el nodo  $j$ .

$\gamma'$ : Peso unitario sumergido.

### Variación del módulo de reacción en profundidad y por la excavación

Teniendo en cuenta que para el análisis de ISE desarrollado en este trabajo no se aplicará la EMAH de la metodología de Zeevaert (1980) ya que los módulos de reacción del suelo son conocidos, se presenta a continuación la definición de este parámetro para la condición inicial y durante el proceso de excavación.

Debido a que no existe perturbación alguna en el suelo para la condición inicial, los módulos de reacción de los resortes tanto del frente  $k_f$  como del respaldo  $k_b$  se estiman a partir de la propuesta de Palmer y Thompson (1948) en función de mediciones de estructuras instrumentadas o ensayos de placa de carga, de la profundidad de la estructura y del tipo de suelo.

$$k_h = k_L \left( \frac{z}{L} \right)^n \quad (3.26)$$

Donde,

$k_h$ : Módulo de reacción horizontal

$k_L$ : Módulo de reacción a la profundidad  $L$  medido en campo

$L$ : Profundidad total de la estructura

$n$ : Coeficiente empírico según tipo de suelo

$z$ : Profundidad a la cual se estima el módulo de reacción horizontal.

En general, el valor de  $n$  en la Ecuación 3.26 se toma como cero para arcillas y uno para arenas, sin embargo, otros autores como Davisson y Prakash (1963) proponen un valor de 0.15 para arcillas en condiciones no drenadas debido a los efectos de plastificación del suelo en las zonas mas cercanas a la superficie.

De forma similar al cambio en los esfuerzos en el frente de la tablestaca por efecto de la excavación, los módulos de reacción presentan una degradación no lineal que varía en profundidad al manifestar un mayor cambio en cercanía de la zona perturbada. Por tanto, para realizar una representación simplificada de este comportamiento se plantea una función hipérbolica como la mostrada en la Figura 3.12, que simula la variación del módulo de reacción en cada etapa de descarga para los resortes del frente.

La definición matemática de esta función se realizó mediante la ecuación canónica de la hipérbola con centro en el punto  $O$ , vértice en el eje  $z$  en el punto  $a$  y vértice en el eje  $k$  en el punto  $b$  la cual es presentada en la Ecuación 3.27.

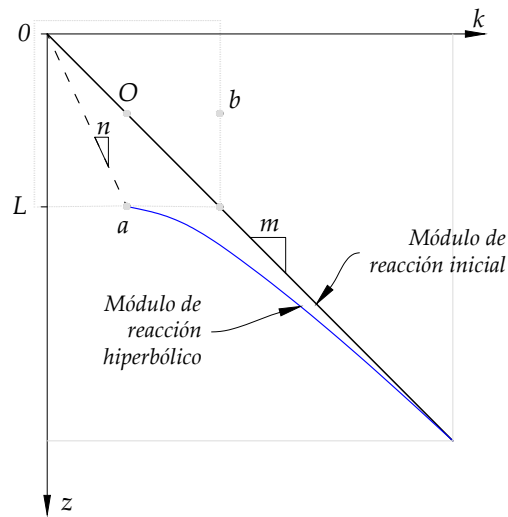


FIGURA 3.12: Función hiperbólica para el módulo de reacción

$$\frac{(z - y)^2}{a^2} - \frac{(k - x)^2}{b^2} = 1 \quad (3.27)$$

Donde,

$z$ : Profundidad de análisis

$k$ : Módulo de reacción a calcular

$a$ : Vértice en el eje  $z$

$b$ : Vértice en el eje  $k$

$x$ : Localización del centro en el eje  $k$

$y$ : Localización del centro en el eje  $z$

Al despejar el valor de  $k$  se obtiene que:

$$k = \frac{b\sqrt{-a^2 + y^2 - 2yz + z^2}}{a} + x \quad (3.28)$$

Así pues, los parámetros de la hipérbola se definieron a partir de la geometría de la Figura 3.12 para un nivel de excavación  $L$ , un módulo de reacción inicial con pendiente  $m$  calculado a partir de la metodología de Palmer y Thompson (1948) y un módulo de reacción reducido

con pendiente  $n = 2m$  utilizado para la zona excavada , se tiene que:

$$\begin{aligned} a &= L - \frac{mL}{n} \\ b &= \frac{L}{m} - \frac{L}{n} \\ x &= \frac{L}{n} \\ y &= \frac{mL}{n} \end{aligned}$$

Por lo tanto, la función hiperbólica para el valor del módulo de reacción  $k$  se expresa así:

$$\begin{aligned} k &= \frac{\frac{L}{m} - \frac{L}{n} \sqrt{-\left(L - \frac{mL}{n}\right)^2 + \left(\frac{mL}{n}\right)^2 - 2\left(\frac{mL}{n}\right)z + z^2}}{L - \frac{mL}{n}} + \frac{L}{n} \\ k &= \frac{\sqrt{m^2 n^3 (L - z)(2Lm - Ln - nz)} + Lm^2 n}{m^2 n^2} \end{aligned} \quad (3.29)$$

### 3.2.3. Análisis de la tablestaca en el proceso de excavación

De acuerdo con lo planteado en la discretización del dominio, el proceso de excavación se realiza por medio de etapas sucesivas que son representadas al eliminar las bandas que corresponden al frente de excavación. Por este motivo, cada una de estas etapas de excavación requieren de un análisis iterativo que involucra la descarga del suelo, la eliminación del resorte en el frente, el cálculo de las reacciones resultantes en cada nodo, la evaluación de la plastificación del suelo y la evaluación del cambio de la rigidez del suelo por efecto de la excavación.

#### Esfuerzos críticos

Para la definición de los esfuerzos críticos que actúan en la tablestaca se utilizaron los coeficientes de presión lateral de tierras según el planteamiento de Rankine (1857) presentado en el capítulo 2 en la sección 2.2.1.

Así mismo, ya que estos esfuerzos representan la condición límite que puede llegar a soportar el suelo, no pueden existir valores mayores a  $\sigma_p$  o menores que  $\sigma_a$  en la distribución de esfuerzos de una tablestaca. Dicho esto, en la Figura 3.13 se presenta un esquema del comportamiento no lineal de los esfuerzos a lo largo de la tablestaca, los cuales se diferencian de los asumidos en los métodos de diseño convencionales y en el utilizado para el predimensionamiento de la estructura.

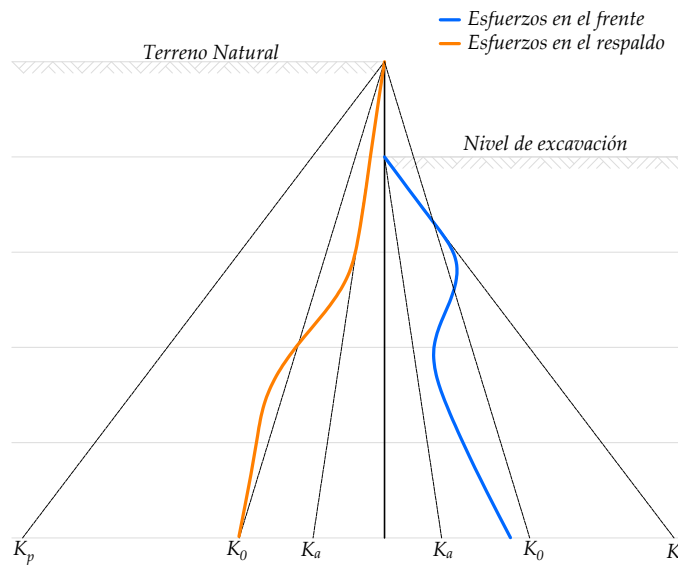


FIGURA 3.13: Distribución no lineal de esfuerzos en una tablestaca

Como se observa en la Figura 3.13, la distribución de esfuerzos críticos cambia en el frente de la excavación debido a que estos están en función de la profundidad efectiva del suelo. Sin embargo, la condición  $K_0$  en el frente no cambia ya que esta representa el estado inicial del suelo antes de ser perturbado por la excavación.

Teniendo en cuenta que los esfuerzos activos para suelos cohesivos involucran valores negativos hasta la profundidad de la grieta de tracción, se plantea la sustitución de el coeficiente  $K_a$  por un  $K_a^*$  hasta una profundidad de  $2z_c$  como se muestra en la Figura 3.14, con el fin de evaluar una condición mas real en la que esta grieta es rellenada por el mismo suelo que colapsa hacia la tablestaca y se deforma por efecto de la descarga.

De esta manera, al igualar el esfuerzo activo en el punto A para un suelo granular y cohesivo se puede calcular el valor del coeficiente  $K_a$  definido en la Ecuación 3.30.

$$\begin{aligned}
 2z_c\gamma K_a - 2c'\sqrt{K_a} &= 2z_c\gamma K_a^* \\
 K_a^* &= \frac{2z_c\gamma K_a - 2c'\sqrt{K_a}}{2z_c\gamma} \\
 K_a^* &= \frac{K_a}{2}
 \end{aligned}
 \tag{3.30}$$

### Proceso de descarga de la tablestaca

Al llevar a cabo la primera etapa de excavación, se genera una descarga en el frente de la tablestaca que corresponde a quitar el esfuerzo de la condición  $K_0$  hasta la profundidad

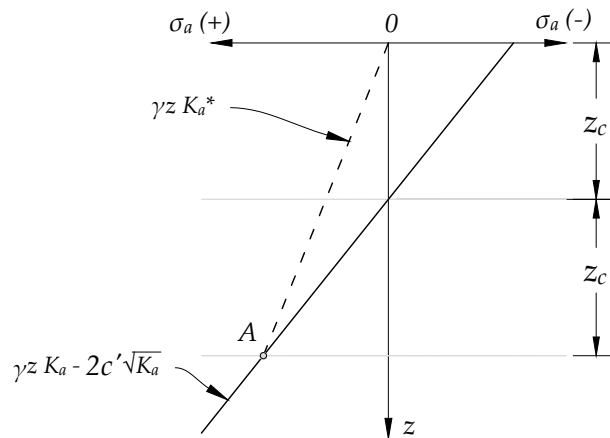
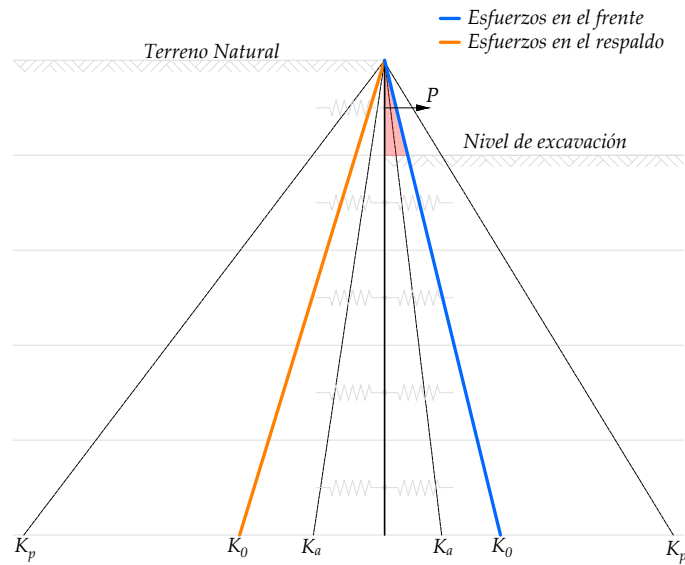


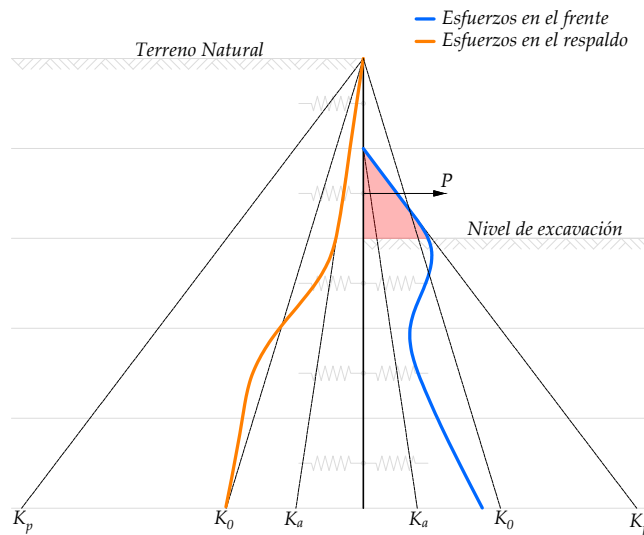
FIGURA 3.14: Corrección del esfuerzo activo

$\lambda_A$ . Esta descarga puede ser representada al aplicar una fuerza equivalente en el nodo de la banda  $A$  pero con dirección contraria, cuyo valor es igual al esfuerzo  $\sigma_{h0}$  por el valor del área del diagrama de empujes respectivo como se muestra en la Figura 3.15a. Adicionalmente, ya que el suelo fue retirado del frente de la tablestaca, se elimina también el resorte del frente y se deja únicamente el resorte del respaldo.

Así mismo, para las siguientes etapas de excavación se debe realizar este mismo procedimiento con la única diferencia que ya no se parte de una condición inicial  $K_0$ , sino que se toma el esfuerzo resultante de la etapa anterior en el frente de la tablestaca para realizar la descarga. Este procedimiento se ilustra gráficamente en la Figura 3.15b.



(A) Primera etapa de descarga



(B) Otras etapas

FIGURA 3.15: Análisis de las etapas de descarga

Debido a que el proceso de descarga involucra retirar el suelo del frente de la tablestaca hasta una profundidad definida, los esfuerzos críticos activos y pasivos se movilizan iniciando desde la nueva profundidad de excavación como en el caso presentado en la Figura 3.15b. Esta variación de esfuerzos en el frente de la estructura permite analizar una condición cada vez más crítica en la cual se genera la plastificación de la estructura y un aumento significativo en las deformaciones.

### 3.2.4. Cálculo de las reacciones nodales a partir de EMIH

Con el fin de evaluar el comportamiento del suelo y la tablestaca al someterlos a un proceso de descarga, se utiliza la metodología de Zeevaert (1980) para determinar la solución de las reacciones en cada elemento nodal bajo una condición de libre giro en ambos extremos de la tablestaca.

Al tener en cuenta que los valores de los módulos de rigidez de los resortes ( $k_j$ ) son conocidos y definidos como funciones conocidas de la profundidad, se plantean a continuación los análisis de la ecuación matricial de interacción horizontal EMIH.

**Condición  $X_i = 0$**

Se realiza el equilibrio de fuerzas y momentos se encuentran las reacciones  $R_{a0}$  y  $R_{b0}$  para la condición  $X_i = 0$ .

$$R_{a0} = \frac{P(\lambda + h)}{h} \quad (3.31)$$

$$R_{b0} = \frac{P\lambda}{h} \quad (3.32)$$

**a) Por la deflexión de la tablestaca debido a la descarga horizontal P aplicada en la banda excavada**

Se define la ecuación de momento mediante funciones de discontinuidad se integran dos veces se obtiene la ecuación de deflexión de la tablestaca con sus respectivas constantes de integración.

$$\begin{aligned} M(x) &= EI \frac{d^2 \Delta_{io}^P}{dx^2} = -R_{b0} \langle x \rangle^1 + R_{a0} \langle x - h \rangle^1 - P \langle x - (h + \lambda) \rangle^1 \\ &= EI \frac{d^2 \Delta_{io}^P}{dx^2} = -\frac{P\lambda}{h} \langle x \rangle^1 + \frac{P(\lambda + h)}{h} \langle x - h \rangle^1 - P \langle x - (h + \lambda) \rangle^1 \\ \theta(x) &= EI \frac{d \Delta_{io}^P}{dx} = -\frac{P\lambda}{2h} \langle x \rangle^2 + \frac{P(\lambda + h)}{2h} \langle x - h \rangle^2 - \frac{P}{2} \langle x - (h + \lambda) \rangle^2 + C_1 \\ EI \Delta_{io}^P &= -\frac{P\lambda}{6h} \langle x \rangle^3 + \frac{P(\lambda + h)}{6h} \langle x - h \rangle^3 - \frac{P}{6} \langle x - (h + \lambda) \rangle^3 + C_1 x + C_2 \end{aligned} \quad (3.33)$$

Al analizar las condiciones de frontera en la Ecuación 3.33 para los apoyos de la viga se obtiene:

Si  $x = 0$ ;  $\Delta_{io}^P = 0$ ; entonces,

$$C_2 = 0 \quad (3.34)$$

Si  $x = h$ ;  $\Delta_{io}^P = 0$ ; entonces,

$$C_1 = \frac{P\lambda h}{6} \quad (3.35)$$

Luego, se reemplazan las ecuaciones 3.34 y 3.35 en la Ecuación 3.33 se obtiene la ecuación de deflexión para la condición  $X_i = 0$ .

$$EI\Delta_{io}^P = -\frac{P\lambda}{6h}\langle x \rangle^3 + \frac{P(\lambda + h)}{6h}\langle x - h \rangle^3 - \frac{P}{6}\langle x - (h + \lambda) \rangle^3 + \frac{P\lambda h}{6}x \quad (3.36)$$

Si  $x = z_i$  y  $z_i \leq h$ :

$$EI\Delta_{io}^P = -\frac{P\lambda}{6}\left(z_i^3 - z_i h^2\right) \quad (3.37)$$

Si  $x = z_i$  y  $z_i > h$ :

$$EI\Delta_{io}^P = -\frac{P\lambda}{6h}\{h(h - z_i)(h - 3z_i)\} - \frac{P}{6}\{(z_i - h - \lambda)^3 - (z_i - h)^3\} \quad (3.38)$$

### b) Por desplazamiento de la tablestaca al ceder los apoyos $a$ y $b$

A partir del análisis de los desplazamientos que se producen en los apoyos  $a$  y  $b$  por efecto de la aplicación de la carga  $P$  se puede obtener la Ecuación 3.39 que es una expresión general de este comportamiento.

$$\Delta_{io}^A = \frac{R_{a0}}{k_a}\psi - \frac{R_{b0}}{k_b}\xi \quad (3.39)$$

Donde,

$R_{a0}$ : Reacción en el punto  $a$

$R_{b0}$ : Reacción en el punto  $b$

$k_a$ : Módulo de reacción en el punto  $a$

$k_b$ : Módulo de reacción en el punto  $b$

$\xi = \frac{h - z_i}{h}$ : Distancia desde  $b$  hasta el punto analizado

$\psi = 1 - \xi$ : Distancia desde  $a$  hasta el punto analizado

Al reemplazar las ecuaciones 3.31 y 3.32 se obtiene:

$$EI\Delta_{io}^A = \frac{(EI)P}{h^2} \left[ \frac{z_i(\lambda + h)}{k_a} - \frac{\lambda(h - z_i)}{k_b} \right] \quad (3.40)$$

Finalmente, a partir del análisis de  $\Delta_{io}^P$  y  $\Delta_{io}^A$  se puede obtener el desplazamiento horizontal en un punto  $i$  para la condición  $X_i = 0$ .

$$(EI)\Delta_{io} = (\Delta_{io}^P - \Delta_{io}^A)(EI) \quad (3.41)$$

**Condición  $X_i = 1$** 

Para el análisis de esta condición se evalúa la influencia de una carga unitaria aplicada en el nodo  $i$  sobre los resortes  $j$  que no hacen parte de la condición  $X_i = 0$ , por lo tanto, al realizar equilibrio de fuerzas y momentos se obtienen las reacciones  $R_{a1}$  y  $R_{b1}$  utilizadas para el calculo de las deformaciones unitarias. Sin embargo, debido a que el resorte del nodo  $a$  no se localiza en el extremo superior de la estructura sino que se encuentra en el primer nodo bajo el nivel de excavación para cada etapa de descarga, la reacción  $R_{b1}$  presenta valores diferentes cuando  $z_i < h$  y  $z_i > h$ .

$$R_{a1} = \frac{z_i}{h} \quad (3.42)$$

$$R_{b1} = \frac{z_i - h}{h}, z_i > h \quad (3.43)$$

$$R_{b1} = \frac{h - z_i}{h}, z_i < h \quad (3.44)$$

**a) Por la deflexión de la tablestaca por la aplicación de la carga unitaria**

Al determinar la ecuación de momento mediante funciones de discontinuidad al integrar dos veces se obtiene la ecuación de deflexión de la tablestaca con sus respectivas constantes de integración en función de la profundidad de aplicación de la carga.

