Contact information

Web http://www.kuleuven.be/bwm/edt
Email mattias.schevenels@bwk.kuleuven.be
Phone +32 16 32 16 82
Fax +32 16 32 19 88
Address Mattias Schevenels
Department of Civil Engineering, K.U.Leuven
Kasteelpark Arenberg 40, B-3001 Leuven, Belgium

Revision history

June 2008 Third printing Extended with theoretical background and numerical examples.
March 2009 Fourth printing Revised for EDT Version 2.1 Build 15.
May 2009 Fifth printing Revised for EDT Version 2.1 Build 16.
June 2009 Sixth printing Revised for EDT Version 2.1 Build 17.
October 2009 Eighth printing Revised for EDT Version 2.2 Build 19.

Copyright

(C) 2007–2009 K.U.Leuven, Structural Mechanics

The software described in this document is distributed under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this document may be reproduced in any form by print, photoprint, microfilm, or any other means without written permission from the publisher.

Acknowledgements

The ElastoDynamics Toolbox has been developed with the financial support of the Education Council and the Research Council of K.U.Leuven, the Research Foundation - Flanders, and the Institute for the Promotion of Innovation through Science and Technology in Flanders.
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Fourier and Hankel transformation algorithms</td>
<td>59</td>
</tr>
<tr>
<td>6.1</td>
<td>Fourier transformations</td>
<td>60</td>
</tr>
<tr>
<td>6.2</td>
<td>Hankel transformations</td>
<td>65</td>
</tr>
<tr>
<td>7</td>
<td>Site amplification</td>
<td>67</td>
</tr>
<tr>
<td>7.1</td>
<td>Modelling site amplification</td>
<td>68</td>
</tr>
<tr>
<td>7.2</td>
<td>Transfer functions</td>
<td>71</td>
</tr>
<tr>
<td>7.3</td>
<td>Linear site response analysis</td>
<td>72</td>
</tr>
<tr>
<td>7.4</td>
<td>Equivalent linear site response analysis</td>
<td>76</td>
</tr>
<tr>
<td>7.5</td>
<td>Frequency dependent equivalent linear site response analysis</td>
<td>81</td>
</tr>
<tr>
<td>8</td>
<td>Surface waves</td>
<td>89</td>
</tr>
<tr>
<td>8.1</td>
<td>Transcendental eigenvalue problem</td>
<td>90</td>
</tr>
<tr>
<td>8.2</td>
<td>Quadratic eigenvalue problem</td>
<td>94</td>
</tr>
<tr>
<td>8.3</td>
<td>Wavenumber content of the Green’s function</td>
<td>97</td>
</tr>
<tr>
<td>9</td>
<td>Forced vibration problems</td>
<td>101</td>
</tr>
<tr>
<td>9.1</td>
<td>Point load</td>
<td>102</td>
</tr>
<tr>
<td>9.2</td>
<td>Line load</td>
<td>107</td>
</tr>
<tr>
<td>9.3</td>
<td>Spatially harmonic line load</td>
<td>109</td>
</tr>
<tr>
<td>9.4</td>
<td>Spatially harmonic traction varying in $x$-direction</td>
<td>110</td>
</tr>
<tr>
<td>9.5</td>
<td>Spatially harmonic traction varying in $x$- and $y$-direction</td>
<td>111</td>
</tr>
<tr>
<td>9.6</td>
<td>Disk load</td>
<td>112</td>
</tr>
<tr>
<td>10</td>
<td>Functions — By category</td>
<td>115</td>
</tr>
<tr>
<td>10.1</td>
<td>Direct stiffness method</td>
<td>115</td>
</tr>
<tr>
<td>10.2</td>
<td>Thin layer method</td>
<td>115</td>
</tr>
<tr>
<td>10.3</td>
<td>Integral transformations</td>
<td>116</td>
</tr>
<tr>
<td>10.4</td>
<td>Amplification problems</td>
<td>116</td>
</tr>
<tr>
<td>10.5</td>
<td>Surface waves</td>
<td>116</td>
</tr>
<tr>
<td>10.6</td>
<td>Forced vibration problems</td>
<td>117</td>
</tr>
<tr>
<td>10.7</td>
<td>Visualization</td>
<td>118</td>
</tr>
<tr>
<td>11</td>
<td>Functions — Alphabetical list</td>
<td>119</td>
</tr>
<tr>
<td>References</td>
<td></td>
<td>229</td>
</tr>
</tbody>
</table>
Chapter 1

Getting started

This chapter describes how to start using the ElastoDynamics Toolbox (EDT) for MATLAB. The chapter is organized as follows:

**About the ElastoDynamics Toolbox (p. 1)**

The ElastoDynamics Toolbox for MATLAB is briefly introduced.

**Obtaining and installing EDT (p. 1)**

This section explains where to get EDT and how to install it.

**Terms of use (p. 3)**

The terms and conditions to use of EDT are summarized. It is shown how to refer to EDT in scientific publications.

**Scope and structure of this document (p. 3)**

The aim of this user’s guide is delineated and the organization of the text is clarified.

1.1 About the ElastoDynamics Toolbox

The ElastoDynamics Toolbox is a MATLAB toolbox to model wave propagation in layered media. It is based on the direct stiffness method and the thin layer method [9, 12]. The toolbox can be used to solve a variety of problems governed by wave propagation in the soil:

**Site amplification** Site amplification is an important issue in the assessment of the seismic hazard at sites where the top soil layers are particularly soft. In such cases, the seismic motion at the surface can be much higher than the outcrop motion due to resonance of the soft layers.
**Surface waves** Surface waves are the natural modes of vibration of a layered medium. These waves travel along the free surface, or along an interface between layers, while they decay exponentially with depth. Surface waves are dispersive: their phase velocity is frequency dependent due to the variation of the dynamic soil properties with depth. This is the basis of the Spectral Analysis of Surface Waves (SASW) method, where the dispersion curve of one or more surface waves is used to identify the dynamic soil properties as a function of depth.

**Forced vibration problems** The calculation of the forced response of the soil due to a unit load (i.e. the Green’s functions of the soil) is the basis of the boundary element method, which can be used to calculate foundation impedance curves or to model dynamic soil-structure interaction.

EDT consists of a large number of MATLAB functions that can be categorized as follows:

1. Shape functions and stiffness matrices used in the direct stiffness method.
2. Shape functions and stiffness matrices used in the thin layer method.
3. Fourier and Hankel transformation algorithms.
4. Amplification problems.
5. Surface waves.
7. Visualization of a wave field.

The user can interact with EDT at a low level of abstraction (categories 1–3) or a high level of abstraction (categories 4–7). Due to this multi-level approach, the toolbox is suitable for educational purposes and for use in a research environment: the high level functions allow for an easy and efficient implementation of many common problems, while the low level functions facilitate customization and the development of new techniques.

The core of EDT is implemented in C++ using the MATLAB MEX interface in order to achieve both a seamless integration with MATLAB and a high numerical efficiency. The MEX-files are compiled with the GNU Compiler Collection (GCC), which has excellent cross-platform capabilities. As a result, the toolbox can be used both in a Windows and in a Linux environment.

### 1.2 Obtaining and installing EDT

EDT can be downloaded from the internet at [http://www.kuleuven.be/bwm/edt](http://www.kuleuven.be/bwm/edt). It is distributed as a ZIP archive, which should be extracted to a directory
on the hard disk, e.g. C:\Program Files\MATLAB\R2009b\toolbox\edt. After extraction of the ZIP archive, this directory should contain a number of P-files, M-files, and MEX-files.

The EDT directory must be added to the MATLAB path to make the toolbox functions available in MATLAB:
- In MATLAB, click on ‘File’, ‘Set Path...’
- Click on ‘Add Folder’ and select the EDT directory.
- Save the path and close the dialog window.

A license file license.dat is required to use the toolbox. This file must reside in the EDT directory or one of its parent directories.

You can either buy a permanent license, or request a free trial license for one month. Type edtlicense at the MATLAB command prompt for instructions on how to obtain a permanent or trial license.

1.3 Terms of use

EDT is distributed under a license agreement, and can be used or copied only under the terms of the license agreement. An important aspect of the license agreement is the commitment to include a reference to EDT in scientific publications presenting results obtained with EDT.

How to refer to EDT in scientific publications?

Scientific publications presenting results obtained with EDT must include a proper reference to EDT. In order to refer to EDT, please add the following article to the list of references:

http://dx.doi.org/10.1016/j.cageo.2008.10.012

1.4 Scope and structure of this document

The present document is the user’s guide to EDT version 2.2 build 19. It is a combination of a reference guide, providing an overview of all functions of EDT, and a tutorial, presenting a set of examples to illustrate the use of EDT. This document is not meant as a textbook on elastodynamic wave propagation.
The theoretical background of wave propagation is only considered where this is necessary to define the functionality of EDT in a unique way.

This document is composed of the following chapters:

**Chapter 1. Getting started (p. 1)**
The aim of this chapter is to get the user started with EDT and the accompanying user’s guide. The installation procedure of EDT is explained, the terms of use of EDT are clarified, and the scope and structure of the user’s guide are discussed.

**Chapter 2. Definitions and conventions (p. 7)**
In this chapter, the definitions and conventions used in EDT are explained. An overview of the relevant material properties is given, the definition of a layered soil profile is explained, and the integral transformations used in EDT are discussed.

**Chapter 3. Waves in layered media (p. 17)**
EDT uses the direct stiffness method and the thin layer method to model waves in layered media. Both methods are based on a decomposition of the wave field into a series of problems governed by plane wave propagation. In this chapter, the solution procedure for problems of plane wave propagation is outlined, and the decomposition of one-dimensional, two-dimensional, and three-dimensional wave fields is addressed.

**Chapter 4. The direct stiffness method (p. 33)**
This chapter focuses on the calculation of stiffness matrices and shape functions for layered media according to the direct stiffness method. The shape functions are exact solutions of the wave equation in the frequency-wavenumber domain. As a result, wave propagation is treated exactly in the direct stiffness method.

**Chapter 5. The thin layer method (p. 51)**
This chapter focuses on the calculation of stiffness matrices and shape functions for layered media according to the thin layer method, which is an alternative to the direct stiffness method. In the thin layer method, linear shape functions are used, and wave propagation is therefore treated in an approximative way.

**Chapter 6. Fourier and Hankel transformation algorithms (p. 59)**
The direct stiffness method and the thin layer method are based on the solution of the wave equation in the frequency-wavenumber domain. This chapter describes the integral transformation algorithms included in EDT to transform a function from the time-space domain to the frequency-wavenumber domain and vice versa.

**Chapter 7. Site amplification (p. 67)**
This chapter addresses the solution of site amplification problems with EDT. It is shown how to compute the amplification of an incident shear wave or dilatational wave in a layered medium for various angles of incidence.

**Chapter 8. Surface waves (p. 89)**
This chapter focuses on surface waves in layered media. The computation
of surface waves involves the solution of an eigenvalue problem, which is transcendental if the layered medium is modelled with the direct stiffness method and quadratic if the thin layer method is used. EDT includes algorithms to solve the eigenvalue problem in both cases.

**Chapter 9. Forced vibration problems (p. 101)**
In this chapter, the focus is on the calculation of the forced response of a layered medium due to a unit load (i.e. the Green’s functions of the medium) with the direct stiffness method. EDT includes a large number of algorithms to compute the 1D, 2D, 2.5D, and 3D Green’s functions of layered media.

**Chapter 10. Functions — By category (p. 115)**
This chapter gives an overview of all functions in EDT, organized by category.

**Chapter 11. Functions — Alphabetical list (p. 119)**
This chapter consists of an alphabetical list of the functions in EDT. The syntax, the input and output arguments, and the use of each function are described in detail. The information provided in this chapter is also accessible at the MATLAB prompt through the `help` command.

The chapters 3–9 are elaborated as independent entities, that is, they can be read in any order without diminishing the capacity to understand the remaining chapters. It is therefore unnecessary to read the chapters on the low level functions (chapters 3–6) for users interested in the high level functions (chapters 7–9) of EDT.
Chapter 2

Definitions and conventions

This chapter gives an overview of the definitions, conventions, and notations used in this user’s guide and in EDT. The chapter is organized as follows:

Elastodynamic wave propagation (p. 7)
A brief introduction on elastodynamic wave propagation in layered media is given and the relevant material properties are discussed.

Definition of a soil profile (p. 9)
The definition of a layered soil profile in EDT is explained.

Coordinate systems (p. 11)
The coordinate systems used to define a wave field in EDT are introduced.

Governing equations (p. 13)
The equations governing elastodynamic wave propagation are briefly reviewed. These equations relate the displacements, strains, and stresses in an elastodynamic medium.

Integral transformations (p. 11)
The Fourier transformations, Hankel transformations, and Fourier series expansions relating the time-space domain and the frequency-wavenumber domain are defined.

2.1 Elastodynamic wave propagation

In the field of seismic wave propagation, the soil is most often modelled as an elastodynamic medium. The wave field in an elastodynamic medium can be expressed as a superposition of plane waves [2]. Two types of plane waves exist:
dilatational and shear waves. In the dilatational (or: longitudinal, irrotational, primary, P) waves, the particles move parallel to the wave propagation direction. In the shear (or: transverse, equivoluminal, rotational, secondary, S) waves, the particles move perpendicular to the wave propagation direction. The contribution of the shear waves to the wave field can be further decomposed into a component in a horizontal plane (SH-waves) and a component in a vertical plane (SV-waves).

In a horizontally layered elastic medium, a (partial) conversion of energy between P-waves and SV-waves occurs at the interfaces between layers. However, the propagation of P-waves and SV-waves is completely uncoupled from the propagation of SH-waves. As a result, all problems of elastodynamic wave propagation in layered media can be decomposed into a problem governed by P-SV-wave propagation and a problem governed by SH-wave propagation, which can be solved independently. This strategy results in a more efficient solution and is therefore followed in EDT.

The phase velocities are $C_s$ for the shear waves and $C_p$ for the dilatational waves. The phase $C_s$ and $C_p$ are related to the Lamé constants $\lambda$ and $\mu$ and the density $\rho$:

\[
C_s = \sqrt{\frac{\mu}{\rho}} \quad (2.1)
\]
\[
C_p = \sqrt{\frac{\lambda + 2\mu}{\rho}} \quad (2.2)
\]

The Lamé constants $\lambda$ and $\mu$ are related to the Young’s modulus $E$ and the Poisson’s ratio $\nu$:

\[
\lambda = \frac{E\nu}{(1 + \nu)(1 - 2\nu)} \quad (2.3)
\]
\[
\mu = \frac{E}{2(1 + \nu)} \quad (2.4)
\]

The ratio $s = C_s/C_p$ of the shear and dilatational wave velocities only depends on the Poisson’s ratio $\nu$:

\[
s = \sqrt{\frac{1 - 2\nu}{2 - 2\nu}} \quad (2.5)
\]

Or, equivalently:

\[
\nu = \frac{2s^2 - 1}{2s^2 - 2} \quad (2.6)
\]

As waves propagate through the medium, their amplitude decreases. This attenuation is due to geometrical damping and material damping. Geometrical
or radiation damping is due to the expansion of the wavefronts, resulting in the spreading of energy over an increasing area. In a homogeneous halfspace, the geometrical attenuation of waves due to point sources is proportional to \( r^n \), with \( r \) the travel distance and \( n = -0.5 \) for Rayleigh waves, \( n = -1 \) for body waves at depth, and \( n = -2 \) for body waves along the surface [20]. Geometrical attenuation is not directly related to the material properties of the medium. Material damping is related to the dissipation of energy. Energy is dissipated through several mechanisms, such as friction between solid particles in the skeleton and relative motion between the skeleton and the pore fluid. Their combined effect is represented by a material damping model. In structural mechanics, a viscous damping model is frequently used. The effect of viscous damping can be explained by means of a sprung mass system subjected to a harmonic displacement with unit amplitude. The energy dissipated by the system per cycle is proportional to the velocity. Viscous damping is consequently rate dependent: the energy dissipation increases with the frequency. In soil dynamics, material damping is usually assumed to be rate independent in the low frequency range. Rate independent material damping is sometimes referred to as hysteretic material damping [13, 14], although viscous damping also involves a hysteresis effect.

The correspondence principle [19, 24] is applied in EDT to model material damping. This principle states that a viscoelastic material can be modelled in the frequency domain as an equivalent elastic material with modified elastic constants. The application of the correspondence principle results in the use of complex Lamé constants \( \lambda^* \) and \( \mu^* \) and, consequently, complex wave velocities \( C_s^* \) and \( C_p^* \). The complex Lamé constants \( \lambda^* \) and \( \mu^* \) are given by:

\[
\lambda^* + 2\mu^* = (\lambda + 2\mu)(1 \pm 2D_p i) \tag{2.7}
\]

\[
\mu^* = \mu(1 \pm 2D_s i) \tag{2.8}
\]

where \( D_p \) and \( D_s \) represent the hysteretic material damping ratio for the dilatational waves and the shear waves, respectively. In the case of rate independent damping, the damping ratios \( D_p \) and \( D_s \) are frequency independent. In equations (2.7) and (2.8), the plus signs are selected if the frequency is positive, while the minus signs are used if the frequency is negative. For a zero frequency, the Lamé constants are not modified and remain real-valued.

### 2.2 Definition of a soil profile

In EDT, the soil is modelled as a horizontally layered medium assembled from (homogeneous) layer and halfspace elements. Both a halfspace element directed upwards and a halfspace element directed downwards are available.

Each element is characterized by a thickness \( h \), a (real-valued) shear wave velocity \( C_s \), a (real-valued) dilatational wave velocity \( C_p \), a shear damping ratio \( D_s \), a
dilatational damping ratio $D_p$, and a density $\rho$. In order to model a soil profile consisting of $n$ elements, these parameters must be specified in MATLAB as vectors with $n$ entries, where the $k$-th entry corresponds to the $k$-th element, starting with the top element. If a parameter assumes an identical value for all elements, it can alternatively be specified as a scalar.

The top element can be a halfspace element directed upwards, while the bottom element can be a halfspace element directed downwards. By convention, the thickness $h$ of a halfspace element directed upwards is $-\infty$, and the thickness $h$ of a halfspace element directed downwards is $\infty$.

**Example 2.1: Definition of a layered halfspace.**

![Figure 2.1: A soil profile consisting of two layers on a halfspace.](image)

This example demonstrates how to define a soil profile in MATLAB for use with EDT. A soil profile consisting of two homogeneous layers on a homogeneous halfspace is considered (figure 2.1). Table 2.1 gives an overview of the properties of the layers and the halfspace.

<table>
<thead>
<tr>
<th>Layer</th>
<th>$h$ [m]</th>
<th>$C_s$ [m/s]</th>
<th>$C_p$ [m/s]</th>
<th>$D_s$ [-]</th>
<th>$D_p$ [-]</th>
<th>$\rho$ [kg/m$^3$]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.97</td>
<td>143</td>
<td>286</td>
<td>0.03</td>
<td>0.03</td>
<td>1800</td>
</tr>
<tr>
<td>2</td>
<td>1.90</td>
<td>168</td>
<td>336</td>
<td>0.03</td>
<td>0.03</td>
<td>1800</td>
</tr>
<tr>
<td>3</td>
<td>$\infty$</td>
<td>259</td>
<td>518</td>
<td>0.03</td>
<td>0.03</td>
<td>1800</td>
</tr>
</tbody>
</table>

**Table 2.1: Properties of the layers and the halfspace.**

The following MATLAB code is used to define this soil profile:

```matlab
% SOIL PROPERTIES
h = [0.97 1.90 inf]; % Element thickness [m]
Cs = [143 168 259]; % Shear wave velocity [m/s]
Cp = [286 336 518]; % Dilatational wave velocity [m/s]
Ds = 0.03; % Shear damping ratio [-]
Dp = 0.03; % Dilatational damping ratio [-]
rho = 1800; % Density [kg/m^3]
```
The thickness $h$, shear wave velocity $C_s$, and dilatational wave velocity $C_p$ of the elements are defined by the vectors $h$, $C_s$, and $C_p$. These vectors have three entries, corresponding to the three elements in the soil profile, starting with the top layer. The shear damping ratio $D_s$, dilatational damping ratio $D_p$, and density $\rho$ are identical for all three elements and are defined by the scalars $D_s$, $D_p$, and $\rho$.

**Example 2.2: Definition of a homogeneous fullspace.**

![Figure 2.2: A homogeneous fullspace modelled with two halfspace elements.](image)

This example shows how to model a homogeneous fullspace with EDT. The fullspace is assembled from two halfspace elements: a halfspace element directed upwards and a halfspace element directed downwards:

```plaintext
SOIL PROPERTIES
h = [-inf inf]; % Element thickness [m]
C_s = 150; % Shear wave velocity [m/s]
C_p = 300; % Dilatational wave velocity [m/s]
D_s = 0.03; % Shear damping ratio [-]
D_p = 0.03; % Dilatational damping ratio [-]
rho = 1800; % Density [kg/m^3]
```

All material properties $C_s$, $C_p$, $D_s$, $D_p$, and $\rho$ are identical for both halfspace elements and are therefore defined as scalars.

### 2.3 Coordinate systems

For three-dimensional problems, either a Cartesian coordinate system $(x, y, z)$ or a cylindrical coordinate system $(r, \theta, z)$ is used in EDT. For two-dimensional problems, a Cartesian coordinate system $(x, y, z)$ is used and wave propagation takes place in the $(x, z)$-plane. Both the Cartesian and the cylindrical coordinate
system are right-handed. In both cases, the $z$-axis is vertical (i.e. perpendicular to the stratification of the medium) and points downwards.

Two types of coordinate systems are used: an element coordinate system and a global coordinate system. The former is used to describe the wave field in a single element, while the latter is used to describe the wave field in a (layered) medium.

![Figure 2.3: Definition of the element coordinate system for (a) a layer element, (b) a halfspace element directed downwards, and (c) a halfspace element directed upwards.](image)

The origin of the element coordinate system coincides with the upper boundary of the element for layer elements and for halfspace elements directed downwards. For halfspace elements directed upwards, the origin is located at the lower boundary of the element.

The origin of the global coordinate system is located at the soil’s surface, unless the soil profile contains a halfspace element directed upwards. If the soil profile contains such a halfspace element, the origin of the coordinate system is located at the lower boundary of this element.

Figures 2.3 and 2.4 illustrate the definition of the element coordinate system and the global coordinate system in various cases.
2.4 Governing equations

This section briefly reviews the equations relating the displacements, strains, and stresses in an elastodynamic medium. The equations are first formulated in a Cartesian coordinate system \((x, y, z)\) and then in a cylindrical coordinate system \((r, \theta, z)\). The Cartesian coordinate system is used to model two-dimensional wave propagation (in the \((x, z)\)-plane), while the cylindrical coordinate system is used to model three-dimensional wave propagation.

2.4.1 Cartesian coordinates

The components \(\epsilon_{ij}\) of the small strain tensor \(\epsilon\) are related to the displacements \(u\) by the following linearized strain-displacement relations:

\[
\begin{bmatrix}
\epsilon_{xx} \\
\epsilon_{yy} \\
\epsilon_{zz} \\
\epsilon_{xy} \\
\epsilon_{yz} \\
\epsilon_{zx}
\end{bmatrix} =
\begin{bmatrix}
\frac{\partial}{\partial x} & 0 & 0 \\
0 & \frac{\partial}{\partial y} & 0 \\
0 & 0 & \frac{\partial}{\partial z} \\
\frac{1}{2} \frac{\partial}{\partial y} & \frac{1}{2} \frac{\partial}{\partial z} & 0 \\
\frac{1}{2} \frac{\partial}{\partial z} & \frac{1}{2} \frac{\partial}{\partial x} & 0 \\
\frac{1}{2} \frac{\partial}{\partial x} & \frac{1}{2} \frac{\partial}{\partial y} & 0
\end{bmatrix}
\begin{bmatrix}
u_x \\
u_y \\
u_z
\end{bmatrix}
\]

(H2.9)

Hooke’s law relates the Cauchy stress tensor \(\sigma\) to the small strain tensor \(\epsilon\):

\[
\begin{bmatrix}
\sigma_{xx} \\
\sigma_{yy} \\
\sigma_{zz} \\
\sigma_{xy} \\
\sigma_{yz} \\
\sigma_{zx}
\end{bmatrix} =
\begin{bmatrix}
\lambda + 2\mu & \lambda & 0 & 0 & 0 \\
\lambda & \lambda + 2\mu & \lambda & 0 & 0 \\
0 & \lambda & \lambda + 2\mu & 0 & 0 \\
0 & 0 & 0 & 2\mu & 0 \\
0 & 0 & 0 & 0 & 2\mu \\
0 & 0 & 0 & 0 & \lambda + 2\mu
\end{bmatrix}
\begin{bmatrix}
\epsilon_{xx} \\
\epsilon_{yy} \\
\epsilon_{zz} \\
\epsilon_{xy} \\
\epsilon_{yz} \\
\epsilon_{zx}
\end{bmatrix}
\]

(2.10)

The dynamic equilibrium of the elastic medium is expressed as:

\[
\begin{bmatrix}
\frac{\partial}{\partial x} & 0 & 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} \\
0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\
0 & 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial x} & 0 & \frac{\partial}{\partial y} \\
\end{bmatrix}
\begin{bmatrix}
\sigma_{xx} \\
\sigma_{yy} \\
\sigma_{zz} \\
\sigma_{xy} \\
\sigma_{yz} \\
\sigma_{zx}
\end{bmatrix} + \rho \frac{\partial^2}{\partial t^2}
\begin{bmatrix}
u_x \\
u_y \\
u_z
\end{bmatrix}
\]

(2.11)

where \(\rho [b_x, b_y, b_z]^T = \rho \mathbf{b}\) are the body forces. In the direct stiffness method and the thin layer method, used in EDT, the body forces \(\rho \mathbf{b}\) are not taken into account.

The Navier equations are obtained by the introduction of equations (2.9) and (2.10) in the equilibrium equation (2.11). The Navier equations are subsequently
reformulated in terms of wave potentials instead of displacements and solved through integral transformations from the time-space domain to the frequency-wavenumber domain. The integral transformations are briefly reviewed in section 2.5. A more detailed discussion on the solution of the Navier equations is beyond the scope of this document but can be found in references [3, 25].

### 2.4.2 Cylindrical coordinates

In a cylindrical coordinate system, the linear strain-displacement relations are:

\[
\begin{bmatrix}
\varepsilon_{rr} \\
\varepsilon_{\theta\theta} \\
\varepsilon_{zz} \\
\varepsilon_{r\theta} \\
\varepsilon_{\theta z} \\
\varepsilon_{z r}
\end{bmatrix} = \begin{bmatrix}
\frac{\partial}{\partial r} & 0 & 0 \\
\frac{1}{r} \frac{\partial}{\partial \theta} & \frac{1}{r} & 0 \\
0 & 0 & \frac{\partial}{\partial z} \\
\frac{1}{r} \frac{\partial}{\partial \theta} & 0 & \frac{1}{r} \left( \frac{\partial^2}{\partial \theta^2} - \frac{1}{r^2} \right) \\
\frac{1}{r} \frac{\partial}{\partial \theta} & 0 & \frac{1}{r} \frac{\partial}{\partial z} \\
\frac{1}{r} \frac{\partial}{\partial \theta} & 0 & \frac{1}{r} \frac{\partial}{\partial z}
\end{bmatrix}
\begin{bmatrix}
u_r \\
u_\theta \\
u_z
\end{bmatrix}
\]

The constitutive equations are:

\[
\begin{bmatrix}
s_{rr} \\
s_{\theta\theta} \\
s_{zz} \\
s_{r\theta} \\
s_{\theta z} \\
s_{z r}
\end{bmatrix} = \begin{bmatrix}
\lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\
\lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\
\lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\
0 & 0 & 0 & 2\mu & 0 & 0 \\
0 & 0 & 0 & 0 & 2\mu & 0 \\
0 & 0 & 0 & 0 & 0 & 2\mu
\end{bmatrix}
\begin{bmatrix}
\varepsilon_{rr} \\
\varepsilon_{\theta\theta} \\
\varepsilon_{zz} \\
\varepsilon_{r\theta} \\
\varepsilon_{\theta z} \\
\varepsilon_{z r}
\end{bmatrix}
\]

The equilibrium equations are:

\[
\begin{bmatrix}
\frac{1}{r} + \frac{\partial}{\partial z} \\
\frac{1}{r} + \frac{\partial}{\partial z} & \frac{1}{r^2} & 0 & \frac{1}{r} \frac{\partial}{\partial \theta} & 0 & \frac{\partial}{\partial z} \\
0 & 0 & \frac{\partial}{\partial z} & 0 & \frac{1}{r} \frac{\partial}{\partial \theta} & \frac{1}{r} + \frac{\partial}{\partial z}
\end{bmatrix}
\begin{bmatrix}
s_{rr} \\
s_{\theta\theta} \\
s_{zz} \\
s_{r\theta} \\
s_{\theta z} \\
s_{z r}
\end{bmatrix} + \rho \begin{bmatrix}
b_r \\
b_\theta \\
b_z
\end{bmatrix} = \rho \left( \frac{\partial^2}{\partial t^2} \begin{bmatrix}
u_r \\
u_\theta \\
u_z
\end{bmatrix} + \begin{bmatrix}
\varepsilon_{rr} \\
\varepsilon_{\theta\theta} \\
\varepsilon_{zz} \\
\varepsilon_{r\theta} \\
\varepsilon_{\theta z} \\
\varepsilon_{z r}
\end{bmatrix} \right)
\]

Introduction of the constitutive equations and the strain-displacement relations in the equilibrium equations leads to the Navier equations.

### 2.5 Integral transformations

The direct stiffness method and the thin layer method are based on the solution of the wave equation in the frequency-wavenumber domain. Depending
on the geometry of the problem (two-dimensional or three-dimensional), the frequency-wavenumber domain is related to the time-space domain through Fourier transformations, Hankel transformations, and/or Fourier series expansions. These transformations are defined in various ways in the literature. The definitions used in EDT are given in the following subsections.

### 2.5.1 Fourier transformations

The relation between the time domain representation $f(t)$ and the frequency domain representation $\hat{f}(\omega)$ of a function $f$ is given by:

\[
\hat{f}(\omega) = F[f(t); \omega] = \int_{-\infty}^{\infty} e^{-i\omega t} f(t) \, dt
\]

\[
f(t) = F^{-1} [\hat{f}(\omega); t] = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\omega t} \hat{f}(\omega) \, d\omega
\]

where $F$ and $F^{-1}$ denote the forward and inverse Fourier transformation, respectively.

The relation between the space domain representation $f(x)$ and the wavenumber domain representation $\hat{f}(k_x)$ of a function $f$ is slightly different:

\[
\hat{f}(k_x) = F[f(x); k_x] = \int_{-\infty}^{\infty} e^{ik_x x} f(x) \, dx
\]

\[
f(x) = F^{-1} [\hat{f}(k_x); x] = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-ik_x x} \hat{f}(k_x) \, dk_x
\]

where $J_n$ is an $n$-th order Bessel function of the first kind.

### 2.5.2 Hankel transformations

The relation between the space domain representation $f(r)$ and the wavenumber domain representation $\hat{f}(k_r)$ of a function $f$ is given by:

\[
\hat{f}(k_r) = H_n[f(r); k_r] = \int_{0}^{\infty} r J_n(k_r r) f(r) \, dr
\]

\[
f(r) = H_n^{-1} [\hat{f}(k_r); r] = \int_{0}^{\infty} k_r J_n(k_r r) f(k_r) \, dk_r
\]

where $J_n$ is an $n$-th order Bessel function of the first kind and $H_n$ and $H_n^{-1}$ denote the forward and inverse Hankel transformation of order $n$, respectively.
2.5.3 Fourier series expansions

The Fourier series expansion of a function $f(\theta)$ is defined as:

$$f(\theta) = \sum_{n=0}^{\infty} f_c(n) \cos(n\theta) + f_s(n) \sin(n\theta)$$  \hspace{1cm} (2.21)

The Fourier coefficients $f_c(n)$ and $f_s(n)$ are given by:

$$f_c(n) = a_n \int_{0}^{2\pi} f(\theta) \cos(n\theta) \, d\theta$$ \hspace{1cm} (2.22)

$$f_s(n) = a_n \int_{0}^{2\pi} f(\theta) \sin(n\theta) \, d\theta$$ \hspace{1cm} (2.23)

where the coefficient $a_n$ is equal to $1/(2\pi)$ if $n = 0$ and $1/\pi$ if $n \neq 0$. If the function $f(\theta)$ is even, the Fourier coefficients $f_s(n)$ vanish and equation (2.21) reduces to a Fourier cosine series expansion. If the function $f(\theta)$ is odd, the Fourier coefficients $f_c(n)$ vanish and equation (2.21) reduces to a Fourier sine series expansion.
Chapter 3

Waves in layered media

EDT uses the direct stiffness method and the thin layer method to model waves in layered media. Both methods are based on a decomposition of the wave field into a series of problems governed by plane wave propagation. In this chapter, the solution procedure for problems of plane wave propagation is outlined, and the decomposition of one-dimensional, two-dimensional, and three-dimensional wave fields is addressed.

This chapter is organized as follows:

**Plane wave propagation (p. 18)**
This section focuses on the application of the direct stiffness method to problems governed by plane wave propagation. The propagation of plane waves is a fundamental problem in the direct stiffness method. More complicated cases, such as two-dimensional and three-dimensional problems, are decomposed into a series of problems governed by plane wave propagation.

**One-dimensional wave propagation (p. 23)**
In this section, it is pointed out that one-dimensional wave propagation is a special case of plane wave propagation. The procedure formulated in section 3.1 is therefore directly applicable to problems governed by one-dimensional wave propagation.

**Two-dimensional wave propagation (p. 24)**
This section addresses the application of the direct stiffness method to problems governed by two-dimensional wave propagation. Using a Fourier transformation from the horizontal coordinate $x$ to the horizontal wavenumber $k_x$, the problem is decomposed into a series of problems of plane wave propagation, which have to be solved according to the procedure outlined in section 3.1.
Three-dimensional wave propagation (p. 26)

Three-dimensional problems are tackled in a similar way as two dimensional problems: using integral transformations from the space domain to the wavenumber domain, the problem is decomposed into a series of problems of plane wave propagation, to which the procedure discussed in section 3.1 is applied.

The aim of this chapter is to give a general overview of the methodology followed in EDT to solve problems of wave propagation in layered media. The ingredients used in the solution procedure are discussed in more detail in the following chapters. Chapters 4 and 5 address the calculation of the stiffness matrices and shape functions for a layered medium according to the direct stiffness method and the thin layer method. Chapter 6 focuses on the algorithms used in EDT to evaluate the Fourier and Hankel transformations involved in the solution of two-dimensional and three-dimensional problems.

3.1 Plane wave propagation

The propagation of plane waves in a layered medium is a fundamental problem in the frame of the direct stiffness method. More complicated cases, such as two-dimensional and three-dimensional problems, are decomposed into a series of problems governed by plane wave propagation. This section therefore focuses on the application of the direct stiffness method to problems of plane wave propagation.

3.1.1 Harmonic plane wave propagation

In the direct stiffness method, the medium is subdivided into layer and halfspace elements corresponding to the physical stratification of the medium. Due to the use of exact solutions of the wave equation as shape functions, wave propagation is treated exactly and it is unnecessary to subdivide the layers into smaller elements. A further subdivision might only be necessary to account for the load distribution, since the loads must be applied at the interfaces between elements.

Figure 3.1 shows a wave field \( u(x, z, t) \) in a layered medium. The wave field \( u(x, z, t) \) is induced by a harmonic load distribution on the interfaces between elements. The load \( p^j(x, t) \) on the \( j \)-th interface reads as:

\[
p^j(x, t) = \tilde{p}^j e^{i(\omega t - k_z x)} + \tilde{p}^j* e^{-i(\omega t - k_z x)}
\]

(3.1)

The load vector \( p^j(x, t) \) represents the force per unit area, the vector \( \tilde{p}^j \) denotes the complex amplitude of the load at the \( j \)-th interface, and the vector \( \tilde{p}^j* \) denotes the complex conjugate of \( \tilde{p}^j \). The load varies harmonically in time and space (in
the $x$-direction). The variation is determined by the circular frequency $\omega$ and the horizontal wavenumber $k_x$.

The resulting wave field $\mathbf{u}(x,z,t)$ consists of plane waves travelling in the $(x,z)$-plane. In each layer $i$, four plane waves can be distinguished: an incoming and an outgoing P-wave (denoted by $P^I_i$ and $P^R_i$ in figure 3.1), and an incoming and an outgoing S-wave (denoted by $S^I_i$ and $S^R_i$ in figure 3.1). In the underlying halfspace, only two waves exist: an outgoing P-wave and an outgoing S-wave. The S-waves consist of a vertically polarized component (the SV-wave) and a horizontally polarized component (the SH-wave).

The angles $\theta_s$ and $\theta_p$ of the S-waves and the P-waves with respect to the $x$-axis are determined by the horizontal wavenumber $k_x$ and the wavenumbers $k_s = \omega/C_s$ and $k_p = \omega/C_p$:

\begin{align}
  k_x &= k_s \cos \theta_s \\
  k_x &= k_p \cos \theta_p
\end{align}

Figure 3.2 demonstrates the relation between the angles $\theta_s$ and $\theta_p$ and the wavelengths $\lambda_s = 2\pi/k_s$, $\lambda_s = 2\pi/k_p$, and $\lambda_p = 2\pi/k_p$.

The displacement field $\mathbf{u}(x,z,t)$ can be decomposed as follows:

$$
\mathbf{u}(x,z,t) = \tilde{\mathbf{u}}(z)e^{i(\omega t - k_x x)} + \tilde{\mathbf{u}}^*(z)e^{-i(\omega t - k_x x)}
$$

The vector $\tilde{\mathbf{u}}(z)$ denotes the complex amplitude of the wave field at depth $z$.

The direct stiffness method is based on the use of a stiffness matrix that relates the load amplitudes $\tilde{p}^j$ and the displacement amplitudes $\tilde{\mathbf{u}}(z)$ at the interfaces.
In order to obtain a symmetric stiffness matrix, use is made of a modified load vector $\tilde{p}^j$ and a modified displacement vector $\tilde{u}(z)$, obtained by a multiplication of the $z$-component with the imaginary unit $i$:

$$\tilde{p}^j = T \tilde{p}^j \quad (3.5)$$
$$\tilde{u}(z) = T \tilde{u}(z) \quad (3.6)$$

where:

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & i \end{bmatrix} \quad (3.7)$$

The loads $\tilde{p}^j$ and the displacements $\tilde{u}(z)$ at the interfaces are collected in two vectors $\tilde{P}$ and $\tilde{U}$ of length $3N$, where $N$ is the number of interfaces:

$$\tilde{P} = \{\tilde{p}_x^1, \tilde{p}_y^1, \tilde{p}_z^1, \tilde{p}_x^2, \tilde{p}_y^2, \tilde{p}_z^2, \ldots, \tilde{p}_x^N, \tilde{p}_y^N, \tilde{p}_z^N\}^T \quad (3.8)$$
$$\tilde{U} = \{\tilde{u}_x^1, \tilde{u}_y^1, \tilde{u}_z^1, \tilde{u}_x^2, \tilde{u}_y^2, \tilde{u}_z^2, \ldots, \tilde{u}_x^N, \tilde{u}_y^N, \tilde{u}_z^N\}^T \quad (3.9)$$

Here, $\tilde{p}^j_i$ and $\tilde{u}^j_i$ denote the loads and the displacements in direction $i$ at the $j$-th interface.

The vectors $\tilde{P}$ and $\tilde{U}$ are related by the stiffness matrix $\tilde{K}$ of the layered medium. The stiffness matrix $\tilde{K}$ is assembled from element stiffness matrices $\tilde{K}^e$ and depends on the layer thicknesses $h$, the material properties $C_s$, $C_p$, $D_s$, $D_p$, and $\rho$, the horizontal wavenumber $k_x$, and the frequency $\omega$. In EDT, the stiffness matrix $\tilde{K}$ is calculated either with the direct stiffness method, or with the thin layer method. The direct stiffness method is addressed in chapter 4, the thin layer method in chapter 5.

The dynamic equilibrium equation reads as follows:

$$\tilde{P} = \tilde{K} \tilde{U} \quad (3.10)$$

The equilibrium equation (3.10) allows for the calculation of the interface displacements $\tilde{U}$ due to the loads $\tilde{P}$ on the interfaces.
The amplitudes $\tilde{u}(z)$ of the displacements in the interior of the elements are subsequently calculated from the interface displacements $\tilde{U}$:

$$\tilde{u}(z) = \tilde{N}(z)\tilde{U} \quad (3.11)$$

Here, the matrix $\tilde{N}(z)$ contains the shape functions for the displacements. The shape functions $\tilde{N}(z)$ are assembled from element based shape functions $\tilde{N}^e(z)$ and $\tilde{B}^e(z)$ and depend on the same parameters as the stiffness matrix $\tilde{K}$. The calculation of the shape functions $\tilde{N}(z)$ with EDT is addressed in chapters 4 and 5. The displacement field $u(x, z, t)$ is computed from the modified displacement amplitudes $\tilde{u}(z)$ according to equations (3.4) and (3.6): The actual displacement amplitudes $\tilde{u}(z)$ at depth $z$ are obtained from the modified displacement amplitudes $\tilde{u}(z)$ by means of a multiplication of the $z$-component with a factor $-i$:

$$\tilde{u}(z) = T^{-1}\tilde{u}(z) \quad (3.12)$$

The displacement field $u(x, z, t)$ is eventually obtained according to equation (3.4):

$$u(x, z, t) = \tilde{u}(z)e^{i(\omega t - k_x x)} + \tilde{u}^*(z)e^{-i(\omega t - k_x x)} \quad (3.13)$$

The amplitudes $\tilde{t}(z)$ of the tractions $t(x, z, t) = \{\sigma_{xx}, \sigma_{yy}, \sigma_{zz}\}^T$ on a horizontal plane are calculated in a similar way:

$$\tilde{t}(z) = \tilde{B}(z)\tilde{U} \quad (3.14)$$

where $\tilde{B}(z)$ is a matrix containing the shape functions for the tractions. Chapters 4 and 4 address the calculation of the shape functions $\tilde{B}(z)$ with the direct stiffness method and the thin layer method, respectively. The vector $\tilde{t}(z)$ denotes the modified traction amplitudes, where the $z$-component is multiplied with the imaginary unit $i$. The actual traction amplitudes $\tilde{t}(z)$ are obtained from the following equation:

$$\tilde{t}(z) = T^{-1}\tilde{t}(z) \quad (3.15)$$

The tractions $t(x, z, t)$ are subsequently obtained as:

$$t(x, z, t) = \tilde{t}(z)e^{i(\omega t - k_x x)} + \tilde{t}^*(z)e^{-i(\omega t - k_x x)} \quad (3.16)$$

The horizontal components $\epsilon_{xx}$, $\epsilon_{yy}$, and $\epsilon_{xy}$ of the strain tensor $\epsilon(x, z, t)$ can be derived from equation (3.13) using the strain-displacement relations in equation (2.9):

$$\begin{cases}
\epsilon_{xx} \\
\epsilon_{yy} \\
\epsilon_{xy}
\end{cases} = L\tilde{u}(z)e^{i(\omega t - k_x x)} + L^*\tilde{u}^*(z)e^{-i(\omega t - k_x x)} \quad (3.17)$$
where:

\[
L = \begin{bmatrix}
-ik_x & 0 & 0 \\
0 & 0 & 0 \\
0 & -\frac{1}{2}k_x & 0
\end{bmatrix}
\]  
(3.18)

The vertical components \( \epsilon_{zz}, \epsilon_{zy}, \) and \( \epsilon_{zz} \) of the strain tensor \( \epsilon(x, z, t) \) and the horizontal components \( \sigma_{xx}, \sigma_{yy}, \) and \( \sigma_{xy} \) of the stress tensor \( \sigma(x, z, t) \) are finally calculated using Hooke’s law given in equation (2.10), which can be reformulated as:

\[
\begin{bmatrix}
\sigma_{xx} \\
\sigma_{yy} \\
\sigma_{xy} \\
\epsilon_{zz} \\
\epsilon_{zy} \\
\epsilon_{zx}
\end{bmatrix} =
\begin{bmatrix}
\frac{4\mu(\lambda+\mu)}{\lambda+2\mu} & \frac{2\lambda\mu}{\lambda+2\mu} & \frac{\lambda}{\lambda+2\mu} & 0 & 0 & 0 \\
\frac{2\lambda\mu}{\lambda+2\mu} & \frac{4\mu(\lambda+\mu)}{\lambda+2\mu} & \frac{\lambda}{\lambda+2\mu} & 0 & 0 & 0 \\
-\frac{\lambda+2\mu}{\lambda+2\mu} & -\frac{\lambda+2\mu}{\lambda+2\mu} & \frac{\lambda}{\lambda+2\mu} & 0 & 0 & 0 \\
0 & 0 & 0 & 2\mu & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{2\mu} & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{2\mu}
\end{bmatrix}
\begin{bmatrix}
\epsilon_{xx} \\
\epsilon_{yy} \\
\epsilon_{xy} \\
\sigma_{zz} \\
\sigma_{yz} \\
\sigma_{zx}
\end{bmatrix}
\]  
(3.19)

### 3.1.2 Transient plane wave propagation

Problems governed by transient plane wave propagation are solved in the frequency domain by means of a forward Fourier transformation from the time \( t \) to the circular frequency \( \omega \).

A transient wave field \( u(x, z, t) \) is considered. The wave field is induced by a transient load distribution on the interfaces between elements. The load on the \( j \)-th interface is denoted by the vector \( p^j(x, t) \). The vector \( p^j(x, t) \) is harmonic with respect to the \( x \)-coordinate:

\[
p^j(x, t) = \hat{p}^j(t)e^{-ik_x x} + \hat{p}^j*(t)e^{ik_x x}
\]  
(3.20)

where \( \hat{p}^j(t) \) is the time dependent complex amplitude of the load vector \( p^j(x, t) \) and \( k_x \) determines the harmonic variation of the load vector with respect to the \( x \)-coordinate. Equation (3.20) is transformed to the frequency domain by means of a forward Fourier transformation from the time \( t \) to the circular frequency \( \omega \):

\[
\hat{p}^j(x, \omega) = \hat{p}^j(\omega)e^{-ik_x x} + \hat{p}^j*(\omega)e^{ik_x x}
\]  
(3.21)

The displacement field \( u(x, z, t) \) induced by the loads \( p^j(x, t) \) is also harmonic with respect to the \( x \)-coordinate:

\[
u(x, z, t) = \hat{u}(z, t)e^{-ik_x x} + \hat{u}^*(z, t)e^{ik_x x}
\]  
(3.22)

This equation is also transformed to the frequency domain:

\[
\hat{u}(x, z, \omega) = \hat{u}(z, \omega)e^{-ik_x x} + \hat{u}^*(z, \omega)e^{ik_x x}
\]  
(3.23)
The $z$-components of the load vector $\tilde{p}^j(\omega)$ and the displacement vector $\tilde{u}(z,\omega)$ are multiplied with the imaginary unit $i$ in order to obtain the modified load vector $\tilde{p}^j(\omega)$ and the modified displacement vector $\tilde{u}(z,\omega)$.

For each frequency $\omega$, a time-harmonic problem has to be solved. The procedure outlined in subsection 3.1.1 is followed: the displacements $\tilde{u}(z,\omega)$ and the tractions $\tilde{t}(z,\omega)$ induced by the loads $\tilde{p}^j(\omega)$ are computed using the stiffness matrix $\tilde{K}$ and the shape functions $\tilde{N}(z)$ and $\tilde{B}(z)$.

The $z$-components of the resulting modified displacement and traction vectors $\tilde{u}(z,\omega)$ and $\tilde{t}(z,\omega)$ are multiplied with a factor $-i$ in order to obtain the actual displacement and traction vectors $\tilde{u}(z,\omega)$ and $\tilde{t}(z,\omega)$.

Next, the displacements $\tilde{u}(z,\omega)$ and tractions $\tilde{t}(z,\omega)$ are transformed to the time domain by means of an inverse Fourier transformation from the circular frequency $\omega$ to the time $t$. In the time domain, the resulting displacements and tractions are denoted by $\tilde{u}(z,t)$ and $\tilde{t}(z,t)$.

The displacement field $u(x,z,t)$ is subsequently obtained according to equation (3.22). Similarly, the tractions $t(x,z,t)$ are computed as:

$$t(x,z,t) = \tilde{t}(z,t)e^{-ik_x x} + \tilde{t}^*(z,t)e^{ik_x x} \quad (3.24)$$

The horizontal components $\epsilon_{xx}$, $\epsilon_{yy}$, and $\epsilon_{xy}$ of the strain tensor $\epsilon(x,z,t)$ are calculated as:

$$\begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{xy} \end{bmatrix} = L\tilde{u}(z,t)e^{-ik_x x} + L^*\tilde{u}^*(z,t)e^{ik_x x} \quad (3.25)$$

where the matrix $L$ is given by equation (3.18).

The vertical components $\epsilon_{xz}$, $\epsilon_{zy}$, and $\epsilon_{zz}$ of the strain tensor $\epsilon(x,z,t)$ and the horizontal components $\sigma_{xx}$, $\sigma_{yy}$, and $\sigma_{xy}$ of the stress tensor $\sigma(x,z,t)$ are finally calculated using Hooke’s law, according to equation (3.19).

### 3.2 One-dimensional wave propagation

One-dimensional wave propagation is a special case of plane wave propagation where the waves propagate in the vertical direction (i.e. perpendicular to the stratification of the medium). One-dimensional wave propagation occurs if the excitation $p^j(x,t)$ is invariable in the horizontal direction, i.e. if the horizontal wavenumber $k_x$ is equal to zero. The procedure to model plane wave propagation, discussed in the previous section, is therefore directly applicable to one-dimensional problems, provided that a zero horizontal wavenumber $k_z$ is used.
3.3 Two-dimensional wave propagation

A two-dimensional wave field can be considered as a superposition of plane waves. In the direct stiffness method, a two-dimensional problem is therefore decomposed into a series of problems governed by plane wave propagation. For each of these problems, the procedure discussed in section 3.1 is applied. The decomposition of the wave field is accomplished by means of a Fourier transformation from the horizontal coordinate $x$ to the horizontal wavenumber $k_x$.

3.3.1 Harmonic two-dimensional wave propagation

This subsection focuses on the modelling of a two-dimensional wave field in the $(x,z)$-plane, induced by a time-harmonic load distribution on the interfaces between elements. The load on the $j$-th interface is denoted by the vector $\mathbf{p}^j(x,t)$. The vector $\mathbf{p}^j(x,t)$ is time-harmonic:

$$\mathbf{p}^j(x,t) = \hat{\mathbf{p}}^j(x)e^{i\omega t} + \hat{\mathbf{p}}^j(x) e^{-i\omega t}$$  \hspace{1cm} (3.26)

where $\hat{\mathbf{p}}^j(x)$ is the space dependent complex amplitude of the load vector $\mathbf{p}^j(x,t)$ and $\omega$ is the circular excitation frequency. Equation (3.26) is transformed to the wavenumber domain by means of a forward Fourier transformation from the horizontal coordinate $x$ to the horizontal wavenumber $k_x$:

$$\hat{\mathbf{p}}^j(k_x,t) = \tilde{\mathbf{p}}^j(k_x)e^{i\omega t} + \tilde{\mathbf{p}}^j(k_x) e^{-i\omega t}$$  \hspace{1cm} (3.27)

The displacement field $\mathbf{u}(x,z,t)$ induced by the loads $\mathbf{p}^j(x,t)$ is also time-harmonic:

$$\mathbf{u}(x,z,t) = \hat{\mathbf{u}}(x,z)e^{i\omega t} + \hat{\mathbf{u}}^*(x,z)e^{-i\omega t}$$  \hspace{1cm} (3.28)

This equation is also transformed to the wavenumber domain:

$$\hat{\mathbf{u}}(k_x,z,t) = \tilde{\mathbf{u}}(k_x,z)e^{i\omega t} + \tilde{\mathbf{u}}^*(k_x,z)e^{-i\omega t}$$  \hspace{1cm} (3.29)

The $z$-components of the load vector $\hat{\mathbf{p}}^j(k_x)$ and the displacement vector $\hat{\mathbf{u}}(k_x,z)$ are multiplied with the imaginary unit $i$ in order to obtain the modified load vector $\tilde{\mathbf{p}}^j(k_x)$ and the modified displacement vector $\tilde{\mathbf{u}}(k_x,z)$.

For each horizontal wavenumber $k_x$, a problem of plane wave propagation has to be solved. The procedure outlined in subsection 3.1.1 is followed: the displacements $\tilde{\mathbf{u}}(k_x,z)$ and the tractions $\tilde{\mathbf{t}}(k_x,z)$ induced by the loads $\tilde{\mathbf{p}}^j(k_x)$ are computed using the stiffness matrix $\tilde{\mathbf{K}}$ and the shape functions $\tilde{\mathbf{N}}(z)$ and $\tilde{\mathbf{B}}(z)$.

The $z$-components of the resulting modified displacement and traction vectors $\tilde{\mathbf{u}}(k_x,z)$ and $\tilde{\mathbf{t}}(k_x,z)$ are multiplied with a factor $-i$ in order to obtain the actual displacement and traction vectors $\hat{\mathbf{u}}(k_x,z)$ and $\hat{\mathbf{t}}(k_x,z)$.
Next, the displacements $\tilde{u}(k_x, z)$ and tractions $\tilde{t}(k_x, z)$ are transformed to the space domain by means of an inverse Fourier transformation from the horizontal wavenumber $k_x$ to the horizontal coordinate $x$. In the space domain, the resulting displacements and tractions are denoted by $\hat{u}(x, z)$ and $\hat{t}(x, z)$.

The horizontal components $\hat{\epsilon}_{xx}$, $\hat{\epsilon}_{yy}$, and $\hat{\epsilon}_{xy}$ of the strain tensor $\hat{\epsilon}(x, z)$ are calculated by means of the following inverse Fourier transformation:

$$\begin{bmatrix} \hat{\epsilon}_{xx} \\ \hat{\epsilon}_{yy} \\ \hat{\epsilon}_{xy} \end{bmatrix} = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-ik_x x} L \tilde{u}(k_z, z) \, dk_z \tag{3.30}$$

where the matrix $L$ is given by equation (3.18).

The vertical components $\hat{\epsilon}_{zx}$, $\hat{\epsilon}_{zy}$, and $\hat{\epsilon}_{zz}$ of the strain tensor $\hat{\epsilon}(x, z)$ and the horizontal components $\hat{\sigma}_{xx}$, $\hat{\sigma}_{yy}$, and $\hat{\sigma}_{xy}$ of the stress tensor $\hat{\sigma}(x, z)$ can be calculated using Hooke’s law, according to equation (3.19).

The time dependent displacement field $u(x, z, t)$ is subsequently obtained according to equation (3.28). Similarly, the time dependent strains $\epsilon(x, z, t)$ and stresses $\sigma(x, z, t)$ are computed as:

$$\sigma(x, z, t) = \hat{\sigma}(x, z) e^{-i\omega t} + \hat{\sigma}^*(x, z) e^{i\omega t} \tag{3.31}$$

$$\epsilon(x, z, t) = \hat{\epsilon}(x, z) e^{-i\omega t} + \hat{\epsilon}^*(x, z) e^{i\omega t} \tag{3.32}$$

### 3.3.2 Transient two-dimensional wave propagation

Problems governed by transient two-dimensional wave propagation are solved by means of a double Fourier transformation, from the time-space domain to the frequency-wavenumber domain.

A transient wave field $u(x, z, t)$ is considered. The wave field is induced by a transient load distribution on the interfaces between elements. The load on the $j$-th interface is denoted by the vector $p^j(x, t)$. The load vector $p^j(x, t)$ is transformed to the frequency-space domain by means of a forward Fourier transformation from the time $t$ to the frequency $\omega$. In the frequency-space domain, the load vector is denoted by $\hat{p}^j(x, \omega)$. This transformation is followed by a forward Fourier transformation from the horizontal coordinate $x$ to the horizontal wavenumber $k_x$. The resulting load vector in the frequency-wavenumber domain is denoted by $\tilde{p}^j(k_x, \omega)$.

The displacement field $u(x, z, t)$ is transformed in a similar way. In the frequency-space domain, the displacement field is denoted by $\hat{u}(x, z, \omega)$. In the frequency-wavenumber domain, it is denoted by $\tilde{u}(k_x, z, \omega)$.
The $z$-components of the load vector $\tilde{p}_j(k_x, \omega)$ and the displacement vector $\tilde{u}(k_x, z, \omega)$ are multiplied with the imaginary unit $i$ in order to obtain the modified load vector $\tilde{p}_j(k_x, \omega)$ and the modified displacement vector $\tilde{u}(k_x, z, \omega)$.

For each frequency $\omega$ and each horizontal wavenumber $k_x$, a problem of plane wave propagation has to be solved. The procedure outlined in subsection 3.1.1 is followed: the displacements $\tilde{u}(k_x, z, \omega)$ and the tractions $\tilde{t}(k_x, z, \omega)$ induced by the loads $\tilde{p}_j(k_x, \omega)$ are computed using the stiffness matrix $\tilde{K}$ and the shape functions $\tilde{N}(z)$ and $\tilde{B}(z)$.

The $z$-components of the resulting modified displacement and traction vectors $\tilde{u}(k_x, z, \omega)$ and $\tilde{t}(k_x, z, \omega)$ are multiplied with a factor $-i$ in order to obtain the actual displacement and traction vectors $\tilde{u}(k_x, z, \omega)$ and $\tilde{t}(k_x, z, \omega)$.

Next, the displacements $\tilde{u}(k_x, z, \omega)$ and tractions $\tilde{t}(k_x, z, \omega)$ are transformed to the space domain by means of an inverse Fourier transformation from the horizontal wavenumber $k_x$ to the horizontal coordinate $x$. In the space domain, the resulting displacements and tractions are denoted by $\hat{u}(x, z, \omega)$ and $\hat{t}(x, z, \omega)$.

The horizontal components $\hat{\varepsilon}_{xx}$, $\hat{\varepsilon}_{yy}$, and $\hat{\varepsilon}_{xy}$ of the strain tensor $\hat{\varepsilon}(x, z, \omega)$ are calculated by means of the following inverse Fourier transformation:

$$
\begin{pmatrix}
\hat{\varepsilon}_{xx} \\
\hat{\varepsilon}_{yy} \\
\hat{\varepsilon}_{xy}
\end{pmatrix} = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-ik_x x} L \tilde{u}(k_x, z, \omega) dk_x
$$

(3.33)

where the matrix $L$ is given by equation (3.18).

The vertical components $\hat{\varepsilon}_{zx}$, $\hat{\varepsilon}_{zy}$, and $\hat{\varepsilon}_{zz}$ of the strain tensor $\hat{\varepsilon}(x, z)$ and the horizontal components $\hat{\sigma}_{xx}$, $\hat{\sigma}_{yy}$, and $\hat{\sigma}_{xy}$ of the stress tensor $\hat{\sigma}(x, z)$ can be calculated using Hooke’s law, according to equation (3.19).

Finally, the displacements $\hat{u}(x, z, \omega)$, strains $\hat{\varepsilon}(x, z, \omega)$, and stresses $\hat{\sigma}(x, z, \omega)$ are transformed to the time-space domain by means of an inverse Fourier transformation from the circular frequency $\omega$ to the time $t$.

### 3.4 Three-dimensional wave propagation

A three-dimensional wave field can be considered as a superposition of plane waves. In the direct stiffness method, a three-dimensional problem is therefore decomposed into a series of problems governed by plane wave propagation. For each of these problems, the procedure discussed in section 3.1 is applied. Three-dimensional problems are formulated in a cylindrical coordinate system $(r, \theta, z)$, and the decomposition of the wave field is accomplished by means of a Fourier series expansion from the circumferential coordinate $\theta$ to the circumferential wavenumber...
n, followed by a Hankel transformation from the radial coordinate \( r \) to the radial wavenumber \( k_r \).

In the following subsections, the transformation from the time-space domain to the frequency-wavenumber domain is elaborated for wave fields that are either symmetric or antisymmetric with respect to the plane \( \theta = 0 \). Asymmetric problems can be solved by means of a decomposition into a symmetric and an antisymmetric component.

### 3.4.1 Harmonic three-dimensional wave propagation

In this subsection, a three-dimensional wave field is considered. The wave field is modelled in cylindrical coordinates \( (r, \theta, z) \). It is induced by a harmonic load distribution on the interfaces between elements. The load on the \( j \)-th interface is denoted by the vector \( \hat{p}^j(r, \theta, t) \). This vector is time-harmonic:

\[
\hat{p}^j(r, \theta, t) = \hat{p}^j(r, \theta)e^{i\omega t} + \hat{p}^j(r, \theta)e^{-i\omega t}
\]  

(3.34)

where \( \hat{p}^j(r, \theta) \) is the space dependent amplitude of the load vector \( \hat{p}^j(r, \theta, t) \) and \( \omega \) is the circular excitation frequency. Equation (3.34) is transformed to the wavenumber domain:

\[
\hat{\tilde{p}}^j(k_r, n, t) = \tilde{p}^j(k_r, n)e^{i\omega t} + \tilde{p}^j(k_r, n)e^{-i\omega t}
\]  

(3.35)

The load vector \( \tilde{p}^j(k_r, n) \) in the wavenumber domain is related to the load vector \( \hat{p}^j(r, \theta) \) in the space domain by the following integral transformations:

\[
\hat{\tilde{p}}^j(k_r, n) = a_n \int_{0}^{\infty} r C_n(k_r, r) \int_{0}^{2\pi} T_n(\theta) \hat{p}^j(r, \theta) d\theta dr
\]  

(3.36)

\[
\hat{p}^j(r, \theta) = \sum_{n=0}^{\infty} T_n(\theta) \int_{0}^{\infty} k_r C_n(k_r, r) \hat{\tilde{p}}^j(k_r, n) dk_r
\]  

(3.37)

where the coefficient \( a_n \) is equal to \( 1/(2\pi) \) if \( n = 0 \) and \( 1/\pi \) if \( n \neq 0 \). The matrix \( T_n(\theta) \) is the kernel of the integral transformation in the circumferential direction.