- Cuando  $z_i > h$

$$\begin{aligned} M(x) &= EI \frac{d^2 \bar{S}'_{ji}}{dx^2} = R_{b1} \langle x \rangle^1 - R_{a1} \langle x - h \rangle^1 + 1 \langle x - z_i \rangle^1 \\ &= EI \frac{d^2 \bar{S}'_{ji}}{dx^2} = \frac{z_i - h}{h} \langle x \rangle^1 - \frac{z_i}{h} \langle x - h \rangle^1 + \langle x - z_i \rangle^1 \\ \theta(x) &= EI \frac{d \bar{S}'_{ji}}{dx} = \frac{z_i - h}{2h} \langle x \rangle^2 - \frac{z_i}{2h} \langle x - h \rangle^2 + \frac{1}{2} \langle x - z_i \rangle^2 + C_1 \\ EI \bar{S}'_{ji} &= \frac{z_i - h}{6h} \langle x \rangle^3 - \frac{z_i}{6h} \langle x - h \rangle^3 + \frac{1}{6} \langle x - z_i \rangle^3 + C_1 x + C_2 \end{aligned} \quad (3.45)$$

Cuando se analizan las condiciones de frontera para la Ecuación 3.45 para los apoyos de la viga se obtiene:

Si  $x = 0$ ;  $EI \bar{S}'_{ji} = 0$ ; entonces,

$$C_2 = 0 \quad (3.46)$$

Si  $x = 0$ ;  $EI \bar{S}'_{ji} = h$ ; entonces,

$$C_1 = -\frac{h^2(z_i - h)}{6h} \quad (3.47)$$

Posteriormente se reemplazan las ecuaciones 3.46 y 3.47 en la Ecuación 3.45 se obtiene la Ecuación de deflexión para la condición  $X_i = 1$  cuando  $z_i > h$ .

$$EI\bar{S}'_{ji} = \frac{z_i - h}{6h}\langle x \rangle^3 - \frac{z_i}{6h}\langle x - h \rangle^3 + \frac{1}{6}\langle x - z_i \rangle^3 - \frac{h^2(z_i - h)}{6h}x \quad (3.48)$$

Si  $x = z_j$ ,  $z_j < z_i$  y  $z_j < h$ :

$$EI\bar{S}'_{ji} = \frac{z_i - h}{6h}z_j^3 - \frac{h^2(z_i - h)}{6h}z_j \quad (3.49)$$

Si  $x = z_j$ ,  $z_j < z_i$  y  $z_j > h$ :

$$EI\bar{S}'_{ji} = \frac{z_i - h}{6h}z_j^3 - \frac{z_i}{6h}(z_j - h)^3 - \frac{h^2(z_i - h)}{6h}z_j \quad (3.50)$$

Si  $x = z_j$ ,  $z_j > z_i$  y  $z_j < h$ :

$$EI\bar{S}'_{ji} = \frac{z_i - h}{6h}z_j^3 + \frac{1}{6}(z_j - z_i)^3 - \frac{h^2(z_i - h)}{6h}z_j \quad (3.51)$$

Si  $x = z_j$ ,  $z_j > z_i$  y  $z_j > h$ :

$$EI\bar{S}'_{ji} = \frac{z_i - h}{6h}z_j^3 - \frac{z_i}{6h}(z_j - h)^3 + \frac{1}{6}(z_j - z_i)^3 - \frac{h^2(z_i - h)}{6h}z_j \quad (3.52)$$

■ Cuando  $z_i < h$

$$\begin{aligned} M(x) &= EI \frac{d^2 \bar{S}'_{ji}}{dx^2} = -R_{b1} \langle x \rangle^1 - R_{a1} \langle x - h \rangle^1 + 1 \langle x - z_i \rangle^1 \\ &= EI \frac{d^2 \bar{S}'_{ji}}{dx^2} = -\frac{h - z_i}{h} \langle x \rangle^1 - \frac{z_i}{h} \langle x - h \rangle^1 + \langle x - z_i \rangle^1 \\ \theta(x) &= EI \frac{d \bar{S}'_{ji}}{dx} = -\frac{h - z_i}{2h} \langle x \rangle^2 - \frac{z_i}{2h} \langle x - h \rangle^2 + \frac{1}{2} \langle x - z_i \rangle^2 + C_1 \\ EI\bar{S}'_{ji} &= -\frac{h - z_i}{6h} \langle x \rangle^3 - \frac{z_i}{6h} \langle x - h \rangle^3 + \frac{1}{6} \langle x - z_i \rangle^3 + C_1 x + C_2 \end{aligned} \quad (3.53)$$

Se analizan las condiciones de frontera en la Ecuación 3.53 para los apoyos de la viga y obtiene:

Si  $x = 0$ ;  $EI\bar{S}'_{ji} = 0$ ; entonces,

$$C_2 = 0 \quad (3.54)$$

Si  $x = 0$ ;  $E\bar{S}'_{ji} = h$ ; entonces,

$$C_1 = \frac{z_i}{6h}(z_i^2 + 2h^2 - 3hz_i) \quad (3.55)$$

Al reemplazar las ecuaciones 3.54 y 3.55 en la Ecuación 3.53 se obtiene la ecuación de deflexión para la condición  $X_i = 1$  cuando  $z_i < h$ .

$$E\bar{S}'_{ji} = -\frac{h-z_i}{6h}\langle x \rangle^3 - \frac{z_i}{6h}\langle x-h \rangle^3 + \frac{1}{6}\langle x-z_i \rangle^3 + \frac{z_i}{6h}(z_i^2 + 2h^2 - 3hz_i)x \quad (3.56)$$

Si  $x = z_j$ ,  $z_j < z_i$  y  $z_j < h$ :

$$E\bar{S}'_{ji} = -\frac{h-z_i}{6h}z_j^3 + \frac{z_i}{6h}(z_i^2 + 2h^2 - 3hz_i)z_j \quad (3.57)$$

Si  $x = z_j$ ,  $z_j < z_i$  y  $z_j > h$ :

$$E\bar{S}'_{ji} = -\frac{h-z_i}{6h}z_j^3 - \frac{z_i}{6h}(z_j-h)^3 + \frac{z_i}{6h}(z_i^2 + 2h^2 - 3hz_i)z_j \quad (3.58)$$

Si  $x = z_j$ ,  $z_j > z_i$  y  $z_j < h$ :

$$E\bar{S}'_{ji} = -\frac{h-z_i}{6h}z_j^3 + \frac{1}{6}(z_j-z_i)^3 + \frac{z_i}{6h}(z_i^2 + 2h^2 - 3hz_i)z_j \quad (3.59)$$

Si  $x = z_j$ ,  $z_j > z_i$  y  $z_j > h$ :

$$E\bar{S}'_{ji} = -\frac{h-z_i}{6h}z_j^3 - \frac{z_i}{6h}(z_j-h)^3 + \frac{1}{6}(z_j-z_i)^3 + \frac{z_i}{6h}(z_i^2 + 2h^2 - 3hz_i)z_j \quad (3.60)$$

### b) Por el giro de la tablestaca al ceder los apoyos $a$ y $b$

Al tener en cuenta los desplazamientos que se producen los resortes de los puntos  $j$  por la aplicación de la carga unitaria en  $i$ , se puede obtener una expresión similar al análisis de la condición  $X_i = 0$  como se muestra a continuación:

$$\bar{S}''_{ji} = \frac{R_{a1}}{k_a}\psi - \frac{R_{b1}}{k_b}\xi \quad (3.61)$$

Donde,

$R_{a1}$ : Reacción en el punto  $a$  por la carga unitaria aplicada en  $i$

$R_{b1}$ : Reacción en el punto  $b$  por la carga unitaria aplicada en  $i$

$k_a$ : Módulo de reacción en el punto  $a$

$k_b$ : Módulo de reacción en el punto  $b$

$\xi = \frac{h-z_j}{h}$  : Distancia desde  $b$  hasta el punto analizado  $z_j$

$\psi = 1 - \xi$ : Distancia desde  $a$  hasta el punto analizado  $z_j$

Se reemplazan las ecuaciones 3.42 y 3.44:

$$EI\bar{S}_{ji}'' = \frac{(EI)}{h^2} \left[ \frac{z_i z_j}{k_a} + \frac{(h - z_i)(h - z_j)}{k_b} \right] \quad (3.62)$$

### c) Por la deformación del suelo debido a la carga unitaria aplicada en el punto $i$

Debido a que la carga unitaria aplicada sobre un nodo implica la deformación del resorte que simula el comportamiento del suelo, a partir de la metodología de Winkler (1867) se define lo siguiente:

$$EI\bar{S}_{ij}''' = \frac{EI}{k_i} \quad (3.63)$$

Finalmente, se tiene en cuenta la definición de los desplazamientos  $\bar{S}_{ij}'$ ,  $\bar{S}_{ij}''$  y  $\bar{S}_{ij}'''$ , se puede obtener el desplazamiento horizontal unitario para la condición  $X_i = 1$ .

$$\bar{S}_{ij} = \bar{S}_{ij}' + \bar{S}_{ij}'' + \bar{S}_{ij}''' \quad (3.64)$$

Una vez determinados los desplazamientos horizontales  $\Delta_{i0}$  de la condición  $X_i = 0$  y los desplazamientos unitarios  $\bar{S}_{ji}$  de la condición  $X_i = 1$ , se aplica la ecuación matricial de interacción presentada en la Ecuación 3.65 para obtener las reacciones en cada nodo de la estructura y realizar la verificación de plasticidad según los esfuerzos críticos que se alcanzaron en cada etapa de descarga.

$$[\bar{S}_{ji}] \{X_i\} = \{\Delta_{i0}\} \quad (3.65)$$

Debido a que el proceso de cálculo se realiza de forma incremental, los valores finales de cargas, esfuerzos, cortantes, momentos y deformaciones para cada etapa de excavación se determinan mediante la suma de cada parámetro en la etapa anterior y el incremento calculado en la etapa de excavación actual como se presenta en las siguientes ecuaciones.

$$Q_i = Q_{i-1} + \Delta Q_i \quad (3.66)$$

$$q_i = q_{i-1} + \Delta q_i \quad (3.67)$$

$$V_i = V_{i-1} + \Delta V_i \quad (3.68)$$

$$M_i = M_{i-1} + \Delta M_i \quad (3.69)$$

$$\delta_i = \delta_{i-1} + \Delta \delta_i \quad (3.70)$$

### 3.2.5. Análisis de plastificación

Cuando se considera la redistribución de esfuerzos generados en la tablestaca por efecto de la excavación, es muy frecuente que los empujes tanto al frente como en el respaldo alcancen una condición crítica que producen la plastificación de algunos de los nodos evaluados.

Como los nodos bajo el nivel de excavación tienen dos resortes y sobre el nivel de excavación solo uno en el respaldo de la tablestaca, se pueden presentar diferentes casos de plastificación en los cuales se afecta cada resorte por separado, o ambos al mismo tiempo. Así mismo, dadas las características de los esfuerzos críticos a lo largo de la estructura, se presentan diferentes condiciones activas y pasivas al depender de la localización del nodo en profundidad que deben evaluarse por separado.

De acuerdo a lo anterior, los casos de plastificación se resumen de la siguiente manera:

- Plastificación del resorte del respaldo para condición activa

Este caso se presenta en los resortes del respaldo de la tablestaca sobre el nivel de excavación debido a la transición de los esfuerzos hacia la condición activa y a la eliminación del resorte del frente por efecto de la descarga. Ya que los esfuerzos necesarios para llegar a la condición activa son mucho menores que en la condición pasiva del frente de la estructura, se pueden plastificar nodos hasta profundidades mucho mayores que dependen de la rigidez del suelo. En la Figura 3.16 se pueden apreciar en color rojo los resortes plastificados bajo estas condiciones.

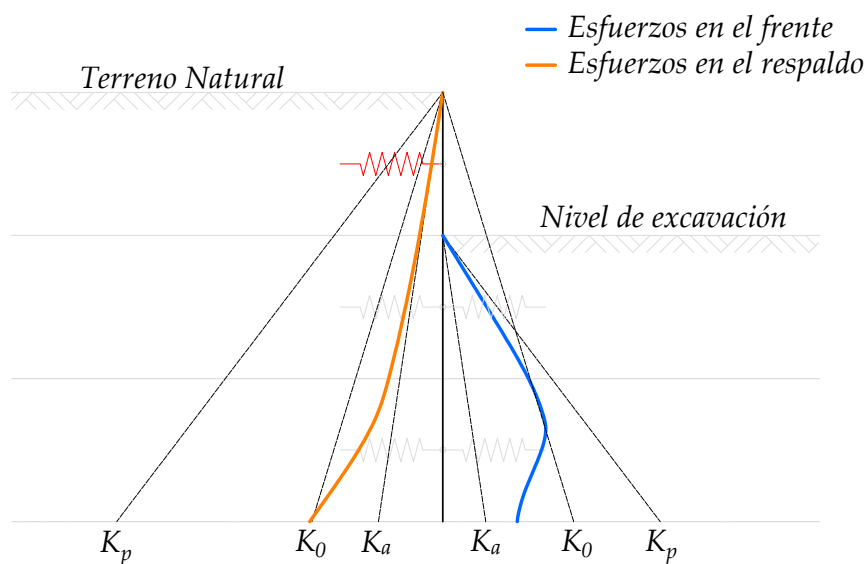


FIGURA 3.16: Plastificación del resorte del respaldo para condición activa

- Plastificación del resorte del frente para condición pasiva

Este caso se puede presentar cuando el resorte del frente de la tablestaca inmediatamente bajo el nivel de excavación se plastifica para una condición pasiva debido a la movilización de los esfuerzos críticos hacia el frente por efecto de la descarga. En la Figura 3.17 se puede apreciar la distribución de esfuerzos en la tablestaca y se señala en color rojo el resorte plastificado.

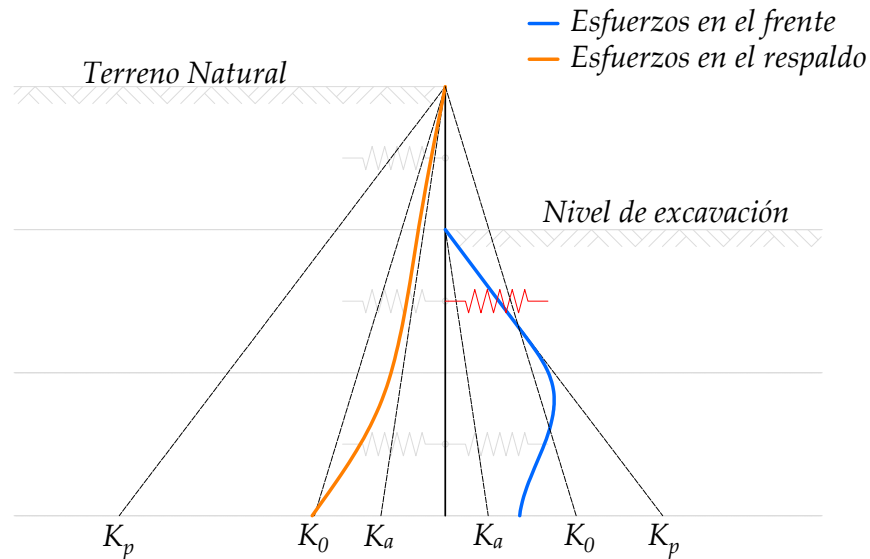


FIGURA 3.17: Plastificación del resorte del frente para condición pasiva

- Plastificación del resorte del frente para condición pasiva y el resorte del respaldo en condición activa

Este caso se presenta cuando el nodo del frente y el respaldo de la tablestaca se plastifican simultáneamente bajo una condición pasiva y activa respectivamente. Ya que sobre el nivel de excavación solamente existen resortes en el respaldo, este caso se presenta únicamente en los resortes bajo el nivel de excavación. En la Figura 3.18 se presenta de manera gráfica la distribución de esfuerzos que genera esta condición y se resalta en color rojo los resortes plastificados.

- Plastificación del resorte del frente por condición activa cuando el nodo evaluado está muy por debajo del nivel de excavación

A diferencia de los casos anteriores que se presentan en los nodos superiores o los adyacentes al nivel de excavación, la plastificación de los resortes del frente para una condición activa se puede presentar en los nodos inferiores cuando casi la totalidad

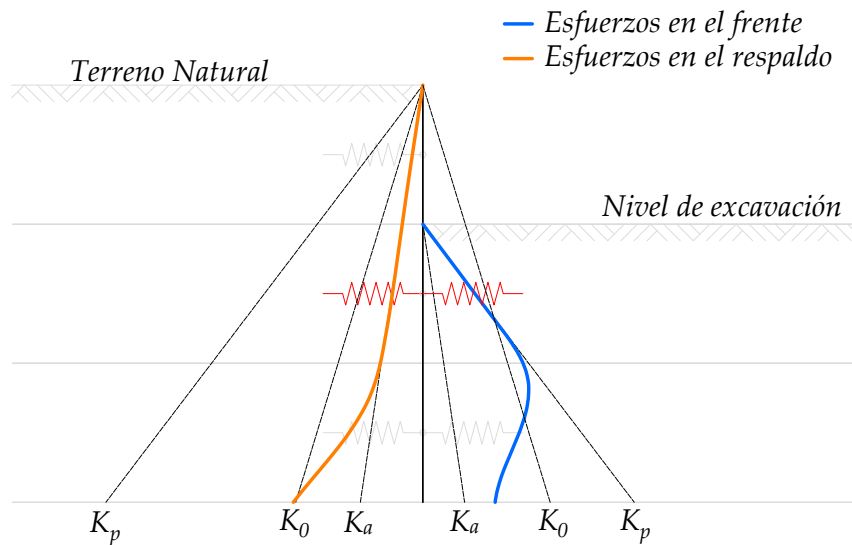


FIGURA 3.18: Plastificación del resorte del frente para condición pasiva y el resorte del respaldo en condición activa

de los nodos ya se ha plastificado, esto puede observarse en la Figura 3.19. Aunque este caso representa una condición inestable de la tablestaca, se tuvo en cuenta en los análisis con el fin de evaluar la totalidad de nodos que se pueden plastificar antes de considerar la falla de la estructura.

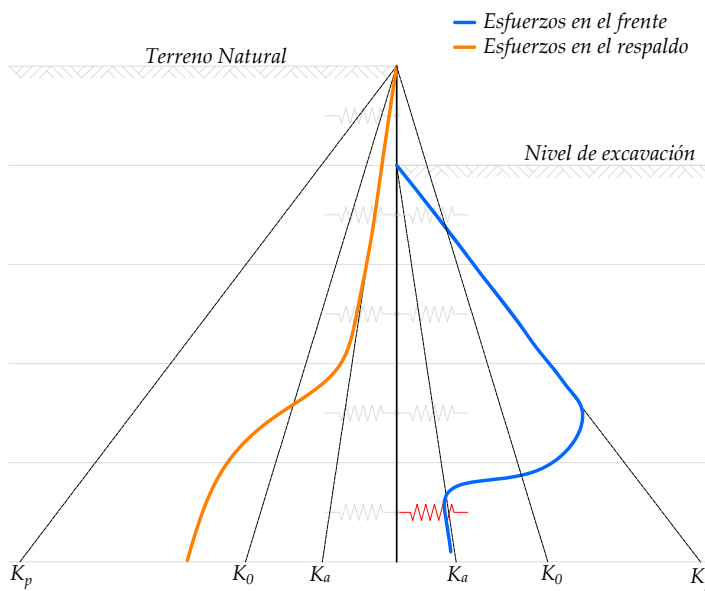


FIGURA 3.19: Plastificación del resorte del frente por condición activa

Para considerar la plastificación de los resortes en la metodología de interacción suelo estructura, se analiza la diferencia ente los esfuerzos calculados en cada etapa de excavación y los esfuerzos límites activos y pasivos según el caso de plastificación que se presenta. De esta manera, ya que el proceso de cálculo se realiza de manera incremental, la definición del esfuerzo de plastificación es la siguiente:

$$(q_{ep})_i = (q)_{i-1} - (q_{crit})_i \quad (3.71)$$

Donde,

$(q_{ep})_i$ : Esfuerzo de plastificación en la etapa actual de descarga  $i$ .

$(q)_{i-1}$ : Esfuerzo final obtenido en la etapa de descarga anterior  $i - 1$ .

$(q_{crit})_i$ : Esfuerzo crítico activo o pasivo en la etapa actual de descarga  $i$

Ya que cada resorte puede plastificarse de forma independiente, el cálculo de este esfuerzo se realiza directamente para los resortes individuales. Sin embargo, cuando se plastifica el resorte del respaldo y del frente simultáneamente se debe calcular un esfuerzo neto para todo el nodo.

Zeevaert (1980) plantea que este esfuerzo de plastificación  $q_{ep}$  puede representarse en la metodología de interacción suelo estructura mediante la aplicación de una carga estructural fija igual a  $R_{ep} = \lambda q_{ep}$  en el nodo involucrado y la eliminación del resorte en el nodo plastificado. De esta manera, para el análisis de EMIH en la condición  $X_i = 0$  se sigue tomando el apoyo del nodo  $b$  que corresponde al punto más profundo, pero el nodo  $a$  localizado bajo el nivel de excavación se va profundizando hasta el siguiente nodo mientras continúe la plastificación de la estructura. La definición de esta condición de carga puede apreciarse de manera gráfica en la Figura 3.20.

### 3.2.6. Reducción de la rigidez del suelo en el frente de la tablestaca

A partir de la variación del módulo de reacción en profundidad por efecto de la excavación, se plantea a continuación una metodología para la evaluación de este efecto en el aumento de las deformaciones mediante la aplicación de una carga estructural equivalente en los nodos excavados.

Para simular la condición de excavación de la tablestaca se quitan resortes en el frente mientras se aplica en su lugar una fuerza equivalente de descarga, y debido a la disminución de los esfuerzos verticales en el frente, los módulos de los resortes del frente se reducen. Por consiguiente, al terminar una etapa de excavación los estados de esfuerzos de una unidad de tablestaca se definen como  $(f_j)_i$ . En la Figura 3.21 se presenta esquemáticamente las variables utilizadas para la definición de los esfuerzos finales en una unidad de tablestaca.

De acuerdo con la Figura 3.21, el esfuerzo resultante en la tablestaca debe ser:

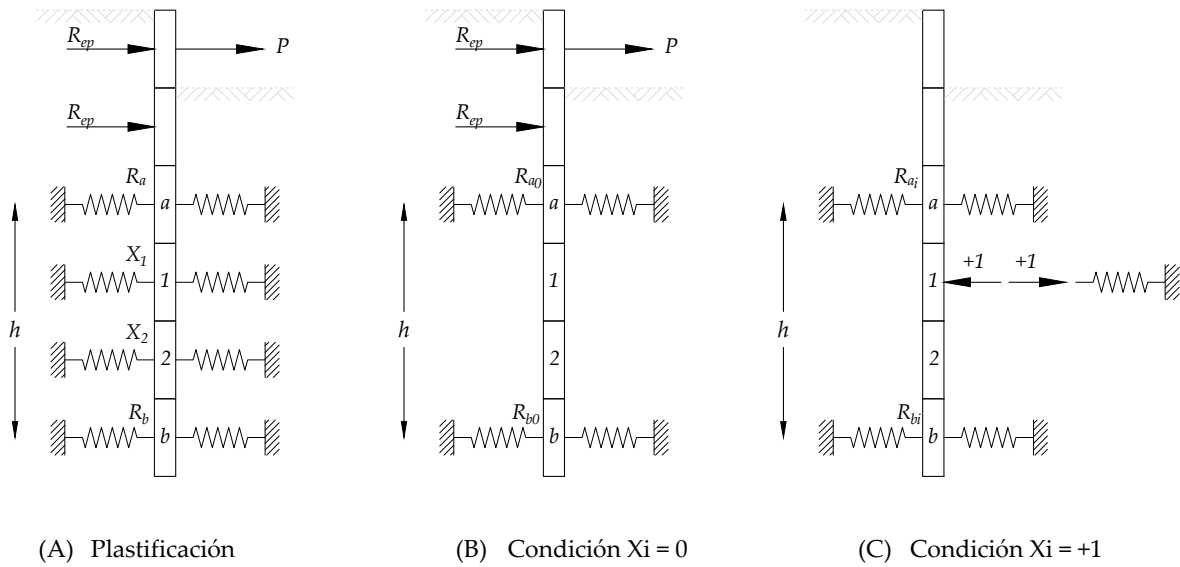


FIGURA 3.20: Modificación de EMIH por la plastificación de nodos.

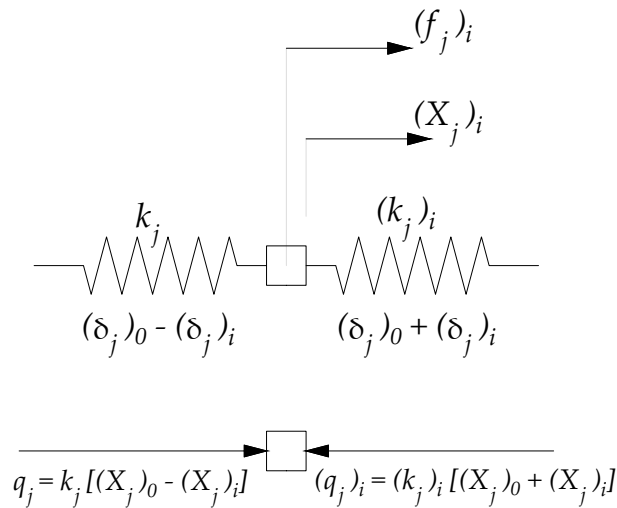


FIGURA 3.21: Condición corriente de esfuerzos para el análisis de pérdida de rigidez.

$$(f_j)_i = (k_j)_i [(\delta_j)_0 + (\delta_j)_i] - k_j [(\delta_j)_0 - (\delta_j)_i]$$

$$(f_j)_i = [(k_j)_i - k_j] (\delta_j)_0 + [(k_j)_i + k_j] (\delta_j)_i$$

Se reemplaza el valor de  $(\delta_j)_0$  según la Ecuación 3.25,

$$(f_j)_i = [(k_j)_i - k_j] \frac{\gamma' z_j K_0}{k_j} + [(k_j)_i + k_j] (\delta_j)_i \quad (3.72)$$

Luego de alcanzar el estado anterior, se procede a profundizar la excavación y se generan así los siguientes efectos:

1. Retirar los resortes comprometidos y aplicar allí unas cargas equivalentes a los que había antes pero en sentido contrario.
2. Disminuir el módulo de rigidez del resorte del frente de la excavación. Al igual que en el caso anterior, ese efecto se reemplaza por cambiar el módulo de  $(k_j)_i$  a  $(k_j)_{i+1}$ , se reduce y aplica un incremento de carga en el nodo también hacia adelante de la tablestaca.

De esta manera, el incremento de esfuerzo debido al cambio de la rigidez de los resortes por efecto de la excavación es el siguiente:

$$\Delta f_j = [(k_j)_i - (k_j)_{i+1}] [(\delta_j)_0 - (\delta_j)_i] \quad (3.73)$$

Por otra parte, al reducir la rigidez al frente de la tablestaca se genera un desplazamiento hacia el frente que es equivalente a aplicar un incremento de fuerza en el elemento considerado. Este incremento será:

$$\Delta F_j = \frac{\lambda}{8} (\Delta f_{j-1} + 6\Delta f_j + \Delta f_{j+1}) \quad (3.74)$$

Así pues, el nuevo estado luego de profundizar la excavación, se determina al aplicar una carga equivalente por descarga en los nodos de la excavación y se retiran los resortes respectivos. Adicionalmente, se deben disminuir las rigideces de los resortes en el frente de la tablestaca y aplican en los nodos respectivos el incremento de carga  $\Delta F_j$  como una carga estructural fija de la misma manera que en el análisis de plastificación.

### 3.3. Algoritmo de diseño

Dado que los planteamientos presentados tienen un componente matricial e iterativo muy fuerte, se requiere la aplicación de técnicas numéricas robustas que deben ser desarrolladas mediante la programación de un algoritmo de diseño específico. Por lo tanto, para dar solución al problema se realizó la programación de una herramienta de cálculo en el lenguaje Python 3 que fue seleccionado por tener una licencia libre, por su facilidad en el manejo de programación orientada a objetos y la aplicación de interfaces gráficas mediante la librería PyQt5.

Con el fin de facilitar la programación del algoritmo de diseño, se dividió el código en tres partes con las siguientes características:

- Una agrupación de todos los procedimientos de la metodología de interacción suelo estructura denominado *SP\_SEI.py* por sus siglas en inglés *Sheet Pile - Soil Structure Interaction*.
- La definición de la interfaz gráfica en PyQt5 utilizando la herramienta QtDesigner en un programa denominado *appSPSEI\_ui.py*.
- Un código que enlaza la interfaz gráfica con la metodología de interacción suelo estructura denominado *main.py*.

De esta manera, los parámetros de entrada necesarios para la ejecución del algoritmo son los siguientes:

1. Geometría y propiedades de la tablestaca
  - a) Longitud total de la tablestaca
  - b) Módulo de elasticidad
  - c) Inercia o módulo de sección de la tablestaca
2. Propiedades del suelo
  - a) Ángulo de fricción
  - b) Cohesión
  - c) Peso unitario
  - d) Factor de rigidez inicial de los resortes
3. Parámetros del modelo
  - a) Profundidad de excavación
  - b) Número de etapas de excavación
  - c) Número total de nodos a evaluar
  - d) Espesor de las bandas

Una vez definidos los parámetros iniciales se realiza el proceso de cálculo iterativo mostrado en la Figura 3.22.

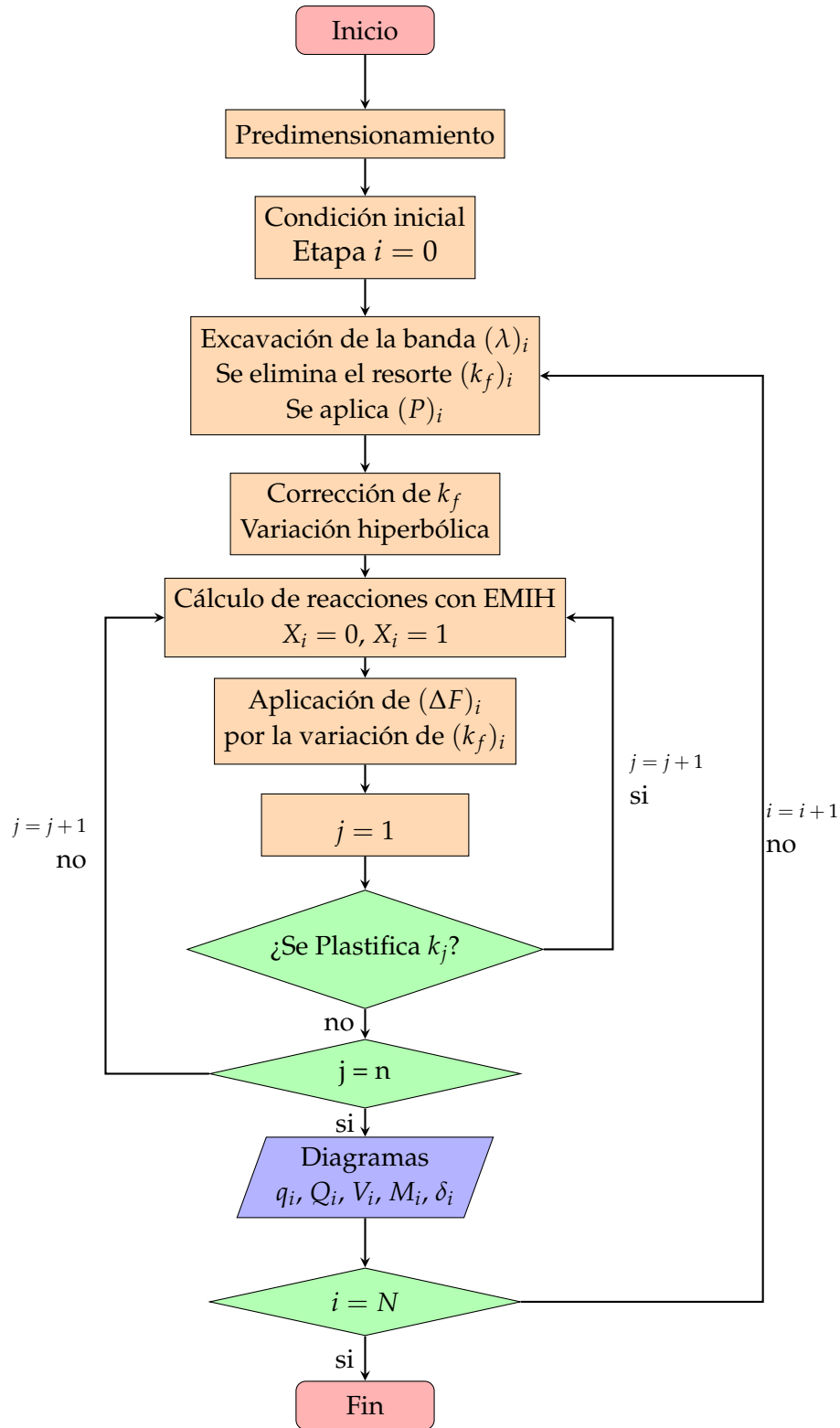


FIGURA 3.22: Diagrama de flujo del algoritmo de diseño

Así mismo, la secuencia de cálculo del algoritmo de diseño se puede resumir de la siguiente manera:

1. Se realiza el predimensionamiento de la estructura para definir la longitud de empotramiento, el módulo de elasticidad y el módulo de inercia a partir de los diagramas de cortante y momento.
2. Se definen las condiciones iniciales del modelo como los esfuerzos críticos, los módulos de reacción del suelo y su respectiva variación por efecto de la descarga, el número de nodos, número de etapas de excavación, espesor de las bandas, la profundidad de la excavación y las propiedades del suelo.
3. Se realiza la excavación de la banda  $(\lambda)_i$  mediante la eliminación del resorte del frente y la aplicación de la descarga  $P_i$ .
4. Se realiza la corrección de los módulos de reacción del frente por efecto de la excavación mediante el planteamiento hiperbólico.
5. Se calculan las reacciones con EMIH.
6. Se aplican las cargas estructurales  $(\Delta F)_i$  por la disminución de la rigidez de los resortes en el frente.
7. Se evalúa si hay plastificación de alguno de los nodos.
8. Si hay plastificación se vuelve al paso 5, de lo contrario se continua con el procedimiento.
9. Se obtienen los diagramas de esfuerzo, carga, cortante, momento y desplazamiento para la etapa de excavación  $i$ .
10. Se suman los incrementos de la etapa a los totales acumulados y se continua con la siguiente etapa de descarga volviendo al paso 3.

## Capítulo 4

# Aplicación del algoritmo de diseño

A partir del algoritmo de diseño definido anteriormente, en este capítulo se presentan algunos ejemplos de la metodología de ISE para tablestacas sometidas a un proceso de excavación. Debido a la importancia del cambio de la rigidez del suelo en el frente de la estructura y la presencia de cohesión en el suelo para la definición de los esfuerzos críticos, se evaluaron cuatro condiciones diferentes para establecer la influencia de estos parámetros en la respuesta de la tablestaca. De esta manera, las condiciones evaluadas son las siguientes:

- Condición 1: Suelo puramente friccionante ( $\phi' \neq 0, c' = 0$ ) sin cambio en la rigidez de los resortes del frente de la tablestaca.
- Condición 2: Suelo con cohesión ( $\phi' \neq 0, c' \neq 0$ ) sin cambio en la rigidez de los resortes del frente de la tablestaca.
- Condición 3: Suelo puramente friccionante ( $\phi' \neq 0, c' = 0$ ) con cambio en la rigidez de los resortes del frente de la tablestaca.
- Condición 4: Suelo con cohesión ( $\phi' \neq 0, c' \neq 0$ ) con cambio en la rigidez de los resortes del frente de la tablestaca.

### 4.1. Parámetros de entrada

Los parámetros del suelo definidos para la aplicación del algoritmo de diseño son los siguientes:

#### 1. Propiedades del suelo

- a) Ángulo de fricción:  $28^\circ$
- b) Cohesión:
  - 1) Condición 1 y 3  $c' = 0 \text{ kPa}$
  - 2) Condición 2 y 4  $c' = 10 \text{ kPa}$
- c) Peso unitario:  $14 \text{ kN/m}^3$
- d) Factor de rigidez inicial de los resortes:  $8000 \text{ kPa/m}$

Para realizar los análisis de las cuatro condiciones mencionadas, se utilizó el programa SP\_SEI y la interfaz gráfica para predimensionar la estructura definiendo la geometría y las propiedades iniciales de la tablestaca para una profundidad de excavación de 3.0 m . En la Figura 4.1 se puede observar una captura de pantalla de la interfaz gráfica en donde se ingresan los parámetros del suelo para realizar el predimensionamiento y así obtener la longitud de empotramiento mínima que será utilizada en el análisis de ISE.

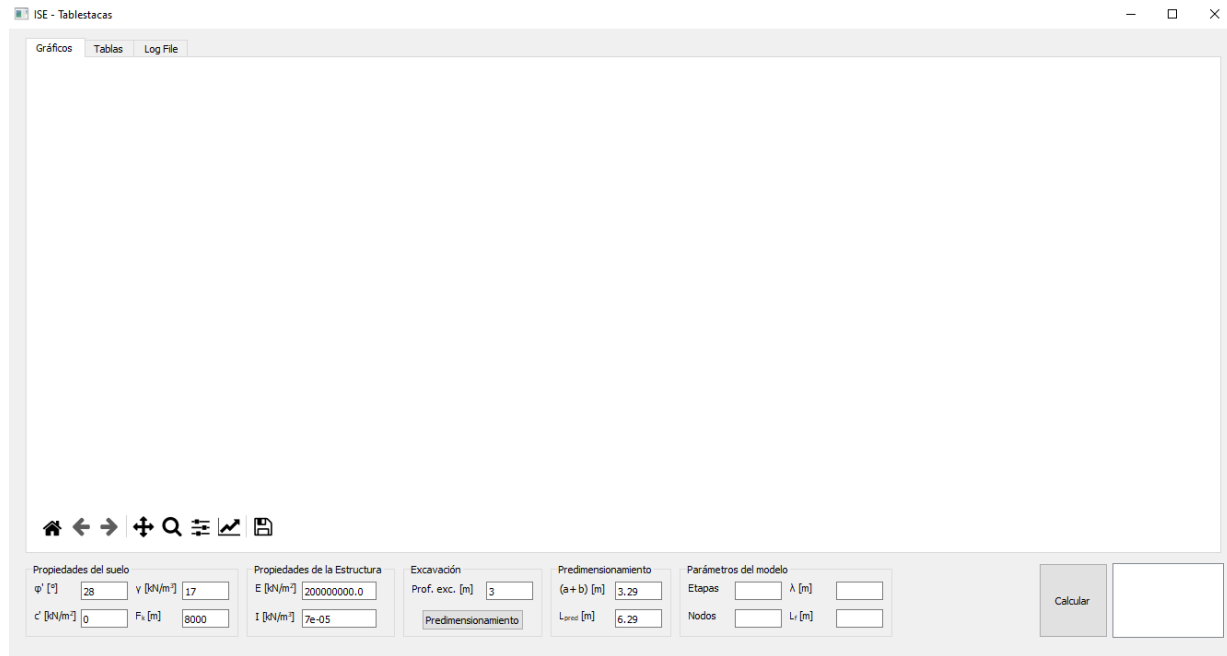


FIGURA 4.1: Predimensionamiento - Interfaz gráfica de diseño

De esta manera, según el predimensionamiento realizado para diferentes parámetros del suelo, se obtiene un valor de empotramiento de 3.29 m. Por lo tanto, para manejar bandas con anchos iguales y facilitar el cálculo del algoritmo se definen los siguientes parámetros del modelo:

### 1. Geometría y propiedades de la tablestaca

- a) Longitud total de la tablestaca: 7.0 m
- b) Módulo de elasticidad: 200 MPa para una tablestaca metálica
- c) Inercia o módulo de sección de la tablestaca:  $70e-6 \text{ m}^4 / \text{m}$

### 2. Parámetros del modelo

- a) Profundidad de excavación: 3.0 m
- b) Número de etapas de excavación: 6 etapas
- c) Número total de nodos a evaluar: 14 nodos
- d) Espesor de las bandas: 0.5 m

## 4.2. Resultados

Una vez definidos los parámetros del modelo se utiliza el algoritmo de diseño para obtener los diagramas de esfuerzos, cortantes, momentos y desplazamientos para cada una de las seis etapas de excavación. Todos estos resultados fueron graficados mediante la interfaz del programa SP\_SEI como se muestra en la Figura 4.2.

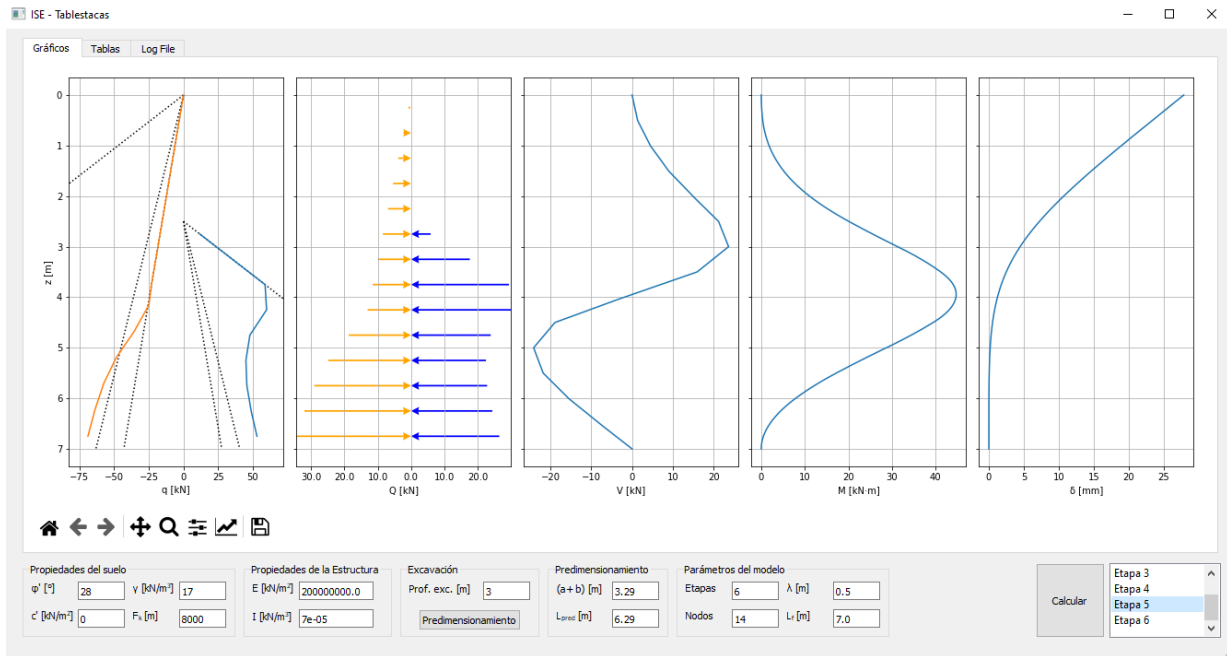


FIGURA 4.2: Pestaña de diagramas - Interfaz gráfica de diseño

Desde la Figura 4.3 hasta la Figura 4.8, se presenta gráficamente la comparación de las cuatro condiciones de análisis.

A partir de los resultados obtenidos para cada una de las cuatro condiciones, se observa un comportamiento coherente y acorde con la respuesta esperada para la distribución de empujes, cortantes, momentos y deformaciones de la tablestaca. En general, los diagramas de cortantes presentan un valor máximo localizado a la misma profundidad de la etapa de excavación y un valor mínimo a una profundidad mayor que ayuda a equilibrar el cambio de esfuerzos que se produce en cada descarga. En el caso de los diagramas de momentos se evidencia que sus valores máximos también cambian con respecto al nivel de excavación pero se profundizan cada vez más para condiciones más desfavorables como es el caso de suelos sin cohesión y con cambio de rigidez por efecto de la excavación.