In the symmetric case, the matrix \( T_n(\theta) \) is defined as:

\[
T_n(\theta) = \begin{bmatrix} 
\cos(n\theta) & 0 & 0 \\
0 & -\sin(n\theta) & 0 \\
0 & 0 & \cos(n\theta)
\end{bmatrix}
\]  

(3.38)

In the antisymmetric case, the matrix \( T_n(\theta) \) is defined as:

\[
T_n(\theta) = \begin{bmatrix} 
\sin(n\theta) & 0 & 0 \\
0 & \cos(n\theta) & 0 \\
0 & 0 & \sin(n\theta)
\end{bmatrix}
\]  

(3.39)
The matrix $C_n(k_r, r)$ is the kernel of the integral transformation in the radial direction. This matrix is given by:

$$
C_n(k_r, r) = \begin{bmatrix}
J'_n(k_r r) & \frac{2n}{k_r r} J_n(k_r r) & 0 \\
\frac{n}{k_r r} J_n(k_r r) & J_n(k_r r) & 0 \\
0 & 0 & -J_n(k_r r)
\end{bmatrix}
$$

(3.40)

where $J_n(k_r r)$ is an $n$-th order Bessel function of the first kind and $J'_n(k_r r)$ denotes the derivative of the function $J_n(k_r r)$ with respect to its argument $k_r r$. The matrix $C_n(k_r, r)$ can be elaborated using the following recurrence relations:

$$
J_{n-1}(k_r r) + J_{n+1}(k_r r) = 2n \frac{2n}{k_r r} J_n(k_r r)
$$

(3.41)

$$
J_{n-1}(k_r r) - J_{n+1}(k_r r) = 2J'_n(k_r r)
$$

(3.42)

The displacement field $u(r, \theta, z, t)$ induced by the loads $p^j(r, \theta, z, t)$ is also time-harmonic:

$$
u(r, \theta, z, t) = \hat{u}(r, \theta, z)e^{i\omega t} + \hat{u}^*(r, \theta, z)e^{-i\omega t}
$$

(3.43)

This equation is also transformed to the wavenumber domain:

$$
\tilde{u}(k_r, n, z, t) = \hat{u}(k_r, n, z)e^{i\omega t} + \hat{u}^*(k_r, n, z)e^{-i\omega t}
$$

(3.44)

The displacement vector $u(k_r, n, z)$ in the wavenumber domain is related to the displacement vector $u(r, \theta, z)$ in the space domain by the following integral transformations:

$$
\tilde{u}(k_r, n, z) = a_n \int_0^\infty rC_n(k_r, r) \int_0^{2\pi} T_n(\theta)u(r, \theta, z) d\theta dr
$$

(3.45)

$$
\hat{u}(r, \theta, z) = \sum_{n=0}^\infty T_n(\theta) \int_0^\infty k_r C_n(k_r, r) \tilde{u}(k_r, n, z) dk_r
$$

(3.46)

For each radial wavenumber $k_r$ and each circumferential wavenumber $n$, a problem of plane wave propagation has to be solved. The procedure outlined in subsection 3.1.1 is followed: the displacements $u(k_r, n, z)$ and the tractions $\tilde{t}(k_r, n, z)$ induced by the loads $\tilde{p}^j(k_r, n)$ are computed by means of the stiffness matrix $\tilde{K}$ and the shape functions $N(z)$ and $B(z)$, using a horizontal wavenumber $k_x$ equal to the radial wavenumber $k_r$. The circumferential wavenumber $n$ has no influence on the stiffness matrices or the shape functions. As a consequence, the stiffness matrices and shape functions have to be calculated only once (and the stiffness matrix has to be inverted only once) for all circumferential wavenumbers $n$.

It should be noted that, in contrast to the procedure followed in the two-dimensional case, the $z$-component of the load, displacement, and traction vectors is not multiplied with the imaginary unit $i$ in the three-dimensional case.
Next, the displacements $\tilde{u}(k_r, n, z)$ are transformed to the space domain by means of equation (3.46). Similarly, the tractions $\tilde{t}(k_r, n, z)$ are transformed according to the following pair of integral transformations:

$$\tilde{t}(k_r, n, z) = a_n \int_0^\infty r C_n(k_r, r) \int_0^{2\pi} T_n(\theta) \tilde{t}(r, \theta, z) d\theta dr$$

(3.47)

$$\tilde{u}(r, \theta, z) = \sum_{n=0}^\infty T_n(\theta) \int_0^\infty k_r C_n(k_r, r) \tilde{u}(k_r, n, z) dk_r$$

(3.48)

The horizontal components $\hat{\epsilon}_{xx}$, $\hat{\epsilon}_{yy}$, and $\hat{\epsilon}_{xy}$ of the strain tensor $\hat{\epsilon}(r, \theta, z)$ are calculated as follows:

$$\begin{cases}
\hat{\epsilon}_{rr} = \sum_{n=0}^\infty T'_n(\theta) \int_0^\infty k_r C'_n(k_r, r) \hat{u}(k_r, n, z) dk_r \\
\hat{\epsilon}_{\theta\theta} = \sum_{n=0}^\infty T'_n(\theta) \int_0^\infty k_r C'_n(k_r, r) \hat{u}(k_r, n, z) dk_r \\
\hat{\epsilon}_{r\theta} = \sum_{n=0}^\infty T'_n(\theta) \int_0^\infty k_r C'_n(k_r, r) \hat{u}(k_r, n, z) dk_r
\end{cases}$$

(3.49)

In the symmetric case, the matrix $T'_n(\theta)$ is defined as:

$$T'_n(\theta) = \begin{bmatrix}
\cos(n\theta) & 0 & 0 \\
0 & \cos(n\theta) & 0 \\
0 & 0 & -\sin(n\theta)
\end{bmatrix}$$

(3.50)

In the antisymmetric case, the matrix $T'_n(\theta)$ is defined as:

$$T'_n(\theta) = \begin{bmatrix}
\sin(n\theta) & 0 & 0 \\
0 & \sin(n\theta) & 0 \\
0 & 0 & \cos(n\theta)
\end{bmatrix}$$

(3.51)

The matrix $C'_n(k_r, r)$ is defined as:

$$\begin{bmatrix}
k_r J''_n(k_r) & -\frac{n}{k_r} J'_n(k_r) + \frac{n}{r} J'_n(k_r) \\
\frac{n}{k_r} J'_n(k_r) & \frac{n^2}{k_r^2} J_n(k_r) - \frac{n}{k_r} J'_n(k_r) & 0 \\
\frac{n}{k_r} J'_n(k_r) & \frac{n^2}{k_r^2} J_n(k_r) - \frac{n}{k_r} J'_n(k_r) & 0
\end{bmatrix}$$

(3.52)

where $J'_n(k_r)$ and $J''_n(k_r)$ denote the first and second derivative of the Bessel function $J_n(k_r)$ with respect to its argument $k_r$.

The vertical components $\hat{\epsilon}_{zx}$, $\hat{\epsilon}_{zy}$, and $\hat{\epsilon}_{zz}$ of the strain tensor $\hat{\epsilon}(r, \theta, z)$ and the horizontal components $\hat{\sigma}_{xx}$, $\hat{\sigma}_{yy}$, and $\hat{\sigma}_{xy}$ of the stress tensor $\hat{\sigma}(r, \theta, z)$ can be
calculated using Hooke’s law:

\[
\begin{pmatrix}
\sigma_{rr} \\
\sigma_{\theta\theta} \\
\sigma_{zz} \\
\sigma_{r\theta} \\
\epsilon_{zz} \\
\epsilon_{zr}
\end{pmatrix} =
\begin{pmatrix}
\frac{4\mu(\lambda+\mu)}{\lambda+2\mu} & \frac{2\mu}{\lambda+2\mu} & \frac{\lambda}{\lambda+2\mu} & 0 & 0 & 0 \\
\frac{2\mu}{\lambda+2\mu} & \frac{4\mu(\lambda+\mu)}{\lambda+2\mu} & \frac{\lambda}{\lambda+2\mu} & 0 & 0 & 0 \\
0 & 0 & \frac{\lambda}{\lambda+2\mu} & 0 & 0 & 0 \\
0 & 0 & 0 & 2\mu & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{2\mu} & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{2\mu}
\end{pmatrix}
\begin{pmatrix}
\epsilon_{rr} \\
\epsilon_{\theta\theta} \\
\epsilon_{zz} \\
\epsilon_{r\theta} \\
\epsilon_{zz} \\
\epsilon_{zr}
\end{pmatrix}
\]

(3.53)

The time dependent displacement field \(u(r, \theta, z, t)\) is subsequently obtained according to equation (3.43). Similarly, the time dependent strains \(\epsilon(r, \theta, z, t)\) and stresses \(\sigma(r, \theta, z, t)\) are computed as:

\[
\sigma(r, \theta, z, t) = \tilde{\sigma}(r, \theta, z)e^{-i\omega t} + \hat{\sigma}^*(r, \theta, z)e^{i\omega t}
\]

(3.54)

\[
\epsilon(r, \theta, z, t) = \tilde{\epsilon}(r, \theta, z)e^{-i\omega t} + \hat{\epsilon}^*(r, \theta, z)e^{i\omega t}
\]

(3.55)

### 3.4.2 Transient three-dimensional wave propagation

This subsection focuses on transient problems governed by three-dimensional wave propagation. The problem is formulated in a cylindrical coordinate system \((r, \theta, z)\) and solved by means of a Fourier transformation from the time domain to the frequency domain, followed by a Fourier series expansion in the circumferential direction and a Hankel transformation in the radial direction.

A transient wave field \(u(r, \theta, z, t)\) is considered. The wave field is induced by a transient load distribution on the interfaces between elements. The load on the \(j\)-th interface is denoted by the vector \(\mathbf{p}^j(r, \theta, t)\). The load vector \(\mathbf{p}^j(r, \theta, t)\) is transformed to the frequency-space domain by means of a forward Fourier transformation from the time \(t\) to the circular frequency \(\omega\). In the frequency-space domain, the vector is denoted by \(\tilde{\mathbf{p}}^j(r, \theta, \omega)\). The load vector \(\tilde{\mathbf{p}}^j(r, \theta, \omega)\) is further transformed to the frequency-wavenumber domain, according to the following equations:

\[
\tilde{\mathbf{p}}^j(k_r, n, \omega) = a_n \int_0^\infty r C_n(k_r, r) \int_0^{2\pi} T_n(\theta) \tilde{\mathbf{p}}^j(r, \theta, \omega) d\theta dr
\]

(3.56)

\[
\hat{\mathbf{p}}^j(r, \theta, \omega) = \sum_{n=0}^\infty T_n(\theta) \int_0^\infty k_r C_n(k_r, r, \omega) \hat{\mathbf{p}}^j(k_r, n, \omega) dk_r
\]

(3.57)

where the coefficient \(a_n\) is equal to \(1/(2\pi)\) if \(n = 0\) and \(1/\pi\) if \(n \neq 0\). The matrices \(T_n(\theta)\) and \(C_n(k_r, r)\) are defined in the same way as in the previous subsection.

The displacement field \(u(r, \theta, z, t)\) is transformed in a similar way. First, the vector \(u(r, \theta, z, t)\) is transformed from the time-space domain to the frequency-space domain, where it is denoted by \(\hat{u}(r, \theta, z, \omega)\). Next, the displacement vector
is transformed to the frequency-wavenumber domain according to the following equations:

\[
\hat{u}(k_r, n, z, \omega) = a_n \int_0^\infty r C_n(k_r, r) \int_0^{2\pi} T_n(\theta) \hat{u}(r, \theta, z, \omega) d\theta dr
\]  
(3.58)

\[
\hat{u}(r, \theta, z, \omega) = \sum_{n=0}^\infty T_n(\theta) \int_0^\infty k_r C_n(k_r, r) \hat{u}(k_r, n, z, \omega) dk_r
\]  
(3.59)

For each frequency \(\omega\), each radial wavenumber \(k_r\), and each circumferential wavenumber \(n\), a problem of plane wave propagation has to be solved. The procedure outlined in subsection 3.1.1 is followed: the displacements \(\hat{u}(k_r, n, z, \omega)\) and the tractions \(\hat{t}(k_r, n, z, \omega)\) induced by the loads \(\hat{p}_j(k_r, n, \omega)\) are computed by means of the stiffness matrix \(\tilde{K}\) and the shape functions \(\tilde{N}(z)\) and \(\tilde{B}(z)\), using a horizontal wavenumber \(k_x\) equal to the radial wavenumber \(k_r\).

Next, the displacements \(\hat{u}(k_r, n, z, \omega)\) are transformed to the space domain by means of equation (3.59). Similarly, the tractions \(\hat{t}(k_r, n, z, \omega)\) are transformed according to the following pair of integral transformations:

\[
\hat{t}(k_r, n, z, \omega) = a_n \int_0^\infty r C_n(k_r, r) \int_0^{2\pi} T_n(\theta) \hat{t}(r, \theta, z, \omega) d\theta dr
\]  
(3.60)

\[
\hat{t}(r, \theta, z, \omega) = \sum_{n=0}^\infty T_n(\theta) \int_0^\infty k_r C_n(k_r, r) \hat{t}(k_r, n, z, \omega) dk_r
\]  
(3.61)

The horizontal components \(\hat{\varepsilon}_{xx}, \hat{\varepsilon}_{yy},\) and \(\hat{\varepsilon}_{xy}\) of the strain tensor \(\hat{\varepsilon}(r, \theta, z, \omega)\) are calculated as follows:

\[
\begin{cases}
\hat{\varepsilon}_{rr} \\
\hat{\varepsilon}_{\theta\theta} \\
\hat{\varepsilon}_{r\theta}
\end{cases}
= \sum_{n=0}^\infty T_n'(\theta) \int_0^\infty k_r C_n'(k_r, r) \hat{u}(k_r, n, z, \omega) dk_r
\]  
(3.62)

The matrices \(T_n'(\theta)\) and \(C_n'(k_r, r)\) are defined in the same way as in the previous subsection.

The vertical components \(\hat{\varepsilon}_{xz}, \hat{\varepsilon}_{zy},\) and \(\hat{\varepsilon}_{zz}\) of the strain tensor \(\hat{\varepsilon}(r, \theta, z, \omega)\) and the horizontal components \(\hat{\sigma}_{xx}, \hat{\sigma}_{yy},\) and \(\hat{\sigma}_{xy}\) of the stress tensor \(\hat{\sigma}(r, \theta, z, \omega)\) can be calculated using Hooke’s law, according to equation (3.53).

Finally, the displacements \(\hat{u}(r, \theta, z, \omega)\), strains \(\hat{\varepsilon}(r, \theta, z, \omega)\), and stresses \(\hat{\sigma}(r, \theta, z, \omega)\) are transformed to the time-space domain by means of an inverse Fourier transformation from the circular frequency \(\omega\) to the time \(t\).
Chapter 4

The direct stiffness method

This chapter focuses on the direct stiffness method [9, 12]. This method is used in EDT to model wave propagation in a horizontally layered medium. The direct stiffness method method is based on the transfer matrix approach, initially proposed by Thomson [28] and Haskell [8], and recast into a stiffness matrix formulation by Kausel and Roëssé [12]. The method has also been referred to as a spectral element formulation by Doyle [4, 5, 6] and Rizzi and Doyle [22, 23].

The direct stiffness method is similar to the finite element method: a layered medium is modelled as an assembly of homogeneous layer and halfspace elements, the field variables within the elements are represented by shape functions, and element stiffness matrices are used to express the relation between displacements and stresses at the element boundaries. The method is based on a transformation from the time-space domain to the frequency-wavenumber domain. In the frequency-wavenumber domain, exact solutions can be obtained for the Navier equations governing wave propagation in a homogeneous layer or a homogeneous halfspace. These exact solutions are used as shape functions in the direct stiffness method. As a result, wave propagation is treated exactly and there is no need to subdivide homogeneous layers into multiple layer elements. This is in contrast to the thin layer method [12], where polynomial shape functions are used. The thin layer method is introduced in the next chapter.

This chapter is organized as follows:

**Stiffness matrices (p. 34)**

This section focuses on the calculation of stiffness matrices for layered media with EDT. The stiffness matrices for a halfspace element and for a layer element are discussed, and the assembly of the global stiffness matrix for a layered medium is clarified. The stiffness matrices relate the loads and the displacements on the interfaces between elements.
The Direct Stiffness Method

Shape functions (p. 42)
This section addresses the shape functions used in the direct stiffness method. Both the shape functions for the displacements and the shape functions for the tractions (on a horizontal plane) are discussed. These functions can be used to determine the displacements and the tractions in the interior of the elements if the interface displacements are known.

4.1 Stiffness matrices

4.1.1 The halfspace element

The halfspace element models the propagation of waves in a semi-infinite halfspace. Both a halfspace element directed downwards and a halfspace element directed upwards are available in EDT. By default, only outgoing waves in the halfspace element are taken into account. However, EDT provides an option to define a halfspace element that accounts for incoming waves instead of outgoing waves. Such a halfspace element is useful for the calculation of wave amplification (chapter 7).

\[ \tilde{P}_e = \{ \tilde{p}_{x1}, \tilde{p}_{y1}, \tilde{p}_{z1} \}^T \]

\[ \tilde{U}_e = \{ \tilde{u}_{x1}, \tilde{u}_{y1}, \tilde{u}_{z1} \}^T \]

Figure 4.1: The loads \( \tilde{P}_e \) and displacements \( \tilde{U}_e \) at the surface of (a) a halfspace element directed downwards and (b) a halfspace element directed upwards.

Figure 4.1 shows the loads \( \tilde{P}_e = \{ \tilde{p}_{x1}, \tilde{p}_{y1}, \tilde{p}_{z1} \}^T \) and displacements \( \tilde{U}_e = \{ \tilde{u}_{x1}, \tilde{u}_{y1}, \tilde{u}_{z1} \}^T \) at the surface of a halfspace element.

The loads \( \tilde{P}_e \) are related to the displacements \( \tilde{U}_e \) through the element stiffness matrix \( K_e \):

\[ K_e \tilde{U}_e = \tilde{P}_e \] (4.1)
Numerical expressions for the stiffness matrix $\tilde{K}$ of the halfspace element can be found in references [3, 9, 12, 25].

Due to the uncoupling of P-SV-waves and SH-waves, the coupling terms between the $x$ or $z$-direction and the $y$-direction vanish in the stiffness matrix $\tilde{K}$. The relation (4.1) between the loads $\tilde{P}$ and the displacements $\tilde{U}$ and can therefore be decomposed as:

$$
\tilde{K}_{PSV}^{c} \tilde{U}_{PSV} = \tilde{P}_{PSV}
$$

(4.2)

$$
\tilde{K}_{SH}^{c} \tilde{U}_{SH} = \tilde{P}_{SH}
$$

(4.3)

where $\tilde{U}_{PSV} = \{\tilde{u}_{x1}, \tilde{u}_{z1}\}$ and $\tilde{P}_{PSV} = \{\tilde{p}_{x1}, \tilde{p}_{z1}\}$ are related to P-SV-wave propagation and $\tilde{U}_{SH} = \{\tilde{u}_{y1}\}$ and $\tilde{P}_{SH} = \{\tilde{p}_{y1}\}$ are related to SH-wave propagation. The $2 \times 2$ matrix $\tilde{K}_{PSV}^{c}$ and the $1 \times 1$ matrix $\tilde{K}_{SH}^{c}$ are submatrices of the element stiffness matrix $\tilde{K}^{c}$. The use of the uncoupled element stiffness matrices $\tilde{K}_{PSV}^{c}$ and $\tilde{K}_{SH}^{c}$ leads to two smaller systems of equations that can be solved more efficiently. The matrices $\tilde{K}_{PSV}^{c}$ and $\tilde{K}_{SH}^{c}$ can be calculated with EDT using the functions $ke_{dsmpsv}$ and $ke_{dsmsh}$, respectively.

Example 4.1: The stiffness matrix of a halfspace.

This example demonstrates how to compute the element stiffness matrix $\tilde{K}_{PSV}^{c}$ for a halfspace element with EDT. The following MATLAB code is used to calculate the stiffness matrix $\tilde{K}_{PSV}^{c}$ for a halfspace with a shear wave velocity $C_s = 500 \text{ m/s}$, a dilatational wave velocity $C_p = 2000 \text{ m/s}$, a material damping ratio $\beta = 0.05$ for both shear and dilatational waves, and a density $\rho = 1800 \text{ kg/m}^3$. The stiffness matrix $\tilde{K}_{PSV}^{c}$ is computed for a frequency $\omega = 100 \text{ rad/s}$ and a horizontal wavenumber $k = 10 \text{ rad/m}$.

```matlab
% HALFSPACE PROPERTIES
h = inf; \% Element thickness [m]
Cs = 500; \% Shear wave velocity [m/s]
Cp = 2000; \% Dilatational wave velocity [m/s]
Ds = 0.05; \% Shear damping ratio [-]
Dp = 0.05; \% Dilatational damping ratio [-]
rho = 1800; \% Density [kg/m^3]

% FREQUENCY AND WAVENUMBER
omega = 100; \% Frequency [rad/s]
k = 0.5; \% Horizontal wavenumber [rad/m]

% STIFFNESS MATRIX FOR P-SV WAVES
Ke = ke_dsmpsv(h,Cs,Cp,Ds,Dp,rho,k,omega); \% Stiffness matrix [m^3/N]
```

The following stiffness matrix $\tilde{K}_{PSV}^{c}$ is obtained in MATLAB:
4.1.2 The layer element

The layer element models the propagation of waves in a layer of finite thickness \( h \), where both incoming and outgoing waves propagate.

\[
\begin{bmatrix}
1.0 \times 10^8 & 4.0668 + 0.4243i & 0.4127 + 0.0256i \\
0.4127 + 0.0256i & 3.7461 + 0.4246i \\
\end{bmatrix}
\]

Figure 4.2: The loads \( \tilde{P}^e \) and displacements \( \tilde{U}^e \) at the boundaries of a layer element.

Figure 4.2 shows the loads \( \tilde{P}^e = \{ \tilde{p}^e_x, \tilde{p}^e_y, \tilde{p}^e_z \} \) and displacements \( \tilde{U}^e = \{ \tilde{u}^e_x, \tilde{u}^e_y, \tilde{u}^e_z \} \) at the boundaries of a layer element. The loads \( \tilde{P}^e \) are related to the displacements \( \tilde{U}^e \) through the element stiffness matrix \( \tilde{K}^e \):

\[
\tilde{K}^e \tilde{U}^e = \tilde{P}^e
\]

Numerical expressions for the stiffness matrix \( \tilde{K}^e \) of the layer element can be found in references [3, 9, 12, 25].

In a similar way as for the halfspace element, the relation (4.4) between the loads \( \tilde{P}^e \) and the displacements \( \tilde{U}^e \) and can be decomposed as:

\[
\begin{align*}
\tilde{K}^e_{PSV} \tilde{U}^e_{PSV} &= \tilde{P}^e_{PSV} \\
\tilde{K}^e_{SH} \tilde{U}^e_{SH} &= \tilde{P}^e_{SH}
\end{align*}
\]

where \( \tilde{U}^e_{PSV} = \{ \tilde{u}^e_x, \tilde{u}^e_y, \tilde{u}^e_z \} \) and \( \tilde{P}^e_{PSV} = \{ \tilde{p}^e_x, \tilde{p}^e_y, \tilde{p}^e_z \} \) are related to P-SV-wave propagation and \( \tilde{U}^e_{SH} = \{ \tilde{u}^e_y \} \) and \( \tilde{P}^e_{SH} = \{ \tilde{p}^e_y \} \) are
related to SH-wave propagation. The $4 \times 4$ matrix $K_{PSV}^e$ and the $2 \times 2$ matrix $K_{SH}^e$ are submatrices of the element stiffness matrix $K^e$. The matrices $K_{PSV}^e$ and $K_{SH}^e$ can be calculated with EDT using the functions $ke_{dsmpsV}$ and $ke_{dsmsH}$, respectively.

Example 4.2: Vertical harmonic wave propagation in a layer on bedrock.

This example considers one-dimensional harmonic wave propagation in a homogeneous layer on bedrock. The layer is loaded at the top with a harmonic force per unit area. The amplitude of the excitation is $\bar{p} = \{0, 0, 1\}^T$. This excitation induces vertical P-waves in the layer. In the following, the amplitude $\bar{u}_e$ of the vertical displacement at the top of the layer is calculated for different excitation frequencies.

The thickness of the layer is $h = 10$ m, the shear wave velocity is $C_s = 200$ m/s, the dilatational wave velocity is $C_p = 400$ m/s, the density is $\rho = 1800$ kg/m³, and the damping ratio in both shear and volumetric deformation is $D_s = D_p = 0.05$.

The following MATLAB code computes the element stiffness matrix $K_{PSV}$ for the layer at 801 frequencies from 0 Hz to 80 Hz. The horizontal wavenumber $k_x$ is equal to zero as vertical wave propagation is considered. The matrix $K_{PSV}$ relates the loads $\bar{P}_{PSV} = \{\bar{p}_{x1}, \bar{p}_{x2}, \bar{p}_{x2}^0\}$ and the displacements $\bar{U}_{PSV} = \{\bar{u}_e^{x1}, \bar{u}_e^{x2}, \bar{u}_e^{x2}, \bar{u}_e^{x2}\}$ at the boundaries of the layer:

$$
\begin{pmatrix}
\bar{p}_{x1} \\
\bar{p}_{x2}
\end{pmatrix} =
\begin{pmatrix}
\bar{K}_{1x1}^e & \bar{K}_{1x2}^e & \bar{K}_{2x1}^e & \bar{K}_{2x2}^e \\
\bar{K}_{2x1}^e & \bar{K}_{2x2}^e & \bar{K}_{5x1}^e & \bar{K}_{5x2}^e \\
\bar{K}_{5x1}^e & \bar{K}_{5x2}^e & \bar{K}_{1x1}^e & \bar{K}_{1x2}^e \\
\bar{K}_{1x1}^e & \bar{K}_{1x2}^e & \bar{K}_{2x1}^e & \bar{K}_{2x2}^e
\end{pmatrix}
\begin{pmatrix}
\bar{u}_e^{x1} \\
\bar{u}_e^{x2}
\end{pmatrix}
$$

(4.7)

As a layer on bedrock is considered, the displacements $\bar{u}_e^{x2}$ and $\bar{u}_e^{x2}$ at the bottom are zero, and equation (4.7) reduces to:

$$
\begin{pmatrix}
\bar{p}_{x1} \\
\bar{p}_{x2}
\end{pmatrix} =
\begin{pmatrix}
\bar{K}_{1x1}^e & \bar{K}_{1x2}^e \\
\bar{K}_{2x1}^e & \bar{K}_{2x2}^e
\end{pmatrix}
\begin{pmatrix}
\bar{u}_e^{x1} \\
\bar{u}_e^{x2}
\end{pmatrix}
$$

(4.8)

For each frequency, equation (4.8) is solved to obtain the vertical displacement $\bar{u}_e^{x1}$ at the top of the layer induced by the loads $\bar{p}_{x1}^0 = 0$ N/m² and $\bar{p}_{x2}^0 = 1$ N/m².

% SOIL PROPERTIES

<table>
<thead>
<tr>
<th>h</th>
<th>10;</th>
<th>Layer thickness [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_s</td>
<td>200;</td>
<td>Shear wave velocity [m/s]</td>
</tr>
<tr>
<td>C_p</td>
<td>400;</td>
<td>Dilatational wave velocity [m/s]</td>
</tr>
<tr>
<td>D_s</td>
<td>0.05;</td>
<td>Shear damping ratio [-]</td>
</tr>
<tr>
<td>D_p</td>
<td>0.05;</td>
<td>Dilatational damping ratio [-]</td>
</tr>
<tr>
<td>rho</td>
<td>1800;</td>
<td>Density [kg/m³]</td>
</tr>
</tbody>
</table>
% FREQUENCY AND WAVENUMBER
k = 0; % Horizontal wavenumber [rad/m]
f = 0:0.1:80; % Frequency [Hz]
omega = 2*pi*f; % Frequency [rad/s]
nFreq = length(f); % Number of frequencies

% EXCITATION
P = [0;1];

% COMPUTE DISPLACEMENT FOR EACH FREQUENCY
uz = zeros(nFreq,1);
for iFreq=1:nFreq
    Ke = ke_dsmpsv(h,Cs,Cp,Ds,Dp,rho,k,omega(iFreq));
    Ke = Ke(1:2,1:2);
    U = Ke\P;
    uz(iFreq) = U(2);
end

% PLOT RESULT
figure;
plot(f,abs(uz));
xlabel('Frequency [Hz]');
ylabel('Displacement [m/(N/m^2)]');

![Graph](image_url)

**Figure 4.3:** Vertical displacement $\tilde{u}_{z1}(z)$ at the top of the layer ($z = 0$ m).

The resulting displacement $\tilde{u}_{z1}$ is shown in figure 4.3. At 0 Hz, the static response $h/M = 3.472 \times 10^{-8}$ m is obtained, where $M = \rho C_p^2 = 288 \times 10^6$ N/m² is the constrained modulus of the layer. Resonance occurs at odd multiples of the first resonance frequency $C_p/(4h) = 10$ Hz of the layer. Due to material damping, the response at the resonance frequencies is finite and the peak values decrease with increasing frequency.
4.1.3 Assembly of equations

The propagation of waves in a horizontally layered medium can be modelled with a number of layer elements coupled to one or two halfspace elements (figure 4.4). The loads and the displacements on interface \( i \) between elements \( i-1 \) and \( i \) are denoted by \( \tilde{p}_i \) and \( \tilde{u}_i \), respectively. The equilibrium of the layered medium is expressed as:

\[
\begin{bmatrix}
\tilde{K}_{11}^1 & \tilde{K}_{12}^1 \\
\tilde{K}_{21}^1 & \tilde{K}_{22}^1 + \tilde{K}_{11}^2 \\
\tilde{K}_{21}^2 & \tilde{K}_{22}^2 + \tilde{K}_{11}^3 \\
\title{\vdots} & \ddots
\end{bmatrix}
\begin{bmatrix}
\tilde{u}_1^1 \\
\tilde{u}_2^1 \\
\tilde{u}_3^1 \\
\vdots
\end{bmatrix}
= \begin{bmatrix}
\tilde{p}_1^1 \\
\tilde{p}_2^1 \\
\tilde{p}_3^1 \\
\vdots
\end{bmatrix}
\tag{4.9}
\]

In this equation, \( \tilde{K}_{ij}^e \) refers to the partition of the stiffness matrix \( \tilde{K}^e \) of element \( e \) that relates the displacements at interface \( i \) to the loads on interface \( j \), where \( i \) and \( j \) are 1 for the top interface and 2 for the bottom interface.