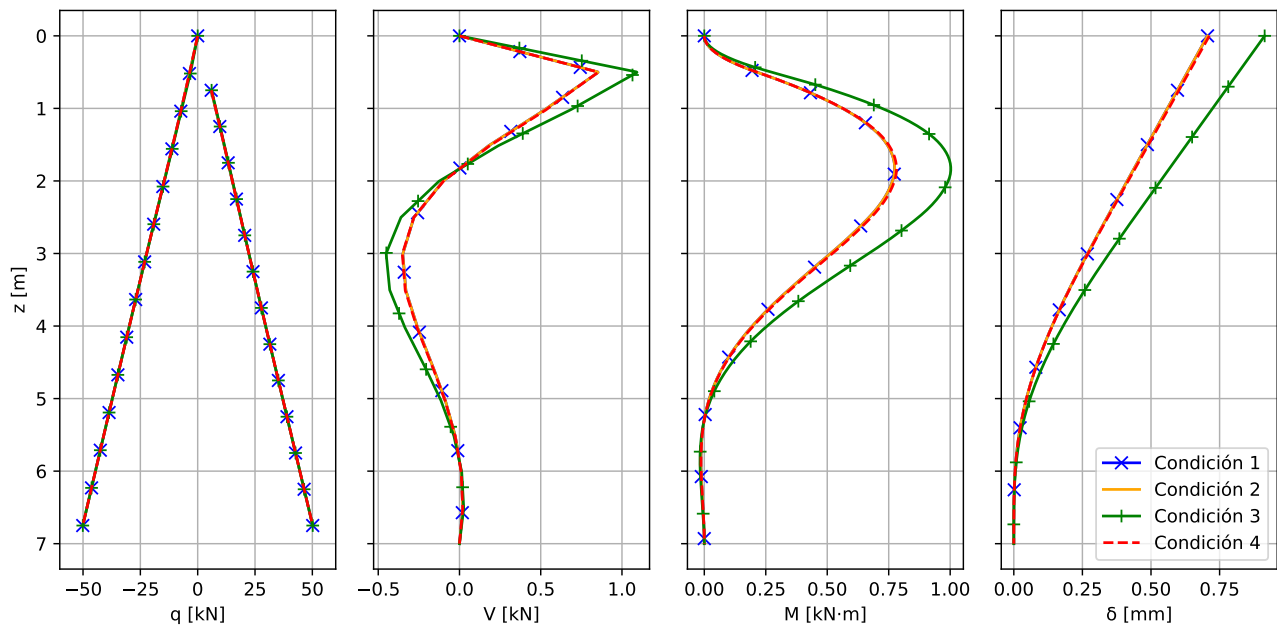


FIGURA 4.3: Diagramas para la etapa de excavación No. 1

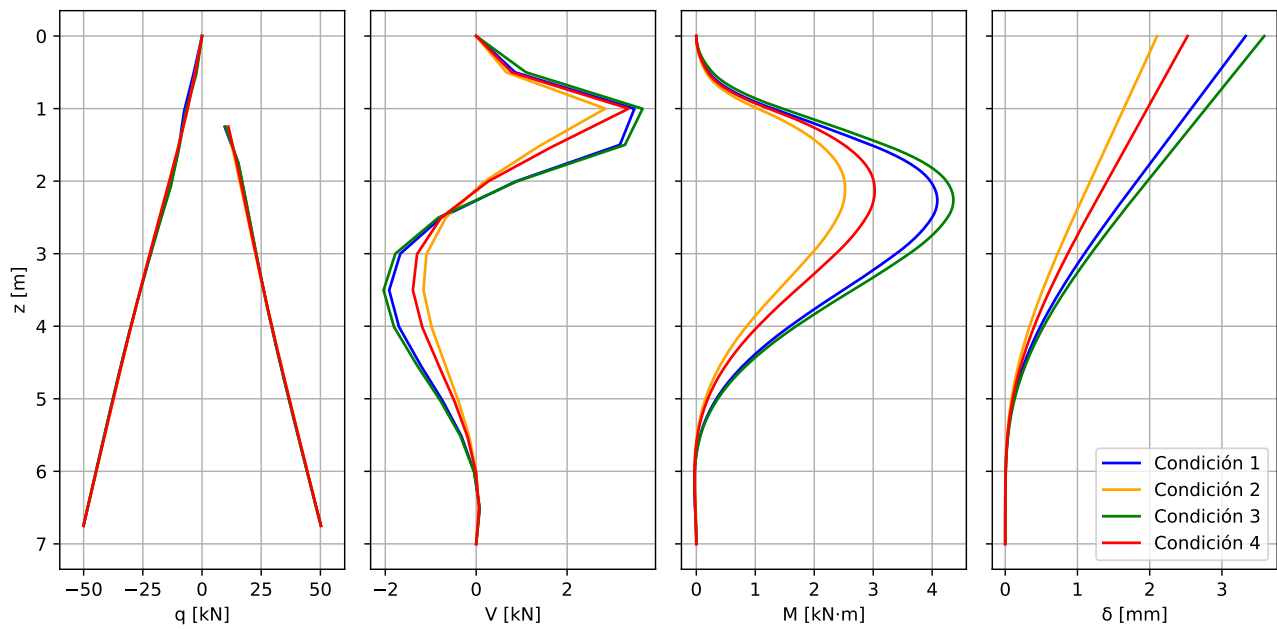


FIGURA 4.4: Diagramas para la etapa de excavación No. 2

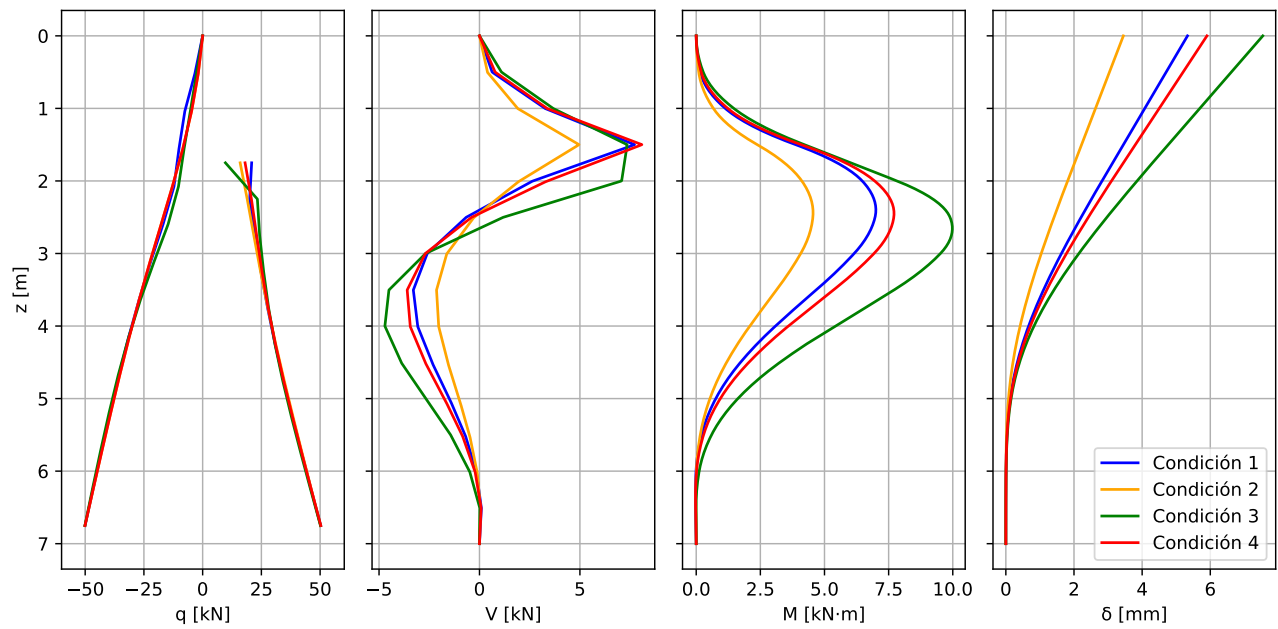


FIGURA 4.5: Diagramas para la etapa de excavación No. 3

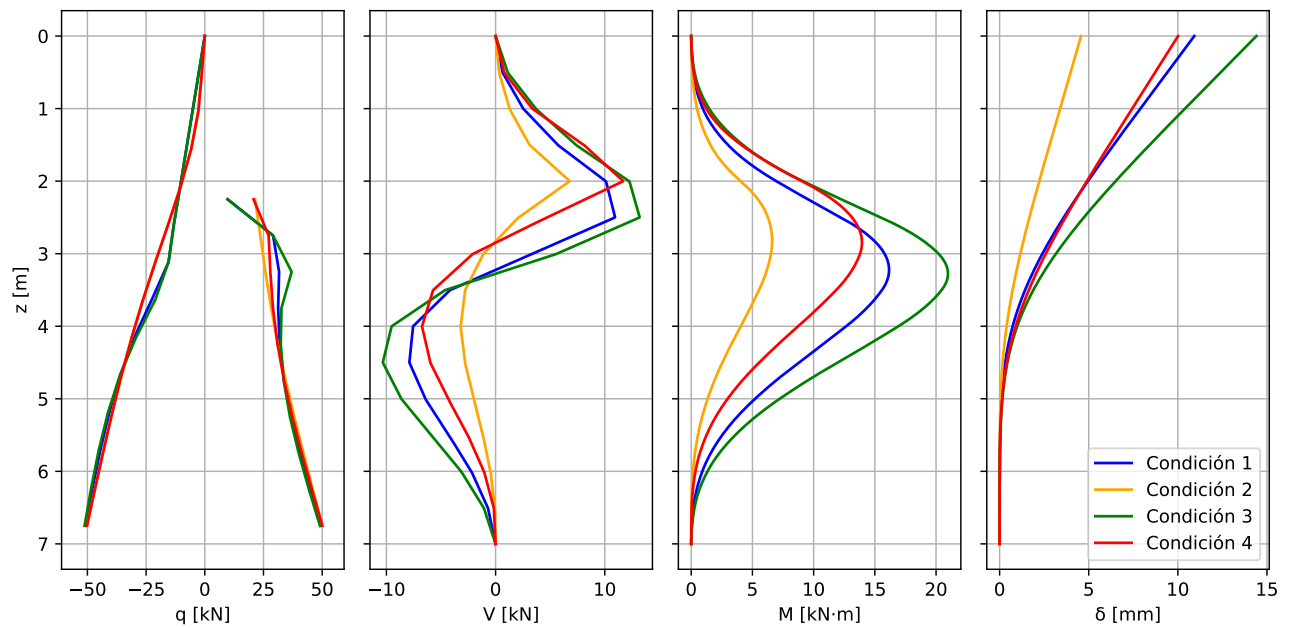


FIGURA 4.6: Diagramas para la etapa de excavación No. 4

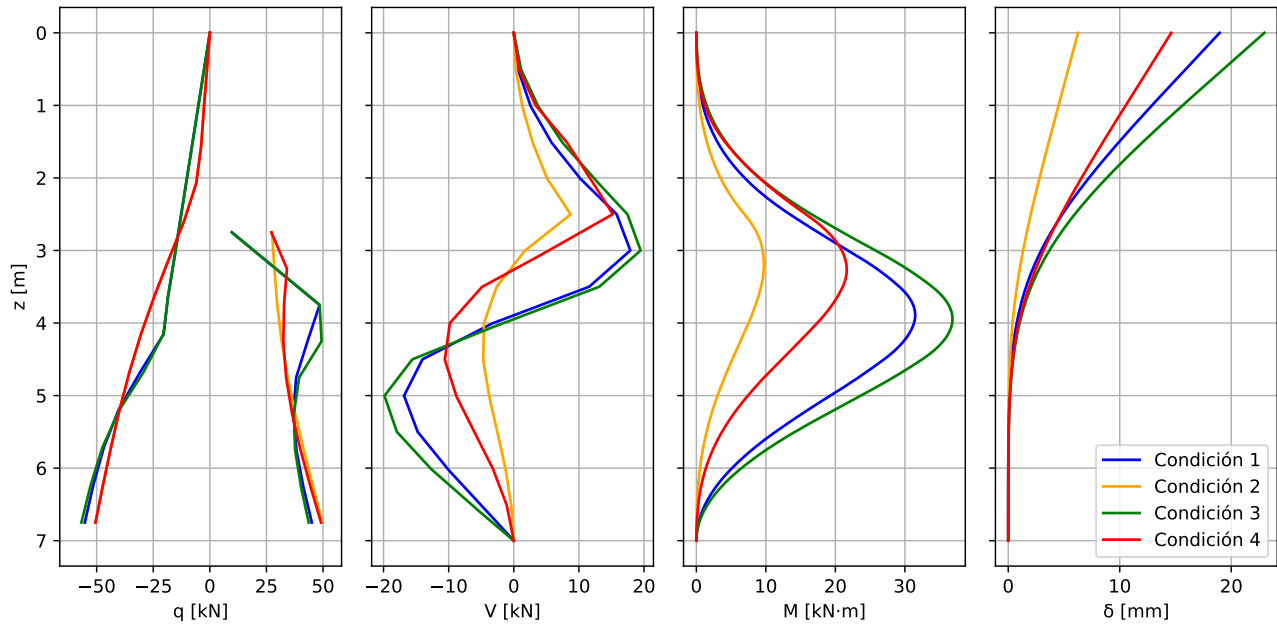


FIGURA 4.7: Diagramas para la etapa de excavación No. 5

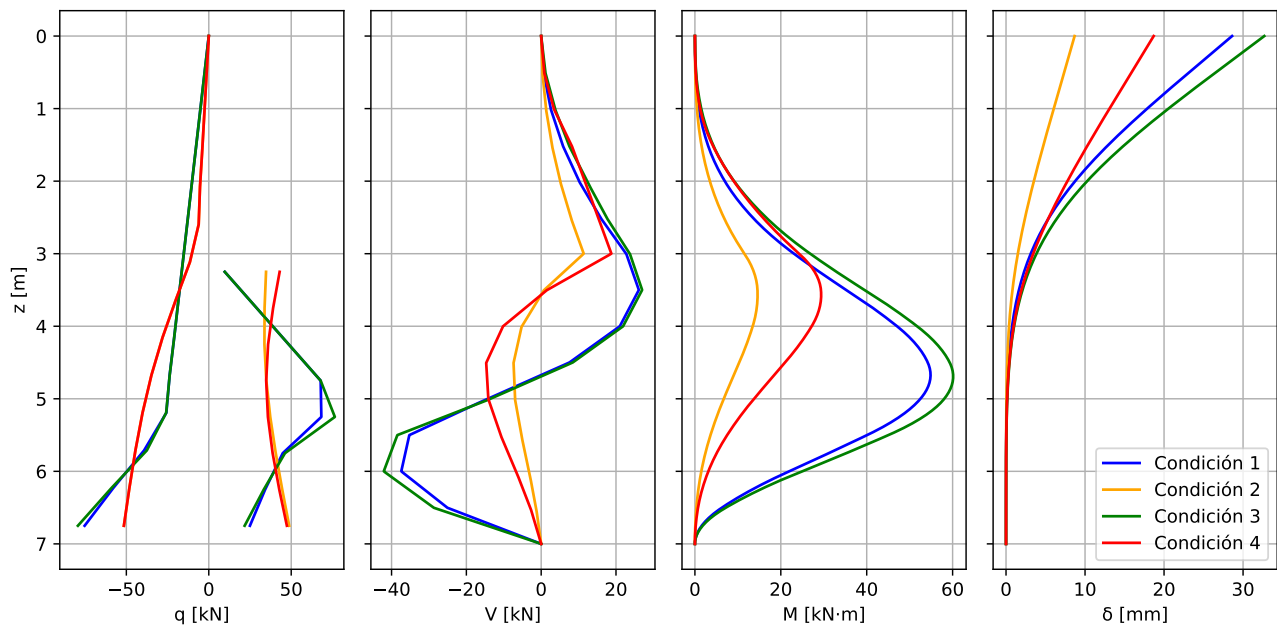


FIGURA 4.8: Diagramas para la etapa de excavación No. 6

Así mismo, como se observa en la Figura 4.3 los resultados obtenidos para la primera etapa de descarga muestran un comportamiento similar entre las condiciones 1, 2 y 4 (línea azul, naranja y roja respectivamente) ya que las dos primeras no tienen en cuenta el cambio de rigidez de los resortes de la tablestaca, y la cuarta condición al tener cohesión presenta una mayor resistencia que reduce el efecto del cambio en la rigidez de los resortes del frente. Sin embargo, a medida que se continúa con la excavación como en las figuras 4.4 a 4.8, se observa que este cambio en la rigidez disminuye cada vez más la resistencia del suelo ocasionando que los empujes del frente de la tablestaca lleguen más fácilmente a la condición de falla pasiva, y por consiguiente, se produzca la plastificación de un mayor número de nodos. Igualmente, al revisar los desplazamientos en todas las etapas de excavación es posible observar que los valores obtenidos al considerar el cambio en la rigidez de los resortes en el frente de la tablestaca pueden llegar a duplicarse al compararlos con un planteamiento que no tiene en cuenta este efecto.

Teniendo en cuenta que para un suelo con cohesión la condición crítica de esfuerzos para el caso pasivo aumenta, al realizar la excavación en el frente de la tablestaca es mucho más difícil que los empujes alcancen la falla y plastifiquen el suelo. Por lo tanto, la distribución de empujes de las condiciones 2 y 4 para la última etapa de descarga, como la mostrada en la Figura 4.8, permite identificar que la cohesión aporta en gran medida a la resistencia del suelo aún cuando se realice la disminución de la rigidez del suelo por efecto de la excavación.

Al analizar el estado de los resortes en la última etapa de excavación, se observa que de los 22 resortes disponibles (sin contar con los seis resortes eliminados en la excavación) para las condiciones 2 y 4 se plastifican 6 resortes de los cuales 5 son del respaldo, lo cual equivale a un 27.3% del total. Del mismo modo, para las condiciones 1 y 3 se plastifican 16 resortes de los cuales 11 son del respaldo y 5 del frente, lo cual equivale a un 72.7% del total. Esta distribución de resortes plastificados se presenta de manera gráfica en la Figura 4.9.

De acuerdo con lo anterior, se puede establecer que las condiciones 2 y 4 son estables ya que aún quedan muchos resortes que mantienen el equilibrio de la estructura, sin embargo, en el caso de las condiciones 1 y 3 la estructura es inestable debido a que no es posible continuar con el análisis de plastificación al necesitarse como mínimo dos nodos con resortes sin fallar para realizar el cálculo de EMIH. De esta manera, para los parámetros de entrada definidos se obtiene que las condiciones 1 y 3 pueden llegar a soportar 3.0 m de excavación mientras que las condiciones 2 y 4 pueden llegar solo hasta los 2.50 m de excavación que corresponde a la quinta etapa de excavación donde solo se plastifican 11 de los 23 resortes disponibles.

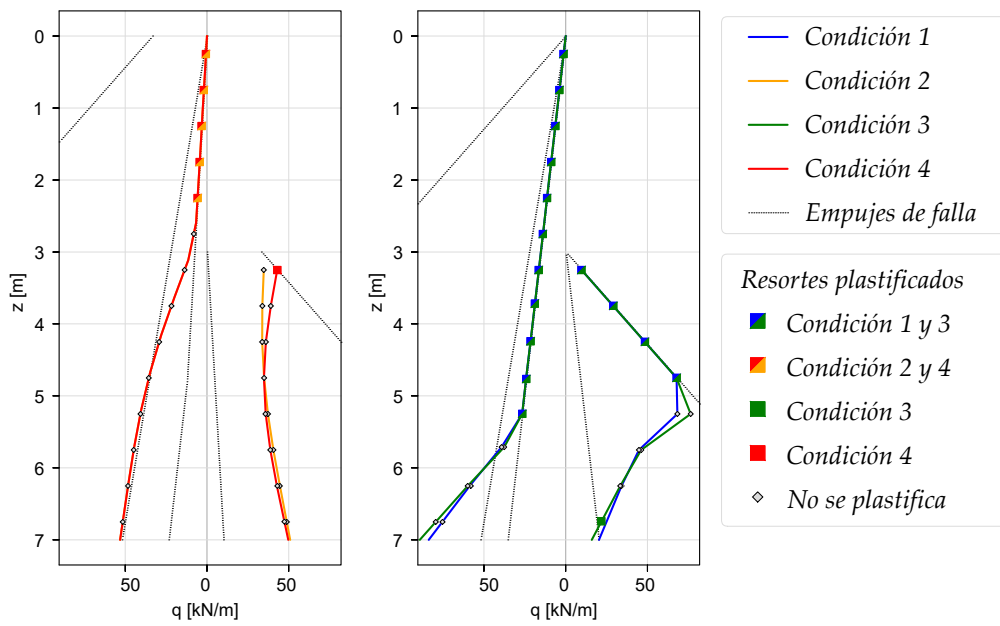


FIGURA 4.9: Resortes plastificados en la última etapa de excavación

## Capítulo 5

# Conclusiones y Recomendaciones

La aplicación del método propuesto permitió evidenciar que los esfuerzos tanto al frente como en el respaldo de la tablestaca se comportan normalmente de forma no lineal. Este comportamiento está definido por una transición de los esfuerzos del respaldo de la tablestaca desde una condición  $K_0$  hacia condición activa para los nodos superiores y desde una condición  $K_0$  hacia una condición pasiva para los nodos inferiores. Así mismo, para el frente de la tablestaca, la transición se da desde una condición  $K_0$  hacia una condición pasiva para los nodos bajo el nivel de excavación.

Al tener en cuenta las variaciones de las rigideces y los esfuerzos de fluencia en el frente de la tablestaca para las condiciones analizadas, se puede observar un aumento significativo de las deformaciones en cada etapa de descarga. De esta manera, es evidente que los diseños que contemplan únicamente un análisis de esfuerzos últimos no son suficientes para determinar el comportamiento real de estas estructuras.

Ya que la metodología realiza un análisis incremental de esfuerzos y deformaciones, se encontró que la tablestaca presenta zonas con falla local en los nodos plastificados, sin embargo, dependiendo de la resistencia del suelo, se pueden encontrar muchas zonas estables que evitan el colapso de la estructura. Así pues, este análisis incremental permite definir hasta que profundidad de excavación se puede llegar con una tablestaca de determinadas dimensiones.

Por otro lado, se demostró que la metodología propuesta es muy competente para analizar las deformaciones de una tablestaca. Esto se debe a que evalúa adecuadamente el proceso de deformación elastoplástico en cada nodo que se genera al aumentar el nivel de excavación y al reducir la rigidez de los resortes en el frente. Por lo tanto, a partir de estos análisis se pueden verificar las profundidades máximas de empotramiento de una tablestaca que garantizan la estabilidad de una excavación.

Una vez realizados los análisis para todas las etapas de excavación se puede utilizar la metodología planteada para optimizar el diseño de una tablestaca por medio de la verificación de los valores de esfuerzos cortantes y momentos de fluencia, y así determinar los módulos de sección mínimos que deben tener estas estructuras.

Teniendo en cuenta que en la práctica profesional y en muchos trabajos de investigación se suelen utilizar diferentes tipos de software disponibles en el mercado para el cálculo de estas estructuras, se resalta la importancia de la programación propia de los algoritmos de diseño ya que esto permite involucrar variables que no se encuentran por defecto en otros programas y pueden ser optimizados para incluir nuevos parámetros.

Aunque la metodología desarrollada en este trabajo plantea un primer acercamiento al análisis de interacción suelo estructura de tablestacas que involucran la variación de la rigidez del suelo en procesos de excavación efectuados de forma incremental por medio de etapas sucesivas de descarga, es posible realizar mejoras importantes tanto a la metodología como a la programación del algoritmo de diseño mediante el desarrollo de los siguientes aspectos:

- Calibración de la propuesta hiperbólica en la definición de los módulos de reacción del suelo a partir de la instrumentación de tablestacas.
- Aplicación de EMAH en el proceso de cálculo.
- Involucrar el efecto del tiempo mediante la simulación del suelo con unidades reológicas viscosas o viscoelásticas.
- Evaluar el comportamiento dinámico de la tablestaca.
- Aunque en este trabajo se evaluaron los cambios en la rigidez del suelo producidos únicamente por la excavación en el frente de la tablestaca, existen otros fenómenos que pueden reducir aún más la rigidez del suelo y producir un aumento en las deformaciones. Uno de ellos, es el flujo de agua por efecto del cambio en los niveles piezométricos, en donde el desplazamiento del agua en el respaldo genera una compresión del suelo aumentando su resistencia efectiva y en el frente una tensión que disminuye la resistencia efectiva del suelo.
- Como la metodología planteada considera un aumento significativo en los desplazamientos horizontales de la tablestaca, se pueden realizar análisis con anclajes o entibados para restringir el movimiento y mejorar la resistencia general de la estructura.