Equation (4.9) is rewritten as:

\[
\tilde{K}\tilde{U} = \tilde{P}
\tag{4.10}
\]

where the matrix \( \tilde{K} \) is the assembled stiffness matrix for the layered medium and the vectors \( \tilde{P} \) and \( \tilde{U} \) collect the loads \( \tilde{p}_i \) and the displacements \( \tilde{u}_i \) at all interfaces between elements.
From a computational point of view, it is advantageous to exploit the uncoupling of P-SV and SH-waves and to assemble separate stiffness matrices $\tilde{K}_{PSV}$ and $\tilde{K}_{SH}$ from the corresponding element stiffness matrices $\tilde{K}_{ePSV}$ and $\tilde{K}_{eSH}$ introduced in equations (4.2), (4.3), (4.5), and (4.6). The equilibrium of the layered medium is then expressed as:

$$\tilde{K}_{PSV} \tilde{U}_{PSV} = \tilde{P}_{PSV}$$

$$\tilde{K}_{SH} \tilde{U}_{SH} = \tilde{P}_{SH}$$

where $\tilde{P}_{PSV}$ and $\tilde{U}_{PSV}$ collect the loads and displacements in $x$ and $z$-direction and $\tilde{P}_{SH}$ and $\tilde{U}_{SH}$ collect the loads and displacements in $y$-direction.

EDT provides three different methods to obtain the stiffness matrices $\tilde{K}_{PSV}$ and $\tilde{K}_{SH}$. The first method is to calculate the element stiffness matrices $\tilde{K}_{ePSV}$ and $\tilde{K}_{eSH}$ are individually using the functions $ke_{dsmpsv}$ and $ke_{dsmsh}$. The assembly of the global stiffness matrices $\tilde{K}_{PSV}$ and $\tilde{K}_{SH}$ from the element stiffness matrices $\tilde{K}_{ePSV}$ and $\tilde{K}_{eSH}$ is performed using standard MATLAB functions. The second method also starts with the calculation of the element stiffness matrices $\tilde{K}_{ePSV}$ and $\tilde{K}_{eSH}$ using the functions $ke_{dsmpsv}$ and $ke_{dsmsh}$. The global stiffness matrices $\tilde{K}_{PSV}$ and $\tilde{K}_{SH}$ are assembled by means of the EDT functions $asmk_{psv}$ and $asmk_{sh}$. The third method is the most convenient method: the global stiffness matrices $\tilde{K}_{PSV}$ and $\tilde{K}_{SH}$ are calculated directly using the functions $k_{dsmpsv}$ and $k_{dsmsh}$. Internally, these functions perform both the calculation of the element stiffness matrices $\tilde{K}_{ePSV}$ and $\tilde{K}_{eSH}$ and the assembly of the global stiffness matrices $\tilde{K}_{PSV}$ and $\tilde{K}_{SH}$.

**Example 4.3: Stiffness matrices for a layered medium.**

In this example, the stiffness matrix for P-SV waves in a layered medium, consisting of two layers on a halfspace, is computed in three different ways. First, the element stiffness matrices are assembled using standard MATLAB functions.

```matlab
% SOIL PROPERTIES
h = [0.97 1.90 inf];  % Element thickness [m]
Cs = [143 168 259];  % Shear wave velocity [m/s]
Cp = [286 336 518];  % Dilatational wave velocity [m/s]
Ds = 0.03;            % Shear damping ratio [-]
Dp = 0.03;            % Dilatational damping ratio [-]
rho = 1800;          % Density [kg/m^3]

% FREQUENCY AND WAVENUMBER
omega = 100;          % Frequency [rad/s]
k = 0.5;              % Horizontal wavenumber [rad/m]

% ELEMENT STIFFNESS MATRICES
```
Second, the element stiffness matrices are assembled with the EDT function `asmk_psv`.

**SOIL PROPERTIES**

- $h = [0.97 \ 1.90 \ \text{inf}]$: Element thickness [m]
- $Cs = [143 \ 168 \ 259]$; Shear wave velocity [m/s]
- $Cp = [286 \ 336 \ 518]$; Dilatational wave velocity [m/s]
- $Ds = 0.03$; Shear damping ratio [-]
- $Dp = 0.03$; Dilatational damping ratio [-]
- $\rho = 1800$; Density [kg/m$^3$]

**FREQUENCY AND WAVENUMBER**

- $\omega = 100$; Frequency [rad/s]
- $k = 0.5$; Horizontal wavenumber [rad/m]

**ELEMENT STIFFNESS MATRICES**

- $K1 = \text{ke_dsmpsv}(h(1),Cs(1),Cp(1),Ds,Dp,\rho,k,\omega)$;
- $K2 = \text{ke_dsmpsv}(h(2),Cs(2),Cp(2),Ds,Dp,\rho,k,\omega)$;
- $K3 = \text{ke_dsmpsv}(h(3),Cs(3),Cp(3),Ds,Dp,\rho,k,\omega)$;

**GLOBAL STIFFNESS MATRIX**

- $K = \text{asmk_psv}(K1,K2,K3)$;

Third, the EDT function `k_dsmpsv` is used to compute the global stiffness matrix directly.

**SOIL PROPERTIES**

- $h = [0.97 \ 1.90 \ \text{inf}]$: Element thickness [m]
- $Cs = [143 \ 168 \ 259]$; Shear wave velocity [m/s]
- $Cp = [286 \ 336 \ 518]$; Dilatational wave velocity [m/s]
- $Ds = 0.03$; Shear damping ratio [-]
- $Dp = 0.03$; Dilatational damping ratio [-]
- $\rho = 1800$; Density [kg/m$^3$]

**FREQUENCY AND WAVENUMBER**
\[
\omega = 100; \quad k = 0.5; \quad \% \text{Frequency [rad/s}}
\]

\[
k = \text{dsmpsv}(h,Cs,Cp,Ds,Dp,rho,k,omega);
\]

In all three cases, the same global stiffness matrix \(K_{PSV}\) is obtained:

\[
K = 1.0e+008 *
\]

Columns 1 through 3

\[
\begin{array}{cccc}
0.4217 + 0.0286i & -0.0801 - 0.0044i & -0.3340 - 0.0185i \\
-0.0801 - 0.0044i & 1.4248 + 0.0892i & -0.2667 - 0.0153i \\
-0.3340 - 0.0185i & -0.2667 - 0.0153i & 0.8259 + 0.0583i \\
0.2667 + 0.0153i & -1.4676 - 0.0865i & 0.0179 + 0.0016i \\
0 & 0 & -0.1430 - 0.0067i \\
0 & 0 & 0.3222 + 0.0171i \\
\end{array}
\]

Columns 4 through 6

\[
\begin{array}{cccc}
0.2667 + 0.0153i & 0 & 0 \\
-1.4676 - 0.0865i & 0 & 0 \\
0.0179 + 0.0016i & -0.1430 - 0.0067i & 0.3222 + 0.0171i \\
2.3091 + 0.1502i & -0.3222 - 0.0171i & -0.9351 - 0.0537i \\
-0.3222 - 0.0171i & 1.2068 + 0.0902i & 0.3995 + 0.0142i \\
-0.9351 - 0.0537i & 0.3995 + 0.0142i & 1.4375 + 0.1243i \\
\end{array}
\]

It should be noted, however, that the functions \(asmk_{psv}\) and \(k_{dsmpsv}\) return this matrix in sparse form.

### 4.2 Shape functions

#### 4.2.1 Shape functions for the displacements

The solution of the equilibrium equations (4.11) and (4.12) leads to the displacements \(\tilde{U}_e\) at the boundaries of each element \(e\). The displacements \(\tilde{u}(z)\) in the interior of element \(e\) can subsequently be computed by evaluation of the
element shape functions $\mathbf{\tilde{N}}_e(z)$:

$$\mathbf{\tilde{u}}(z) = \mathbf{\tilde{N}}_e(z) \mathbf{\tilde{U}}_e$$  \hspace{1cm} (4.13)

Due to the uncoupling of P-SV-waves and SH-waves, this equation can be reformulated as:

$$\mathbf{\tilde{u}}_{PSV}(z) = \mathbf{\tilde{N}}_{PSV}^e(z) \mathbf{\tilde{U}}_{PSV}_e$$  \hspace{1cm} (4.14)

$$\mathbf{\tilde{u}}_{SH}(z) = \mathbf{\tilde{N}}_{SH}^e(z) \mathbf{\tilde{U}}_{SH}_e$$  \hspace{1cm} (4.15)

where $\mathbf{\tilde{u}}_{PSV}(z) = \{\tilde{u}_x(z), \tilde{u}_z(z)\}^T$ and $\mathbf{\tilde{u}}_{SH}(z) = \{\tilde{u}_y(z)\}$. Considering each component $\tilde{u}_i(z)$ of the displacement vector $\mathbf{\tilde{u}}(z)$ separately, the following equations are obtained:

$$\tilde{u}_x(z) = \mathbf{\tilde{N}}_{PSV}^e(z) \mathbf{\tilde{U}}_{PSV}_e$$  \hspace{1cm} (4.16)

$$\tilde{u}_y(z) = \mathbf{\tilde{N}}_{SH}^e(z) \mathbf{\tilde{U}}_{SH}_e$$  \hspace{1cm} (4.17)

$$\tilde{u}_z(z) = \mathbf{\tilde{N}}_{PSV}^e(z) \mathbf{\tilde{U}}_{PSV}_e$$  \hspace{1cm} (4.18)

where the vector $\mathbf{\tilde{N}}_e^i(z)$ collects the shape functions for the displacements in element $e$ in direction $i$.

In EDT, the element shape functions $\mathbf{\tilde{N}}_{PSV}^e(z)$ and $\mathbf{\tilde{N}}_{SH}^e(z)$ are computed with the function ne_dampsv. This function evaluates the shape functions at $N$ user defined depths $z$ and returns the result as a matrix with dimensions $N \times 2$ for halfspace elements and $N \times 4$ for layer elements. The depth $z$ has to be specified in the element coordinate system. The shape functions $\mathbf{\tilde{N}}_{PSV}^e(z)$ are computed in a similar way with the function ne_dampsh. The resulting matrix has dimensions $N \times 1$ for halfspace elements and $N \times 2$ for layer elements.

**Example 4.4: The shape functions of a layer element.**

In this example, the shape functions $\mathbf{\tilde{N}}_{PSV}^e(z)$ for the vertical displacement $\tilde{u}_z(z)$ in a layer element are computed. The thickness of the layer is 10 m, the shear wave velocity is $C_s = 200$ m/s, the dilatational wave velocity is $C_p = 400$ m/s, the damping ratio in both shear and volumetric deformation is $D_s = D_p = 0.05$, and the density is $1800$ kg/m$^3$. The shape functions are calculated for a frequency $f = 20$ Hz and a horizontal wavenumber $k_x = 0.05$ rad/m. The horizontal wavenumber $k_x$ determines the angles $\theta_s$ and $\theta_p$ of the S-waves and the P-waves with respect to the $x$-axis. According to equations (3.2) and (3.3), these angles are equal to $\theta_s = 85^\circ$ and $\theta_p = 81^\circ$. The direction of the waves is therefore close to vertical.

```matlab
% SOIL PROPERTIES
h = 10; % Layer thickness [m]
Cs = 200; % Shear wave velocity [m/s]
```
Cp = 400; % Dilatational wave velocity [m/s]
Ds = 0.05; % Shear damping ratio [-]
Dp = 0.05; % Dilatational damping ratio [-]
rho = 1800; % Density [kg/m^3]

% SAMPLING PARAMETERS
k = 0.05; % Horizontal wavenumber [rad/m]
z = [0:0.01:10]; % Receiver depth [m]
f = 20; % Frequency [Hz]
omega = 2*pi*f; % Frequency [rad/s]

% SHAPE FUNCTIONS FOR P-SV WAVE PROPAGATION
[Nxe,Nze]=ne_dmpsv(h,Cs,Cp,Ds,Dp,rho,k,omega,z);

% PLOT RESULT
for n=1:4
    figure;
    plot(real(Nze(:,n)),z,imag(Nze(:,n)),z);
    xlabel('Displacement [-]');
    ylabel('Depth [m]');
end

The resulting shape functions \( \tilde{N}_e^z(z) \) are shown in figure 4.5. The shape functions \( \tilde{N}_{e_1}^z(z) \) and \( \tilde{N}_{e_3}^z(z) \) (figures 4.5a and 4.5c) relate the vertical displacement \( \tilde{u}_e(z) \) at depth \( z \) to the horizontal displacements \( \tilde{u}_{e_1}^z \) and \( \tilde{u}_{e_2}^z \) at the top and the bottom of the element, respectively. Analogously, the shape functions \( \tilde{N}_{e_2}^z(z) \) and \( \tilde{N}_{e_4}^z(z) \) (figures 4.5b and 4.5d) relate the vertical displacement \( \tilde{u}_e(z) \) at depth \( z \) to the vertical displacements \( \tilde{u}_{e_1}^z \) and \( \tilde{u}_{e_2}^z \) at the top and the bottom of the element, respectively. It is clear from figures 4.5a and 4.5c that the relation between the vertical displacement \( \tilde{u}_e(z) \) at depth \( z \) and the horizontal displacements \( \tilde{u}_{e_1}^z \) and \( \tilde{u}_{e_2}^z \) at the boundaries of the element is relatively weak. This is explained by the propagation direction of the waves in the layer, which is close to vertical. In the case of perfectly vertical wave propagation, the vertical displacement \( \tilde{u}_e(z) \) at depth \( z \) is independent of the horizontal displacements \( \tilde{u}_{e_1}^z \) and \( \tilde{u}_{e_2}^z \) at the element boundaries and the shape functions \( \tilde{N}_{e_1}^z(z) \) and \( \tilde{N}_{e_3}^z(z) \) shown in figures 4.5a and 4.5c are equal to zero.
SHAPE FUNCTIONS

The displacements $\tilde{u}(z)$ at depth $z$ in a layered medium can be calculated as:

$$\tilde{u}_x(z) = \tilde{N}_x(z) \tilde{U}_{PSV} \quad (4.19)$$
$$\tilde{u}_y(z) = \tilde{N}_y(z) \tilde{U}_{SH} \quad (4.20)$$
$$\tilde{u}_z(z) = \tilde{N}_z(z) \tilde{U}_{PSV} \quad (4.21)$$

where the vectors $\tilde{U}_{PSV}$ and $\tilde{U}_{SH}$ collect the displacements at all interfaces between elements and the vectors $\tilde{N}_i(z)$ are global shape functions assembled from the element shape functions $\tilde{N}_e(z)$.

In EDT, the global shape functions $\tilde{N}_i(z)$ can be obtained with the functions n_dmpsv and n_dshm. These functions assemble the global shape functions from the element shape functions and evaluate them at $N$ user defined depths $z$. The depth $z$ is the depth in the global coordinate system.

**Figure 4.5**: Real (solid line) and imaginary (dashed line) part of the shape functions (a) $\tilde{N}_{e1}(z)$, (b) $\tilde{N}_{e2}(z)$, (c) $\tilde{N}_{e3}(z)$, and (d) $\tilde{N}_{e4}(z)$ for the vertical displacements $\tilde{u}_z(z)$ in a layer element.
4.2.2 Shape functions for the tractions

The tractions \( \hat{t}(z) \) on a horizontal plane at depth \( z \) in an element \( e \) can be computed using the element shape functions \( \hat{B}^e(z) \):

\[
\hat{t}(z) = \hat{B}^e(z) \hat{U}^e \tag{4.22}
\]

where the vector \( \hat{U}^e \) collects the displacements at the boundaries of the element \( e \). Due to the uncoupling of P-SV-waves and SH-waves, equation (4.22) can be reformulated as:

\[
\hat{t}_{PSV}(z) = \hat{B}^e_{PSV}(z) \hat{U}^e_{PSV} \tag{4.23}
\]

\[
\hat{t}_{SH}(z) = \hat{B}^e_{SH}(z) \hat{U}^e_{SH} \tag{4.24}
\]

where \( \hat{t}_{PSV}(z) = \{ \hat{t}_x(z), \hat{t}_z(z) \}^T \) and \( \hat{t}_{SH}(z) = \{ \hat{t}_y(z) \} \). Considering each component \( \hat{t}_i(z) \) of the traction vector \( \hat{t}(z) \) separately, the following equations are obtained:

\[
\hat{t}_x(z) = \hat{B}^e_x(z) \hat{U}^e_{PSV} \tag{4.25}
\]

\[
\hat{t}_y(z) = \hat{B}^e_y(z) \hat{U}^e_{SH} \tag{4.26}
\]

\[
\hat{t}_z(z) = \hat{B}^e_z(z) \hat{U}^e_{PSV} \tag{4.27}
\]

where the vector \( \hat{B}^e_i(z) \) collects the shape functions for the tractions in element \( e \) in direction \( i \).

In EDT, the element shape functions \( \hat{B}^e_x(z) \) and \( \hat{B}^e_y(z) \) are computed with the function \texttt{be_dsmpsv}. This function evaluates the shape functions at \( N \) user defined depths \( z \) and returns the result as a matrix with dimensions \( N \times 2 \) for halfspace elements and \( N \times 4 \) for layer elements. The depth \( z \) has to be specified in the element coordinate system. The shape functions \( \hat{B}^e_y(z) \) are computed in a similar way with the function \texttt{be_dsmsh}. The resulting matrix has dimensions \( N \times 1 \) for halfspace elements and \( N \times 2 \) for layer elements.

The tractions \( \hat{t}(z) \) on a horizontal plane at depth \( z \) in a layered medium can be calculated as:

\[
\hat{t}_x(z) = \hat{B}^e_x(z) \hat{U}_{PSV} \tag{4.28}
\]

\[
\hat{t}_y(z) = \hat{B}^e_y(z) \hat{U}_{SH} \tag{4.29}
\]

\[
\hat{t}_z(z) = \hat{B}^e_z(z) \hat{U}_{PSV} \tag{4.30}
\]

where the vectors \( \hat{U}_{PSV} \) and \( \hat{U}_{SH} \) collect the displacements at all interfaces between elements and the vectors \( \hat{B}_i(z) \) are global shape functions assembled from the element shape functions \( \hat{B}^e_i(z) \).

The global shape functions \( \hat{B}_i(z) \) can be obtained in EDT with the functions \texttt{b_dsmpsv} and \texttt{b_dsmsh}. These functions assemble the global shape functions from
the element shape functions and evaluate them at \( N \) user defined depths \( z \). The depth \( z \) is the depth in the global coordinate system.

**Example 4.5: Vertical transient wave propagation in a layer on a halfspace.**

This example considers one-dimensional transient wave propagation in a layer on a halfspace. The thickness of the layer is \( h = 10 \text{ m} \), the Young’s modulus is \( E = 192 \times 10^6 \text{ N/m}^2 \), the Poisson’s ratio is \( \nu = 0.333 \), the density is \( \rho = 1800 \text{ kg/m}^3 \), and the damping ratio in both shear and volumetric deformation is \( D_s = D_p = 0.05 \). According to equations (2.1–2.4), the shear wave velocity of the layer is \( C_s = 200 \text{ m/s} \) and the dilatational wave velocity is \( C_p = 400 \text{ m/s} \). The Young’s modulus of the halfspace is \( E = 1320 \times 10^6 \text{ N/m}^2 \) and the Poisson’s ratio is \( \nu = 0.466 \). The other material properties of the halfspace are identical to those of the layer. The shear wave velocity of the halfspace is \( C_s = 500 \text{ m/s} \) and the dilatational wave velocity is \( C_p = 2000 \text{ m/s} \).

The top of the layer is loaded with a transient force \( \mathbf{p}(t) = \{0, 0, p_z(t)\}^T \) per unit area. The time history of the resulting vertical velocity \( v_z(z, t) \) and the vertical traction \( t_z(z, t) \) on a horizontal plane is computed at different depths \( z \).

The loading function \( p_z(t) \) is a Ricker pulse defined as:

\[
p_z(t) = \left[ 2 \left( \frac{\pi(t - t_s)}{t_d} \right)^2 - 1 \right] \exp \left[ - \left( \frac{\pi(t - t_s)}{t_d} \right)^2 \right]
\]

(4.31)

where \( t_d = 0.01 \text{ s} \) is the characteristic period and \( t_s = 0.1 \text{ s} \) the time shift. Figure 4.6 shows the time history and frequency content of the Ricker pulse. The spectrum has a maximum at a frequency \( 1/t_d = 100 \text{ Hz} \).

\[\begin{array}{c}
\text{(a)} \\
\text{(b)}
\end{array}\]

**Figure 4.6:** (a) Time history \( p_z(t) \) and (b) frequency content \( \hat{p}_z(\omega) \) of the loading function.
The loading function \( p_z(t) \) is sampled in the time domain and transformed to the frequency domain by means of an FFT algorithm. In the frequency domain, the vertical velocity \( \hat{v}_z(z, \omega) \) and the vertical traction \( \hat{t}_z(z, \omega) \) on a horizontal plane are calculated. The velocity \( \hat{v}_z(z, \omega) \) is obtained from the displacement \( \hat{u}_z(z, \omega) \) by a multiplication with the factor \( i\omega \). The positive spectra of the velocity \( \hat{v}_z(z, \omega) \) and the traction \( \hat{t}_z(z, \omega) \) are actually calculated, while the negative spectra are determined using the following equations:

\[
\hat{v}_z(z, -\omega) = \hat{v}_z^\ast(z, \omega) \tag{4.32}
\]

\[
\hat{t}_z(z, -\omega) = \hat{t}_z^\ast(z, \omega) \tag{4.33}
\]

where an asterisk denotes the complex conjugate. The time domain response is finally obtained by means of an inverse FFT algorithm.

The sampling interval \( \Delta t \) is 0.001 s. For this interval, the Nyquist frequency \( f_{Nyq} = 1/(2\Delta t) = 500 \text{ Hz} \) covers the bandwidth of the loading function (figure 4.6b). The number of samples is \( N = 1024 \), resulting in a time window \( T = N\Delta t = 1.024 \text{ s} \) and a resolution \( \Delta f = 1/T = 0.9766 \text{ Hz} \) in the frequency domain.

% SOIL PROPERTIES

\( h = [10 \text{ inf}] \); % Layer thickness [m]
\( Cs = [200 500] \); % Shear wave velocity [m/s]
\( Cp = [400 2000] \); % Dilatational wave velocity [m/s]
\( Ds = 0.05 \); % Shear damping ratio [-]
\( Dp = 0.05 \); % Dilatational damping ratio [-]
\( \rho = 1800 \); % Density [kg/m^3]

% SAMPLING PARAMETERS

\( k = 0 \); % Horizontal wavenumber [rad/m]
\( z = [0:1:20] \); % Receiver depths [m]
\( nRec = \text{length}(z) \); % Number of receivers [-]
\( dt = 0.001 \); % Time sampling interval [s]
\( N = 1024 \); % Number of time samples [-]
\( T = N*dt \); % Time window [s]
\( df = 1/T \); % Frequency sampling interval [Hz]
\( t = [0:N-1]*dt \); % Time sampling [s]
\( f = [0:N/2-1]*df \); % Frequency sampling (positive spectrum) [Hz]
\( \omega = 2*pi*f \); % Frequency sampling (positive spectrum) [rad/s]
\( nFreq = \text{length}(f) \); % Number of positive frequencies [-]

% EXCITATION

\( ts = 0.1 \); % Time shift of Ricker pulse [s]
\( td = 0.01 \); % Characteristic period of Ricker pulse [s]
\( pz = (2*(pi*(t-ts)/td).^2-1).*exp(-(pi*(t-ts)/td).^2); \) % Ricker pulse
\( Pz = \text{fft}(pz) \); % Frequency content of Ricker pulse
SHAPE FUNCTIONS

Pz = Pz(1:nFreq); \% Positive spectrum of vertical load
P = zeros(4,nFreq); \% Load vector for each positive frequency
P(2,:) = Pz; \% Load vector for each positive frequency

\% COMPUTE RESPONSE FOR EACH POSITIVE FREQUENCY
Vz = zeros(nRec,nFreq);
Tz = zeros(nRec,nFreq);
for iFreq=2:nFreq
    K = k_dampsv(h,Cs,Cp,Ds,Dp,rho,k,omega(iFreq));
    [Nx,Nz] = n_dampsv(h,Cs,Cp,Ds,Dp,rho,k,omega(iFreq),z);
    [Bx,Bz] = b_dampsv(h,Cs,Cp,Ds,Dp,rho,k,omega(iFreq),z);
    U = K\P(:,iFreq);
    Vz(:,iFreq) = i*omega(iFreq)*Nz*U;
    Tz(:,iFreq) = Bz*U;
end

\% RESPONSE IN TIME DOMAIN
Vz = [Vz,zeros(nRec,1),conj(Vz(:,end:-1:2))]; \% Add negative spectrum
Tz = [Tz,zeros(nRec,1),conj(Tz(:,end:-1:2))]; \% Add negative spectrum
Vz = ifft(Vz,[1,2]); \% Transform along rows
Tz = ifft(Tz,[1,2]); \% Transform along rows

\% PLOT VELOCITY
figure;
wiggle(z,t,Vz);
xlabel('Depth [m]');
ylabel('Time [s]');

\% PLOT TRACTION
figure;
wiggle(z,t,Tz);
xlabel('Depth [m]');
ylabel('Time [s]');

The time history of the vertical velocity $v_z(z,t)$ and the vertical traction $t_z(z,t)$ on a horizontal plane at 21 depths $z$ is shown in figure 4.7. From time $t = 0$ s to time $t = 0.1$ s, when the pulse is applied, the velocity $v_z(z,t)$ and the traction $t_z(z,t)$ remain zero. The wave velocity $C_p$ of the layer is equal to 400 m/s; the wave therefore travels from the top to the bottom of the layer in a period of 0.025 s. At the bottom of the layer, the wave is partly refracted into the halfspace and partly reflected back into the layer. The impedance $\rho C_p$ of the halfspace is higher than the impedance of the layer. As a result, the velocity $v_z(z,t)$ switches sign as the wave is reflected. The sign of the traction $t_z(z,t)$ does not change. At the surface, the wave is reflected. The surface is a free boundary; as a result, the traction $t_z(z,t)$ switches sign as the wave is reflected and the sign of the velocity $v_z(z,t)$
remains unchanged. Both in the layer and the halfspace, the waves are attenuated due to material damping. Additional attenuation occurs due to the radiation of energy into the halfspace.

Figure 4.7: Time history of (a) the velocity $v_z(z,t)$ and (b) the vertical traction $t_z(z,t)$ on a horizontal plane at 21 depths $z$ in a layer on a halfspace loaded by a Ricker pulse.
Chapter 5

The thin layer method

The thin layer method [12] is an alternative to the direct stiffness method for wave propagation in layered media. It is based on the use of polynomial shape functions to represent the vertical variation of displacements and tractions. Compared to the direct stiffness method, the thin layer method leads to mathematically more tractable stiffness matrices involving only polynomial functions instead of transcendental functions. Due to its approximative nature, the thin layer method requires a small thickness of the layer elements compared to the smallest relevant wavelength. Furthermore, the method is only applicable to media with a finite thickness, with either free-free boundary conditions or supported by a rigid stratum. A hybrid formulation, where thin layer elements are coupled to a halfspace element, offers a solution to model a layered halfspace, but again leads to transcendental functions in the stiffness matrix.

This chapter is organized as follows:

**Stiffness matrices (p. 52)**
This section addresses the calculation of stiffness matrices with EDT according to the thin layer method. The stiffness matrix of a layer element is discussed, as well as the assembly of the global stiffness matrix of a layered medium.

**Shape functions (p. 56)**
This section addresses the shape functions used in the direct stiffness method. Both the shape functions for the displacements and the shape functions for the tractions (on a horizontal plane) are discussed. These functions can be used to determine the displacements and the tractions in the interior of the elements if the interface displacements are known.
5.1 Stiffness matrices

5.1.1 The element stiffness matrix

While the direct stiffness method allows for the use of layer and halfspace elements, the thin layer method only provides a layer element.

\[ \tilde{K}_e \tilde{U}^e = \tilde{F}^e \]  

(5.1)

Figure 5.1 shows the loads \( \tilde{F}^e = (\tilde{P}^e_1, \tilde{P}^e_2, \tilde{P}^e_3, \tilde{P}^e_4) \) and displacements \( \tilde{U}^e = (\tilde{u}^e_1, \tilde{u}^e_2, \tilde{u}^e_3) \) at the boundaries of a layer element. In a similar way as in the direct stiffness method, the loads \( \tilde{F}^e \) are related to the displacements \( \tilde{U}^e \) through the element stiffness matrix \( \tilde{K}_e \):

\[ \tilde{K}_e \tilde{U}^e = \tilde{F}^e \]

(5.1)

The relation (5.1) between the loads \( \tilde{F}^e \) and the displacements \( \tilde{U}^e \) and can be decomposed as:

\[ \begin{align*}
\tilde{K}^e_{PSV} \tilde{U}^e_{PSV} &= \tilde{F}^e_{PSV} \\
\tilde{K}^e_{SH} \tilde{U}^e_{SH} &= \tilde{F}^e_{SH}
\end{align*} \]

(5.2)  

(5.3)

where \( \tilde{U}^e_{PSV} = (\tilde{u}^e_1, \tilde{u}^e_2, \tilde{u}^e_3) \) and \( \tilde{F}^e_{PSV} = (\tilde{P}^e_1, \tilde{P}^e_2, \tilde{P}^e_3) \) are related to P-SV-wave propagation and \( \tilde{U}^e_{SH} = (\tilde{u}^e_1, \tilde{u}^e_2) \) and \( \tilde{F}^e_{SH} = (\tilde{P}^e_1, \tilde{P}^e_2) \) are related to SH-wave propagation. The \( 4 \times 4 \) matrix \( \tilde{K}^e_{PSV} \) and the \( 2 \times 2 \) matrix \( \tilde{K}^e_{SH} \) are submatrices of the element stiffness matrix \( \tilde{K}^e \).

Due to the use of linear shape functions for the displacements, the element stiffness matrix \( \tilde{K}^e \) only involves polynomial functions of the horizontal wavenumber \( k_x \) and the circular frequency \( \omega \). This is in contrast to the direct stiffness method, where exact solutions of the wave equation are used as shape functions and the stiffness matrix consequently involves transcendental functions of the wavenumber \( k_x \) and
the frequency $\omega$. In the thin layer method, the element stiffness matrix $\tilde{K}^e$ can be decomposed as:

$$\tilde{K}^e = k_x^2 \tilde{A}^e + k_x \tilde{B}^e + \tilde{C}^e - \omega^2 \tilde{M}^e$$ (5.4)

where the matrices $\tilde{A}^e$, $\tilde{B}^e$, $\tilde{G}^e$, and $\tilde{M}^e$ are independent of the wavenumber $k_x$ and the frequency $\omega$. Analytical expressions for these matrices are presented in references [11, 12]. Equation (5.4) is valid for positive frequencies $\omega > 0$. For negative frequencies $\omega < 0$, the matrices $\tilde{A}^e$, $\tilde{B}^e$, $\tilde{G}^e$, and $\tilde{M}^e$ must be replaced with their complex conjugates. For zero frequencies $\omega = 0$, only the real part of the matrices $\tilde{A}^e$, $\tilde{B}^e$, $\tilde{G}^e$, and $\tilde{M}^e$ has to be taken into account.

The matrices $\tilde{K}^e_{PSV}$ and $\tilde{K}^e_{SH}$ of the element stiffness matrix $\tilde{K}^e$ can be decomposed in a similar way:

$$\tilde{K}^e_{PSV} = k_x^2 \tilde{A}^e_{PSV} + k_x \tilde{B}^e_{PSV} + \tilde{G}^e_{PSV} - \omega^2 \tilde{M}^e_{PSV}$$ (5.5)

$$\tilde{K}^e_{SH} = k_x^2 \tilde{A}^e_{SH} + k_x \tilde{B}^e_{SH} + \tilde{G}^e_{SH} - \omega^2 \tilde{M}^e_{SH}$$ (5.6)

The matrices $\tilde{A}^e_{PSV}$, $\tilde{B}^e_{PSV}$, $\tilde{G}^e_{PSV}$, and $\tilde{M}^e_{PSV}$ can be calculated with EDT using the function $ke_{tlmPSV}$. Similarly, the matrices $\tilde{A}^e_{SH}$, $\tilde{B}^e_{SH}$, $\tilde{G}^e_{SH}$, and $\tilde{M}^e_{SH}$ can be computed with the function $ke_{tlmSH}$. EDT also provides a function $addk_tlm$ to compute the stiffness matrices $\tilde{K}^e_{PSV}$ and $\tilde{K}^e_{SH}$ according to equations (5.5) and (5.6). This function exploits the identical sparsity pattern of the matrices involved, so improving the efficiency of the calculation.

### 5.1.2 Assembly of equations

This subsection focuses on the global stiffness matrix $\tilde{K}$ for a layered medium consisting of a number of layer elements. The assembly of the global stiffness matrix $\tilde{K}$ from the element stiffness matrices $\tilde{K}^e$ proceeds along the same lines as in the direct stiffness method (subsection 4.1.3). In terms of the global stiffness matrix $\tilde{K}$, the equilibrium of the layered medium is expressed as:

$$\tilde{K}\tilde{U} = \tilde{P}$$ (5.7)

where the vectors $\tilde{P}$ and $\tilde{U}$ collect the loads $\tilde{p}^i$ and the displacements $\tilde{u}^i$ at all interfaces $i$ between elements. The stiffness matrix $\tilde{K}$ has the following form:

$$\tilde{K} = k_x^2 \tilde{A} + k_x \tilde{B} + \tilde{G} - \omega^2 \tilde{M}$$ (5.8)

where the global matrices $\tilde{A}$, $\tilde{B}$, $\tilde{G}$, and $\tilde{M}$ are composed of the corresponding element matrices $\tilde{A}^e$, $\tilde{B}^e$, $\tilde{G}^e$, and $\tilde{M}^e$.

In view of the uncoupling of P-SV-waves and SH-waves, it is advantageous to assemble separate stiffness matrices $\tilde{K}_{PSV}$ and $\tilde{K}_{SH}$, so that equation (5.7) can be
reformulated as:

\[
\begin{align*}
\tilde{K}_{PSV} \tilde{U}_{PSV} &= \tilde{P}_{PSV} \\
\tilde{K}_{SH} \tilde{U}_{SH} &= \tilde{P}_{SH}
\end{align*}
\] (5.9)

\[
\begin{align*}
\tilde{K}_{PSV} &= \tilde{k}_x^2 \tilde{A}_{PSV} + \tilde{k}_x \tilde{B}_{PSV} + \tilde{G}_{PSV} - \omega^2 \tilde{M}_{PSV} \\
\tilde{K}_{SH} &= \tilde{k}_y^2 \tilde{A}_{SH} + \tilde{k}_y \tilde{B}_{SH} + \tilde{G}_{SH} - \omega^2 \tilde{M}_{SH}
\end{align*}
\] (5.10)

where \( \tilde{P}_{PSV} \) and \( \tilde{U}_{PSV} \) collect the loads and displacements in \( x \) and \( z \)-direction and \( \tilde{P}_{SH} \) and \( \tilde{U}_{SH} \) collect the loads and displacements in \( y \)-direction.

The stiffness matrices \( \tilde{K}_{PSV} \) and \( \tilde{K}_{SH} \) are given by:

\[
\begin{align*}
\tilde{K}_{PSV} &= \tilde{k}_x^2 \tilde{A}_{PSV} + \tilde{k}_x \tilde{B}_{PSV} + \tilde{G}_{PSV} - \omega^2 \tilde{M}_{PSV} \\
\tilde{K}_{SH} &= \tilde{k}_y^2 \tilde{A}_{SH} + \tilde{k}_y \tilde{B}_{SH} + \tilde{G}_{SH} - \omega^2 \tilde{M}_{SH}
\end{align*}
\] (5.11) (5.12)

The function \( \text{addk_tlm} \) can be used to evaluate these equations in an efficient manner.

The global matrices \( \tilde{A}_{PSV}, \tilde{B}_{PSV}, \tilde{G}_{PSV}, \) and \( \tilde{M}_{PSV} \) in equation (5.11) are assembled from the element matrices \( \tilde{A}_{PSV}^e, \tilde{B}_{PSV}^e, \tilde{G}_{PSV}^e, \) and \( \tilde{M}_{PSV}^e \). Similarly, the global matrices \( \tilde{A}_{SH}, \tilde{B}_{SH}, \tilde{G}_{SH}, \) and \( \tilde{M}_{SH} \) in equation (5.12) are assembled from the element matrices \( \tilde{A}_{SH}^e, \tilde{B}_{SH}^e, \tilde{G}_{SH}^e, \) and \( \tilde{M}_{SH}^e \).

The matrices \( \tilde{A}_{PSV}, \tilde{B}_{PSV}, \tilde{G}_{PSV}, \tilde{M}_{PSV}, \tilde{A}_{SH}, \tilde{B}_{SH}, \tilde{G}_{SH}, \) and \( \tilde{M}_{SH} \) can be obtained with EDT in three different ways. The first method is to use the functions \( \text{ke_tlmpsv} \) and \( \text{ke_tlmsh} \) to compute the element matrices \( \tilde{A}_{PSV}^e, \tilde{B}_{PSV}^e, \tilde{G}_{PSV}^e, \) and \( \tilde{M}_{PSV}^e \). Subsequently, these matrices are used in an assembly procedure based on standard MATLAB functions. The second method also starts with the calculation of the element matrices \( \tilde{A}_{PSV}^e, \tilde{B}_{PSV}^e, \tilde{G}_{PSV}^e, \) \( \tilde{M}_{PSV}^e, \tilde{A}_{SH}^e, \tilde{B}_{SH}^e, \tilde{G}_{SH}^e, \) and \( \tilde{M}_{SH}^e \) using the functions \( \text{ke_tlmpsv} \) and \( \text{ke_tlmsh} \). Next, the assembly is performed by means of the functions \( \text{asmk_psv} \) and \( \text{asmk_sh} \). The third method is the most convenient method: the global matrices \( \tilde{A}_{PSV}, \tilde{B}_{PSV}, \tilde{G}_{PSV}, \tilde{M}_{PSV}, \tilde{A}_{SH}, \tilde{B}_{SH}, \tilde{G}_{SH}, \) and \( \tilde{M}_{SH} \) are computed directly by means of the functions \( \text{k_tlmpsv} \) and \( \text{k_tlmsh} \). Internally, these functions perform both the calculation of the element matrices and the assembly of the global matrices.

**Example 5.1: Vertical harmonic wave propagation in a layer on bedrock.**

This example reconsiders the problem introduced in example 4.2 (p. 37), where one-dimensional harmonic wave propagation in a homogeneous layer on bedrock is studied. In the present example, the problem is solved by means of the thin layer method instead of the direct stiffness method.

The use of the thin layer method implies that the medium must be subdivided into a number of layer elements with a thickness that is sufficiently small compared to the wavelength of the waves in the medium. In the present example, the maximum frequency considered is 40 Hz. At this frequency, the wavelength of the P-waves
is $\lambda_p = C_p/f = 5$ m. The medium is therefore subdivided into 40 layer elements, with a thickness of 0.25 m, which is small compared to the wavelength $\lambda_p$.

The following MATLAB code is used. First, the thin layer matrices $\tilde{A}_{PSV}, \tilde{B}_{PSV}, \tilde{G}_{PSV},$ and $\tilde{M}_{PSV}$ are assembled. The last two rows and columns of these matrices, corresponding to the bottom interface, are eliminated in order to comply with the clamped boundary condition. Next, for all frequencies, the stiffness matrix $\tilde{K}_{PSV}$ is computed using the thin layer matrices, and the equilibrium equations are solved. Finally, the vertical response at the surface is plotted.

```matlab
% SOIL PROPERTIES
nElt = 40; % Number of elements [-]
H = 10; % Total thickness [m]
h = repmat(H/nElt,nElt,1); % Element thickness [m]
Cs = 141; % Shear wave velocity [m/s]
Cp = 200; % Dilatational wave velocity [m/s]
Ds = 0.02; % Shear damping ratio [-]
Dp = 0.02; % Dilatational damping ratio [-]
rho = 2000; % Density [kg/m^3]

% FREQUENCY AND WAVENUMBER
k = 0; % Horizontal wavenumber [rad/m]
f = 0:0.1:40; % Frequency [Hz]
omega = 2*pi*f; % Frequency [rad/s]
nFreq = length(f); % Number of frequencies

% EXCITATION
P = zeros(2*nElt,1);
P(2) = 1;

% THIN LAYER MATRICES
[A,B,G,M] = k_tlmpsv(h,Cs,Cp,Ds,Dp,rho);
A=A(1:end-2,1:end-2);
B=B(1:end-2,1:end-2);
G=G(1:end-2,1:end-2);
M=M(1:end-2,1:end-2);

% COMPUTE DISPLACEMENT FOR EACH FREQUENCY
uz = zeros(nFreq,1);
for iFreq=1:nFreq
    K = addk_tlm(A,B,G,M,k,omega(iFreq));
    U = K\P;
    uz(iFreq) = U(2);
end
```
The results of the calculation are shown in figure 5.2. This figure also shows the response obtained with 8 instead of 40 layer elements. Peaks are visible at the resonance frequencies of the layer. The first resonance frequency equals 5 Hz. At this frequency, the results obtained with 8 elements are very similar to the results obtained with 40 elements. This is no longer the case at the fourth resonance frequency, which equals 35 Hz. The peak corresponding to this resonance frequency is well reproduced with the 40 element model, but not with the 8 element model, where it occurs at a higher frequency. This is explained by the large element thickness in the 8 element model compared to the wavelength of the P-waves at 35 Hz. As in the finite element method, the use of relatively large elements generally leads to an overestimation of the stiffness and, consequently, the resonance frequencies.

5.2 Shape functions

5.2.1 Shape functions for the displacements

The solution of the equilibrium equations (5.9) and (5.10) leads to the displacements $\tilde{U}^e$ at the boundaries of each element $e$. The displacements $\tilde{u}(z)$ in the
interior of element \( e \) can subsequently be computed by evaluation of the element
shape functions \( \tilde{N}^e(z) \):

\[
\tilde{u}(z) = \tilde{N}^e(z) \tilde{U}^e
\]  

(5.13)

Due to the uncoupling of P-SV-waves and SH-waves, this equation can be
reformulated as:

\[
\tilde{u}_{PSV}(z) = \tilde{N}_{PSV}^e(z) \tilde{U}_{PSV}^e
\]  

(5.14)

\[
\tilde{u}_{SH}(z) = \tilde{N}_{SH}^e(z) \tilde{U}_{SH}^e
\]  

(5.15)

where \( \tilde{u}_{PSV}(z) = \{\tilde{u}_x(z),\tilde{u}_z(z)\}^T \) and \( \tilde{u}_{SH}(z) = \{\tilde{u}_y(z)\} \). Considering each
component \( \tilde{u}_i(z) \) of the displacement vector \( \tilde{u}(z) \) separately, the following
equations are obtained:

\[
\tilde{u}_x(z) = \tilde{N}_x^e(z) \tilde{U}_{PSV}^e
\]  

(5.16)

\[
\tilde{u}_y(z) = \tilde{N}_y^e(z) \tilde{U}_{SH}^e
\]  

(5.17)

\[
\tilde{u}_z(z) = \tilde{N}_z^e(z) \tilde{U}_{PSV}^e
\]  

(5.18)

where the vector \( \tilde{N}_i^e(z) \) collects the shape functions for the displacements in
element \( e \) in direction \( i \).

In EDT, the element shape functions \( \tilde{N}_i^e(z) \) are computed with the function \texttt{ne_tlmsh}. This function evaluates the shape functions at \( N \) user defined
depths \( z \) and returns the result as a matrix with dimensions \( N \times 4 \). The depth \( z \) has to be specified in the element coordinate system. The shape functions \( \tilde{N}_i^e(z) \) are computed in a similar way with the function \texttt{ne_tlmsh}. The resulting matrix
has dimensions \( N \times 2 \).

The displacements \( \tilde{u}(z) \) at depth \( z \) in a layered medium can be calculated as:

\[
\tilde{u}_x(z) = \tilde{N}_x(z) \tilde{U}_{PSV}
\]  

(5.19)

\[
\tilde{u}_y(z) = \tilde{N}_y(z) \tilde{U}_{SH}
\]  

(5.20)

\[
\tilde{u}_z(z) = \tilde{N}_z(z) \tilde{U}_{PSV}
\]  

(5.21)

where the vectors \( \tilde{U}_{PSV} \) and \( \tilde{U}_{SH} \) collect the displacements at all interfaces between elements and the vectors \( \tilde{N}_i(z) \) are global shape functions assembled
from the element shape functions \( \tilde{N}_i^e(z) \).

In EDT, the global shape functions \( \tilde{N}_i(z) \) can be obtained with the functions \texttt{n_tlmsh}. These functions assemble the global shape functions from
the element shape functions and evaluate them at \( N \) user defined depths \( z \). The
depth \( z \) must be specified the global coordinate system.
5.2.2 Shape functions for the tractions

The tractions $\mathbf{t}(z)$ on a horizontal plane at depth $z$ in an element $e$ can be computed using the element shape functions $\mathbf{B}^e(z)$:

$$\mathbf{t}(z) = \mathbf{B}^e(z) \mathbf{U}^e$$  \hspace{1cm} (5.22)

where the vector $\mathbf{U}^e$ collects the displacements at the boundaries of the element $e$. Due to the uncoupling of P-SV-waves and SH-waves, equation (5.22) can be reformulated as:

$$\mathbf{t}_{PSV}(z) = \mathbf{B}_{PSV}^e(z) \mathbf{U}_{PSV}^e$$  \hspace{1cm} (5.23)

$$\mathbf{t}_{SH}(z) = \mathbf{B}_{SH}^e(z) \mathbf{U}_{SH}^e$$  \hspace{1cm} (5.24)

where $\mathbf{t}_{PSV}(z) = \{t_x(z), t_z(z)\}^T$ and $\mathbf{t}_{SH}(z) = \{t_y(z)\}$. Considering each component $\mathbf{t}_i(z)$ of the traction vector $\mathbf{t}(z)$ separately, the following equations are obtained:

$$t_x(z) = \mathbf{B}_x^e(z) \mathbf{U}_{PSV}^e$$  \hspace{1cm} (5.25)

$$t_y(z) = \mathbf{B}_y^e(z) \mathbf{U}_{SH}^e$$  \hspace{1cm} (5.26)

$$t_z(z) = \mathbf{B}_z^e(z) \mathbf{U}_{PSV}^e$$  \hspace{1cm} (5.27)

where the vector $\mathbf{B}_i^e(z)$ collects the shape functions for the tractions in element $e$ in direction $i$.

In EDT, the element shape functions $\mathbf{B}_i^e(z)$ are computed with the function $\text{be}_{\text{tlmpsv}}$. This function evaluates the shape functions at $N$ user defined depths $z$ and returns the result as a matrix with dimensions $N \times 4$. The depth $z$ has to be specified in the element coordinate system. The shape functions $\mathbf{B}_i^e(z)$ are computed in a similar way with the function $\text{be}_{\text{tlmsh}}$. The resulting matrix has dimensions $N \times 2$.

The tractions $\mathbf{t}(z)$ on a horizontal plane at depth $z$ in a layered medium can be calculated as:

$$\mathbf{t}_x(z) = \mathbf{B}_x(z) \mathbf{U}_{PSV}$$  \hspace{1cm} (5.28)

$$\mathbf{t}_y(z) = \mathbf{B}_y(z) \mathbf{U}_{SH}$$  \hspace{1cm} (5.29)

$$\mathbf{t}_z(z) = \mathbf{B}_z(z) \mathbf{U}_{PSV}$$  \hspace{1cm} (5.30)

where the vectors $\mathbf{U}_{PSV}$ and $\mathbf{U}_{SH}$ collect the displacements at all interfaces between elements and the vectors $\mathbf{B}_i(z)$ are global shape functions assembled from the element shape functions $\mathbf{B}_i^e(z)$.

The global shape functions $\mathbf{B}_i(z)$ can be obtained in EDT with the functions $\text{b}_{\text{tlmpsv}}$ and $\text{b}_{\text{tlmsh}}$. These functions assemble the global shape functions from the element shape functions and evaluate them at $N$ user defined depths $z$. The depth $z$ must be specified in the global coordinate system.
Chapter 6

Fourier and Hankel transformation algorithms

The solution of problems governed by two-dimensional or three-dimensional wave propagation in layered media is based on an integral transformation from the space domain to the wavenumber domain. In the two-dimensional case, a Fourier transformation is performed from the horizontal coordinate $x$ to the horizontal wavenumber $k_x$. In the three-dimensional case, a Fourier series expansion is performed from the circumferential coordinate $\theta$ to the circumferential wavenumber $n$, followed by a Hankel transformation from the radial coordinate $r$ to the radial wavenumber $k_r$.

This chapter focuses on the Fourier and Hankel transformation algorithm developed by Talman [27]. This algorithm is based on a logarithmic change of variables to express the Fourier or Hankel transformation as a convolution, which is efficiently evaluated using an FFT algorithm. The original algorithm is improved through the use of a window and a filter to mitigate artifacts caused by the Gibbs phenomenon [25].

Transformation algorithms based on a logarithmic sampling scheme are particularly efficient for problems involving considerably different length scales. This is the case for the Green’s functions of a layered medium, which are computed in the wavenumber domain and subsequently transformed to the space domain. Often, both the response in the near field and the radiation of waves to the far field are of interest, and the Green’s functions have to be computed over a large wavenumber range. In the high wavenumber range, the Green’s functions are very smooth, and the sampling interval can be relaxed, resulting in a logarithmic sampling scheme.

This chapter is composed of the following sections:
Fourier transformations (p. 60)
This section focuses on the Fourier transformation of functions in logarithmic variables. First, the decomposition of a Fourier transformation into a Fourier sine transformation and a Fourier cosine transformation is briefly reviewed. Next, the evaluation of the Fourier sine and cosine transformations of functions in logarithmic variables is discussed.

Hankel transformations (p. 65)
This section addresses the Hankel transformation of functions in logarithmic variables.

6.1 Fourier transformations

This section addresses the numerical evaluation of the Fourier transformations introduced in subsection 2.5.1.

The forward and inverse Fourier transformations between the spatial coordinate \(x\) and the wavenumber \(k_x\) are defined as:

\[
\hat{f}(k_x) = \int_{-\infty}^{\infty} e^{ik_x x} f(x) \, dx \tag{6.1}
\]

\[
f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-ik_x x} \hat{f}(k_x) \, dk_x \tag{6.2}
\]

Using the identity \(e^{\pm ik_x x} = \cos(k_x x) \pm i \sin(k_x x)\), and since the sine and cosine functions are respectively odd and even, equations (6.1) and (6.2) are equivalent to:

\[
\hat{f}(k_x) = \int_{0}^{\infty} \cos(k_x x) f(-x) \, dx + \int_{0}^{\infty} \cos(k_x x) f(x) \, dx
\]

\[-i \int_{0}^{\infty} \sin(k_x x) f(-x) \, dx + i \int_{0}^{\infty} \sin(k_x x) f(x) \, dx \tag{6.3}
\]

\[
f(x) = \frac{1}{2\pi} \int_{0}^{\infty} \cos(k_x x) \hat{f}(-k_x) \, dx + \frac{1}{2\pi} \int_{0}^{\infty} \cos(k_x x) \hat{f}(k_x) \, dx
\]

\[+ \frac{i}{2\pi} \int_{0}^{\infty} \sin(k_x x) \hat{f}(-k_x) \, dx - \frac{i}{2\pi} \int_{0}^{\infty} \sin(k_x x) \hat{f}(k_x) \, dx \tag{6.4}
\]

These equations can be simplified if the functions \(f(x)\) and/or \(\hat{f}(k_x)\) have specific properties, e.g. if they are real or imaginary, or odd or even.
The forward and inverse Fourier transformations between the time $t$ and the circular frequency $\omega$ are defined as:

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} e^{-i\omega t} f(t) \, dt$$  \hspace{1cm} (6.5)$$
$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\omega t} \hat{f}(\omega) \, d\omega$$  \hspace{1cm} (6.6)$$

These equations are reformulated as:

$$\hat{f}(\omega) = \int_{0}^{\infty} \cos(\omega t) f(-t) \, dt + \int_{0}^{\infty} \cos(\omega t) f(t) \, dt$$  
$$+ i \int_{0}^{\infty} \sin(\omega t) f(-t) \, dt - i \int_{0}^{\infty} \sin(\omega t) f(t) \, dt$$  \hspace{1cm} (6.7)$$
$$f(t) = \frac{1}{2\pi} \int_{0}^{\infty} \cos(\omega t) \hat{f}(-\omega) \, d\omega + \frac{1}{2\pi} \int_{0}^{\infty} \cos(\omega t) \hat{f}(\omega) \, d\omega$$  
$$- \frac{i}{2\pi} \int_{0}^{\infty} \sin(\omega t) \hat{f}(-\omega) \, d\omega + \frac{i}{2\pi} \int_{0}^{\infty} \sin(\omega t) \hat{f}(\omega) \, d\omega$$  \hspace{1cm} (6.8)$$

It is now clear that the Fourier transformations in equations (6.1), (6.2), (6.5), and (6.6) can be expressed in terms of integral transformations of the following type:

$$\hat{f}_c(k) = \int_{0}^{\infty} \cos(kx) f(x) \, dx$$  \hspace{1cm} (6.9)$$
$$\hat{f}_s(k) = \int_{0}^{\infty} \sin(kx) f(x) \, dx$$  \hspace{1cm} (6.10)$$

The numerical evaluation of equation (6.9) is addressed in the following. Equation (6.10) can be evaluated in a similar way.

A change of variables is performed using $k_x = k_0 e^v$ and $x = x_0 e^{-w}$, so that equation (6.9) becomes:

$$\hat{f}_c(k_0 e^v) = \int_{-\infty}^{\infty} \cos(k_0 x_0 e^{v-w}) f(x_0 e^{-w}) x_0 e^{-w} \, dw$$  \hspace{1cm} (6.11)$$
$$= x_0 e^{-\frac{v}{2}} \int_{-\infty}^{\infty} \cos(k_0 x_0 e^{v-w}) e^{\frac{v-w}{2}} f(x_0 e^{-w}) e^{-\frac{w}{2}} \, dw$$  \hspace{1cm} (6.12)$$

The reference wavenumber $k_0$ and distance $x_0$ may be chosen arbitrarily. The factor $e^{-\frac{v}{2}} e^{\frac{v}{2}}$ is introduced in equation (6.12) to improve numerical stability [27]. Equation (6.12) is reformulated as:

$$\hat{f}_c(k_0 e^w) = x_0 e^{-\frac{w}{2}} \chi(w)$$  \hspace{1cm} (6.13)$$

where the function $\chi(w)$ is defined as the following convolution:

$$\chi(w) = \varphi(w) * \psi(w)$$  \hspace{1cm} (6.14)$$
The functions $\varphi(w)$ and $\psi(w)$ are given by:

$$\varphi(w) = f(x_0 e^{-w})e^{-\frac{w^2}{2}}$$

(6.15)

$$\psi(w) = \cos(k_0 x_0 e^w)e^{\frac{w^2}{2}}$$

(6.16)

The convolution theorem is used to evaluate equation (6.14). The Fourier transformations of the kernel $\psi(w)$ and the function $\varphi(w)$ are calculated analytically and with an FFT algorithm, respectively. The product of both Fourier transformations is transformed back by means of an inverse FFT in order to obtain the function $\chi(w)$. The use of an FFT algorithm requires that the $w$-axis is sampled with a constant sampling interval $\Delta w$. This implies that the $k_x$-axis and the $x$-axis must be sampled according to a logarithmic scheme, so that:

$$\ln \left( \frac{k_{n+1}}{k_n} \right) = \Delta w$$

(6.17)

$$\ln \left( \frac{x_{n+1}}{x_n} \right) = \Delta w$$

(6.18)

EDT provides the functions `logffcos` and `logffsin` to evaluate the Fourier sine and cosine transformations in equations (6.9) and (6.10) according to the procedure outlined above. These functions require the input sampling $x_n$ to be logarithmic, while they use spline interpolation to compute the value $\tilde{f}_c(k_x)$ (or $\tilde{f}_s(k_x)$) at any output point $k_x$.

The sampling $x_n$ is determined by three parameters: the minimum $x_{\text{min}}$, the maximum $x_{\text{max}}$, and the number of samples $N$. The parameters $x_{\text{min}}$ and $x_{\text{max}}$ determine the accuracy of the transformed function $\tilde{f}(k_x)$ in the high and the low wavenumber range, respectively. The parameter $N$ must be chosen so that the function $f(x)$ is sampled at a sufficiently fine resolution.

In order to avoid aliasing in the computation of the convolution in equation (6.14), the function $f(x)$ is padded at both sides. The amount of padding is determined automatically by the functions `logffcos` and `logffsin`, following a procedure outlined in reference [25]. The function $f(x)$ is extrapolated proportionally to $x^{q_1}$ as $x$ tends to 0 and proportionally to $x^{q_2}$ as $x$ tends to $\infty$. The exponents $q_1$ and $q_2$ are determined from the first two and the last two samples of the function $f(x)$.

### Example 6.1: The response of a layered halfspace due to a harmonic line load.

This example considers two-dimensional harmonic wave propagation in a layer on a halfspace. The layer has a shear wave velocity $C_s = 200 \text{ m/s}$, a dilatational wave velocity $C_p = 400 \text{ m/s}$, a damping ratio $D_s = D_p = 0.05$ in both shear and volumetric deformation, and a density $\rho = 1800 \text{ kg/m}^3$. The halfspace has a shear wave velocity $C_s = 500 \text{ m/s}$ and a dilatational wave velocity $C_p = 2000 \text{ m/s}$. The other properties of the halfspace are identical to those of the layer.
The layered halfspace is loaded at the free surface with a vertical harmonic line load. The amplitude of the load is \( \hat{p} = \{0, 0, \delta(x)\}^T \), where \( \delta(x) \) denotes the Dirac delta function. The excitation frequency is \( f = 100 \text{ Hz} \).

Following the procedure outlined in subsection 3.3.1, the load \( \hat{p} \) is first transformed from the space domain to the wavenumber domain. In the wavenumber domain, the load is given by \( \hat{p}(k_x) = \{0, 0, 1\}^T \). Next, the vertical component \( \tilde{u}_z(k_x, z) \) of the response \( \tilde{u}(k_x, z) \) in the wavenumber domain is computed: the EDT function \texttt{green} \_\texttt{zz} is used to compute the displacement \( \tilde{u}(k_x, z) \) for 800 logarithmically spaced values of the horizontal slowness \( p_x = k_x/\omega \) between \( 10^{-5} \text{s/m} \) and \( 10^{3} \text{s/m} \). More details on the use of the function \texttt{green} \_\texttt{zz} are given in chapter 9, along with guidelines for the horizontal wavenumber (or horizontal slowness) sampling. Finally, the response \( \hat{u}_z(x, z) \) is transformed from the wavenumber domain to the space domain by means of an inverse Fourier transformation:

\[
\hat{u}_z(x, z) = \frac{1}{2\pi} e^{-ik_x x} \tilde{u}_z(k_x, z) \, dk_x \tag{6.19}
\]

which can be reformulated according to equation (6.4):

\[
\hat{u}_z(x, z) = \frac{1}{2\pi} \int_{0}^{\infty} \cos(k_x x)\tilde{u}_z(-k_x, z) \, dx + \frac{1}{2\pi} \int_{0}^{\infty} \cos(k_x x)\tilde{u}_z(k_x, z) \, dx \\
+ \frac{i}{2\pi} \int_{0}^{\infty} \sin(k_x x)\tilde{u}_z(-k_x, z) \, dx - \frac{i}{2\pi} \int_{0}^{\infty} \sin(k_x x)\tilde{u}_z(k_x, z) \, dx \tag{6.20}
\]

The wave field \( \hat{u}(x, z) \) induced by the line load \( \hat{p}(x) = \{0, 0, \delta(x)\}^T \) is symmetric with respect to the \((y, z)\)-plane. As a consequence, (1) the function \( \hat{u}_z(x, z) \) is even with respect to \( x \), so that the sine terms in equation (6.20) must be zero, and (2) the function \( \tilde{u}_z(k_x, z) \) is even with respect to \( k_x \), so that the first and the second integral in equation (6.20) are identical. Equation (6.20) therefore reduces to:

\[
\hat{u}_z(x, z) = \frac{1}{\pi} \int_{0}^{\infty} \cos(k_x x)\tilde{u}_z(k_x, z) \, dx \tag{6.21}
\]

This integral transformation is evaluated by means of the function \texttt{logffcos}.