## Apéndice A

# Código del algoritmo de diseño

### A.1. SP\_SEI.py

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import scipy.optimize as op
4 from scipy.interpolate import interp1d
5 from scipy.interpolate import InterpolatedUnivariateSpline as IUS
6
7 def pred(phi,c,pe_f,gamma): #Predimensionamiento de la estructura
8     Kp, Ka, K0 = K_LEC(phi)
9     a = pe_f/4
10    D = 1
11    while D > 0.001:
12
13        f = lambda b: (0.5*gamma*Ka*((pe_f+a)**2-2*a*b -b**2)+ c*np.sqrt(Ka)*(2*b
14        - (pe_f+a)))+(
15            0.5*gamma*Kp*(2*b*(pe_f+a)+b**2 -a**2) + c*np.sqrt(Kp)*(2*b-a))
16
17        g = lambda b: (-(1/6)*gamma*Ka*((pe_f+a)**3 +3*a*b**2 +2*b**3)+(1/3)*c*np
18        .sqrt(Ka)*((
19            (pe_f+a)**2 + 3*b**2) + (1/6)*gamma*Kp*((pe_f+a)*3*b**2 + 2*b**3
20            + a**3
21            )+ (1/3)*c*np.sqrt(Kp)*(3*b**2 +a**2))
22
23        x0 = np.array([0.001,10])
24        rf = np.zeros_like(x0)
25        rg = np.zeros_like(x0)
26        for i in range(2):
27            rf[i]= op.newton(f,x0[i],tol=1.e-12,maxiter=1000)
28            rg[i]= op.newton(g,x0[i],tol=1.e-12,maxiter=1000)
29
30        D = rg[1] - rf[1]
31
32        a = a + 0.01
33        a = np.round(a,2)
34        b = np.round(rf[1],2)
35    return a,b

```

```

34 def K_LEC(phi): #Coeficientes de presiones laterales de tierra (Lateral Earth
    Coeficient)
35     Kp = (1+np.sin(np.radians(phi)))/(1-np.sin(np.radians(phi)))
36     Ka = 1/Kp
37     K0 = 1-np.sin(np.radians(phi))
38     return Kp, Ka, K0
39
40 def dim_nodos (): #Dimensionamiento de los nodos
41     n = pfe/st
42     nd_sup = st
43     nd_inf = nd_t - nd_sup
44     m = (a+b)/nd_inf
45     return n, m , nd_sup, nd_inf
46
47 def Z(): #Longitud desde la superficie hasta cada nodo
48     z = np.zeros([nd_t,1])
49     zf = np.zeros([nd_t,1])
50     z_ij = np.zeros([nd_t,1])
51     bd = np.zeros([nd_t,1])
52     for i in range (nd_t):
53         if i == 0:
54             z[i,0] = n/2
55             zf[i,0] = 0
56             z_ij[i,0] = L-n
57             bd[i,0] = n
58         elif i < nd_sup:
59             z[i,0] = z[i-1,0] + n
60             zf[i,0] = zf[i-1,0]+n
61             z_ij [i,0] = z_ij[i-1,0] - n
62             bd [i,0] = n
63         elif i == nd_sup:
64             z[i,0] = z[i-1,0] + 0.5*n + 0.5*m
65             zf[i,0] = zf[i-1,0] + 0.5*n + 0.5*m
66             z_ij [i,0] = z_ij[i-1,0] - 0.5*n - 0.5*m
67             bd [i,0] = m
68         elif i > nd_sup:
69             z [i,0] = z[i-1,0] + m
70             zf [i,0] = zf[i-1,0] + m
71             z_ij [i,0] = z_ij[i-1,0] - m
72             bd [i,0] = m
73     return z, z_ij , zf, bd
74
75 def K_calc (nd_t, gm, c, z, phi, K): #Cálculo de esfuerzos iniciales , activos y
    pasivos
76     qz_0 = np.zeros([nd_t,1])
77     q0 = np.zeros([nd_t,1])
78     qa = np.zeros([nd_t,1])
79     qp = np.zeros([nd_t,1])
80
81     Kp, Ka, K0 = K_LEC(phi)
82     KA = Ka/2
83     ZC= (4*c)/(gm*np.sqrt(Ka))
84

```

```

85     for i in range (nd_t):
86         qz_0[i,0] = gm * z[i]
87         q0[i,0] = qz_0[i,0] * K0
88         qp[i,0] = qz_0[i,0] * Kp + 2*c*np.sqrt(Kp)
89         if z[i] <= ZC:
90             qa[i,0] = qz_0[i,0] * KA
91         else:
92             qa[i,0] = qz_0[i,0] * Ka - 2*c*np.sqrt(Ka)
93
94     if K == "K0":
95         return q0
96     elif K == "Ka":
97         return qa
98     elif K == "Kp":
99         return qp
100
101 def ver_z(zif): #Verificación de la profundidad z en el frente por el cambio en
102     la rigidez de los resortes
103     if c_r == False:
104         zif = zif
105     elif c_r == True:
106         zif = zif
107     return zif
108
109 def prof_f (nd_t, nd_sup, phase, z): #Cambio de profundidad en el frente por
110     efecto de la excavación
111     zi_f = np.zeros([nd_t,1])
112     for i in range (nd_t):
113         if phase == 0:
114             zi_f[i,0] = z[i,0]
115         else:
116             if i == 0:
117                 zi_f [i,0] = 0
118             elif i < nd_sup:
119                 zi_f [i,0] = z[i-1,0] + n - phase*n
120             elif i == nd_sup:
121                 zi_f [i,0] = z[i-1,0] + 0.5*n + 0.5*m - phase*n
122             elif i > nd_sup:
123                 zi_f [i,0] = z[i-1,0] + m - phase*n
124         if zi_f [i,0] < 0:
125             zi_f[i,0] = 0
126     return zi_f
127
128 def k_srm(phase, z, zif, c_r, Fk, A, B, F, K, hh): #Cálculo del módulo de reacció
129     n de la subrasante
130     kf = np.zeros([nd_t,1])
131     kb = np.zeros([nd_t,1])
132     ki = np.zeros([nd_t,1])
133
134     #NO HAY CAMBIO DE RIGIDEZ DE LOS RESORTES DEL FRENTE POR LA EXCAVACIÓN
135     if c_r == False:
136         zif = z
137     for i in range (nd_t):

```

```

135         if phase == 0:
136             kf[i,0] = Fk*(zif[i,0]/L)**1
137         elif phase > 0 and phase <= i:
138             kf[i,0] = Fk*(zif[i,0]/L)**1
139             kb[i,0] = Fk*(z[i,0]/L)**1
140             ki[i] = kb[i]+kf[i]
141     else:
142     # SI HAY CAMBIO DE LA RIGIDEZ DE LOS RESORTES DEL FRENTE POR LA EXCAVACIÓN
143     for i in range (nd_t):
144         if phase == 0:
145             kf[i,0] = Fk*(zif[i,0]/L)**1
146             kb[i,0] = Fk*(z[i,0]/L)**1
147         elif phase > 0 and phase <= i:
148             if z[i,0] <= hh:
149                 kf[i,0] = z[i,0]/N
150             else:
151                 kf[i,0] = (B*np.sqrt(-A**2 + K**2 -2*K*z[i,0] + z[i,0]**2))/A
+ F
152                 kb[i,0] = Fk*(z[i,0]/L)**1
153                 ki[i] = kb[i]+kf[i]
154     return kf, kb, ki
155
156 def initLC (ph, nd_t, n, qi): #Condición inicial de carga
157     Pi = np.zeros([nd_t,1])
158     Pi[ph,0] = np.abs(qi[ph,0])*bd[ph,0]
159     return Pi
160
161 def d_ioP (P, lmd, zi, h):
162     dio_P = np.zeros([nd_t,1])
163     for i in range (nd_t):
164         if zi[i,0] <= h:
165             dio_P[i,0] = -((P*lmd)/(6*h))*(zi[i,0]**3 - zi[i,0]*h**2)
166         else:
167             dio_P[i,0] = -(((P*lmd)/(6*h))*(h*(h-zi[i,0])*(h-3*zi[i,0])) + (P/6)
* ((zi[i,0]-h-lmd)**3 - (zi[i,0]-h)**3))
168     return dio_P
169
170 def d_ioA (P, lmd, zi, h, ki, ph):
171     dio_A = np.zeros([nd_t,1])
172     for i in range (nd_t):
173         dio_A [i,0] = ((E*I*P)/(h**2))*(zi[i,0]*(lmd+h)/ki[ph,0] - lmd*(h-zi[i
,0])/ki[nd_t-1,0])
174     return dio_A
175
176 def Sji_1 (z, h):
177     sji_1 = np.zeros([nd_t,nd_t])
178     for i in range (nd_t):
179         for j in range (nd_t):
180             if z[i,0] <= h:
181                 if z[j,0] <= z[i,0] and z[j,0] < h:
182                     sji_1 [i,j]= (-((h-z[i,0])/h)*(z[j,0])**3 + (z[i,0]/h)*(z[i
,0]**2 + 2*h**2 - 3*h*z[i,0])*z[j,0])/6
183                 elif z[j,0] > z[i,0] and z[j,0] < h:

```

```

184         sji_1[i,j]= ((-((h-z[i,0])/h)*(z[j,0])**3 + (z[i,0]/h)*(z[i
,0]**2 + 2*h**2 - 3*h*z[i,0])*z[j,0])/6 + (1/6)*(z[j,0]-z[i,0])**3)
185
186         if z[i,0] > h:
187             if z[j,0]<= z[i,0] and z[j,0] <= h:
188                 sji_1[i,j] = ((1/(6*h))*(z[i,0]-h)*z[j,0]**3 - (h/6)*(z[i,0]-
h)*z[j,0])
189                 sji_1[j,i] = sji_1[i,j]
190             elif z[j,0]<= z[i,0] and z[j,0] > h:
191                 sji_1[i,j]= ((1/(6*h))*(z[i,0]-h)*z[j,0]**3 - (z[i,0]/(6*h))
*(z[j,0]-h)**3 - (h/6)*(z[i,0]-h)*z[j,0])
192             elif z[j,0] > z[i,0] and z[j,0] < h:
193                 sji_1[i,j]= ((1/(6*h))*(z[i,0]-h)*z[j,0]**3 - (h/6)*(z[i,0]-h
)*z[j,0] + (1/6)*(z[j,0]-z[i,0])**3)
194             elif z[j,0] > z[i,0] and z[j,0] > h:
195                 sji_1[i,j]= ((1/(6*h))*(z[i,0]-h)*z[j,0]**3 - (z[i,0]/(6*h))
*(z[j,0]-h)**3 - (h/6)*(z[i,0]-h)*z[j,0] + (1/6)*(z[j,0]-z[i,0])**3)
196         return sji_1
197
198 def Sji_2 (ki,z,h,ph):
199     sji_2 = np.zeros([nd_t, nd_t])
200     for j in range (nd_t):
201         for i in range (nd_t):
202             if z[j,0] < h and z[j,0] > 0:
203                 sji_2 [i,j] = (E*I/h**2)*((z[j,0]*z[i,0])/ki[ph,0] + (h-z[j,0])*(
h-z[i,0])/ki[nd_t-1,0])
204             elif z[j,0] > h:
205                 sji_2 [i,j] = (E*I/h**2)*((z[j,0]*z[i,0])/ki[ph,0] + (h-z[j,0])*(
h-z[i,0])/ki[nd_t-1,0])
206     return sji_2
207
208 def Sii_3 (ki):
209     sii_3 = np.zeros([nd_t, nd_t])
210     for i in range (nd_t):
211         sii_3 [i,i] = E*I/ki[i,0]
212     return sii_3
213
214 def dio (P, lmd, zi, h, ki, ph): #Cálculo final de dio para Xi=0
215     dio_P = d_ioP(P, lmd, zi, h)
216     dio_A = d_ioA(P, lmd, zi, h, ki, ph)
217     dio = (dio_P - dio_A)
218     return dio, dio_P, dio_A
219
220 def sji (zi, h, ki, ph):
221     sji_1 = Sji_1(zi, h)
222     sji_2 = Sji_2(ki, zi, h, ph)
223     sii_3 = Sii_3(ki)
224     sji = (sji_1 + sji_2 + sii_3)
225     return sji, sji_1, sji_2, sii_3
226
227
228 def HEMI (dio, sji): #Cálculo de ISE para obtener los valores de Xi
229     Xi = np.linalg.inv(sji)@dio

```

```

230     return Xi
231
232 def Vi_Mi (P, Xi, DXj, Vi, Mi):
233     for i in range (nd_t):
234         if i == 0:
235             Vi[i,0] = Xi[i,0] + P[i,0] + DXj[i,0]
236             Mi[i,0] = 0
237         elif i > 0:
238             Vi[i,0] = Xi[i,0] + Vi[i-1,0] + P[i,0] + DXj[i,0]
239             Mi[i,0] = Mi[i-1,0] + Vi[i-1,0] * (z[i,0]-z[i-1,0])
240     return Vi, Mi
241
242 def V_M (X, DXj, V, M):
243     QQ = np.zeros([nd_t,1])
244     VV = np.zeros([nd_t,1])
245     for i in range (nd_t):
246         if i == 0:
247             QQ[i,0] = (X[i,0] - DXj[i,0])/n
248             VV[i,0] = QQ[i,0]*n
249
250         else:
251             QQ[i,0] = (X[i,0] - DXj[i,0])/n
252             VV[i,0] = VV[i-1,0] + QQ[i,0]*n
253     VV = np.insert(VV, 0, 0)
254     VV = VV.flatten()
255
256     Vstep = interp1d(ZF, VV)
257     FF = Vstep(XX)
258     INTER_V = IUS(XX, FF, k = 1)
259     for i in range (nd_t):
260         V[i,0] = INTER_V(zf[i,0])
261
262     MM = np.zeros([len(XX),1])
263     for i in range (len(XX)):
264         MM [i,0] = INTER_V.integral(0,XX[i])
265     MM = MM.flatten()
266     INTER_M = IUS(XX, MM, k = 1)
267     for i in range (nd_t):
268         M[i,0] = INTER_M(z[i,0])
269
270     return V, M, INTER_V, INTER_M
271
272 def CXqf (Xqf ,phase): # Corrección de cargas y esfuerzos finales al frente por la
273     excavación
274     for i in range (nd_t):
275         if i < phase:
276             Xqf[i,0] = 0
277     return Xqf
278
279 def d_plast(Xplst, sji, ph):
280     dpl = np.zeros([nd_t, nd_t])
281     for j in range (nd_t):
282         dpl[0:,j] = Xplst[j,0]*sji[0:,j]

```

```

282     return dpl
283
284 def d_corr(sji_1, sji_2, sii_3, X):
285     for i in range (ph_j):
286         for j in range (ph_j):
287             sii_3 [i,j] = 0
288     sji = (sji_1 + sji_2 + sii_3)
289     DA = (sji@X)/(E*I)
290     return DA
291
292 def d_calc(z,INTER_M,n, nd_t):
293     GIRO = np.zeros ([nd_t,1])
294     DESP = np.zeros ([nd_t,1])
295
296     GG = np.zeros ([len(XX),1])
297     for i in range (len(XX)):
298         GG[i,0] =INTER_M.integral(0,XX[i])
299     GG = GG.flatten ()
300     INTER_G = IUS(XX,GG, k=1)
301     for i in range (nd_t):
302         GIRO[i,0] = INTER_G(z[i,0])
303
304     DD = np.zeros ([len(XX),1])
305     for i in range (len(XX)):
306         DD[len(XX)-1-i,0] = INTER_G.integral(0,XX[i])/(E*I)
307     DD = DD.flatten ()
308     INTER_D = IUS(XX,DD, k=1)
309     for i in range (nd_t):
310         DESP[i,0] = INTER_D(z[i,0])
311     return DESP, INTER_D
312
313
314
315 v = {} #Diccionario de variables
316 diag = {}
317 # =====
318 # DEFINICIÓN DE LOS ESFUERZOS PARA K0, Ka y Kp EN EL FRENTE Y RESPALDO
319 # =====
320 z, zi, zf, bd = Z() #Longitud desde la superficie hasta cada nodo
321 ZF = np.insert(zf, nd_t,L)
322 ZF = ZF.flatten ()
323 XX = np.linspace(0, L, 1000)
324 v['qa_b'] = -K_calc(nd_t, gm, c, z, phi, 'Ka') #Esfuerzo activo en el respaldo
325 v['q0_b'] = -K_calc(nd_t, gm, c, z, phi, 'K0') #Esfuerzo en condición K0 en el
    respaldo
326 v['qp_b'] = -K_calc(nd_t, gm, c, z, phi, 'Kp') #Esfuerzo en condición pasiva en
    el respaldo
327 for ph in range (st+1):
328     v['ZF_'+str(ph)] = z
329     v['zif_'+str(ph)] = prof_f (nd_t, nd_sup, ph, z) #Profundidad del frente para
    cada etapa de excavación
330 v['qa_f_'+str(ph)] = K_calc(nd_t, gm, c, ver_z(v['zif_'+str(ph)]), phi, 'Ka')
    #Esfuerzo activo en el frente para cada etapa de excavación

```

```

331 v['qp_f'+str(ph)] = K_calc(nd_t, gm, c, ver_z(v['zif_'+str(ph)]), phi, 'Kp')
#Esfuerzo en condición K0 en el frente para cada etapa de excavación
332 v['q0_f'+str(ph)] = K_calc(nd_t, gm, c, ver_z(v['zif_'+str(ph)]), phi, 'K0')
#Esfuerzo en condición pasiva en el frente para cada etapa de excavación
333
334 ph = 0
335 while ph <= st:
336     hh=zf[ph,0]
337     A = hh - M*hh/N
338     B = (hh/M) - (hh/N)
339     F = hh/N
340     K = M*hh/N
341 # =====
342 # DEFINICIÓN DE LOS MÓDULOS DE REACCIÓN DE LA SUBRASANTE
343 # =====
344 v['kf_'+str(ph)], v['kb_'+str(ph)], v['ki_'+str(ph)] = k_srm(ph, z, v['zif_'+
str(ph)], c_r, Fk, A, B, F, K, hh) #Definición de los módulos de reacción de
la subrasante
345 v['kfp1_'+str(ph)], v['kbp1_'+str(ph)], v['kipl_'+str(ph)] = k_srm(ph, z, v['
zif_'+str(ph)], c_r, Fk, A, B, F, K, hh)
346 if c_r == True:
347     v['dj0'] = -v['q0_b'] * bd / v['kb_'+str(ph)] #Definición de la deformaci
ón inicial de cada resorte
348 # =====
349 # CÁLCULO DEL ESTADO INICIAL DEL SUELO (ETAPA 0 DE EXCAVACIÓN)
350 # =====
351 if ph == 0:
352     v['qbi_'+str(ph)] = np.zeros([nd_t,1])
353     v['qb_'+str(ph)] = -K_calc (nd_t,gm,c,z, phi, 'K0')
354     v['qfi_'+str(ph)] = np.zeros([nd_t,1])
355     v['qf_'+str(ph)] = K_calc (nd_t,gm,c,z, phi, 'K0')
356     v['qi_'+str(ph)] = np.zeros([nd_t,1])
357     v['q_'+str(ph)] = v['qb_'+str(ph)] + v['qf_'+str(ph)]
358     v['Xbi_'+str(ph)] = np.zeros([nd_t,1])
359     v['Xfi_'+str(ph)] = np.zeros([nd_t,1])
360     v['Xi_'+str(ph)] = v['Xbi_'+str(ph)] + v['Xfi_'+str(ph)]
361     v['Xb_'+str(ph)] = v['qb_'+str(ph)] * bd
362     v['Xf_'+str(ph)] = v['qf_'+str(ph)] * bd
363     v['X_'+str(ph)] = v['Xb_'+str(ph)] + v['Xf_'+str(ph)]
364     v['Mi_'+str(ph)] = np.zeros([nd_t,1])
365     v['M_'+str(ph)] = np.zeros([nd_t,1])
366     v['di_'+str(ph)] = np.zeros([nd_t,1])
367     v['d_'+str(ph)] = np.zeros([nd_t,1])
368     v['Vi_'+str(ph)] = np.zeros([nd_t,1])
369     v['V_'+str(ph)] = np.zeros([nd_t,1])
370
371     v['Pi_'+str(ph)] = initLC(ph, nd_t, n, v['qf_'+str(ph)])
372     ph+=1
373 # =====
374 # CÁLCULO DE LAS SIGUIENTES ETAPAS DE EXCAVACIÓN (VARIABLES GENERALES q,Q,V,M,D)
375 # =====
376 elif ph <= st:

```