% SOIL PROPERTIES
h = [10 inf]; % Element thickness [m]
Cs = [200 500]; % Shear wave velocity [m/s]
Cp = [400 2000]; % Dilatational wave velocity [m/s]
Ds = 0.05; % Shear damping ratio [-]
Dp = 0.05; % Dilatational damping ratio [-]
rho = 1800; % Density [kg/m^3]

% SAMPLING PARAMETERS
\[ f = 100; \quad \% \text{Frequency [Hz]} \]
\[ \text{omega} = 2\pi f; \quad \% \text{Frequency [rad/s]} \]
\[ \text{p} = \logspace(-5,3,800); \quad \% \text{Horizontal slowness [s/m]} \]
\[ \text{k} = \text{omega} \times \text{p}; \quad \% \text{Horizontal wavenumber [rad/m]} \]
\[ \text{zs} = 0; \quad \% \text{Source depth [m]} \]
\[ x = 0.01:0.01:10; \quad \% \text{Receiver distance [m]} \]
\[ z = 0; \quad \% \text{Receiver depth [m]} \]

\% GREEN'S FUNCTION IN FREQUENCY-WAVENUMBER DOMAIN
\[ \text{Ugzz} = \text{green_zz(h,Cs,Cp,Ds,Dp,rho,zs,p,z,omega);} \]

\% TRANSFORMATION TO FREQUENCY-SPACE DOMAIN
\[ \text{ugzz} = \frac{1}{\pi} \logffcos(\text{Ugzz},k,x); \]

\% PLOT RESULT
\begin{verbatim}
figure;
plot(x,real(ugzz),x,imag(ugzz));
xlabel('Distance [m]');
ylabel('Displacement [m/(N/m)]');
\end{verbatim}

Figure 6.1: Real (solid line) and imaginary (dashed line) part of the vertical displacement \( \hat{u}_z(x,z) \) at the surface (\( z = 0 \) m) of a halfspace loaded by a harmonic line load.

The resulting displacement \( \hat{u}_z(x,z) \) at the surface (\( z = 0 \) m) of the layered halfspace is shown in figure 6.1. The response at the surface is dominated by the Rayleigh wave. The Rayleigh wave velocity is \( C_R = 186.5 \text{ m/s} \) [18], and the Rayleigh wavelength is therefore \( \lambda_R = 1.865 \text{ m} \). This wavelength can easily be identified in figure 6.1.
6.2 Hankel transformations

This section focuses on the numerical evaluation of the Hankel transformations introduced in subsection 2.5.2. The forward and inverse Hankel transformations of order $n$ between the spatial coordinate $r$ and the wavenumber $k$ are defined as:

$$\mathcal{H}_n(f)(kr) = \int_0^\infty r J_n(kr) f(r) \, dr$$  \hspace{1cm} (6.22)

$$f(r) = \int_0^\infty k J_n(kr) f(kr) \, dk$$  \hspace{1cm} (6.23)

where $J_n$ is an $n$-th order Bessel function of the first kind.

It is clear from equations (6.22) and (6.23) that the forward and the inverse Hankel transformation are mathematically identical. The focus is therefore restricted to the forward transformation in the following.

A change of variables is performed in equation (6.22) using $kr = k_0e^v$ and $r = r_0e^{-w}$:

$$\mathcal{H}_n(f)(k_0e^v) = \int_{-\infty}^{\infty} r_0e^{-w} J_n(k_0r_0e^{v-w}) \mathcal{F}(r_0e^{-w}) r_0e^{-w} \, dw$$  \hspace{1cm} (6.24)

$$= r_0^2e^{-aw} \int_{-\infty}^{\infty} J_n(k_0r_0e^{v-w}) e^{\alpha(v-w)} \mathcal{F}(r_0e^{-w}) e^{-(2-a)w} \, dw$$  \hspace{1cm} (6.25)

where $k_0$ and $r_0$ can be chosen arbitrarily, as in the logarithmic Fourier transformation algorithm. The factor $e^{-aw}e^{aw}$ is introduced in equation (6.25) to ensure absolute integrability in the following [27]. Equation (6.25) is reformulated as:

$$\hat{f}(k_0e^w) = r_0^2e^{-aw} \chi(w)$$  \hspace{1cm} (6.26)

where the function $\chi(w)$ is defined as the following convolution:

$$\chi(w) = \varphi(w) * \psi(w)$$  \hspace{1cm} (6.27)

The functions $\varphi(w)$ and $\psi(w)$ are defined as:

$$\varphi(w) = \mathcal{F}(r_0e^{-w}) e^{-(2-a)w}$$  \hspace{1cm} (6.28)

$$\psi(w) = J_n(k_0r_0e^{w}) e^{aw}$$  \hspace{1cm} (6.29)

The convolution theorem is used to evaluate equation (6.27) in a similar way as in the logarithmic Fourier transformation algorithm. This is only possible if the function $\psi(w)$ is absolutely integrable, which is the case if $-n < a < 1.5$ [1, 27]. Within this range, the exponent $a$ can be chosen freely, but optimal performance is achieved if it is chosen so that the function $\varphi(w)$ in equation (6.28) is asymptotically constant if $w$ tends to $\pm\infty$. 
EDT provides the function `fht` to evaluate the (forward and inverse) Hankel transformation of order $n$ according to the procedure outlined in this section. The use of the function `fht` is similar to the use of the functions `logffcos` and `logffsin` introduced in the previous section: the input sampling $r_n$ must be logarithmic, the function $\hat{f}(k_r)$ can be obtained at any output point $k_r$, and the function $f(r)$ is padded at both sides in order to avoid aliasing. Padding is achieved by extrapolation of the function $f(r)$ proportionally to $r^{q_1}$ as $r$ tends to 0 and to $r^{q_2}$ as $r$ tends to $\infty$. The default value for both exponents $q_1$ and $q_2$ is $a - 2$, which implies that the function $\varphi(w)$ is padded using a constant value.
Chapter 7

Site amplification

Site amplification is an important issue in the assessment of the seismic hazard at sites where the top soil layers are particularly soft. In such cases, the seismic motion at the surface can be much higher than the outcrop motion due to resonance of the soft layers.

While the soil is often modelled as a linear (visco)elastic material, the constitutive behavior of soil is actually nonlinear. Under strong earthquake motions, it might be important to account for this nonlinearity. Soil typically exhibits a softening nonlinearity, or a decrease in modulus as strain increases. Increasing strains also cause progressively larger hysteresis in the stress-strain relation, leading to strain-dependent wave attenuation.

Three strategies can be followed regarding the nonlinear constitutive behavior of the soil: (1) the stress-strain relation is linearized, allowing for the use of a linear model, (2) the nonlinearity is accounted for through the use of an equivalent linear model, where the layer properties are iteratively modified as a function of the effective strain level, and (3) a fully nonlinear calculation is performed, using a time integration procedure.

In the equivalent linear approach, the equivalent soil properties are usually assumed to be frequency independent (this is the case in software such as SHAKE2000 and ProShake). However, this assumption often lead to an underestimation of the response in the higher frequency range. An alternative methodology based on a frequency dependent equivalent linear material model has therefore been developed by Kausel and Assimaki [10].

In EDT, the soil is modelled using the direct stiffness method or the thin layer method, which are based on a linear viscoelastic material model. As a consequence, EDT only allows for linear or equivalent linear calculations. In the equivalent
linear approach, both the use of frequency independent and the use of frequency
dependent equivalent soil properties is supported.

This chapter addresses the calculation of site amplification with EDT. It is
subdivided into the following sections:

**Modelling site amplification (p. 68)**
In this section, the amplification of an incident plane wave in a layered medium
is addressed. The governing equations are derived and the modelling of site
amplification with the direct stiffness method is discussed.

**Transfer functions (p. 71)**
This section focuses on the use of the algorithms provided by EDT compute the
amplification of an incident harmonic P-wave, SV-wave, or SH-wave in a layered
halfspace.

**Linear site response analysis (p. 72)**
This section addresses the computation of the seismic motion at the surface of a
layered soil from a seismogram representing the outcrop motion. A linear material
model is assumed.

**Equivalent linear site response analysis (p. 76)**
This section is similar to the previous one, but the soil is modelled as an equivalent
linear material.

**Frequency dependent equivalent linear site response analysis (p. 81)**
In this section, a similar site response analysis is performed, using an equivalent
linear material model with frequency dependent equivalent soil properties.

### 7.1 Modelling site amplification

This section addresses the modelling of an incident harmonic plane wave impinging
on a layered medium. The direct stiffness method is used, which implies that the
constitutive behavior of the medium is assumed to be linear.

The problem is visualized in figure 7.1a. An incoming wave travels through the
underlying halfspace and reaches the layer-halfspace interface. At the interface,
the incoming wave is partly reflected back into the halfspace and partly refracted
into the layers. Subsequent reflections and refractions at the interfaces between
layers may give rise to resonance effects, resulting in a relatively high response
at the surface of the layered medium. The aim of a site response analysis is to
compute the displacement amplitude at the free surface (or at a depth z), given
the amplitude of the incident wave at the point where it reaches the layer-halfspace
interface.
The incident wave is a plane harmonic P-wave or S-wave, characterized by an angle of incidence $\theta$ and a circular frequency $\omega$. The resulting wave field $u(x, z, t)$ in the layered medium can be expressed as:

$$u(x, z, t) = \tilde{u}(z)e^{i(\omega t - k_x x)} + \tilde{u}^*(z)e^{-i(\omega t - k_x x)}$$  \hspace{1cm} (7.1)

The (complex) horizontal wavenumber $k_x$ is determined from the angle of incidence $\theta$, according to equation (3.3) (for an incident P-wave) or (3.2) (for an incident S-wave). The vector $\tilde{u}(z)$ denotes the complex amplitude of the wave field at depth $z$. The calculation of the displacement field $\tilde{u}(z)$ is outlined in the following.

Figure 7.1: Decomposition of (a) the total wave field $\tilde{u}$ into (b) the wave field $\tilde{u}_0$ with zero interface displacements and (c) the diffracted wave field $\tilde{u}_d$.

In the direct stiffness method, it is impossible to account for the incident wave field in a direct way. The problem is therefore decomposed into two more tractable subproblems, as shown in figure 7.1. In subproblem (1), the incident wave is taken into account and the layer-halfspace interface is clamped. In subproblem (2), the motion of the layer-halfspace interface is taken into account. In this way, the total displacement field $\tilde{u}(z)$ is decomposed as follows:

$$\tilde{u}(z) = \tilde{u}_0(z) + \tilde{u}_d(z)$$  \hspace{1cm} (7.2)

where $\tilde{u}_0(z)$ and $\tilde{u}_d(z)$ denote the displacement field in subproblem (1) and (2), respectively.

According to the superposition principle, the external load $\tilde{p}$ on the layer-halfspace interface allows for a similar decomposition:

$$\tilde{p} = \tilde{p}_0 + \tilde{p}_d$$  \hspace{1cm} (7.3)

The load $\tilde{p}$ is equal to zero (no external forces are applied, the excitation only consists of the incident wave). The component $\tilde{p}_0$ that occurs in subproblem (1) can be considered as a reaction force due to the clamped boundary condition (figure 7.1b). In order to satisfy equation (7.3), the component $\tilde{p}_d$ in subproblem (2) must be equal to $-\tilde{p}_0$. 
The displacement field $\tilde{u}_0(z)$ and the reaction force $\tilde{p}_0$ in subproblem (1) can not be determined in a direct way. This subproblem is therefore further decomposed into two problems (1a) and (1b), as shown in figure 7.2.

In subproblem (1a), only the incident wave $\tilde{u}_i(z)$ is considered. The occurrence of a reflected wave is suppressed by means of an external force $\tilde{p}_i$ applied at the surface of the halfspace element. The force $\tilde{p}_i$ is computed from the displacement $\tilde{u}_i(z)$ at the surface of the halfspace, which follows from the angle of incidence and the amplitude of the incident wave. The computation of the force $\tilde{p}_i$ is performed by means of the direct stiffness method, using a halfspace element that accounts for incoming waves.

In subproblem (1b), only the reflected wave $\tilde{u}_r(z)$ is considered. The reflected wave is defined so that the displacement $\tilde{u}_0(z) = \tilde{u}_i(z) + \tilde{u}_r(z)$ vanishes at the surface of the layer element. The resulting force $\tilde{p}_r$ at the surface of the halfspace element is computed by means of the direct stiffness method, using a halfspace element that accounts for outgoing waves.

The load $\tilde{p}_0$ in the subproblem (1), shown in figure 7.1b, is subsequently obtained by superposition of the loads $\tilde{p}_i$ and $\tilde{p}_r$.

Next, subproblem (2), shown in figure 7.1c, is solved: the displacement field $\tilde{u}_d(z)$ induced by the load $\tilde{p}_d = -\tilde{p}_0$ is computed by means of the direct stiffness method.

The total displacement $\tilde{u}(z)$ at a depth $z$ is finally obtained by superposition of the displacements in the subproblems (1a), (1b), and (2):

$$\tilde{u}(z) = \tilde{u}_i(z) + \tilde{u}_r(z) + \tilde{u}_d(z)$$ (7.4)
7.2 Transfer functions

EDT provides the functions \texttt{amp\_p}, \texttt{amp\_sv}, and \texttt{amp\_sh} to compute the amplification of an incident P-wave, SV-wave, or SH-wave in a layered halfspace, following the procedure explained in the previous section.

\textbf{Example 7.1: Amplification of an incident P-wave in a layered halfspace.}

This example addresses the amplification of an incident P-wave in a layered soil. The soil consists of a homogeneous layer with a thickness of 10 m on a homogeneous halfspace. The shear wave velocity $C_s$ is 200 m/s in the layer and 500 m/s in the halfspace. The dilatational wave velocity $C_d$ is 400 m/s in the layer and 2000 m/s in the halfspace. The damping ratio $D_s = D_p$ for both shear waves and dilatational waves is 0.05. The density of the soil is $\rho = 1800 \text{ kg/m}^3$.

The amplification of an incident P-wave is computed with the EDT function \texttt{amp\_p}. Three angles of incidence $\theta$ are considered: 90°, 60°, and 30°. The amplification factor is computed in the frequency range from 0 Hz to 50 Hz.

```matlab
% SOIL PROPERTIES
h = [10 inf]; % Element thickness [m]
Cs = [200 500]; % Shear wave velocity [m/s]
Cp = [400 2000]; % Dilatational wave velocity [m/s]
Ds = 0.05; % Shear damping ratio [-]
Dp = 0.05; % Dilatational damping ratio [-]
rho = 1800; % Density [kg/m^3]

% INCIDENT WAVE PROPERTIES
f = [0:0.1:50]; % Frequency [Hz]
omega = 2*pi*f; % Frequency [rad/s]
phi = [90 60 30]/180*pi; % Angle of incidence [rad]

% RECEIVER LOCATION
x = 0; % Horizontal receiver coordinate [m]
z = 0; % Vertical receiver coordinate [m]

% AMPLIFICATION FACTOR
[ux,uz] = amp\_p(h,Cs,Cp,Ds,Dp,rho,phi,x,z,omega);

% PLOT HORIZONTAL RESPONSE
figure;
plot(f,squeeze(abs(ux)));
xlabel('Frequency [Hz]');
ylabel('Vertical amplification [-]');
```

The resulting amplification factor is shown in figure 7.3. For normal incidence ($\theta = 90^\circ$), the horizontal response at the surface is zero. The vertical response varies with the frequency. As the frequency tends to zero, the thickness of the layer is very small compared to the wavelength of the waves in the soil. As a consequence, the influence of the layer vanishes, and the amplification factor tends to a value of 2, which corresponds to the result for a homogeneous halfspace. At the odd multiples of 10 Hz, the amplification factor reaches a maximum, which can be explained by resonance of the layer. For oblique incidence ($\theta < 90^\circ$), the horizontal response at the free surface differs from zero. The vertical response at the free surface decreases as the angle of incidence $\theta$ decreases.

7.3 Linear site response analysis

This section focuses on the modelling of site amplification with a linear material model. The aim of a site response analysis is to compute the response of a layered soil due to a specific earthquake, characterized by a strong motion record
representing the outcrop motion. Usually, it is assumed that the direction of the incident wave is vertical, and a one-dimensional analysis is performed.

A linear site response analysis is performed as follows:

1. Select a seismogram representing the outcrop motion $a_o(t)$.
2. Divide the outcrop motion by two in order to obtain the input motion $a_i(t)$.
3. Compute the Fourier transform $\hat{a}_i(\omega)$ of the input motion $a_i(t)$.
4. Compute the transfer function between the input motion $\hat{a}_i(\omega)$ and the acceleration $\hat{a}(\omega)$ at the surface of the layered soil.
5. Multiply this transfer function by the input motion spectrum $\hat{a}_i(\omega)$ and perform an inverse Fourier transformation to obtain a seismogram $a(t)$ representing the amplified response at the soil’s surface.
6. Compute the response spectrum corresponding to the seismogram $a(t)$.

EDT provides a function $\text{linamp}$ to perform steps 3 to 5, and a function $\text{rspectrum}$ to compute the response spectrum referred to in step 6. These functions are used in the following example.

**Example 7.2: Response of a layered soil due to an earthquake - linear analysis.**

In this example, a linear site response analysis is performed for a layered soil. The soil consists of 4 homogeneous layers on a homogeneous halfspace. Each layer is characterized by a thickness $h$, a shear modulus $\mu$, a density $\rho$, a shear wave velocity $C_s = \sqrt{\mu/\rho}$, and a material damping ratio $D_s$. These properties are given in table 7.1.

<table>
<thead>
<tr>
<th>Layer</th>
<th>$h$ [m]</th>
<th>$\mu_0$ [MPa]</th>
<th>$\rho$ [kg/m$^3$]</th>
<th>$C_{s0}$ [m/s]</th>
<th>$D_{s0}$ [-]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>39.2</td>
<td>2000</td>
<td>140</td>
<td>0.005</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>64.8</td>
<td>2000</td>
<td>180</td>
<td>0.005</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>156.8</td>
<td>2000</td>
<td>280</td>
<td>0.005</td>
</tr>
<tr>
<td>4</td>
<td>18</td>
<td>320.0</td>
<td>2000</td>
<td>400</td>
<td>0.005</td>
</tr>
<tr>
<td>5</td>
<td>$\infty$</td>
<td>1620.0</td>
<td>2000</td>
<td>900</td>
<td>0.005</td>
</tr>
</tbody>
</table>

**Table 7.1:** Dynamic soil properties.

The variation of the shear wave velocity $C_s$ and the material damping ratio $D_s$ with depth is visualized in figure 7.4.
The outcrop motion is represented by a seismogram from the 1999 M 7.4 Kocaeli earthquake. Figure 7.5 shows the seismogram and the corresponding response spectrum. The response spectrum is computed for a structural damping ratio $\xi = 0.05$.

The linear site response analysis is performed by means of the following MATLAB M-file.

```matlab
% OUTCROP MOTION
ao=load('kocaeli.txt');       % Outcrop motion [m/s^2]
F = 200;                      % Sampling frequency [Hz]
dt = 1/F;                     % Sampling interval [s]
N = length(ao);              % Number of samples [-]
t = [0:N-1]*dt;              % Time sampling [s]
```
Figure 7.6 shows the time history $a(t)$ of the acceleration at the soil's surface, and the corresponding response spectrum. For comparison, the outcrop response spectrum is also shown. It is clear that the presence of the soft layers leads to
a higher value of the response spectrum, or, equivalently, to an increase of the response of structures due to an earthquake. For large structures (fundamental eigenperiod larger than 1 s), the influence of the soft layers is less pronounced. The response of these structures mainly depends on the low frequency waves in the soil. The wavelength of the low frequency waves is much larger than the thickness of the soft layers, which explains the limited influence of these layers.

Figure 7.6: (a) Seismogram computed with a linear soil model and (b) corresponding response spectrum (solid line), compared with the outcrop response spectrum (dashed line).

7.4 Equivalent linear site response analysis

Under strong earthquake motions, the constitutive behavior of soil can no longer be considered as linear. Soil typically exhibits a softening nonlinearity, or a decrease in modulus as strain increases. Increasing strains also cause progressively larger hysteresis in the stress-strain relation, leading to strain-dependent wave attenuation. However, the nonlinear response of a layered soil due to an earthquake can still be modelled (in an approximate way) by means of a linear soil model, provided that the shear modulus and the damping ratio of each layer are adjusted as a function of the strain level in the layer.

Various authors have investigated the variation of the shear modulus and the damping ratio with the strain level under cyclic loading. Seed et al. [26] have presented the modulus reduction and damping ratio curves for sandy soils shown in figure 7.7. The modulus reduction curve, shown in figure 7.7a, represents the ratio \( \mu / \mu_0 \) of the (equivalent) shear modulus \( \mu \) and the small-strain shear modulus \( \mu_0 \) as a function of the shear strain. The damping ratio curve, shown in figure 7.7b, represents the damping ratio \( D_\delta \) as a function of the shear strain.
An equivalent linear site response analysis proceeds as follows:

1. Select a seismogram representing the outcrop motion $a_o(t)$.
2. Divide the outcrop motion by two in order to obtain the input motion $a_i(t)$.
3. Subdivide the soil profile in a sufficient number of sublayers to characterize properly the spatial variation of inelastic effects.
4. Assign to each layer an initial shear modulus $\mu_0$ and damping ratio $\beta_0$.
5. Compute the Fourier transform $\hat{a}_i(\omega)$ of the input motion $a_i(t)$.
6. Compute the transfer functions between the input motion $\hat{a}_i(\omega)$ and the strains $\hat{\gamma}(\omega)$ at the center of each layer.
7. Multiply each transfer function by the input motion spectrum $\hat{a}_i(\omega)$ to obtain the strain spectra $\hat{\gamma}(\omega)$ and perform an inverse Fourier transformation to obtain the strain time histories $\gamma(t)$.
8. Determine for each layer the peak strain $\gamma_{\text{max}} = \max(\gamma(t))$.
9. Determine for each layer the effective strain level $\gamma_{\text{eff}}$. The effective strain level $\gamma_{\text{eff}}$ is usually defined as $\gamma_{\text{eff}} = 0.65 \gamma_{\text{max}}$ [17].
10. Evaluate the modulus reduction and damping ratio curves at the effective strain level $\gamma_{\text{eff}}$ to determine the equivalent modulus $\mu$ and damping ratio $\beta$ for each layer. Modify the soil profile accordingly.
11. Compare the peak strains with their values in the previous iteration. Repeat steps 6 to 11 as necessary.
12. Compute the transfer function between the input motion $\hat{a}_i(\omega)$ and the acceleration $\hat{a}(\omega)$ at the surface of the layered soil.

13. Multiply this transfer function by the input motion spectrum $\hat{i}_o(\omega)$ and perform an inverse Fourier transformation to obtain a seismogram $a(t)$ representing the amplified response at the soil’s surface.

14. Compute the response spectrum corresponding to the seismogram $a(t)$.

EDT provides a function \texttt{eqlinamp} to perform steps 5 to 13, and a function \texttt{rspectrum} to compute the response spectrum referred to in step 14. These functions are used in the following example.

\begin{example}
\textbf{Example 7.3: Response of a layered soil due to an earthquake - equivalent linear analysis.}

In this example, the analysis performed in the previous section is reconsidered, now using an equivalent linear soil model. Under small strain conditions, the soil properties are the same as in the previous example (table 7.1). At larger strains, the nonlinear behavior of the soil is accounted for through the use of modulus reduction and damping ratio curves. The curves proposed for sandy soils by Seed et al. [26] are used. These curves are shown in figure 7.7.

The outcrop motion is identical to the motion considered in the previous section. The time history of the outcrop motion and the corresponding response spectrum are shown in figure 7.5.

The equivalent linear site response analysis is performed by means of the following MATLAB M-file. Compared to the previous example, the soil is now subdivided into layers with a thickness of 3 m, so that the spatial variation of the nonlinear effects is properly represented. The modulus reduction and damping ratio curves are defined for a finite number of strain levels; the function \texttt{eqlinamp} computes the actual curves by spline interpolation. In addition to the amplified motion, the function \texttt{eqlinamp} also returns the equivalent shear modulus, the equivalent damping ratio, and the effective strain level in each layer.

% OUTCROP MOTION
ao=load('kocaeli.txt'); % Outcrop motion [m/s^2]
F = 200; % Sampling frequency [Hz]
dt = 1/F; % Sampling interval [s]
N = length(ao); % Number of samples [-]
t = [0:N-1]*dt; % Time sampling [s]

% OUTCROP RESPONSE SPECTRUM
T = logspace(-2,1,400); % Eigenperiods [s]
xi = 0.05; % Structural damping ratio [-]
EQUIVALENT LINEAR SITE RESPONSE ANALYSIS

\[ Sao = \text{rspectrum}(ao,F,T,xi); \]  
% Outcrop response spectrum [m/s^2]

% SMALL STRAIN SOIL PROPERTIES
\[ h = [3 3 3 3 3 3 3 \ldots 3 3 3 3 3 3 \inf]; \]  
% Layer thickness [m]
\[ Cs0 = [140 140 180 180 280 280 280 \ldots 280 280 400 400 400 400 900]; \]  
% Small strain Cs [m/s]
\[ rho = 2000; \]  
% Density [kg/m^3]

% MODULUS REDUCTION AND DAMPING RATIO VERSUS CYCLIC STRAIN AMPLITUDE
\[ gd = [1e-6 2e-6 5e-6 1e-5 2e-5 5e-5 1e-4 \ldots 2e-4 5e-4 1e-3 2e-3 5e-3 1e-2 2e-2 5e-2]; \]  
% Strain amp.
\[ Rd = [1.000 0.998 0.949 0.917 0.832 0.729 \ldots 0.600 0.421 0.291 0.188 0.098 0.036 0.016]; \]  
% Modulus red.
\[ Dd = [0.005 0.008 0.013 0.019 0.025 0.037 0.053 \ldots 0.077 0.120 0.153 0.187 0.226 0.244 0.259 0.273]; \]  
% Damping ratio

% AMPLIFICATION
\[ ai = ao/2; \]  
% Input motion [m/s^2]
\[ [a,Cs,Ds,g] = \text{eqlinamp}(h,Cs0,rho,gd,Rd,Dd,ai,F); \]

% RESPONSE SPECTRUM
\[ Sa = \text{rspectrum}(a,F,T,xi); \]  
% Amplified response spectrum [m/s^2]

% PLOT OUTCROP MOTION
figure;
plot(t,ao);
xlabel('Time [s]');
ylabel('Acceleration [m/s^2]');

% PLOT OUTCROP RESPONSE SPECTRUM
figure;
semilogx(T,Sao);
xlabel('Period [s]');
ylabel('Acceleration [m/s^2]');

% PLOT EQUIVALENT SHEAR MODULUS
figure;
plotprofile(h,Cs);
xlabel('Shear wave velocity [m/s]');
ylabel('Depth [m]');

% PLOT EQUIVALENT DAMPING RATIO
figure;
plotprofile(h,Ds);
xlabel('Material damping ratio [-]');
Figure 7.8 shows the equivalent shear modulus $C_s$ and the equivalent material damping ratio $D_s$. The small strain values of both soil properties are also shown. A softening nonlinearity is observed: the equivalent shear modulus has decreased compared to the small strain value, while the equivalent damping ratio has increased. The nonlinear effect decreases with depth.

Figure 7.8: Equivalent (a) shear wave velocity $C_s$ and (b) material damping ratio $D_s$ as a function of depth (solid lines) compared with the small strain soil properties (dashed lines).

Figure 7.9 shows the time history of the acceleration $a(t)$ at the soil’s surface, and the corresponding response spectrum. For comparison, the response spectrum obtained from the linear analysis and the outcrop response spectrum are also shown. Compared to the results of the linear analysis, the increase of the structural response is smaller, especially for small structures (fundamental eigenperiod smaller than 0.1s). The response of these structures depends to a large extent on the high frequency waves in the soil, with a small wavelength. These waves are strongly attenuated in the equivalent linear model due to the use of a high
equivalent material damping ratio. This effect is not realistic; the attenuation of high frequency waves is much weaker in reality. An alternative equivalent linear approach with frequency dependent soil properties has therefore been developed by Kausel and Assimaki [10]. The frequency dependent approach is addressed in the next section.

![Seismogram and response spectrum](a) (b)

**Figure 7.9:** (a) Seismogram computed with a frequency independent equivalent linear soil model and (b) corresponding response spectrum (solid line), compared with the outcrop response spectrum (dashed line).

### 7.5 Frequency dependent equivalent linear site response analysis

This section focuses on the frequency dependent equivalent linear method developed by Kausel and Assimaki [10]. In this method, the frequency spectrum of the strain in each layer is considered to determine the equivalent dynamic soil properties. This implies that the degradation curves are evaluated at a different strain level for each frequency, resulting in frequency dependent equivalent dynamic soil properties. Kausel and Assimaki [10] demonstrate that this approach leads to more realistic results by means of a comparison with a fully nonlinear calculation.

Following Kausel and Assimaki [10], the modelling of site amplification with a frequency dependent equivalent linear material model involves the following steps:

1. Select a seismogram representing the outcrop motion \( a_o(t) \).
2. Divide the outcrop motion by two in order to obtain the input motion \( a_i(t) \).
3. Compute the Fourier transform $\hat{a}_i(\omega)$ of the input motion $a_i(t)$.

4. Subdivide the soil profile in a sufficient number of sublayers to characterize properly the spatial variation of inelastic effects.

5. Assign to each layer an initial shear modulus $\mu_0$ and damping ratio $\beta_0$.

6. Compute the transfer functions between the input motion $\hat{a}_i(\omega)$ and the strains $\hat{\gamma}(\omega)$ at the center of each layer.

7. Multiply each transfer function by the input motion spectrum $\hat{a}_i(\omega)$ to obtain the strain spectra $\hat{\gamma}(\omega)$ and perform an inverse Fourier transformation to obtain the strain time histories $\gamma(t)$.

8. Determine for each layer the scaled strain spectrum $\hat{\gamma}_{sc}(\omega)$:

$$\hat{\gamma}_{sc}(\omega) = \frac{\gamma_{\text{max}}}{\gamma_0} \hat{\gamma}(\omega) \quad (7.5)$$

where $\gamma_{\text{max}}$ is the peak strain and $\gamma_0$ is the average value of the strain spectrum between zero and the mean frequency $\omega_0$. The peak strain $\gamma_{\text{max}}$ is defined as:

$$\gamma_{\text{max}} = \max(\gamma(t)) \quad (7.6)$$

The average value $\gamma_0$ of the strain spectrum between zero and $\omega_0$ is defined as:

$$\gamma_0 = \frac{1}{\omega_0} \int_0^{\omega_0} |\hat{\gamma}(\omega)| \, d\omega \quad (7.7)$$

The mean frequency $\omega_0$ is given by:

$$\omega_0 = \frac{\int_0^\infty \omega |\hat{\gamma}(\omega)| \, d\omega}{\int_0^\infty |\hat{\gamma}(\omega)| \, d\omega} \quad (7.8)$$

9. Determine for each layer the smoothed strain spectrum $\hat{\gamma}_{sm}(\omega)$:

$$\hat{\gamma}_{sm}(\omega) = \begin{cases} 
\gamma_{\text{max}} & \text{for } \omega \leq \omega_0 \\
\gamma_{\text{max}} \exp\left(-\alpha \frac{\omega}{\omega_0} - 1\right) & \text{for } \omega > \omega_0
\end{cases} \quad (7.9)$$

The parameters $\alpha$ and $\beta$ must be determined by fitting the smoothed strain spectrum $\hat{\gamma}_{sm}(\omega)$ to the scaled strain amplitude spectrum $|\hat{\gamma}_{sc}(\omega)|$. To this end, the following linear least squares problem must be solved:

$$\begin{Bmatrix} \alpha \\ \beta \end{Bmatrix} = \arg \min_{\alpha, \beta} \left( \log \hat{\gamma}_{sm}(\omega) - \log |\hat{\gamma}_{sc}(\omega)| \right) \quad (7.10)$$
It should be noted that equation (7.9) differs from the equation proposed in the original paper; a term $-1$ is added here. This term ensures that the smooth strain spectrum is continuous at the mean frequency $\omega_0$.