```

377     h = zi[ph,0]
378
379     v[ 'Vi_'+str(ph) ] = np.zeros([nd_t,1])
380     v[ 'Mi_'+str(ph) ] = np.zeros([nd_t,1])
381     v[ 'V_'+str(ph) ] = np.zeros([nd_t,1])
382     v[ 'M_'+str(ph) ] = np.zeros([nd_t,1])
383     v[ 'Dqj_'+str(ph) ] = np.zeros([nd_t,1])
384     v[ 'DXj_'+str(ph) ] = np.zeros([nd_t,1])
385
386     P = v[ 'Pi_'+str(ph-1) ][ph-1,0]
387
388     v[ 'dio_'+str(ph) ], v[ 'dioP_'+str(ph) ], v[ 'dioA_'+str(ph) ] = dio(P, (bd[ph
,0]+bd[ph-1,0])/2, zi, h, v[ 'ki_'+str(ph) ], ph)
389     v[ 'sji_'+str(ph) ], v[ 'sji1_'+str(ph) ], v[ 'sji2_'+str(ph) ], v[ 'sii3_'+str(
ph) ] = sji(zi, h, v[ 'ki_'+str(ph) ], ph)
390     v[ 'Xi_'+str(ph) ] = HEMI(v[ 'dio_'+str(ph) ], v[ 'sji_'+str(ph) ])
391     v[ 'di_'+str(ph) ] = v[ 'Xi_'+str(ph) ] / v[ 'ki_'+str(ph) ]
392
393     if c_r == True:
394         v[ 'Dqj_'+str(ph) ] = ((v[ 'kf_'+str(ph-1) ] - v[ 'kf_'+str(ph) ]) * (v[ '
dj0_ ' ] - v[ 'di_'+str(ph-1) ]))
395         v[ 'DXj_'+str(ph) ] = v[ 'Dqj_'+str(ph) ] *bd
396         v[ 'DDA_'+str(ph) ] = (v[ 'sji_'+str(ph) ]@v[ 'DXj_'+str(ph) ])
397         v[ 'dio_'+str(ph) ] = v[ 'dio_'+str(ph) ] - v[ 'DDA_'+str(ph) ]
398         v[ 'Xi_'+str(ph) ] = HEMI(v[ 'dio_'+str(ph) ], v[ 'sji_'+str(ph) ])
399         v[ 'di_'+str(ph) ] = v[ 'Xi_'+str(ph) ] / v[ 'ki_'+str(ph) ]
400
401     v[ 'Xbi_'+str(ph) ] = v[ 'Xi_'+str(ph) ] - v[ 'di_'+str(ph) ]*v[ 'kf_'+str(ph) ]
402     v[ 'Xfi_'+str(ph) ] = v[ 'Xbi_'+str(ph) ] - v[ 'Xi_'+str(ph) ]
403     v[ 'qi_'+str(ph) ] = v[ 'Xi_'+str(ph) ] / bd
404     v[ 'qbi_'+str(ph) ] = v[ 'Xbi_'+str(ph) ] / bd
405     v[ 'qfi_'+str(ph) ] = v[ 'Xfi_'+str(ph) ] / bd
406     v[ 'Vi_'+str(ph) ], v[ 'Mi_'+str(ph) ] = Vi_Mi(v[ 'Pi_'+str(ph-1) ], v[ 'Xi_'+
str(ph) ], v[ 'DXj_'+str(ph) ], v[ 'Vi_'+str(ph) ], v[ 'Mi_'+str(ph) ])
407
408     v[ 'Xb_'+str(ph) ] = v[ 'Xb_'+str(ph-1) ] - v[ 'Xbi_'+str(ph) ]
409     v[ 'Xf_'+str(ph) ] = v[ 'Xf_'+str(ph-1) ] + v[ 'Xfi_'+str(ph) ]
410     v[ 'Xf_'+str(ph) ] = CXqf (v[ 'Xf_'+str(ph) ], ph) #Corrección de las fuerzas
del frente al quitar los resortes de la etapa de excavación
411     v[ 'X_'+str(ph) ] = v[ 'Xb_'+str(ph) ] + v[ 'Xf_'+str(ph) ]
412     v[ 'qb_'+str(ph) ] = v[ 'Xb_'+str(ph) ] / bd
413     v[ 'qf_'+str(ph) ] = v[ 'Xf_'+str(ph) ] / bd
414     v[ 'q_'+str(ph) ] = v[ 'qb_'+str(ph) ] + v[ 'qf_'+str(ph) ]
415     v[ 'V_'+str(ph) ], v[ 'M_'+str(ph) ], v[ 'INTERV_'+str(ph) ], v[ 'INTERM_'+str(
ph) ] = V_M(v[ 'X_'+str(ph) ], v[ 'DXj_'+str(ph) ],v[ 'V_'+str(ph) ], v[ 'M_'+str(ph)
])
416     v[ 'd_'+str(ph) ], v[ 'INTERD_'+str(ph) ] = d_calc(z,v[ 'INTERM_'+str(ph) ], n,
nd_t)
417 # =====
418 # ANÁLISIS DE PLASTIFICACIÓN
419 # =====
420     v[ 'vplst_'+str(ph) ] = np.zeros([nd_t,1])
421     v[ 'dio_plst_'+str(ph) ] = np.zeros([nd_t,1])

```

```

422     v['Xplst_'+str(ph)] = np.zeros([nd_t,1])
423     v['Xplst_b_'+str(ph)] = np.zeros([nd_t,1])
424     v['Xplst_f_'+str(ph)] = np.zeros([nd_t,1])
425     v['dpl'] = np.zeros([nd_t,nd_t])
426     ph_j = ph # Se inicializa la variable que acumula las fases de excavación
para el análisis de plastificación y la respectiva corrección de dio y sji
427     h2 = h
428
429     for i in range (nd_t): #Valores temporales de reacciones para el análisis
de plastificación
430         v['XXi_'+str(ph)+'_'+str(i)] = np.zeros([nd_t,1])
431         v['XXXi_'+str(ph)+'_'+str(i)] = np.zeros([nd_t,1])
432
433         j=0 #Se inicializa la variable j para recorrer cada nodo
434         verif=0
435         while j < nd_t:
436 # =====
437             #Plastificación del resorte frente (Derecha) para condición pasiva
438             if v['qf_'+str(ph)][j,0] > v['qp_f_'+str(ph)][j,0] and v['qb_'+str(ph)
]][j,0] < v['qa_b']][j,0]:
439                 v['vplst_'+str(ph)] [j,0] = 1
440
441                 v['Xplst_'+str(ph)] [j,0] = -(v['qf_'+str(ph-1)][j,0]-v['qp_f_'+
str(ph)][j,0])*bd[j,0]
442                 v['Xplst_f_'+str(ph)] [j,0] = v['Xplst_'+str(ph)] [j,0]
443
444                 if j >= ph_j:
445                     h2= zi[ph_j+1,0]
446                     ph_j = ph_j + 1
447
448                     v['dio_'+str(ph)], v['dioP_'+str(ph)], v['dioA_'+str(ph)] =
dio(P, ((bd[ph_j,0]+bd[ph_j-1,0])/2)+(h-h2), zi, h2, v['ki_'+str(ph)], ph_j)
449                     v['sji_'+str(ph)], v['sji1_'+str(ph)], v['sji2_'+str(ph)], v[
'sii3_'+str(ph)] = sji(zi, h2, v['ki_'+str(ph)], ph_j)
450                     if c_r == True:
451                         v['Dqj_'+str(ph)] = -((v['kf_'+str(ph-1)] - v['kf_'+str(
ph)]) * (v['di_'+str(ph)] - v['di_'+str(ph-1)]))
452                         v['DXj_'+str(ph)] = v['Dqj_'+str(ph)] *bd
453                         v['DDA_'+str(ph)] = (v['sji_'+str(ph)]@v['DXj_'+str(ph)])
454                         v['dio_'+str(ph)] = v['dio_'+str(ph)] - v['DDA_'+str(ph)]
455
456                     v['dpl'] = d_plast(v['Xplst_'+str(ph)], v['sji_'+str(ph)], ph)
457                     v['dio_plst_'+str(ph)] = v['dio_'+str(ph)] - np.sum(v['dpl'],
axis=1, keepdims=1)
458
459                     v['XXi_'+str(ph)+'_'+str(j)] = HEMI(v['dio_plst_'+str(ph)][j
+1:,0], v['sji_'+str(ph)][j+1:,j+1:])
460                     v['XXXi_'+str(ph)+'_'+str(j)] = np.insert(v['XXi_'+str(ph)+'_'+
str(j)], 0, v['Xplst_'+str(ph)] [:j+1,0]).reshape((nd_t,1))
461                     v['Xi_'+str(ph)] = v['XXXi_'+str(ph)+'_'+str(j)]
462
463 # =====

```

```

464         #Plastificación del resorte de atrás (Izquierda) para condición
activa
465         elif v['qf_'+str(ph)][j,0] <= v['qp_f_'+str(ph)][j,0] and v['qb_'+str
(ph)][j,0] > v['qa_b'][j,0] and j <= ph:
466             v['vplst_'+str(ph)][j,0] = 2
467             v['kbpl_'+str(ph)][j,0] = 0
468             v['kipl_'+str(ph)][j,0] = v['kbpl_'+str(ph)][j,0] + v['kfpl_'+str
(ph)][j,0]
469             v['Xplst_'+str(ph)][j,0] = (v['qb_'+str(ph-1)][j,0]-v['qa_b'][j
,0])*bd[j,0]
470             v['Xplst_b_'+str(ph)][j,0] = v['Xplst_'+str(ph)][j,0]
471
472             if j >= ph_j:
473                 h2= zi[ph_j+1,0]
474                 print(h2)
475                 ph_j = ph_j + 1
476
477             v['dio_'+str(ph)], v['dioP_'+str(ph)], v['dioA_'+str(ph)] =
dio(P, ((bd[ph_j,0]+bd[ph_j-1,0])/2)+(h-h2), zi, h2, v['ki_'+str(ph)], ph_j)
478             v['sji_'+str(ph)], v['sji1_'+str(ph)], v['sji2_'+str(ph)], v[
'sii3_'+str(ph)] = sji(zi, h2, v['ki_'+str(ph)], ph_j)
479
480             if c_r == True:
481                 v['Dqj_'+str(ph)] = -((v['kf_'+str(ph-1)] - v['kf_'+str(
ph)])) * (v['di_'+str(ph)] - v['di_'+str(ph-1)])
482                 v['DXj_'+str(ph)] = v['Dqj_'+str(ph)] *bd
483                 v['DDA_'+str(ph)] = (v['sji_'+str(ph)]@v['DXj_'+str(ph)])
484                 v['dio_'+str(ph)] = v['dio_'+str(ph)] - v['DDA_'+str(ph)]
485
486                 v['dpl'] = d_plast(v['Xplst_'+str(ph)], v['sji_'+str(ph)], ph)
487                 v['dio_plst_'+str(ph)] = v['dio_'+str(ph)] - np.sum(v['dpl'],
axis=1, keepdims=1)
488
489                 v['XXi_'+str(ph)+'_'+str(j)] = HEMI(v['dio_plst_'+str(ph)][j
+1:,0], v['sji_'+str(ph)][j+1:,j+1:])
490                 v['XXXi_'+str(ph)+'_'+str(j)] = np.insert(v['XXi_'+str(ph)+'_'+
str(j)], 0, v['Xplst_'+str(ph)][j+1,0]).reshape((nd_t,1))
491                 v['Xi_'+str(ph)] = v['XXXi_'+str(ph)+'_'+str(j)]
492 # =====
493         #Plastificación de ambos resortes (resorte de atrás en condición
activa y resorte del frente en condición pasiva)
494         elif v['qf_'+str(ph)][j,0] > v['qp_f_'+str(ph)][j,0] and v['qb_'+str(
ph)][j,0] >= v['qa_b'][j,0]:
495             v['vplst_'+str(ph)][j,0] = 3
496             v['kbpl_'+str(ph)][j,0] = 0
497             v['kfpl_'+str(ph)][j,0] = 0
498             v['kipl_'+str(ph)][j,0] = v['kbpl_'+str(ph)][j,0] + v['kfpl_'+str
(ph)][j,0]
499             v['Xplst_'+str(ph)][j,0] = (v['qb_'+str(ph-1)][j,0]-v['qa_b'][j
,0])*bd[j,0] + (v['qf_'+str(ph-1)][j,0]-v['qp_f_'+str(ph)][j,0])*bd[j,0]
500             v['Xplst_b_'+str(ph)][j,0] = (v['qb_'+str(ph-1)][j,0]-v['qa_b'][j
,0])*bd[j,0]

```

```

501     v['Xplst_f_'+str(ph)][j,0] = -(v['qf_'+str(ph-1)][j,0]-v['qp_f_'
+str(ph)][j,0])*bd[j,0]
502
503     if j >= ph_j:
504         h2= zi[ph_j+1,0]
505         ph_j = ph_j + 1
506
507     v['dio_'+str(ph)], v['dioP_'+str(ph)], v['dioA_'+str(ph)] =
dio(P, ((bd[ph_j,0]+bd[ph_j-1,0])/2)+(h-h2), zi, h2, v['ki_'+str(ph)], ph_j)
508     v['sji_'+str(ph)], v['sji1_'+str(ph)], v['sji2_'+str(ph)], v[
'sii3_'+str(ph)] = sji(zi, h2, v['ki_'+str(ph)], ph_j)
509
510     if c_r == True:
511         v['Dqj_'+str(ph)] = -((v['kf_'+str(ph-1)] - v['kf_'+str(
ph)])) * (v['di_'+str(ph)] - v['di_'+str(ph-1)])
512         v['DXj_'+str(ph)] = v['Dqj_'+str(ph)] *bd
513         v['DDA_'+str(ph)] = (v['sji_'+str(ph)]@v['DXj_'+str(ph)])
514         v['dio_'+str(ph)] = v['dio_'+str(ph)] - v['DDA_'+str(ph)]
515
516     v['dpl'] = d_plast(v['Xplst_'+str(ph)], v['sji_'+str(ph)], ph)
517     v['dio_plst_'+str(ph)] = v['dio_'+str(ph)] - np.sum(v['dpl'],
axis=1, keepdims=1)
518
519     v['XXi_'+str(ph)+'_'+str(j)] = HEMI(v['dio_plst_'+str(ph)][j
+1:,0], v['sji_'+str(ph)][j+1:,j+1:])
520     v['XXXi_'+str(ph)+'_'+str(j)] = np.insert(v['XXi_'+str(ph)+'_'+
str(j)], 0, v['Xplst_'+str(ph)][j+1,0]).reshape((nd_t,1))
521     v['Xi_'+str(ph)] = v['XXXi_'+str(ph)+'_'+str(j)]
522 # =====
523     #Plastificación del resorte de atrás (Izquierda) para condición
activa cuando el nodo evaluado está por debajo del nivel de excavación
524     elif v['qf_'+str(ph)][j,0] <= v['qp_f_'+str(ph)][j,0] and v['qb_'+str
(ph)][j,0] > v['qa_b'][j,0] and j > ph:
525         v['vplst_'+str(ph)][j,0] = 4
526         v['kbpl_'+str(ph)][j,0] = 0
527         v['kipl_'+str(ph)][j,0] = v['kbpl_'+str(ph)][j,0] + v['kfpl_'+str
(ph)][j,0]
528         v['Xplst_'+str(ph)][j,0] = (v['qb_'+str(ph-1)][j,0]-v['qa_b'][j
,0])*bd[j,0]
529         v['Xplst_b_'+str(ph)][j,0] = v['Xplst_'+str(ph)][j,0]
530
531     v['dio_'+str(ph)], v['dioP_'+str(ph)], v['dioA_'+str(ph)] = dio(P
, ((bd[ph,0]+bd[ph-1,0])/2)+(h-h2), zi, h2, v['ki_'+str(ph)], ph_j)
532     if c_r == True:
533         v['dio_'+str(ph)] = v['dio_'+str(ph)] - v['DDA_'+str(ph)]
534         v['sji_'+str(ph)], v['sji1_'+str(ph)], v['sji2_'+str(ph)], v[
'sii3_'+str(ph)] = sji(zi, h2, v['ki_'+str(ph)], ph_j)
535
536     v['dpl'] = d_plast(v['Xplst_'+str(ph)], v['sji_'+str(ph)], ph)
537     v['dio_plst_'+str(ph)] = v['dio_'+str(ph)] - np.sum(v['dpl'],
axis=1, keepdims=1)
538

```

```

539     v['XXi_'+str(ph)+'_'+str(j)] = HEMI(v['dio_plst_'+str(ph)][j
: ,0], v['sji_'+str(ph)][j:,j:])
540     v['XXXi_'+str(ph)+'_'+str(j)] = np.insert(v['XXi_'+str(ph)+'_'+
str(j)], 0, v['Xplst_'+str(ph)][j,0]).reshape((nd_t,1))
541     v['XXXi_'+str(ph)+'_'+str(j)][j-1,0] = v['XXXi_'+str(ph)+'_'+
str(j)][j-1,0] + v['Xplst_b_'+str(ph)][j,0]
542     v['Xi_'+str(ph)] = v['XXXi_'+str(ph)+'_'+str(j)]
543
544 # =====
545 # CORRECCIÓN DE X, q, V, M y d POR EFECTO DE LA PLASTIFICACIÓN
546 # =====
547     v['di_'+str(ph)] = v['Xi_'+str(ph)] / v['ki_'+str(ph)]
548     for it in range(nd_t):
549         if v['vplst_'+str(ph)][it,0] == 1: #Plastificación adelante en
condición pasiva cuando j <= ph
550             v['di_'+str(ph)][it,0] = v['Xi_'+str(ph)][it,0] / v['kb_'+str
(ph)][it,0]
551             v['Xfi_'+str(ph)][it,0] = -v['Xplst_f_'+str(ph)][it,0]
552             v['Xbi_'+str(ph)][it,0] = -v['Xfi_'+str(ph)][it,0] - v['Xi_'+
str(ph)][it,0] #- v['Xplst_f_'+str(ph)][it,0]
553             elif v['vplst_'+str(ph)][it,0] == 2: #Plastificación atrás en
condición activa cuando j <= ph
554                 v['Xbi_'+str(ph)][it,0] = v['Xplst_b_'+str(ph)][it,0]
555                 v['Xfi_'+str(ph)][it,0] = 0
556             elif v['vplst_'+str(ph)][it,0] == 3: #Plastificación de ambos
resortes (atrás en condición activa y adelante en condición pasiva)
557                 v['Xbi_'+str(ph)][it,0] = v['Xplst_b_'+str(ph)][it,0]
558                 v['Xfi_'+str(ph)][it,0] = v['Xplst_f_'+str(ph)][it,0]
559             elif v['vplst_'+str(ph)][it,0] == 4: #Plastificación atrás en
condición activa cuando j > ph
560                 v['di_'+str(ph)][it,0] = v['Xi_'+str(ph)][it,0] / (v['kf_'+
str(ph)][it,0]*1)
561                 v['Xbi_'+str(ph)][it,0] = v['Xplst_b_'+str(ph)][it,0]
562                 v['Xfi_'+str(ph)][it,0] = v['Xbi_'+str(ph)][it,0] - v['Xi_'+
str(ph)][it,0] - v['Xplst_b_'+str(ph)][it,0]
563             elif v['vplst_'+str(ph)][it,0] == 0: # No hay plastificación
564                 v['di_'+str(ph)][it,0] = v['Xi_'+str(ph)][it,0] / v['ki_'+str
(ph)][it,0]
565                 v['Xbi_'+str(ph)][it,0] = v['Xi_'+str(ph)][it,0] - v['di_'+
str(ph)][it,0]*v['kf_'+str(ph)][it,0]
566                 v['Xfi_'+str(ph)][it,0] = v['Xbi_'+str(ph)][it,0] - v['Xi_'+
str(ph)][it,0]
567
568     v['qi_'+str(ph)] = v['Xi_'+str(ph)] / bd
569     v['qbi_'+str(ph)] = v['Xbi_'+str(ph)] / bd
570     v['qfi_'+str(ph)] = v['Xfi_'+str(ph)] / bd
571     v['Vi_'+str(ph)], v['Mi_'+str(ph)] = Vi_Mi(v['Pi_'+str(ph-1)], v['Xi_
'+str(ph)], v['DXj_'+str(ph)], v['Vi_'+str(ph)], v['Mi_'+str(ph)])
572
573     #Acumulación de cargas por reducción de rigidez en el frente
574     v['DXjF'] = np.zeros([nd_t,1])
575     if c_r == True:
576         for it in range(ph):

```

```

577         v['DXjF'] = v['DXjF'] + v['DXj_' + str(it+1)]
578
579     v['Xb_' + str(ph)] = v['Xb_' + str(ph-1)] - v['Xbi_' + str(ph)]
580     v['Xf_' + str(ph)] = v['Xf_' + str(ph-1)] + v['Xfi_' + str(ph)]
581     v['Xf_' + str(ph)] = CXqf (v['Xf_' + str(ph)], ph) #Corrección de las
fuerzas del frente al quitar los resortes de la etapa de excavación
582     v['X_' + str(ph)] = v['Xb_' + str(ph)] + v['Xf_' + str(ph)]
583     v['qb_' + str(ph)] = v['Xb_' + str(ph)] / bd
584     v['qf_' + str(ph)] = v['Xf_' + str(ph)] / bd
585     v['q_' + str(ph)] = v['qb_' + str(ph)] + v['qf_' + str(ph)]
586     v['V_' + str(ph)], v['M_' + str(ph)], v['INTERV_' + str(ph)], v['INTERM_' +
str(ph)] = VM(v['X_' + str(ph)], v['DXjF'], v['V_' + str(ph)], v['M_' + str(ph)]
587     v['d_' + str(ph)], v['INTERD_' + str(ph)] = d_calc(z, v['INTERM_' + str(ph)
], n, nd_t)
588     v['di_' + str(ph)] = v['d_' + str(ph)] - v['d_' + str(ph-1)]
589
590     #Verificación de plastificación cuando se plastifica atrás y adelante
luego de analizar una plastificación unicamente atrás.
591     if v['qf_' + str(ph)][j,0] > v['qp_f_' + str(ph)][j,0] and v['qb_' + str(ph)
][j,0] >= v['qa_b']][j,0]:
592         if verif >=1:
593             verif = 0
594         else:
595             verif+=1
596             j=j-1
597     elif v['qf_' + str(ph)][j,0] > v['qp_f_' + str(ph)][j,0] and v['qb_' + str(
ph)][j,0] < v['qa_b']][j,0]:
598         if verif >=1:
599             verif = 0
600         else:
601             verif+=1
602             j=j-1
603
604     j+=1
605     # Advertencia de plastificación de nodos y falla de la estructura
606     if ph_j >= nd_t-2:
607         print ("Se plastificaron " + str(ph_j) + " de " + str(nd_t) + " nodos
evaluados.")
608         break
609
610     else:
611         print ("Se realizó el análisis completo para la etapa de excavación No
." + str(ph))
612 # =====
613 # CÁLCULO DE P PARA LA SIGUIENTE ETAPA DE EXCAVACIÓN
614 # =====
615     v['Pi_' + str(ph)] = initLC(ph, nd_t, n, v['qf_' + str(ph)])
616     ph+=1

```

## A.2. appSPSEI\_ui.py

```

1 # Created by: PyQt5 UI code generator 5.9.2
2
3 from PyQt5 import QtCore, QtGui, QtWidgets
4
5 class Ui_MainWindow(object):
6     def setupUi(self, MainWindow):
7         MainWindow.setObjectName("MainWindow")
8         MainWindow.resize(1334, 686)
9         MainWindow.setContextMenuPolicy(QtCore.Qt.PreventContextMenu)
10        self.centralwidget = QtWidgets.QWidget(MainWindow)
11        self.centralwidget.setObjectName("centralwidget")
12        self.pushButton = QtWidgets.QPushButton(self.centralwidget)
13        self.pushButton.setGeometry(QtCore.QRect(1120, 580, 75, 81))
14        self.pushButton.setObjectName("pushButton")
15        self.listWidget = QtWidgets.QListWidget(self.centralwidget)
16        self.listWidget.setGeometry(QtCore.QRect(1200, 580, 121, 81))
17        sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Maximum,
18        QtWidgets.QSizePolicy.Expanding)
19        sizePolicy.setHorizontalStretch(0)
20        sizePolicy.setVerticalStretch(0)
21        sizePolicy.setHeightForWidth(self.listWidget.sizePolicy().
22        hasHeightForWidth())
23        self.listWidget.setSizePolicy(sizePolicy)
24        self.listWidget.setObjectName("listWidget")
25        self.tabWidget = QtWidgets.QTabWidget(self.centralwidget)
26        self.tabWidget.setGeometry(QtCore.QRect(20, 10, 1300, 560))
27        self.tabWidget.setObjectName("tabWidget")
28        self.tab = QtWidgets.QWidget()
29        self.tab.setEnabled(False)
30        self.tab.setObjectName("tab")
31        self.widget_1 = QtWidgets.QWidget(self.tab)
32        self.widget_1.setGeometry(QtCore.QRect(10, 10, 1270, 515))
33        sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Expanding,
34        QtWidgets.QSizePolicy.Preferred)
35        sizePolicy.setHorizontalStretch(0)
36        sizePolicy.setVerticalStretch(0)
37        sizePolicy.setHeightForWidth(self.widget_1.sizePolicy().hasHeightForWidth
38        ())
39        self.widget_1.setSizePolicy(sizePolicy)
40        self.widget_1.setObjectName("widget_1")
41        self.verticalLayout_1 = QtWidgets.QVBoxLayout(self.widget_1)
42        self.verticalLayout_1.setContentsMargins(0, 0, 0, 0)
43        self.verticalLayout_1.setObjectName("verticalLayout_1")
44        self.tabWidget.addTab(self.tab, "")
45        self.tab_2 = QtWidgets.QWidget()
46        self.tab_2.setObjectName("tab_2")
47        self.widget_2 = QtWidgets.QWidget(self.tab_2)
48        self.widget_2.setGeometry(QtCore.QRect(10, 10, 1270, 515))
49        sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Expanding,
50        QtWidgets.QSizePolicy.Preferred)

```

```

46     sizePolicy .setHorizontalStretch(0)
47     sizePolicy .setVerticalStretch(0)
48     sizePolicy .setHeightForWidth(self .widget_2 .sizePolicy () .hasHeightForWidth
    ())
49     self .widget_2 .setSizePolicy (sizePolicy)
50     self .widget_2 .setObjectName ("widget_2")
51     self .verticalLayout_2 = QtWidgets.QVBoxLayout(self .widget_2)
52     self .verticalLayout_2 .setContentsMargins(0, 0, 0, 0)
53     self .verticalLayout_2 .setObjectName ("verticalLayout_2")
54     self .tabWidget .addTab(self .tab_2, "")
55     self .tab_3 = QtWidgets.QWidget()
56     self .tab_3 .setObjectName ("tab_3")
57     self .tablageral = QtWidgets.QTableView(self .tab_3)
58     self .tablageral .setGeometry(QtCore.QRect(10, 10, 1271, 511))
59     self .tablageral .setObjectName ("tablageral")
60     self .tabWidget .addTab(self .tab_3, "")
61     self .tab_4 = QtWidgets.QWidget()
62     self .tab_4 .setObjectName ("tab_4")
63     self .plainTextEdit = QtWidgets.QPlainTextEdit(self .tab_4)
64     self .plainTextEdit .setGeometry(QtCore.QRect(10, 10, 1270, 515))
65     self .plainTextEdit .setObjectName ("plainTextEdit")
66     self .tabWidget .addTab(self .tab_4, "")
67     self .groupBox = QtWidgets.QGroupBox(self .centralwidget)
68     self .groupBox .setGeometry(QtCore.QRect(20, 580, 231, 81))
69     self .groupBox .setObjectName ("groupBox")
70     self .label_3 = QtWidgets.QLabel(self .groupBox)
71     self .label_3 .setGeometry(QtCore.QRect(120, 20, 47, 13))
72     self .label_3 .setObjectName ("label_3")
73     self .line_phi = QtWidgets.QLineEdit(self .groupBox)
74     self .line_phi .setGeometry(QtCore.QRect(60, 20, 51, 20))
75     self .line_phi .setObjectName ("line_phi")
76     self .label = QtWidgets.QLabel(self .groupBox)
77     self .label .setGeometry(QtCore.QRect(10, 20, 47, 13))
78     self .label .setObjectName ("label")
79     self .line_gamma = QtWidgets.QLineEdit(self .groupBox)
80     self .line_gamma .setGeometry(QtCore.QRect(170, 20, 51, 20))
81     self .line_gamma .setObjectName ("line_gamma")
82     self .line_c = QtWidgets.QLineEdit(self .groupBox)
83     self .line_c .setGeometry(QtCore.QRect(60, 50, 51, 20))
84     self .line_c .setObjectName ("line_c")
85     self .label_2 = QtWidgets.QLabel(self .groupBox)
86     self .label_2 .setGeometry(QtCore.QRect(10, 50, 47, 13))
87     self .label_2 .setObjectName ("label_2")
88     self .line_Fk = QtWidgets.QLineEdit(self .groupBox)
89     self .line_Fk .setGeometry(QtCore.QRect(170, 50, 51, 20))
90     self .line_Fk .setObjectName ("line_Fk")
91     self .label_6 = QtWidgets.QLabel(self .groupBox)
92     self .label_6 .setGeometry(QtCore.QRect(120, 50, 47, 13))
93     self .label_6 .setObjectName ("label_6")
94     self .groupBox_2 = QtWidgets.QGroupBox(self .centralwidget)
95     self .groupBox_2 .setGeometry(QtCore.QRect(430, 580, 151, 81))
96     self .groupBox_2 .setObjectName ("groupBox_2")
97     self .line_Prof = QtWidgets.QLineEdit(self .groupBox_2)

```

```
98     self.line_Prof.setGeometry(QtCore.QRect(90, 20, 51, 20))
99     self.line_Prof.setObjectName("line_Prof")
100    self.label_4 = QtWidgets.QLabel(self.groupBox_2)
101    self.label_4.setGeometry(QtCore.QRect(10, 20, 71, 16))
102    self.label_4.setObjectName("label_4")
103    self.predimButton = QtWidgets.QPushButton(self.groupBox_2)
104    self.predimButton.setGeometry(QtCore.QRect(20, 50, 111, 23))
105    self.predimButton.setObjectName("predimButton")
106    self.groupBox_3 = QtWidgets.QGroupBox(self.centralwidget)
107    self.groupBox_3.setGeometry(QtCore.QRect(730, 580, 231, 81))
108    self.groupBox_3.setObjectName("groupBox_3")
109    self.line_Stage = QtWidgets.QLineEdit(self.groupBox_3)
110    self.line_Stage.setGeometry(QtCore.QRect(60, 20, 51, 20))
111    self.line_Stage.setObjectName("line_Stage")
112    self.label_10 = QtWidgets.QLabel(self.groupBox_3)
113    self.label_10.setGeometry(QtCore.QRect(10, 20, 47, 13))
114    self.label_10.setObjectName("label_10")
115    self.line_Nodes = QtWidgets.QLineEdit(self.groupBox_3)
116    self.line_Nodes.setGeometry(QtCore.QRect(60, 50, 51, 20))
117    self.line_Nodes.setObjectName("line_Nodes")
118    self.label_11 = QtWidgets.QLabel(self.groupBox_3)
119    self.label_11.setGeometry(QtCore.QRect(10, 50, 47, 13))
120    self.label_11.setObjectName("label_11")
121    self.line_ld = QtWidgets.QLineEdit(self.groupBox_3)
122    self.line_ld.setGeometry(QtCore.QRect(170, 20, 51, 20))
123    self.line_ld.setObjectName("line_ld")
124    self.label_17 = QtWidgets.QLabel(self.groupBox_3)
125    self.label_17.setGeometry(QtCore.QRect(120, 20, 47, 13))
126    self.label_17.setObjectName("label_17")
127    self.line_Lf = QtWidgets.QLineEdit(self.groupBox_3)
128    self.line_Lf.setGeometry(QtCore.QRect(170, 50, 51, 20))
129    self.line_Lf.setObjectName("line_Lf")
130    self.label_18 = QtWidgets.QLabel(self.groupBox_3)
131    self.label_18.setGeometry(QtCore.QRect(120, 50, 47, 13))
132    self.label_18.setObjectName("label_18")
133    self.groupBox_4 = QtWidgets.QGroupBox(self.centralwidget)
134    self.groupBox_4.setGeometry(QtCore.QRect(590, 580, 131, 81))
135    self.groupBox_4.setObjectName("groupBox_4")
136    self.line_L = QtWidgets.QLineEdit(self.groupBox_4)
137    self.line_L.setEnabled(True)
138    self.line_L.setGeometry(QtCore.QRect(70, 50, 51, 20))
139    self.line_L.setMouseTracking(False)
140    self.line_L.setFocusPolicy(QtCore.Qt.NoFocus)
141    self.line_L.setReadOnly(True)
142    self.line_L.setObjectName("line_L")
143    self.label_14 = QtWidgets.QLabel(self.groupBox_4)
144    self.label_14.setGeometry(QtCore.QRect(10, 50, 47, 13))
145    self.label_14.setObjectName("label_14")
146    self.label_13 = QtWidgets.QLabel(self.groupBox_4)
147    self.label_13.setGeometry(QtCore.QRect(10, 20, 55, 16))
148    self.label_13.setObjectName("label_13")
149    self.line_ab = QtWidgets.QLineEdit(self.groupBox_4)
150    self.line_ab.setEnabled(True)
```

```

151     self.line_ab.setGeometry(QtCore.QRect(70, 20, 51, 20))
152     self.line_ab.setMouseTracking(False)
153     self.line_ab.setFocusPolicy(QtCore.Qt.NoFocus)
154     self.line_ab.setReadOnly(True)
155     self.line_ab.setObjectName("line_ab")
156     self.groupBox_5 = QtWidgets.QGroupBox(self.centralwidget)
157     self.groupBox_5.setGeometry(QtCore.QRect(260, 580, 161, 81))
158     self.groupBox_5.setObjectName("groupBox_5")
159     self.line_E = QtWidgets.QLineEdit(self.groupBox_5)
160     self.line_E.setGeometry(QtCore.QRect(60, 20, 81, 20))
161     self.line_E.setObjectName("line_E")
162     self.label_19 = QtWidgets.QLabel(self.groupBox_5)
163     self.label_19.setGeometry(QtCore.QRect(10, 20, 47, 13))
164     self.label_19.setObjectName("label_19")
165     self.line_In = QtWidgets.QLineEdit(self.groupBox_5)
166     self.line_In.setGeometry(QtCore.QRect(60, 50, 81, 20))
167     self.line_In.setObjectName("line_In")
168     self.label_20 = QtWidgets.QLabel(self.groupBox_5)
169     self.label_20.setGeometry(QtCore.QRect(10, 50, 47, 16))
170     self.label_20.setObjectName("label_20")
171     MainWindow.setCentralWidget(self.centralwidget)
172     self.menubar = QtWidgets.QMenuBar(MainWindow)
173     self.menubar.setGeometry(QtCore.QRect(0, 0, 1334, 21))
174     self.menubar.setObjectName("menubar")
175     MainWindow.setMenuBar(self.menubar)
176     self.statusbar = QtWidgets.QStatusBar(MainWindow)
177     self.statusbar.setObjectName("statusbar")
178     MainWindow.setStatusBar(self.statusbar)
179
180     self.retranslateUi(MainWindow)
181     self.tabWidget.setCurrentIndex(1)
182     QtCore.QMetaObject.connectSlotsByName(MainWindow)
183
184     def retranslateUi(self, MainWindow):
185         _translate = QtCore.QCoreApplication.translate
186         MainWindow.setWindowTitle(_translate("MainWindow", "ISE - Tablestacas"))
187         self.pushButton.setText(_translate("MainWindow", "Calcular"))
188         self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab), _translate("
189     MainWindow", "Graficos (Incrementos)"))
190         self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab_2), _translate(
191     "MainWindow", "Graficos"))
192         self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab_3), _translate(
193     "MainWindow", "Tablas"))
194         self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab_4), _translate(
195     "MainWindow", "Log File"))
196         self.groupBox.setTitle(_translate("MainWindow", "Propiedades del suelo"))
197         self.label_3.setText(_translate("MainWindow", "<html><head/><body><p>\
198     gamma [kN/m<span style=\" vertical-align:super;\>3</span></p></body></html>
199     "))
200         self.label.setText(_translate("MainWindow", "<html><head/><body><p>\phi\'
201     [^\{\\ circ}</p></body></html>"))
202         self.label_2.setText(_translate("MainWindow", "<html><head/><body><p>c\'
203     [kN/m<span style=\" vertical-align:super;\>2</span></p></body></html>"))

```

```

196     self.label_6.setText(_translate("MainWindow", "<html><head/><body><p>F<
span style=\" vertical-align:sub;\">k</span> [m]</p></body></html>"))
197     self.groupBox_2.setTitle(_translate("MainWindow", "Excavación"))
198     self.label_4.setText(_translate("MainWindow", "<html><head/><body><p>Prof
. exc. [m]</p></body></html>"))
199     self.predimButton.setText(_translate("MainWindow", "Predimensionamiento")
)
200     self.groupBox_3.setTitle(_translate("MainWindow", "Parametros del modelo"
))
201     self.label_10.setText(_translate("MainWindow", "<html><head/><body><p>
Etapas</p></body></html>"))
202     self.label_11.setText(_translate("MainWindow", "<html><head/><body><p>
Nodos</p></body></html>"))
203     self.label_17.setText(_translate("MainWindow", "<html><head/><body><p>\
lambda [m]</p></body></html>"))
204     self.label_18.setText(_translate("MainWindow", "<html><head/><body><p>L<
span style=\" vertical-align:sub;\">f</span> [m]</p></body></html>"))
205     self.groupBox_4.setTitle(_translate("MainWindow", "Predimensionamiento"))
206     self.label_14.setText(_translate("MainWindow", "<html><head/><body><p>L<
span style=\" vertical-align:sub;\">>pred</span> [m]</p></body></html>"))
207     self.label_13.setText(_translate("MainWindow", "<html><head/><body><p>(a+
b) [m]</p></body></html>"))
208     self.groupBox_5.setTitle(_translate("MainWindow", "Propiedades de la
Estructura"))
209     self.label_19.setText(_translate("MainWindow", "<html><head/><body><p>E [
kN/m<span style=\" vertical-align:super;\">2</span>]</p></body></html>"))
210     self.label_20.setText(_translate("MainWindow", "<html><head/><body><p>I [
kN/m<span style=\" vertical-align:super;\">3</span>]</p></body></html>"))
211
212
213 if __name__ == "__main__":
214     import sys
215     app = QtWidgets.QApplication(sys.argv)
216     MainWindow = QtWidgets.QMainWindow()
217     ui = Ui_MainWindow()
218     ui.setupUi(MainWindow)
219     MainWindow.show()
220     sys.exit(app.exec_())

```

### A.3. main.py

```

1 import sys
2 import SP_SEI as SP
3 SP.parent = sys.modules[__name__]
4
5
6 import numpy as np
7 from scipy.interpolate import interp1d
8 import pandas as pd
9 from appSPSEI_ui import *
10 from PyQt5.QtGui import QFont, QIcon, QColor
11 from PyQt5 import QtCore, QtGui, QtWidgets
12 from PyQt5.QtCore import Qt
13 from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg as FigureCanvas
14 from matplotlib.backends.backend_qt5agg import NavigationToolbar2QT as
    NavigationToolbar
15 from matplotlib.figure import Figure
16
17
18
19 import matplotlib.ticker as ticker
20 import matplotlib.pyplot as plt
21 from PyQt5 import QtWidgets
22 from PyQt5.QtWidgets import (QApplication, QDialog, QPushButton, QTableWidgetItem,
    QTableWidgetItem, QAbstractItemView, QHeaderView,
23     QMenu,
24     QActionGroup, QAction, QMessageBox, QVBoxLayout,
    QWidget)
25
26
27 class TableModel(QtCore.QAbstractTableModel):
28     def __init__(self, data):
29         super(TableModel, self).__init__()
30         self._data = data
31
32     def data(self, index, role):
33         if role == Qt.DisplayRole:
34             value = self._data.iloc[index.row()][index.column()]
35             if isinstance(value, float):
36                 return "%.3f" % value
37             if isinstance(value, str):
38                 return "%s" % value
39
40             return value
41         if role == Qt.TextAlignmentRole:
42             value = self._data.iloc[index.row()][index.column()]
43
44             if isinstance(value, int) or isinstance(value, float):
45                 # Align right, vertical middle.
46                 return Qt.AlignVCenter + Qt.AlignRight

```

```

47
48     def rowCount(self, index):
49         # The length of the outer list.
50         return self._data.shape[0]
51
52     def columnCount(self, index):
53         # The following takes the first sub-list, and returns
54         # the length (only works if all rows are an equal length)
55         return self._data.shape[1]
56     def headerData(self, section, orientation, role):
57         if role == Qt.DisplayRole:
58             return str(self._data.columns[section])
59         if orientation == Qt.Vertical:
60             return str(self._data.index[section])
61
62 #

```

---

```

63 class MainWindow(QtWidgets.QMainWindow, Ui_MainWindow):
64     def __init__(self, *args, **kwargs):
65         QtWidgets.QMainWindow.__init__(self, *args, **kwargs)
66         self.setupUi(self)
67         self.fig_dict_1 = {}
68         self.fig_dict_2 = {}
69
70         fig_1 = Figure()
71         fig_2 = Figure()
72         self.addmpl_1(fig_1)
73         self.addmpl_2(fig_2)
74         self.figures()
75         self.tabWidget.removeTab(0)
76 # =====
77 # VALORES INICIALES DE LOS QLINEEDIT
78 # =====
79         self.line_phi.setText(str(phi))
80         self.line_c.setText(str(c))
81         self.line_gamma.setText(str(gm))
82         self.line_Fk.setText(str(Fk))
83
84         self.line_E.setText(str(E))
85         self.line_In.setText(str(I))
86
87         self.line_Prof.setText(str(pfe))
88
89         self.line_ab.setText(str(a+b))
90         self.line_L.setText(str(l_pred))
91
92         self.line_Stage.setText(str(st))
93         self.line_Nodes.setText(str(nd_t))
94
95         self.line_ld.setText(str(n))
96         self.line_Lf.setText(str(L))
97

```

```

98 #Events
99     self.listWidget.itemClicked.connect(self.changeFig)
100     self.listWidget.currentItemChanged.connect(self.changeFig)
101     self.pushButton.clicked.connect(self.on_click)
102     self.predimButton.clicked.connect(self.predimB)
103
104 #Initial Values
105 def compute_initial_var (self, phi, c, h, gm, E, I):
106     phi1 = phi
107     c1 = c
108     gm1 = gm
109     E1 = E
110     I1 = I
111     h1 = h
112     return phi1, c1, h1, gm1, E1, I1
113
114 #Adding a plot
115 def addmpl_1 (self, fig):
116     self.canvas_1 = FigureCanvas(fig)
117     self.verticalLayout_1.addWidget(self.canvas_1)
118     self.canvas_1.draw()
119     self.toolbar_1 = NavigationToolbar(self.canvas_1, self.widget_1,
coordinates = True)
120     self.verticalLayout_1.addWidget(self.toolbar_1)
121
122 def addmpl_2 (self, fig):
123     self.canvas_2 = FigureCanvas(fig)
124     self.verticalLayout_2.addWidget(self.canvas_2)
125     self.canvas_2.draw()
126     self.toolbar_2 = NavigationToolbar(self.canvas_2, self.widget_2,
coordinates = True)
127     self.verticalLayout_2.addWidget(self.toolbar_2)
128
129 def addtable (self, tab):
130     self.table = self.tablageneral
131     df = pd.DataFrame()
132     zi2 = SP.z
133     zi2 = np.insert(zi2, 0, 0)
134     df ['Prof [m]'] = zi2.flatten()
135     df ['qb [kN]'] = p_var['qb_'+str(tab)]
136     df ['qf [kN]'] = self.tab_c(p_var['qf_'+str(tab)], tab, 1)
137     df ['Qb [kN]'] = self.tab_c(p_var['Xb_'+str(tab)], tab, 0)
138     df ['Qf [kN]'] = self.tab_c(p_var['Xf_'+str(tab)], tab, 1)
139     df ['V [kN]'] = -p_var['V_'+str(tab)]
140     df ['M [kN m]'] = -p_var['M_'+str(tab)]
141     df ['def [mm]'] = p_var['d_'+str(tab)]*-1000
142
143     data = df
144
145     self.model = TableModel(data)
146     self.table.setModel(self.model)
147
148 def tab_c (self, X, tab, y):

```

```
149     if y == 0:
150         X = np.insert(X,0,0)
151     elif y == 1:
152         for it in range (tab+1):
153             X = np.insert(X,0,0)
154     elif y == 2:
155         for it in range (tab-1):
156             X = np.insert(X,0,0)
157     return X.flatten ()
158
159 # Remove plot
160 def rmmpl_1 (self ,):
161     self.verticalLayout_1.removeWidget(self.canvas_1)
162     self.canvas_1.close ()
163     self.verticalLayout_1.removeWidget(self.toolbar_1)
164     self.toolbar_1.close ()
165
166 def rmmpl_2 (self ,):
167     self.verticalLayout_2.removeWidget(self.canvas_2)
168     self.canvas_2.close ()
169     self.verticalLayout_2.removeWidget(self.toolbar_2)
170     self.toolbar_2.close ()
171
172 #Add list
173 def addfig (self , name, fig1 , fig2):
174     self.fig_dict_1[name] = fig1
175     self.fig_dict_2[name] = fig2
176     self.listWidget.addItem(name)
177
178 #Change fig list
179 def changefig(self , item):
180     text = item.text ()
181     tab = int(text[6:])
182     self.rmmpl_1()
183     self.rmmpl_2()
184     self.addmpl_1(self.fig_dict_1[text])
185     self.addmpl_2(self.fig_dict_2[text])
186     self.addtable(tab)
187
188 def on_click(self):
189     phi3 = self.line_phi.text ()
190     phi3 = float(phi3)
191
192     c3 = self.line_c.text ()
193     c3 = float (c3)
194
195     gm3 = self.line_gm.text ()
196     gm3 = float(gm3)
197
198     E3 = self.line_E.text ()
199     E3 = float(E3)
200
201     I3 = self.line_I.text ()
```