10. Use the smoothed strain spectrum $\hat{\gamma}_{sm}(\omega)$ and the degradation curves to determine the frequency-dependent equivalent modulus $\mu$ and damping ratio $\beta$ for each layer. Modify the soil profile accordingly.

11. Compare the peak strains with their values in the previous iteration. Repeat steps 6 to 11 as necessary.

12. Compute the transfer function between the input motion $\hat{a}_i(\omega)$ and the acceleration $\hat{a}(\omega)$ at the surface of the layered soil.

13. Multiply this transfer function by the input motion spectrum $\hat{a}_i(\omega)$ and perform an inverse Fourier transformation to obtain a seismogram $a(t)$ representing the amplified response at the soil’s surface.

14. Compute the response spectrum corresponding to the seismogram $a(t)$.

EDT provides a function `eqlinampf` to perform a frequency dependent equivalent linear site amplification analysis. The use of this function is clarified in the following example.

**Example 7.4: Response of a layered soil due to an earthquake - frequency dependent equivalent linear analysis.**

The analysis performed in the previous two examples is reconsidered, now using a frequency dependent equivalent linear material model. The soil profile is the same as in the previous example, the same modulus reduction and damping ratio curves are used, and the same outcrop motion is considered.

The analysis is performed by means of the following MATLAB M-file. In addition to the amplified motion, the function `eqlinampf` also returns the equivalent shear modulus, the equivalent damping ratio, the scaled strain amplitude spectrum, and the smoothed strain spectrum in each layer. These variables are all frequency dependent; a vector with the corresponding frequencies is also returned.

```matlab
% OUTCROP MOTION
ao=load('kocaeli.txt'); % Outcrop motion [m/s^2]
F = 200; % Sampling frequency [Hz]
dt = 1/F; % Sampling interval [s]
N = length(ao); % Number of samples [-]
t = [0:N-1]*dt; % Time sampling [s]
```
% OUTCROP RESPONSE SPECTRUM
T = logspace(-2,1,400); % Eigenperiods [s]
xi = 0.05; % Structural damping ratio [-]
Sao = rspectrum(ao,F,T,xi); % Outcrop response spectrum [m/s^2]

% SMALL STRAIN SOIL PROPERTIES
h = [3 3 3 3 3 3 3 3 ... 3 3 3 3 3 3 3 3 inf]; % Layer thickness [m]
Cs0 = [140 140 180 180 180 280 280 280 ... 280 280 400 400 400 400 400 900]; % Small strain Cs [m/s]
rho = 2000; % Density [kg/m^3]

% MODULUS REDUCTION AND DAMPING RATIO VERSUS CYCLIC STRAIN AMPLITUDE
gd = [1e-6 2e-6 5e-6 1e-5 2e-5 5e-5 1e-4 ... 2e-4 5e-4 1e-3 2e-3 5e-3 1e-2 2e-2 5e-2]; % Strain amp.
Rd = [1.000 0.998 0.980 0.949 0.917 0.832 0.729 ... 0.600 0.421 0.291 0.188 0.098 0.036 0.016]; % Modulus red.
Dd = [0.005 0.008 0.013 0.019 0.025 0.037 0.053 ... 0.077 0.120 0.153 0.187 0.226 0.244 0.259 0.273]; % Damping ratio

% AMPLIFICATION
ai = ao/2; % Input motion [m/s^2]
[a,Cs,Ds,G,Gs,f] = eqlinampf(h,Cs0,rho,gd,Rd,Dd,ai,F);

% RESPONSE SPECTRUM
Sa = rspectrum(a,F,T,xi); % Amplified response spectrum [m/s^2]

% PLOT OUTCROP MOTION
figure;
plot(t,ao);
xlabel('Time [s]');
ylabel('Acceleration [m/s^2]');

% PLOT OUTCROP RESPONSE SPECTRUM
figure;
semilogx(T,Sao);
xlabel('Period [s]');
ylabel('Acceleration [m/s^2]');

% PLOT SCALED AND SMOOTHED STRAIN AMPLITUDE SPECTRA FOR LAYER 1
figure;
semilogy(f, G(:,1), '--', f, Gs(:,1), '-');
xlabel('Frequency [Hz]');
ylabel('Strain [-]');

% PLOT SCALED AND SMOOTHED STRAIN AMPLITUDE SPECTRA FOR LAYER 2
Figure 7.10 shows both the scaled strain spectrum $\hat{\gamma}_{sc}(\omega)$ and the smoothed strain spectrum $\hat{\gamma}_{sm}(\omega)$ for the top layer and the second layer. It can be observed that the strain decreases with the frequency, explaining the overestimation of the high frequency wave attenuation in the frequency independent approach. The correspondence between the unsmoothed and the smoothed strain spectrum is acceptable. This confirms the statement by Kausel and Assimaki [10] that the model proposed for the smoothed strain spectrum (equation (7.9)) is suitable for most earthquakes, even after the addition of the term $-1$ that ensures the continuity of the spectrum.
Figure 7.10: Unsmoothed (dots) and smoothed (solid line) spectrum of the strain in the center of (a) the top layer and (b) the second layer.

Figure 7.11 shows the equivalent shear modulus $C_s$ and the equivalent material damping ratio $D_s$. Both properties are frequency dependent and are shown for three different frequencies. For comparison, the values used in the frequency independent approach and the small strain values are also shown. For low frequencies, the soil properties used in the frequency dependent approach are close to the values used in the frequency independent approach. For high frequencies, the difference is larger, and a tendency towards the small strain values is observed.

Figure 7.11: Frequency dependent equivalent (a) shear wave velocity $C_s$ and (b) material damping ratio $D_s$ as a function of depth at 2 Hz (dark gray lines), 4 Hz (medium gray lines), and 8 Hz (light gray lines), compared with the small strain soil properties (dashed lines).

Figure 7.12 shows the time history of the acceleration $a(t)$ at the soil’s surface, and the corresponding response spectrum. The spectra computed with the previous two methods and the outcrop spectrum are also shown. Compared to the results
of the linear analysis, a reduction of the structural response is observed, but not to the same extent as in the frequency independent approach. This indicates that the overestimation of high frequency wave attenuation in an equivalent linear analysis can be avoided through the use of frequency dependent material properties.

Figure 7.12: (a) Seismogram computed with a frequency dependent equivalent linear soil model and (b) corresponding response spectrum (solid line), compared with the outcrop response spectrum (dashed line).
Chapter 8

Surface waves

Surface waves are the natural modes of vibration of a layered medium. The natural modes of vibration correspond to the displacement fields in the layered medium that occur in the absence of an external load. Surface waves travel in the horizontal direction, along the free surface of the medium or along an interface between layers, and are evanescent in the vertical direction. Due to the uncoupling of in-plane and out-of-plane motion in a layered medium, a distinction can be made between in-plane surface waves (or Rayleigh waves) and out-of-plane surface waves (or Love waves).

Surface waves are dispersive: the phase velocity and the attenuation coefficient of a surface wave varies with the frequency. The computation of the dispersion and attenuation curves of a layered halfspace is the key ingredient of the Spectral Analysis of Surface Waves (SASW) method, which is a method to determine the dynamic properties of shallow soil layers [15, 21]. The SASW method consists of an in situ experiment where dispersive surface waves are generated by means of an impact hammer, a falling weight, or a hydraulic shaker. The response at the surface is recorded by a number of geophones or accelerometers and used to determine the dispersion and attenuation curves of the soil. Finally, an inverse problem is solved where the corresponding soil properties are determined.

Surface waves in a layered medium can be computed by means of the direct stiffness method or the thin layer method, by the solution of an eigenvalue problem in terms of the frequency $\omega$ and the horizontal wavenumber $k_x$. If the direct stiffness method is used, the eigenvalue problem is transcendental, has an infinite number of solutions, and must be solved with search techniques. If the thin layer method is used, a quadratic eigenvalue problem in terms of the horizontal wavenumber $k_x$ is obtained for each frequency $\omega$. This problem can be reformulated as a linear eigenvalue problem, which can be solved using standard techniques. An important
The drawback of the thin layer approach, however, is that it is only applicable to a layered soil supported by a rigid stratum.

The phase velocity and attenuation coefficient of the dispersive surface modes can also be derived from the wavenumber content of the Green’s function $\tilde{u}_{zz}$ of the soil, so avoiding the solution of an eigenvalue problem. This method can not be used to identify all surface waves of a layered halfspace, but it is particularly suitable to compute the surface wave that is relevant in the SASW method, i.e. the surface wave that dominates the response due to a vertical load at the soil’s surface.

This chapter focuses on the computation of dispersive surface waves with EDT. It is organized as follows:

**Transcendental eigenvalue problem (p. 90)**
This section addresses the computation of dispersive surface waves in a layered halfspace modelled with the direct stiffness method, through the solution of a transcendental eigenvalue problem.

**Quadratic eigenvalue problem (p. 94)**
This section addresses the computation of dispersive surface waves in a layered stratum by means of the thin layer method, which leads to a quadratic eigenvalue problem.

**Wavenumber content of the Green’s function (p. 97)**
In this section, the phase velocity and the attenuation coefficient of the dominant surface wave in a layered halfspace are determined from the wavenumber content of the Green’s function.

### 8.1 Transcendental eigenvalue problem

This section focuses on the calculation of the dispersive surface waves in a layered halfspace by means of the direct stiffness method. In the direct stiffness method, the equilibrium of the medium is expressed in the frequency-wavenumber domain as:

$$\tilde{P} = \tilde{K}\tilde{U} \quad (8.1)$$

If the load vector $\tilde{P}$ vanishes, non-trivial solutions for the displacements $\tilde{U}$ can be obtained if the determinant of the stiffness matrix $\tilde{K}$ is zero:

$$\det \tilde{K} = 0 \quad (8.2)$$
Due to the uncoupling of in-plane and out-of-plane motion in a layered halfspace, this equation can be reformulated as:

\[
\begin{align*}
\det \mathbf{K}_{PSV} &= 0 \quad (8.3) \\
\det \mathbf{K}_{SH} &= 0 \quad (8.4)
\end{align*}
\]

where \(\mathbf{K}_{PSV}\) and \(\mathbf{K}_{SH}\) are the submatrices of the stiffness matrix \(\mathbf{K}\) that model P-SV-wave and SH-wave propagation, respectively.

Equation (8.3) governs the propagation of in-plane surface waves or Rayleigh waves. This equation corresponds to an eigenvalue problem in terms of the frequency \(\omega\) and the complex horizontal wavenumber \(k_x\), whose imaginary part represents wave attenuation in the horizontal direction. The eigenvalue problem is transcendental, has an infinite number of solutions, and must be solved with search techniques. For each frequency \(\omega\), the phase velocity \(C_R(\omega)\) of the Rayleigh wave is obtained as the ratio \(\omega/\text{Re}[k_x]\) where \((\omega, k_x)\) is a solution of the characteristic equation (8.3). The attenuation coefficient \(A_R(\omega)\) of the Rayleigh wave is computed as \(1/\text{Im}[k_x]\). At a given frequency, multiple Rayleigh waves corresponding to multiple solutions of the eigenvalue problem may exist. The Rayleigh wave with the lowest phase velocity is referred to as the fundamental Rayleigh wave.

Equation (8.4) governs the propagation of out-of-plane surface waves or Love waves. It is solved in a similar way as equation (8.3) in order to obtain the phase velocity \(C_L(\omega)\) and the attenuation coefficient \(A_L(\omega)\) of the Love waves.

EDT contains two functions \texttt{eig_dsmpsv} and \texttt{eig_dsmsh} to solve the eigenvalue problems in equations (8.3) and (8.4). These functions calculate the dispersion curves \(C_R(\omega)\) and \(C_L(\omega)\) and the attenuation curves \(A_R(\omega)\) and \(A_L(\omega)\), as well as the modal displacements \(\tilde{u}(z)\) and tractions \(\tilde{t}(z)\) at the specified depths \(z\).

**Example 8.1: Surface waves in a layered halfspace.**

In this example, the surface waves in a layered soil are computed. The soil consists of a homogeneous layer with a thickness of 3 m on a homogeneous halfspace. The shear wave velocity \(C_s\) is 200 m/s in the layer and 500 m/s in the halfspace. The dilatational wave velocity \(C_d\) is 400 m/s in the layer and 2000 m/s in the halfspace. The damping ratio for both shear and dilatational waves is \(D_s = D_p = 0.05\). The density of the soil is 1800 kg/m\(^3\).

The function \texttt{eig_dsmpsv} is used to determine all surface waves in the frequency range up to 30 Hz. For each surface wave, the dispersion curve \(C_R(\omega)\) and the attenuation curve \(A_R(\omega)\) are calculated. At 10 Hz, the mode shapes of the surface waves are computed.
% SOIL PROPERTIES
h = [10 inf]; % Element thickness [m]
Cs = [200 500]; % Shear wave velocity [m/s]
Cp = [400 2000]; % Dilatational wave velocity [m/s]
Ds = 0.05; % Shear damping ratio [-]
Dp = 0.05; % Dilatational damping ratio [-]
rho = 1800; % Density [kg/m^3]

% SOLVE EIGENVALUE PROBLEM
fMax = 30; % Maximum frequency [Hz]
fMode = 10; % Frequency where the mode shapes are computed [Hz]
zMode = 0:0.02:20; % Depths where the mode shapes are computed [m]
[f,C,A,Ux,Uz] = eig_dsmpsv(h,Cs,Cp,Ds,Dp,rho,fMax,[],fMode,zMode);

% PLOT DISPERSION CURVES
figure;
plot(f,C);
xlabel('Frequency [Hz]');
ylabel('Phase velocity [m/s]');

% PLOT ATTENUATION CURVES
figure;
plot(f,A);
xlabel('Frequency [Hz]');
ylabel('Attenuation coefficient [1/m]');

% PLOT MODE SHAPES
for iMode = 1:2

% HORIZONTAL COMPONENT
figure;
plot(real(Ux(:,:,iMode)),zMode,imag(Ux(:,:,iMode)),zMode);
set(gca,'YDir','reverse');
xlabel('Displacement [\text{-}]');
ylabel('Depth [m]');

% VERTICAL COMPONENT
figure;
plot(real(Uz(:,:,iMode)),zMode,imag(Uz(:,:,iMode)),zMode);
set(gca,'YDir','reverse');
xlabel('Displacement [\text{-}]');
ylabel('Depth [m]');
end
Figure 8.1: (a) Phase velocity $C_R(\omega)$ and (b) attenuation coefficient $A_R(\omega)$ of the first four Rayleigh waves in a layered halfspace.

Figure 8.2: (a) Horizontal and (b) vertical displacement for the first (top) and the second (bottom) Rayleigh wave in a layered halfspace at 10 Hz. The solid line represents the real part, the dashed line the imaginary part.

Figure 8.1 shows the resulting dispersion curves $C_R(\omega)$ and attenuation curves $A_R(\omega)$. In the low frequency range, only a single Rayleigh wave exists. This
is the fundamental Rayleigh wave. Higher order surface waves occur at higher frequencies, which are referred to as the cut-off frequencies of these waves.

The surface waves are dispersive due to the variation of the soil properties with depth. In the low frequency range, the surface wavelength is large and the surface wave reaches deep and stiff layers, resulting in a high phase velocity. As the frequency increases, the surface wavelength decreases and the surface wave travels through shallow and soft layers, resulting in a low phase velocity. This dispersive behaviour is the basis of the SASW method, where the dispersion curve $C_R(\omega)$ of the fundamental Rayleigh wave is used to identify the shear wave velocity $C_s$ as a function of depth. In a similar way, the variation of the damping ratio $D_s$ with depth can be identified from the variation of the attenuation curve $A_R(\omega)$ of the fundamental Rayleigh wave.

At 10 Hz, two surface waves exist. The corresponding modal displacements are shown in figures 8.2. For both surface waves, the horizontal displacement is almost purely imaginary, while the vertical displacement is almost purely real. This implies that both displacement components are $90^\circ$ out of phase and that the particles move in an elliptic path.

### 8.2 Quadratic eigenvalue problem

For a layered stratum modelled with the thin layer method, the equilibrium is expressed as:

\[
\tilde{P} = \tilde{K}\tilde{U} \tag{8.5}
\]

In the absence of an external load $\tilde{P}$, non-trivial solutions $\tilde{U}$ of the equilibrium equation can be obtained if the determinant of the stiffness matrix $\tilde{K}$ is zero:

\[
\det \tilde{K} = 0 \tag{8.6}
\]

As discussed in chapter 5 on the thin layer method, the stiffness matrix $\tilde{K}$ is given by:

\[
\tilde{K} = k^2_{xz}\tilde{A} + k_x\tilde{B} + \tilde{G} - \omega^2\tilde{M} \tag{8.7}
\]

The introduction of equation (8.7) in equation (8.6) leads (for every frequency $\omega$) to a quadratic eigenvalue problem in terms of the wavenumber $k_x$. This eigenvalue problem can be reformulated as a linear eigenvalue problem of the same size [12], which can be solved with standard techniques.

The resulting wavenumbers $k_x$ are complex-valued: the real part represents the harmonic variation with the distance $x$, while the imaginary part represents wave
attenuation in the $x$-direction. For a medium with zero material damping, each wavenumber $k_z$ is either purely real or purely imaginary. The surface waves corresponding to real-valued wavenumbers $k_z$ are propagative in the horizontal direction, while the surface corresponding to imaginary-valued wavenumbers are evanescent. For a medium with non-zero material damping, it is not trivial to differentiate between propagative and evanescent surface waves: the wavenumbers $k_z$ all consist of both a real and an imaginary part, and all surface waves are to a certain extent attenuated.

Due to the uncoupling of in-plane and out-of-plane motion in a layered halfspace, equation (8.6) can be reformulated as:

\[
\begin{align*}
\det \tilde{K}_{PSV} &= 0 \quad (8.8) \\
\det \tilde{K}_{SH} &= 0 \quad (8.9)
\end{align*}
\]

Equations (8.8) and (8.9) govern the propagation of Rayleigh waves and Love waves, respectively.

EDT provides two functions `eig_timpsv` and `eig_tlmsh` to solve the eigenvalue problems in equations (8.3) and (8.4). These functions return the complex wavenumbers $k_z$ characterizing the Rayleigh and Love waves in a layered stratum, as well as the corresponding eigenvectors $\tilde{\phi}$. The eigenvectors $\tilde{\phi}$ collect the displacements at the interfaces between elements, where the vertical displacements are multiplied with the imaginary unit $i$ (see also chapter 3).

**Example 8.2: Surface waves in a homogeneous layer on bedrock.**

In this example, the thin layer method is used to compute the out-of-plane surface waves or Love waves in a homogeneous layer on bedrock. The layer has a thickness $h = 5$ m, a shear wave velocity $C_s = 200$ m/s, and a density $\rho = 1800$ kg/m$^3$. It is subdivided into 50 thin layer elements. Zero material damping is assumed, so that the wavenumbers $k_z$ corresponding to propagative modes are purely real, and the wavenumbers $k_z$ corresponding to evanescent modes are purely imaginary.

First, the EDT function `k_tlmsh` is used to compute the matrices $\tilde{A}_{SH}$, $\tilde{B}_{SH}$, $\tilde{G}_{SH}$, and $\tilde{M}_{SH}$, which constitute the stiffness matrix $\tilde{K}_{SH}$ for out-of-plane wave propagation. The last row and the last column of these matrices are eliminated in order to model the clamped boundary condition at the bottom of the stratum.

Next, the function `eig_tlmsh` is used to compute the complex wavenumbers $k_z$ corresponding to the surface waves in the layer. The computation is performed in a frequency range from 0 Hz to 100 Hz.

Finally, the phase velocity $C_L(\omega)$ of the Love waves is calculated as $C_L = \omega/\text{Re}[k_z]$ for the propagative modes, i.e. the modes characterized by a purely real wavenumber $k_z$. 
% SOIL PROPERTIES
h = repmat(0.1,50,1); % Layer thickness [m]
Cs = 200; % Shear wave velocity [m/s]
Ds = 0; % Shear damping ratio [-]
rho = 1800; % Density [kg/m^3]

% FREQUENCY SAMPLING
f=0:0.1:100; % Frequency [Hz]
omega=2*pi*f; % Frequency [rad/s]

% THIN LAYER STIFFNESS MATRICES
[A,B,G,M] = k_tlmsh(h,Cs,Ds,rho);
A=A(1:end-1,1:end-1);
B=B(1:end-1,1:end-1);
G=G(1:end-1,1:end-1);
M=M(1:end-1,1:end-1);

% SOLVE EIGENVALUE PROBLEM
k = eig_tlmsh(A,G,M,omega); % Wavenumbers [rad/m]
C = repmat(omega,size(k,1),1)./k; % Phase velocity [rad/m]
C(imag(k)~=0) = nan; % Eliminate non-propagative modes

% PLOT COMPLEX WAVENUMBERS FOR ALL MODES
figure;
plot3(real(k),imag(k),f);
xlabel('Re(k_x) [rad/m]);
ylabel('Im(k_x) [rad/m]');
zlabel('Frequency [Hz]');

% PLOT PHASE VELOCITY FOR PROPAGATIVE MODES
figure;
plot(f,C);
xlabel('Frequency [Hz]');
ylabel('Phase velocity [m/s]');

Figure 8.3 shows, for all frequencies \( \omega \), the complex wavenumbers \( k_x \) corresponding to all Love waves. At zero frequency, all wavenumbers \( k_x \) are imaginary, and all surface waves are evanescent. At the frequency \( f_n = \frac{(2n-1)Cs}{4h} \), the wavenumber of the \( n \)-th surface wave becomes real, and the surface wave becomes propagative. The frequency \( f_n \) is referred to as the cut-off frequency of the \( n \)-th surface wave.

Figure 8.4 shows the phase velocity of the propagative Love waves. The cut-off frequencies \( f_n \) of the Love waves are clearly visible in this figure.
8.3 Wavenumber content of the Green’s function

The relevant surface wave mode in the SASW method is the mode that dominates the response recorded at the surface. For soils where the shear modulus increases smoothly with depth, the dominant mode is the fundamental Rayleigh wave. If
the shear modulus varies irregularly with depth (e.g. if the soil contains soft layers overlain by stiffer layers), higher modes may dominate the response [7, 29].

The dominant surface wave under a vertical load at the soil’s surface can be identified from the wavenumber content of the Green’s function \( \tilde{u}_G^{zz}(z' = 0, k_x, z = 0, \omega) \) of the soil. The Green’s function \( \tilde{u}_G^{zz}(z' = 0, k_x, z = 0, \omega) \) represents the vertical response of the soil due to vertical load at the surface, and can be computed with the direct stiffness method or the thin layer method. More details on the calculation of Green’s functions of layered media are given in the following chapter.

The wavenumber content of the Green’s function \( \tilde{u}_G^{zz}(z' = 0, k_x, z = 0, \omega) \) exhibits peaks that reveal the presence of Rayleigh waves. The largest peak corresponds to the dominant surface wave. For each frequency \( \omega \), the phase velocity \( C_R(\omega) \) of the dominant surface wave follows from the position of this peak in the wavenumber content of the Green’s function \( \tilde{u}_G^{zz}(z' = 0, k_x, z = 0, \omega) \). The attenuation coefficient \( A_R \) can be derived from the width of the peak, using the half power bandwidth method.

The EDT function \texttt{rayleigh\_domk} can be used to determine the phase velocity \( C_R(\omega) \) and the attenuation coefficient \( A_R(\omega) \) of the dominant Rayleigh wave in a layered halfspace under a vertical load at the surface. The use of this function is clarified in the following example.

**Example 8.3: Dominant surface wave in a layered halfspace.**

In this example, the dominant Rayleigh wave of a layered soil is computed with EDT, using the function \texttt{rayleigh\_domk}. The soil profile is described in table 8.1. This soil profile is considered as a benchmark profile in different papers that address the dominance of higher modes in the SASW method [16, 30].

<table>
<thead>
<tr>
<th>Layer</th>
<th>( h ) [m]</th>
<th>( C_s ) [m/s]</th>
<th>( C_p ) [m/s]</th>
<th>( D_s ) [-]</th>
<th>( D_p ) [-]</th>
<th>( \rho ) [kg/m(^3)]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>180</td>
<td>300</td>
<td>0.01</td>
<td>0.01</td>
<td>1800</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>120</td>
<td>857</td>
<td>0.01</td>
<td>0.01</td>
<td>1800</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>180</td>
<td>1285</td>
<td>0.01</td>
<td>0.01</td>
<td>1800</td>
</tr>
<tr>
<td>4</td>
<td>( \infty )</td>
<td>360</td>
<td>1323</td>
<td>0.01</td>
<td>0.01</td>
<td>1800</td>
</tr>
</tbody>
</table>

**Table 8.1: Soil properties.**

Two calculations are performed: first, the dispersion curves \( C_R(\omega) \) and the attenuation curves \( A_R(\omega) \) of all Rayleigh waves are computed with the function \texttt{eig\_dmpsv}, which has been introduced in section 8.1. Second, the dispersion and attenuation curves corresponding to the dominant Rayleigh wave is computed with the function \texttt{rayleigh\_domk}. Both calculations are performed in the frequency up to 100 Hz.
% SOIL PROPERTIES
h = [2 4 8 inf]; % Element thickness [m]
Cs = [180 120 180 360]; % Shear wave velocity [m/s]
Cp = [300 857 1285 1323]; % Dilatational wave velocity [m/s]
Ds = 0.01; % Shear damping ratio [-]
Dp = 0.01; % Dilatational damping ratio [-]
rho = 1800; % Density [kg/m^3]

% SOLVE EIGENVALUE PROBLEM
fMax = 100;
[f1,C1,A1] = eig_dsmpsv(h,Cs,Cp,Ds,Dp,rho,fMax);

% DERIVE DISPERSION CURVE OF DOMINANT MODE FROM GREEN'S FUNCTION
f2 = [1:1:100];
[C2,A2] = rayleigh_domk(h,Cs,Cp,Ds,Dp,rho,f2);

% PLOT DISPERSION CURVES
figure;
plot(f1,C1,'b',f2,C2,'r');
xlabel('Frequency [Hz]');
ylabel('Phase velocity [m/s]');

% PLOT ATTENUATION CURVES
figure;
plot(f1,A1,'b',f2,A2,'r');
xlabel('Frequency [Hz]');
ylabel('Attenuation coefficient [1/m]');

Figure 8.5 shows the resulting phase velocity $C_R(\omega)$ and attenuation coefficient
$A_R(\omega)$ of the dominant Rayleigh wave. Above 30 Hz, the dominance of higher
modes can clearly be observed in figure 8.5a.
Figure 8.5: (a) Phase velocity $C_R(\omega)$ and (b) attenuation coefficient $A_R(\omega)$ of the Rayleigh waves in a layered soil. The dashed line is the dominant Rayleigh wave under a vertical load at the soil’s surface.
Chapter 9

Forced vibration problems

This chapter addresses the computation of the response of a layered medium due to an external load. The response is calculated with the direct stiffness method in the frequency-wavenumber domain and subsequently transformed to the frequency-space domain.

While the direct stiffness method can be used to compute the response due to a load with an arbitrary spatial distribution, the focus in this chapter is restricted to the computation of the Green’s functions of the medium. The Green’s functions (or fundamental solutions) represent the response due to a unit load. Given the Green’s functions of a medium, the response due to an arbitrary load distribution can be computed by means of the dynamic reciprocity theorem, which is the basis of the boundary element method.

The chapter is organized as follows:

**Point load (p. 102)**
This section addresses the computation of the displacements and stresses in a layered medium due to a point load. These displacements and stresses correspond to the three-dimensional Green’s functions of the medium.

**Line load (p. 107)**
In this section, the wave field induced by a line load is considered. The resulting displacements and stresses correspond to the two-dimensional Green’s functions.

**Spatially harmonic line load (p. 109)**
This section focuses on the response of a layered medium due to a line load with an amplitude that varies harmonically in space.
Spatially harmonic traction varying in x-direction (p. 110)
This section considers the wave field induced by a traction on a horizontal plane with an amplitude that varies harmonically in one direction.

Spatially harmonic traction varying in x- and y-direction (p. 111)
This section considers the wave field induced by a traction on a horizontal plane with an amplitude that varies harmonically in two directions.

Disk load (p. 112)
This section adresses the computation of the displacements and stresses in a layered medium due to disk loads.

9.1 Point load

Figure 9.1: Point load.

This section addresses the computation of the response of a layered medium due to a time-harmonic point load (figure 9.1). The point load is applied in the direction $e_i$ (i.e. the x-direction, the y-direction, or the z-direction) at the position $(0,0,z')$.

In the frequency-space domain, the load $\hat{p}_j(x,y,z,\omega)$ is given by:

$$\hat{p}_j(x,y,z,\omega) = \delta_{ij}\delta(x)\delta(y)\delta(z-z')$$ (9.1)

where $\delta_{ij}$ is the Kronecker delta and $\delta(x)$ is the Dirac delta function.

The resulting displacements $\hat{u}_j(x,y,z,\omega)$ and stresses $\hat{\sigma}_{jk}(x,y,z,\omega)$ correspond to the three-dimensional Green’s functions of the layered medium, which are denoted by $\hat{u}_{ij}^G(z',x,y,z,\omega)$ and $\hat{\sigma}_{ijk}^G(z',x,y,z,\omega)$.

EDT provides the functions green3d_rec and green3d_cyl to compute the Green’s functions $\hat{u}_{ij}^G(z',x,y,z,\omega)$ and $\hat{\sigma}_{ijk}^G(z',x,y,z,\omega)$. These functions compute all three components of the displacement vector and all six components of the stress vector induced by a load in the x-direction, the y-direction, and the z-direction.

The function green3d_rec returns the results in Cartesian coordinates, while the function green3d_cyl returns the results in cylindrical coordinates.
The functions `green3d_recz` and `green3d_cylz` are similar, but they only compute the response due to a vertical load.

The functions `green3d_reczz` and `green3d_cylzz` only compute the vertical displacement due to a vertical load.

A function `green3d_bem` is also provided. This function is very similar to `green3d_cyl`, but it returns the results in a different format that is more suitable for boundary element calculations.

The Green’s functions of a layered medium are computed in the frequency-wavenumber domain by means of the direct stiffness method, and subsequently transformed to the frequency-space domain by means of an inverse Hankel transformation. A detailed description of the algorithm can be found in reference [25]. The calculation in the wavenumber domain implies that the user must specify a wavenumber sampling. The inverse Hankel transformation is performed by means of the function `fht` (chapter 6), which means that the wavenumber sampling must be logarithmic.