```

202     I3 = float(I3)
203
204     h3 = self.line_h.text()
205     h3 = float(h3)
206
207     self.listWidget.clear()
208     self.rmmpl_1()
209     self.rmmpl_2()
210     self.figures(phi3, c3, h3, gm3, E3, I3)
211
212     def predimB(self):
213         phi = float(self.line_phi.text())
214         c = float(self.line_c.text())
215         pfe = float(self.line_Prof.text())
216         gamma = float(self.line_gamma.text())
217
218         a, b = SP.pred(phi, c, pfe, gamma)
219         l_pred = np.round(pfe + a + b, 2)
220
221         self.line_ab.setText(str(a+b))
222         self.line_L.setText(str(l_pred))
223
224
225     def Corr_graf (X, phase):
226         for i in range (nd_t):
227             if i < phase:
228                 X = np.delete(X, 0, axis = 0)
229         return X
230
231     def figures (self):
232         zi = SP.z
233         zi2 = zi
234         zi2 = np.insert(zi, 0, 0)
235         zi = zi.flatten()
236         zi2 = zi2.flatten()
237         def Corr_def (d):
238             x= (d[0]-d[1])/2
239             df = d[0] + x
240             return df
241
242         st = SP.st
243         i=1
244         while i <= st:
245             p_var[ 'qi_'+str(i) ] = SP.v[ 'qi_'+str(i) ].flatten()
246             p_var[ 'qi_'+str(i) ] = np.insert(p_var[ 'qi_'+str(i) ], 0, 0)
247             p_var[ 'qbi_'+str(i) ] = SP.v[ 'qbi_'+str(i) ].flatten()
248             p_var[ 'qbi_'+str(i) ] = np.insert(p_var[ 'qbi_'+str(i) ], 0, 0)
249             p_var[ 'qfi_'+str(i) ] = MainWindow.Corr_graf(SP.v[ 'qfi_'+str(i) ], i).
flatten()
250             p_var[ 'Xi_'+str(i) ] = SP.v[ 'Xi_'+str(i) ].flatten()
251             p_var[ 'Xbi_'+str(i) ] = SP.v[ 'Xbi_'+str(i) ].flatten()
252             p_var[ 'Xfi_'+str(i) ] = MainWindow.Corr_graf(SP.v[ 'Xfi_'+str(i) ], i).
flatten()

```

```

253     p_var[ 'ZF_'+str(i) ] = MainWindow.Corr_graf(SP.v[ 'ZF_'+str(i) ], i).
flatten()
254     p_var[ 'Mi_'+str(i) ] = SP.v[ 'Mi_'+str(i) ].flatten()
255     p_var[ 'Mi_'+str(i) ] = np.insert(p_var[ 'Mi_'+str(i) ], 0, 0)
256     p_var[ 'Vi_'+str(i) ] = SP.v[ 'Vi_'+str(i) ].flatten()
257     p_var[ 'Vi_'+str(i) ] = np.insert(p_var[ 'Vi_'+str(i) ], 0, 0)
258     p_var[ 'di_'+str(i) ] = SP.v[ 'di_'+str(i) ].flatten()
259     p_var[ 'di_'+str(i) ] = np.insert(p_var[ 'di_'+str(i) ], 0, 0)
260
261
262     p_var[ 'q_'+str(i) ] = SP.v[ 'q_'+str(i) ].flatten()
263     p_var[ 'q_'+str(i) ] = np.insert(p_var[ 'q_'+str(i) ], 0, 0)
264     p_var[ 'qb_'+str(i) ] = SP.v[ 'qb_'+str(i) ].flatten()
265     p_var[ 'qb_'+str(i) ] = np.insert(p_var[ 'qb_'+str(i) ], 0, 0)
266     p_var[ 'qf_'+str(i) ] = MainWindow.Corr_graf(SP.v[ 'qf_'+str(i) ], i).
flatten()
267     p_var[ 'X_'+str(i) ] = SP.v[ 'X_'+str(i) ].flatten()
268     p_var[ 'Xb_'+str(i) ] = SP.v[ 'Xb_'+str(i) ].flatten()
269     p_var[ 'Xf_'+str(i) ] = MainWindow.Corr_graf(SP.v[ 'Xf_'+str(i) ], i).
flatten()
270     p_var[ 'M_'+str(i) ] = -SP.v[ 'M_'+str(i) ].flatten()
271     p_var[ 'M_'+str(i) ] = np.insert(p_var[ 'M_'+str(i) ], 0, 0)
272     p_var[ 'Mstep_'+str(i) ] = -SP.v[ 'INTERM_'+str(i) ](XX)
273     plots[ 'Mstep_'+str(i) ]=p_var[ 'Mstep_'+str(i) ]
274     p_var[ 'V_'+str(i) ] = -SP.v[ 'V_'+str(i) ].flatten()
275     p_var[ 'V_'+str(i) ] = np.insert(p_var[ 'V_'+str(i) ], 0, 0)
276     p_var[ 'Vstep_'+str(i) ] = -SP.v[ 'INTERV_'+str(i) ](XX)
277     plots[ 'Vstep_'+str(i) ]=p_var[ 'Vstep_'+str(i) ]
278     p_var[ 'd_'+str(i) ] = SP.v[ 'd_'+str(i) ].flatten()
279     df = Corr_def(p_var[ 'd_'+str(i) ])
280     p_var[ 'd_'+str(i) ] = np.insert(p_var[ 'd_'+str(i) ], 0, df)
281     p_var[ 'dstep_'+str(i) ] = SP.v[ 'INTERD_'+str(i) ](XX)
282     plots[ 'dstep_'+str(i) ]= p_var[ 'dstep_'+str(i) ]
283     i+=1
284
285 # =====
286 #   SUAVIZADO DE LAS CURVAS
287 # =====
288     mult = 1
289     npoints = nd_t * mult
290     mult_q = 100
291     npoints_q = nd_t * mult_q
292     mult_vm = 1
293     npoints_vm = nd_t * mult_vm
294     for i in range (st):
295         plots[ 'npointsf_'+str(i+1) ] = (nd_t-i-1)*mult
296         plots[ 'npointsfq_'+str(i+1) ] = (nd_t-i-1)*mult_q
297         plots[ 'npointsfvm_'+str(i+1) ] = (nd_t-i-1)*mult_vm
298
299     i=1
300     while i <=st:
301         plots[ 'yqfi_'+str(i) ] = np.linspace(p_var[ 'ZF_'+str(i) ].min(),p_var[ '
ZF_'+str(i) ].max(),plots[ 'npointsf_'+str(i) ])

```

```

302     ff = interp1d(p_var[ 'ZF_'+str(i) ],p_var[ 'qfi_'+str(i) ], kind='
quadratic' )
303     plots[ 'xqfi_'+str(i) ] = ff(plots[ 'yqfi_'+str(i) ])
304
305     plots[ 'yqbi_'+str(i) ] = np.linspace(zi2.min(),zi2.max(),npoints)
306     ff = interp1d(zi2,p_var[ 'qbi_'+str(i) ], kind='quadratic' )
307     plots[ 'xqbi_'+str(i) ] = ff(plots[ 'yqbi_'+str(i) ])
308
309     plots[ 'yXfi_'+str(i) ] = np.linspace( p_var[ 'ZF_'+str(i) ].min(), p_var
[ 'ZF_'+str(i) ].max() ,plots[ 'npointsf_'+str(i) ])
310     ff = interp1d( p_var[ 'ZF_'+str(i) ],p_var[ 'Xfi_'+str(i) ], kind='
quadratic' )
311     plots[ 'xXfi_'+str(i) ] = ff(plots[ 'yXfi_'+str(i) ])
312
313     plots[ 'yXbi_'+str(i) ] = np.linspace(zi.min(),zi.max(),npoints)
314     ff = interp1d(zi,p_var[ 'Xbi_'+str(i) ], kind='quadratic' )
315     plots[ 'xXbi_'+str(i) ] = ff(plots[ 'yXbi_'+str(i) ])
316
317     plots[ 'yVi_'+str(i) ] = np.linspace(zi2.min(),zi2.max(),npoints)
318     ff = interp1d(zi2,p_var[ 'Vi_'+str(i) ], kind='quadratic' )
319     plots[ 'xVi_'+str(i) ] = ff(plots[ 'yVi_'+str(i) ])
320
321     plots[ 'yMi_'+str(i) ] = np.linspace(zi2.min(),zi2.max(),npoints)
322     ff = interp1d(zi2,p_var[ 'Mi_'+str(i) ], kind='quadratic' )
323     plots[ 'xMi_'+str(i) ] = ff(plots[ 'yMi_'+str(i) ])
324
325     plots[ 'ydi_'+str(i) ] = np.linspace(zi2.min(),zi2.max(),npoints)
326     ff = interp1d(zi2,p_var[ 'di_'+str(i) ], kind='quadratic' )
327     plots[ 'xdi_'+str(i) ] = ff(plots[ 'ydi_'+str(i) ])
328
329     #-----
330     plots[ 'yqf_'+str(i) ] = np.linspace(p_var[ 'ZF_'+str(i) ].min(),p_var[ '
ZF_'+str(i) ].max() ,plots[ 'npointsf_'+str(i) ])
331     ff = interp1d(p_var[ 'ZF_'+str(i) ],p_var[ 'qf_'+str(i) ], kind='
quadratic' )
332     plots[ 'xqf_'+str(i) ] = ff(plots[ 'yqf_'+str(i) ])
333
334     plots[ 'yqb_'+str(i) ] = np.linspace(zi2.min(),zi2.max(),npoints)
335     ff = interp1d(zi2,p_var[ 'qb_'+str(i) ], kind='quadratic' )
336     plots[ 'xqb_'+str(i) ] = ff(plots[ 'yqb_'+str(i) ])
337
338     plots[ 'yqb' ] = np.linspace(0,L,npoints)
339     plots[ 'yqbB' ] = np.linspace(0,L,npoints_q)
340     plots[ 'xqab' ] = -SP.K_calc(npoints_q,gm,c,plots[ 'yqbB' ], phi, 'Ka')
341     plots[ 'xq0b' ] = -SP.K_calc(npoints_q,gm,c,plots[ 'yqbB' ], phi, 'K0')
342     plots[ 'xqpb' ] = -SP.K_calc(npoints_q,gm,c,plots[ 'yqbB' ], phi, 'Kp')
343     plots[ 'yqff_'+str(i) ] = np.linspace(0,L-n*i,npoints)
344     plots[ 'yqffF_'+str(i) ] = np.linspace(0,L-n*i,npoints_q)
345     plots[ 'xqaf_'+str(i) ] = SP.K_calc(npoints_q,gm,c,plots[ 'yqffF_'+str(i
) ], phi, 'Ka')
346     plots[ 'xq0f_'+str(i) ] = SP.K_calc(npoints_q,gm,c,plots[ 'yqffF_'+str(i
) ], phi, 'K0')

```

```

347     plots['xqpf_'+str(i)] = SP.K_calc(npoints_q, gm, c, plots['yqffF_'+str(i)
    ]), phi, 'Kp')
348
349     plots['yXf_'+str(i)] = np.linspace(p_var['ZF_'+str(i)].min(), p_var['
ZF_'+str(i)].max(), plots['npointsf_'+str(i)])
350     ff = interp1d(p_var['ZF_'+str(i)], p_var['Xf_'+str(i)], kind='
quadratic')
351     plots['xXf_'+str(i)] = ff(plots['yXf_'+str(i)])
352
353     plots['yXb_'+str(i)] = np.linspace(zi.min(), zi.max(), npoints)
354     ff = interp1d(zi, p_var['Xb_'+str(i)], kind='quadratic')
355     plots['xXb_'+str(i)] = ff(plots['yXb_'+str(i)])
356
357     plots['yV_'+str(i)] = np.linspace(zi2.min(), zi2.max(), npoints_vm)
358     ff = interp1d(zi2, p_var['V_'+str(i)], kind='quadratic')
359     plots['xV_'+str(i)] = ff(plots['yV_'+str(i)])
360
361     plots['yM_'+str(i)] = np.linspace(zi2.min(), zi2.max(), npoints_vm)
362     ff = interp1d(zi2, p_var['M_'+str(i)], kind='quadratic')
363     plots['xM_'+str(i)] = ff(plots['yM_'+str(i)])
364
365     plots['yd_'+str(i)] = np.linspace(zi2.min(), zi2.max(), npoints)
366     ff = interp1d(zi2, p_var['d_'+str(i)], kind='quadratic')
367     plots['xd_'+str(i)] = ff(plots['yd_'+str(i)])
368
369     i+=1
370
371 # =====
372 # GENERACIÓN DE FIGURAS
373 # =====
374     mark = ','
375     m_size = 0
376
377     i=1
378     while i <= st:
379         figs['fig1_'+str(i)] = Figure()
380         (figs['qi_'+str(i)], figs['Xi_'+str(i)], figs['Vi_'+str(i)], figs['
Mi_'+str(i)], figs['di_'+str(i)]) = figs['fig1_'+str(i)].subplots(1,5,sharey=
True)
381
382         figs['qi_'+str(i)].plot(plots['xqfi_'+str(i)], plots['yqfi_'+str(i)],
plots['xqbi_'+str(i)], plots['yqbi_'+str(i)], marker =mark, markersize =
m_size)
383         figs['qi_'+str(i)].set_ylabel('z [m]')
384         figs['qi_'+str(i)].set_xlabel('q [kN]')
385         figs['qi_'+str(i)].grid(True)
386         figs['fig1_'+str(i)].gca().invert_yaxis()
387
388         figs['Xi_'+str(i)].set_xlim(np.min(plots['xXbi_'+str(i)]), np.max([np.
max(plots['xXbi_'+str(i)]), np.max(plots['xXfi_'+str(i)])]))
389         figs['Xi_'+str(i)].quiver(np.zeros([plots['npointsf_'+str(i)],1]),
plots['yXfi_'+str(i)], plots['xXfi_'+str(i)], np.zeros([plots['npointsf_'+str(i)
]),1]), angles='xy', scale_units='xy', scale=1, color='blue', headwidth = 5)

```

```

390     figs[ 'Xi_'+str(i) ].quiver(np.zeros([npoints,1]),plots[ 'yXbi_'+str(i)
],plots[ 'xXbi_'+str(i) ],np.zeros([npoints,1]), angles='xy', scale_units='xy',
    scale=1, color='orange', headwidth = 5)
391     figs[ 'Xi_'+str(i) ].set_xlabel( 'Q [kN]' )
392     figs[ 'Xi_'+str(i) ].grid(True)
393
394     figs[ 'Vi_'+str(i) ].plot(plots[ 'xVi_'+str(i) ], plots[ 'yVi_'+str(i) ],
marker =mark,markersize = m_size)
395     figs[ 'Vi_'+str(i) ].set_xlabel( 'V [kN]' )
396     figs[ 'Vi_'+str(i) ].grid(True)
397
398     figs[ 'Mi_'+str(i) ].plot(plots[ 'xMi_'+str(i) ], plots[ 'yMi_'+str(i) ],
marker =mark,markersize = m_size)
399     figs[ 'Mi_'+str(i) ].set_xlabel( 'M [kNm]' )
400     figs[ 'Mi_'+str(i) ].grid(True)
401
402     figs[ 'di_'+str(i) ].plot(plots[ 'xdi_'+str(i) ]*-1000, plots[ 'ydi_'+str(
i) ], marker =mark,markersize = m_size)
403     figs[ 'di_'+str(i) ].set_xlabel( '\delta [mm]' )
404     figs[ 'di_'+str(i) ].grid(True)
405     figs[ 'fig1_'+str(i) ].set_tight_layout(True)
406
407     #-----
408     figs[ 'fig2_'+str(i) ] = Figure()
409     (figs[ 'q_'+str(i) ], figs[ 'X_'+str(i) ], figs[ 'V_'+str(i) ], figs[ 'M_'+
str(i) ], figs[ 'd_'+str(i) ]) = figs[ 'fig2_'+str(i) ].subplots(1,5,sharey=True)
410
411     figs[ 'q_'+str(i) ].plot(plots[ 'xqab' ], plots[ 'yqbB' ],plots[ 'xq0b' ],
plots[ 'yqbB' ],plots[ 'xqpb' ],plots[ 'yqbB' ], marker =mark,markersize = m_size,
    linestyle = 'dotted', color = 'black')
412     figs[ 'q_'+str(i) ].plot(plots[ 'xqaf_'+str(i) ], plots[ 'yqffF_'+str(i) ]+
n*i, plots[ 'xq0f_'+str(i) ],plots[ 'yqffF_'+str(i) ]+n*i, plots[ 'xqpf_'+str(i) ],
plots[ 'yqffF_'+str(i) ]+n*i, marker =mark,markersize = m_size, linestyle = '
    dotted', color = 'black')
413     figs[ 'q_'+str(i) ].plot(plots[ 'xqf_'+str(i) ], plots[ 'yqf_'+str(i) ],
plots[ 'xqb_'+str(i) ], plots[ 'yqb_'+str(i) ], marker =mark,markersize = m_size)
414     figs[ 'q_'+str(i) ].set_xlim(np.min(plots[ 'xqb_'+str(i) ]*1.2), np.max(
plots[ 'xqf_'+str(i) ])*1.2)
415     figs[ 'q_'+str(i) ].set_ylabel( 'z [m]' )
416     figs[ 'q_'+str(i) ].set_xlabel( 'q [kN]' )
417     figs[ 'q_'+str(i) ].grid(True)
418     figs[ 'fig2_'+str(i) ].gca().invert_yaxis()
419
420
421     figs[ 'X_'+str(i) ].set_xlim(np.min(plots[ 'xXb_'+str(i) ]),np.max(plots[
'xXf_'+str(i) ]))
422     figs[ 'X_'+str(i) ].quiver(plots[ 'xXf_'+str(i) ],plots[ 'yXf_'+str(i) ],-
plots[ 'xXf_'+str(i) ],np.zeros([plots[ 'npointsf_'+str(i) ],1]), angles='xy',
    scale_units='xy', scale=1, color='blue', headwidth = 5)
423     figs[ 'X_'+str(i) ].quiver(plots[ 'xXb_'+str(i) ],plots[ 'yXb_'+str(i) ],-
plots[ 'xXb_'+str(i) ],np.zeros([npoints,1]), angles='xy', scale_units='xy',
    scale=1, color='orange', headwidth = 5)
424     def major_formatter(x, pos):

```

```

425         label = str(-np.round(x,0)) if x < 0 else str(np.round(x,0))
426         return label
427     figs['X_'+str(i)].xaxis.set_major_formatter(major_formatter)
428
429     figs['X_'+str(i)].set_xlabel('Q [kN]')
430     figs['X_'+str(i)].grid(True)
431
432     figs['V_'+str(i)].plot(p_var['Vstep_'+str(i)], XX, marker =mark,
markersize = m_size)#, linestyle = 'dotted', color = 'gray')
433     figs['V_'+str(i)].set_xlabel('V [kN]')
434     figs['V_'+str(i)].grid(True)
435
436     figs['M_'+str(i)].plot(p_var['Mstep_'+str(i)], XX, marker =mark,
markersize = m_size)
437     figs['M_'+str(i)].set_xlabel('M [kNm]')
438     figs['M_'+str(i)].grid(True)
439
440     figs['d_'+str(i)].plot(p_var['dstep_'+str(i)]*-1000, XX, marker =mark
, markersize = m_size)
441     figs['d_'+str(i)].set_xlabel('\delta [mm]')
442     figs['d_'+str(i)].grid(True)
443     figs['fig2_'+str(i)].set_tight_layout(True)
444
445     i+=1
446 # =====
447 # ANADIENDO LAS ETAPAS A LISTA
448 # =====
449     i=1
450     while i<= st:
451         self.addfig('Etapa '+str(i), figs['fig1_'+str(i)], figs['fig2_'+str(i)
))]
452         i+=1
453
454     return
455 # =====
456 # EJECUTANDO EL PROGRAMA
457 # =====
458 if __name__ == "__main__":
459     app = QtWidgets.QApplication([])
460     window = MainWindow()
461     window.show()
462     sys.exit(app.exec_())

```



# Bibliografía

- Bojorque Iñiguez, Jaime (dic. de 2011). «Métodos para el análisis de la estabilidad de pendientes». En: *MASKANA 2*.
- Clayton, C.R.I. y J. Milititsky (1986). «Earth Pressure and Earth-Retaining Structures». En: Surrey University Press. ISBN: 0-903384-51-5.
- Conte, E., Antonello Troncone y M. Vena (abr. de 2017). «A method for the design of embedded cantilever retaining walls under static and seismic loading». En: *Géotechnique 67*, págs. 1-9.
- Coulomb, C. A. (1776). «Essai sur une application des règles des maximis et minimis à quelques problèmes de statique relatifs al architecture». En: Mem. Math. Phys. Acad. R. Sci.
- Das, B. M. (2012). «Fundamentos de ingeniería de cimentaciones». En: CENGANE Learning.
- Davisson, M. T. y S. Prakash (1963). «A review of soil pile behavior». En: *Highway Research Record*.
- Eskandari, Leila y Behzad Kalantari (ene. de 2011). «Basic Types of Sheet Pile Walls and Their Application in the Construction Industry—a Review». En: *Electronic Journal of Geotechnical Engineering EJGE 16*.
- FEMA-P-2091 (2020). *A Practical Guide to Soil-Structure Interaction*. Applied Technology Council.
- Holloway, Redmond (1991). «Foundation and Structural Problems: Solved by Microcomputer». En: Blackwell Science Inc. Cap. 5. Release-deformation flexibility analysis, págs. 34-46. ISBN: 978-0632029242.
- Jaky, J. (1944). «The coeficient of earth preassure at rest». En: *Hungarian Architects and Engrs*.
- Mayne, P. W. y F. H. Kulhawy (1982). « $K_0$  - OCR relationships in soil». En: *Journal of the Geotechnical Engineering Division*.
- Padfield, C. J. y R. J. Mair (1984). «Design of retaining walls embedded in stiff clays». En: *London, UK: Construction Industry Research and Information Association (Ciria)*.
- Palmer, L. A. y J. B. Thompson (1948). «The Earth Pressure and Deflection Along the Embedded Lengths of Piles Subjected to Lateral Thrusts». En: *Proceedings of 2nd Int. Conf. on Soil Mechanics and Foundation Engineering*.
- Rankine, W. (1857). «On the stability of loose earth». En: vol. 147. Philosophical Transactions of the Royal Society of London.
- Rowe, P. W. (1952). «Anchored sheet-pile walls». En: *Proc, ICE 1*.
- Singh, A. P. y K. Chatterjee (2020). «Lateral earth pressure and bending moment on sheet pile walls due to uniform surcharge». En: *Geomechanics and Engineering*.
- Terzaghi, K. (1954). «Anchored bulkheads». En: *Trans. ASCE 119*.
- Winkler, E. (1867). «Die Lehre Von Elasticitaet Und Festigkeit». En: *1 st Edn., H. Dominicus*.

Zeevaert, L (1980). «Interacción suelo-estructura de cimentaciones superficiales y profundas, sujetas a cargas estáticas y sísmicas». En: Editorial Limusa México. Cap. Fuerza estática horizontal en pilas y pilotes, págs. 69-76. ISBN: 968-18-1237-9.