The wavenumber sampling must be passed to the functions `green3d_recz`, `green3d_cylz`, etc. in terms of slowness $p_x$, which is defined as $p_x = k_x/\omega$. The advantage of specifying the slowness $p_x$ instead of the wavenumber $k_x$ is that the same sampling can be used for all frequencies $\omega$, since the relevant wavenumber range for the calculation of the Green’s functions is proportional to the frequency $\omega$.

Due to the use of a logarithmic sampling scheme, the slowness sampling is determined by three parameters: the minimum slowness $p_{min}$, the maximum slowness $p_{max}$, and the number of samples $N$. The minimum slowness $p_{min}$ determines the accuracy of the three-dimensional Green’s functions in the far field, and the maximum slowness $p_{max}$ determines the accuracy in the near field. Good starting values for these parameters are $p_{min} = 10^{-5}/C_{max}$ and $p_{max} = 10^{5}/C_{min}$, where $C_{max}$ is the maximum (dilatational) wave velocity in the medium and $C_{min}$ is the minimum (shear) wave velocity. The number of samples $N$ determines the resolution in the wavenumber domain, which must be fine enough to capture the peaks in the Green’s functions due to the surface waves. If the damping ratio of the medium is small, these peaks are narrow, and a large number of samples is required. Accurate results can be obtained if the number of samples $N$ equals:

$$N = \left(\log_{10} p_{max} - \log_{10} p_{min}\right) \frac{C_{max}}{C_{min}} \frac{1}{D_{min}}$$  \hspace{1cm} (9.2)

$$\approx 2.3 \left(\log_{10} p_{max} - \log_{10} p_{min}\right) \frac{C_{max}}{C_{min}} \frac{1}{D_{min}}$$  \hspace{1cm} (9.3)

Finally, EDT also contains a function `fsgreen3d_cyl` to compute the three-dimensional Green’s functions of a homogeneous fullspace. The result is returned in cylindrical coordinates. The function `fsgreen3d_cyl` uses analytical expressions for the Green’s functions and is not based on the direct stiffness
method. As a consequence, it does not require the user to specify a slowness sampling.

**Example 9.1: The response of a halfspace due to a harmonic point load.**

This example addresses the computation of the displacement field in a homogeneous halfspace due to a vertical harmonic point load at the surface. The halfspace has a shear wave velocity $C_s = 150 \text{ m/s}$, a dilatational wave velocity $C_p = 300 \text{ m/s}$, a density $\rho = 1800 \text{ kg/m}^3$, and a damping ratio $D_s = D_p = 0.025$ in both shear and volumetric deformation. The excitation frequency is 50 Hz.

The response is calculated with the function `green3d_recz`. This function computes the wave field in the frequency-wavenumber domain, and transforms it to the frequency-space domain, using the appropriate integral transformations. The sampling in the wavenumber domain is determined by the slowness vector $p$. This vector is constructed with the standard MATLAB function `logspace` and contains $N = 2000$ logarithmically spaced slowness values between $10^{-7} \text{ s/m}$ and $10^3 \text{ s/m}$.

The function `green3d_recz` returns a multidimensional matrix $u_g$ containing the $x$, $y$, and $z$-component of the displacement field induced by a harmonic point load. Both the real and the imaginary part of the displacement field are visualized by means of the function `waveplot_rec`. In order to obtain a color plot of the displacement norm on an undeformed mesh, the deformation scale is set to zero.

```matlab
% SOIL PROPERTIES
h = inf;  % Element thickness [m]
Cs = 150; % Shear wave velocity [m/s]
Cp = 300; % Dilatational wave velocity [m/s]
Ds = 0.025; % Shear damping ratio [-]
Dp = 0.025; % Dilatational damping ratio [-]
rho = 1800; % Density [kg/m^3]

% SAMPLING
zs = 0;    % Source depth [m]
p = logspace(-7,3,2000); % Slowness [s/m]
x = 0:0.1:20; % Receiver x-coordinates [m]
y = 0;    % Receiver y-coordinates [m]
z = 0:0.1:12; % Receiver z-coordinates [m]
f = 50;    % Frequency [Hz]
omega = 2*pi*f; % Frequency [rad/s]

% GREEN'S DISPLACEMENT FUNCTIONS
ug = green3d_recz(h,Cs,Cp,Ds,Dp,rho,zs,p,x,y,z,omega);

% PLOT RESULT (REAL PART)
```
The results are shown in figure 9.2. Along z-axis, the motion consists of a dilatational wave. A shear window is travelling in an average direction of $45^\circ$.

In a region adjacent to the surface, the Rayleigh wave is clearly visible.

Example 9.2: The response of a halfspace due to an impulsive point load.

This example focuses on the computation of the response of a homogeneous halfspace due to a vertical impulsive point load at the surface. The properties of the halfspace are the same as in the previous example.

First, the function `green3d_recz` is used to obtain the wave field in the frequency-space domain. The calculation is performed for 1500 logarithmically spaced frequencies between $10^{-2}$ Hz and $10^4$ Hz. Next, the wave field is transformed to the time-space domain by means of an inverse Fourier transformation, using the function `logifft`. This function is based on the functions `logffcos` and `logffsin`. 

```matlab
% PLOT RESULT (REAL PART)
figure;
waveplot_rec(x,y,z,real(ug), 'DefScale', 0);
caxis([0 1e-9]);
shading('interp');
xlabel('x [m]');
zlabel('z [m]');
```

```matlab
% PLOT RESULT (IMAGINARY PART)
figure;
waveplot_rec(x,y,z,imag(ug), 'DefScale', 0);
xlabel('x [m]');
zlabel('z [m]');
```

**Figure 9.2:** Norm of the (a) real and (b) imaginary part of the displacement field $\hat{u}(x, y, z, \omega)$ in a homogeneous halfspace loaded by a vertical harmonic point load at the surface.
logffsin introduced in chapter 6. In order to save computer memory, the response is computed separately for each receiver depth, even though this is less efficient in terms of computation time.

\% SOIL PROPERTIES

\h = inf; % Element thickness [m]
\Cs = 150; % Shear wave velocity [m/s]
\Cp = 300; % Dilatational wave velocity [m/s]
\Ds = 0.025; % Shear damping ratio [-]
\Dp = 0.025; % Dilatational damping ratio [-]
\rho = 1800; % Density [kg/m^3]

\% SAMPLING

\zs = 0; % Source depth [m]
\p = logspace(-7,3,2000); % Slowness [s/m]
\x = 0:0.1:20; % Receiver x-coordinates [m]
\y = 0; % Receiver y-coordinates [m]
\z = 0:0.1:12; % Receiver z-coordinates [m]
\nx=length(x); % Number of receiver x-coordinates
\ny=length(y); % Number of receiver y-coordinates
\nz=length(z); % Number of receiver z-coordinates
\f = logspace(-2,4,1500); % Frequency [Hz]
\omega = 2*pi*f; % Frequency [rad/s]
\t = 0.025:0.025:0.100; % Time [s]
\nt = length(t); % Number of time steps

\% GREEN'S FUNCTIONS (CALCULATE PER RECEIVER DEPTH TO SAVE MEMORY)

ug=zeros(3,nx,1,nz,nt);
for \iz=1:nz
  Ug0=green3d_recz(h,Cs,Cp,Ds,Dp,rho,zs,p,x,y,z(iz),omega);
  ug0=logifft(Ug0,omega,t,6);
  ug(:,:,iz,:)=ug0;
end

\% FIGURES

for \it = 1:nt
  figure;
  waveplot_rec(x,y,z,ug(:,:,,:,\it),'DefScale',0);
  xlabel('x [m]');
  zlabel('z [m]');
end

Figure 9.3 shows four snapshots of the wave field induced by the impulsive point load. The figure reveals the presence of four types of waves. The first type is the P-wave, which travels at 300 m/s and is visible at 7.5 m, 15 m, 22.5 m, and 30 m
from the source in the successive snapshots. The second type is the S-wave, with a phase velocity of 150 m/s, visible at 3.75 m, 7.5 m, 11.25 m, and 15 m from the source. The third type is the Rayleigh wave, which travels along the surface at 139.9 m/s. The fourth type is the head wave, which is a shear wave that originates from the interaction of the P-wave with the free surface.

Figure 9.3: Norm of the displacement field $u(x, y, z, t)$ in a homogeneous halfspace loaded by a vertical impulsive point load at the surface at (a) 0.025 s, (b) 0.050 s, (c) 0.075 s, and (d) 0.100 s after the impulse.

9.2 Line load

This section addresses the computation of the response of a layered medium due to a time-harmonic line load, applied along a line parallel to the $y$-axis (figure 9.4). This is a two-dimensional problem: the wave field is invariant in the $y$-direction.

The line load is applied along the line ($x = 0, z = z'$) in the direction $e_i$. In the frequency-space domain, the load $\hat{p}_j(x, z, \omega)$ is given by:

$$\hat{p}_j(x, z, \omega) = \delta_{ij} \delta(x) \delta(z - z')$$  \hspace{1cm} (9.4)
The resulting displacements $\hat{u}_{ij}(x, z, \omega)$ and stresses $\hat{\sigma}_{jk}(x, z, \omega)$ correspond to the two-dimensional Green’s functions of the layered medium, which are denoted by $\hat{u}_{ij}^G(z', x, z, \omega)$ and $\hat{\sigma}_{jk}^G(z', x, z, \omega)$.

Due to the uncoupling of in-plane and out-of-plane motion, a line load in the $y$-direction does not give rise to displacements or stresses in the $x$- or $z$-direction, and vice versa. This implies that the in-plane and the out-of-plane components of the two-dimensional Green’s functions $\hat{u}_{ij}^G(z', x, z, \omega)$ and $\hat{\sigma}_{jk}^G(z', x, z, \omega)$ can be computed independently, and that the coupling terms are zero. In EDT, the in-plane components are computed by the function `green2d_inplane`, while the out-of-plane components are computed by the function `green2d_outofplane`.

The function `green2d_z` is similar to `green2d_inplane`, but it only computes the response due to a vertical load.

The functions `green2d_zz` is also similar, but it only computes the vertical displacement due to a vertical load.

The Green’s functions are computed in the frequency-wavenumber domain by means of the direct stiffness method, and subsequently transformed to the frequency-space domain by means of an inverse Fourier transformation. The calculation in the wavenumber domain implies that the user must specify a wavenumber sampling. The inverse Fourier transformation is performed by means of the functions `logffcos` and `logffsin` (chapter 6), which means that the wavenumber sampling must be logarithmic. In section 9.1, guidelines regarding the wavenumber sampling are given for the computation of the response due to a point load. The same guidelines can be followed in the case of a line load.

EDT also provides the functions `fsgreen2d_inplane` and `fsgreen2d_outofplane` to compute the two-dimensional Green’s functions of a homogeneous fullspace. These functions are based on analytical expressions for the Green’s functions and do not rely on the direct stiffness method. As a consequence, they do not require the user to specify a wavenumber sampling.
9.3 Spatially harmonic line load

This section focuses on the response of a layered medium due to a time-harmonic line load with an amplitude that varies harmonically with respect to the $y$-coordinate (figure 9.5). The variation is characterized by a wavenumber $k_y$.

**Figure 9.5:** Spatially harmonic line load.

The line load is applied along the line $(x = 0, z = z')$ in the direction $\mathbf{e}_i$. In the frequency-space domain, the load $\hat{p}_j(x, y, z, \omega)$ is given by:

$$\hat{p}_j(x, y, z, \omega) = \delta_{ij} \delta(x) \delta(z - z') \cos k_y y$$  (9.5)

Since the load is harmonic with respect to the $y$-coordinate, the resulting displacements $\hat{u}_j(x, y, z, \omega)$ and stresses $\hat{\sigma}_{jk}(x, y, z, \omega)$ are also harmonic:

$$\hat{u}_j(x, y, z, \omega) = \tilde{u}_j(x, z, \omega) \cos k_y y$$  (9.6)

$$\hat{\sigma}_{jk}(x, y, z, \omega) = \tilde{\sigma}_{jk}(x, z, \omega) \cos k_y y$$  (9.7)

The displacements $\tilde{u}_j(x, z, \omega)$ and the stresses $\tilde{\sigma}_{jk}(x, z, \omega)$ are referred to as the Green’s functions in the $(x, k_y)$-domain, which are denoted by $\tilde{u}_{ij}^G(z', x, k_y, z, \omega)$ and $\tilde{\sigma}_{ijk}^G(z', x, k_y, z, \omega)$.

EDT provides a function `greenf` to compute the Green’s functions in the $(x, k_y)$-domain. The Green’s functions are calculated in the frequency-wavenumber domain by means of the direct stiffness method, and subsequently transformed to the $(x, k_y)$-domain by means of an inverse Fourier transformation. The calculation in the wavenumber domain implies that the user must specify a wavenumber sampling. The inverse Fourier transformation is performed by means of the functions `logffcos` and `logffsin` (chapter 6), which means that the wavenumber sampling must be logarithmic. In section 9.1, guidelines regarding the wavenumber sampling are given for the computation of the response due to a point load. The same guidelines can be followed in the case of a line load with a spatially harmonic amplitude.

In addition, EDT contains a function `fsgreenf` to compute the Green’s functions in the $(x, k_y)$-domain for a homogeneous fullspace. This function is based on
analytical expressions for the Green’s functions and does not rely on the direct stiffness method. As a consequence, it does not require the user to specify a wavenumber sampling.

\section{9.4 Spatially harmonic traction varying in $x$-direction}

This section considers the response of a layered medium due to a time-harmonic traction on a horizontal plane, where the amplitude of the traction varies harmonically with respect to the $x$-coordinate (figure 9.6). The harmonic variation is characterized by a wavenumber $k_x$. This is a two-dimensional problem: the wave field is invariant in the $y$-direction.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure9.6.png}
\caption{Spatially harmonic traction varying in $x$-direction.}
\end{figure}

The load is applied at a depth $z'$ in the direction $e_x$. In the frequency-space domain, the load $\hat{p}_j(x, z, \omega)$ is given by:

$$\hat{p}_j(x, z, \omega) = \delta_{ij} \delta(z - z') \cos k_x x$$ \hfill (9.8)

Since the load is harmonic with respect to the $x$-coordinate, the resulting displacements $\hat{u}_j(x, z, \omega)$ and stresses $\hat{\sigma}_{jk}(x, z, \omega)$ are also harmonic:

$$\hat{u}_j(x, z, \omega) = \hat{u}_j(z, \omega) \cos k_x x$$ \hfill (9.9)

$$\hat{\sigma}_{jk}(x, z, \omega) = \hat{\sigma}_{jk}(z, \omega) \cos k_x x$$ \hfill (9.10)

The displacements $\hat{u}_j(z, \omega)$ and the stresses $\hat{\sigma}_{jk}(z, \omega)$ are referred to as the Green’s functions in the wavenumber domain, which are denoted by $\hat{u}_{ij}^G(z', k_x, z, \omega)$ and $\hat{\sigma}_{ijk}^G(z', k_x, z, \omega)$.

Due to the uncoupling of in-plane and out-of-plane motion, a traction in the $y$-direction does not give rise to displacements or stresses in the $x$- or $z$-direction, and vice versa. This implies that the in-plane and the out-of-plane components of the Green’s functions $\hat{u}_{ij}^G(z', k_x, z, \omega)$ and $\hat{\sigma}_{ijk}^G(z', k_x, z, \omega)$ can be computed independently, and that the coupling terms are zero.
In EDT, the in-plane components are computed by the function `green_psv`, while the out-of-plane components are computed by the function `green_sh`. However, these functions do not return all components of the Green’s stress tensor $\tilde{\sigma}_{ijk}(z', k_x, z, \omega)$: only the tractions $\tilde{t}_{ij}(z', k_x, z, \omega) = \tilde{\sigma}_{ijz}(z', k_x, z, \omega)$ on a horizontal plane are computed. The other components of the Green’s stress tensor $\tilde{\sigma}_{ijk}(z', k_x, z, \omega)$ can be obtained according to equations (2.9) and (2.10).

### 9.5 Spatially harmonic traction varying in $x$- and $y$-direction

This section addresses the response of a layered medium due to a time-harmonic traction on a horizontal plane with an amplitude that varies harmonically with respect to both the $x$-coordinate and the $y$-coordinate (figure 9.7). The variation is characterized by two wavenumbers $k_x$ and $k_y$.

![Figure 9.7: Spatially harmonic traction varying in $x$- and $y$-direction.](image)

The load is applied at a depth $z'$ in the direction $e$. In the frequency-space domain, the load $\hat{p}_j(x, y, z, \omega)$ is given by:

$$\hat{p}_j(x, y, z, \omega) = \delta_{ij}\delta(z - z') \cos k_x x \cos k_y y$$  \hspace{1cm} (9.11)

Since the load is harmonic with respect to the $x$-coordinate and the $y$-coordinate, the resulting displacements $\hat{u}_j(x, y, z, \omega)$ and stresses $\hat{\sigma}_{jk}(x, y, z, \omega)$ are also harmonic:

$$\hat{u}_j(x, y, z, \omega) = \tilde{u}_j(z, \omega) \cos k_x x \cos k_y y$$  \hspace{1cm} (9.12)

$$\hat{\sigma}_{jk}(x, z, \omega) = \tilde{\sigma}_{jk}(z, \omega) \cos k_x x \cos k_y y$$  \hspace{1cm} (9.13)

The displacements $\tilde{u}_j(z, \omega)$ and the stresses $\tilde{\sigma}_{jk}(z, \omega)$ are referred to as the Green’s functions in the $(k_x, k_y)$-domain, which are denoted by $\tilde{u}^G_{ij}(z', k_x, k_y, z, \omega)$ and $\tilde{\sigma}^G_{ijk}(z', k_x, k_y, z, \omega)$.

In EDT, the Green’s functions in the $(k_x, k_y)$-domain are computed by the function `greenff`. The computation is performed directly in the wavenumber domain, and no integral transformation is involved.
The function `greenff_z` is similar to `greenff`, but it only computes the response due to a vertical load.

The functions `greenff_zz` is also similar, but it only computes the vertical displacement due to a vertical load.

### 9.6 Disk load

![Disk loads](image)

Figure 9.8: Disk loads corresponding to a unit force in the $x$, $y$, and $z$-direction (cases 1 to 3) and a unit moment around the $x$, $y$, and $z$-axis (cases 4 to 6).

This section addresses the computation of the response of a layered medium due to time-harmonic disk loads (figure 9.8). Six cases are considered, corresponding to a unit force in the $x$, $y$, and $z$-direction (cases 1 to 3) and a unit moment around the $x$, $y$, and $z$-axis (cases 4 to 6).

In the frequency-space domain, the load $\hat{p}_j(x, y, z, \omega)$ is defined for case 1 as:

\[
\hat{p}_x(x, y, z, \omega) = \frac{1}{\pi R^2} H(R^2 - x^2 - y^2) \delta(z - z') \\
\hat{p}_y(x, y, z, \omega) = 0 \\
\hat{p}_z(x, y, z, \omega) = 0
\]

(9.14) (9.15) (9.16)
The load \( \hat{p}_j(x, y, z, \omega) \) is defined for case 2 as:

\[
\begin{align*}
\hat{p}_x(x, y, z, \omega) &= 0 \\
\hat{p}_y(x, y, z, \omega) &= \frac{1}{\pi R^2} H(R^2 - x^2 - y^2) \delta(z - z') \\
\hat{p}_z(x, y, z, \omega) &= 0
\end{align*}
\] (9.17) (9.18) (9.19)

The load \( \hat{p}_j(x, y, z, \omega) \) is defined for case 3 as:

\[
\begin{align*}
\hat{p}_x(x, y, z, \omega) &= 0 \\
\hat{p}_y(x, y, z, \omega) &= 0 \\
\hat{p}_z(x, y, z, \omega) &= \frac{1}{\pi R^2} H(R^2 - x^2 - y^2) \delta(z - z')
\end{align*}
\] (9.20) (9.21) (9.22)

The load \( \hat{p}_j(x, y, z, \omega) \) is defined for case 4 as:

\[
\begin{align*}
\hat{p}_x(x, y, z, \omega) &= 0 \\
\hat{p}_y(x, y, z, \omega) &= 0 \\
\hat{p}_z(x, y, z, \omega) &= \frac{2y}{\pi R^5} H(R^2 - x^2 - y^2) \delta(z - z')
\end{align*}
\] (9.23) (9.24) (9.25)

The load \( \hat{p}_j(x, y, z, \omega) \) is defined for case 5 as:

\[
\begin{align*}
\hat{p}_x(x, y, z, \omega) &= 0 \\
\hat{p}_y(x, y, z, \omega) &= 0 \\
\hat{p}_z(x, y, z, \omega) &= \frac{2x}{\pi R^5} H(R^2 - x^2 - y^2) \delta(z - z')
\end{align*}
\] (9.26) (9.27) (9.28)

The load \( \hat{p}_j(x, y, z, \omega) \) is defined for case 6 as:

\[
\begin{align*}
\hat{p}_x(x, y, z, \omega) &= -\frac{2}{\pi R^4} \frac{y}{R} H(R^2 - x^2 - y^2) \delta(z - z') \\
\hat{p}_y(x, y, z, \omega) &= \frac{2}{\pi R^4} \frac{x}{R} H(R^2 - x^2 - y^2) \delta(z - z') \\
\hat{p}_z(x, y, z, \omega) &= 0
\end{align*}
\] (9.29) (9.30) (9.31)

where \( \delta(x) \) is the Dirac delta function and \( H(x) \) is the Heaviside step function.

EDT provides the functions \texttt{disk3d_rec} and \texttt{disk3d_cyl} to compute the disk load solutions. These functions compute all three components of the displacement vector \( \hat{u}_i(z', x, y, z, \omega) \) and all six components of the stress tensor \( \hat{\sigma}_{ij}(z', x, y, z, \omega) \) induced by the six disk load cases defined above. The function \texttt{disk3d_rec} returns the results in Cartesian coordinates, while the function \texttt{disk3d_cyl} returns the results in cylindrical coordinates.
The functions disk3d_recz and disk3d_cylz are similar, but they only compute the response due to a vertical disk load (case 3).

The disk load solutions of a layered medium are computed in the frequency-wavenumber domain by means of the direct stiffness method, and subsequently transformed to the frequency-space domain by means of an inverse Hankel transformation. The calculation in the wavenumber domain implies that the user must specify a wavenumber sampling. The inverse Hankel transformation is performed by means of the function fht (chapter 6), which means that the wavenumber sampling must be logarithmic. In section 9.1, guidelines regarding the wavenumber sampling are given for the computation of the response due to a point load. The same guidelines can be followed in the case of disk loads.
Chapter 10

Functions — By category

10.1 Direct stiffness method

ke_dsmpsv  Element stiffness matrix (P-SV-waves).
ke_dsmsh   Element stiffness matrix (SH-waves).
k_dsmpsv   Global stiffness matrix (P-SV-waves).
k_dsmsh    Global stiffness matrix (SH-waves).
asmk_psv   Assemble stiffness matrix (P-SV-waves).
asmk_sh    Assemble stiffness matrix (SH-waves).
ne_dsmpsv  Element displacement shape functions (P-SV-waves).
ne_dsmsh   Element displacement shape functions (SH-waves).
n_dsmpsv   Global displacement shape functions (P-SV-waves).
n_dsmsh    Global displacement shape functions (SH-waves).
be_dsmpsv  Element traction shape functions (P-SV-waves).
be_dsmsh   Element traction shape functions (SH-waves).
b_dsmpsv   Global traction shape functions (P-SV-waves).
b_dsmsh    Global traction shape functions (SH-waves).

10.2 Thin layer method

ke_tlm-psv Element stiffness matrix (P-SV-waves).
ke_tlmsh   Element stiffness matrix (SH-waves).
k_tlm-psv  Global stiffness matrix (P-SV-waves).
k_tlmsh   Global stiffness matrix (SH-waves).
addk_tlm   Add components of stiffness matrix.
asmk_psv   Assemble stiffness matrix (P-SV-waves).
asmk_sh Assemble stiffness matrix (SH-waves).
b_tlmpsv Global traction shape functions (P-SV-waves).
b_tlmsh Global traction shape functions (SH-waves).
be_tlmpsv Element traction shape functions (P-SV-waves).
be_tlmsh Element traction shape functions (SH-waves).
n_tlmpsv Global displacement shape functions (P-SV-waves).
n_tlmsh Global displacement shape functions (SH-waves).
ne_tlmpsv Element displacement shape functions (P-SV-waves).
ne_tlmsh Element displacement shape functions (SH-waves).

10.3 Integral transformations

fht Fast Hankel transform.
logffcos Logarithmic fast Fourier cosine transform.
logffsin Logarithmic fast Fourier sine transform.
logfft Logarithmic fast forward Fourier transform of causal signals.
logifft Logarithmic fast inverse Fourier transform of real signals.

10.4 Amplification problems

amp_p Amplification of incident P-waves in a layered halfspace.
amp_sh Amplification of incident SH-waves in a layered halfspace.
amp_sv Amplification of incident SV-waves in a layered halfspace.
linamp Linear 1-D site response analysis.
eqlinamp Equivalent linear 1-D site response analysis.
eqlinampf Frequency dependent equivalent linear 1-D site response analysis.

10.5 Surface waves

eig_dsmpsv Dispersion curves of a layered halfspace (direct stiffness method, P-SV-waves).
eig_dsmsh Dispersion curves of a layered halfspace (direct stiffness method, SH-waves).
10.6 Forced vibration problems

- **eig_tlmpsv**: Stiffness matrix eigen decomposition (thin layer method, P-SV-waves).
- **eig_tlmsh**: Stiffness matrix eigen decomposition (thin layer method, SH-waves).
- **rayleigh_domk**: Dominant Rayleigh wave velocity in a layered soil.

### Green's Functions

- **green3d_rec**: 3D Green's functions in Cartesian coordinates.
- **green3d_cyl**: 3D Green's functions in cylindrical coordinates.
- **green3d_recz**: 3D Green's functions $\tilde{u}^G_{zj}$ and $\tilde{\sigma}^G_{zjk}$ in Cartesian coordinates.
- **green3d_cylz**: 3D Green's functions $\tilde{u}^G_{zj}$ and $\tilde{\sigma}^G_{zjk}$ in cylindrical coordinates.
- **green3d_reczz**: 3D Green's functions in Cartesian coordinates.
- **green3d_cylzz**: 3D Green's functions in cylindrical coordinates.
- **fsgreen3d_cyl**: 3D fullspace Green's functions in cylindrical coordinates.
- **green2d_inplane**: In-plane 2D Green's functions.
- **green2d_outofplane**: Out-of-plane 2D Green's functions.
- **green2d_z**: 2D Green's functions $\tilde{u}^G_{zj}$ and $\tilde{\sigma}^G_{zjk}$.
- **green2d_zz**: 2D Green's function $\tilde{u}^G_{zz}$.
- **fsgreen2d_inplane**: In-plane 2D fullspace Green's functions.
- **fsgreen2d_outofplane**: Out-of-plane 2D fullspace Green's functions.
- **green**: Green's functions in the $(x,k_y)$-domain.
- **fsgreenf**: Fullspace Green's functions in the $(x,k_y)$-domain.
- **green_psv**: Wavenumber domain P-SV Green's functions.
- **green_sh**: Wavenumber domain SH Green's functions.
- **green_z**: Wavenumber domain Green's functions $\tilde{u}^G_{zj}$ and $\tilde{\sigma}^G_{zjk}$.
- **green_zz**: Wavenumber domain Green's functions $\tilde{u}^G_{zz}$ and $\tilde{\sigma}^G_{zz}$.
- **greenff**: Green's functions in the $(k_x,k_y)$-domain.
- **greenff_z**: Green's functions $\tilde{u}^G_{zj}$ and $\tilde{\sigma}^G_{zjk}$ in the $(k_x,k_y)$-domain.
- **greenff_zz**: Green's function $\tilde{u}^G_{zz}$ in the $(k_x,k_y)$-domain.
- **disk3d_rec**: 3D disk load solutions in Cartesian coordinates.
- **disk3d_recz**: 3D vertical disk load solution in Cartesian coordinates.
- **disk3d_cyl**: 3D disk load solutions in cylindrical coordinates.
- **disk3d_cylz**: 3D vertical disk load solution in cylindrical coordinates.
10.7 Visualization

plotprofile  Plot a property of a layered medium as a function of depth.
waveplot_rec  Plot a wave field in Cartesian coordinates.
waveplot_cyl  Plot a wave field in cylindrical coordinates.
wavemovie_rec  Animate a wave field in Cartesian coordinates.
wavemovie_cyl  Animate a wave field in cylindrical coordinates.
wiggle  Plot wiggle traces.
arrows  Arrow plot of a 2D vector field.
addk_tlm

Purpose
Add components of a stiffness matrix computed with the thin layer method.

Syntax
K = ADDK_TLM(A,B,G,M,k,omega)

Input arguments
A, B, G, M  Components of the stiffness matrix according to the thin layer method (nDOF × nDOF).
k  Horizontal wavenumber.
omega  Circular frequency.

Output arguments
K  Stiffness matrix (nDOF × nDOF).

Description
K = ADDK_TLM(A,B,G,M,k,omega) calculates A*k^2 + B*k + G - omega^2*M.
The calculation is performed in an efficient way, exploiting the identical sparsity pattern of the matrices A, B, G, and M.

Example
Example 5.1: Vertical harmonic wave propagation in a layer on bedrock (p. 54).
**amp_p**

**Purpose**
Amplification of incident P-waves in a layered halfspace.

**Syntax**

\[
[ux,uz,tx,tz] = \text{AMP}_P(h,Cs,Cp,Ds,Dp,rho,phi,x,z,omega)
\]

**Input arguments**
- **h** Layer thickness, \(\text{INF}\) for a halfspace \((nElt \times 1)\) or \((1 \times 1)\).
- **Cs** Shear wave velocity \((nElt \times 1)\) or \((1 \times 1)\).
- **Cp** Dilatational wave velocity \((nElt \times 1)\) or \((1 \times 1)\).
- **Ds** Shear damping ratio \((nElt \times 1)\) or \((1 \times 1)\).
- **Dp** Dilatational damping ratio \((nElt \times 1)\) or \((1 \times 1)\).
- **rho** Density \((nElt \times 1)\) or \((1 \times 1)\).
- **phi** Angle of incidence of the P-wave w.r.t. the free surface \((nWave \times 1)\).
- **x** Receiver locations (horizontal coordinate) \((nxRec \times 1)\).
- **z** Receiver locations (vertical coordinate) \((nzRec \times 1)\).
- **omega** Circular frequency \((nFreq \times 1)\).

**Output arguments**
- **ux** Resulting displacements in \(x\)-direction \((nWave \times nxRec \times nzRec \times nFreq)\).
- **uz** Resulting displacements in \(z\)-direction \((nWave \times nxRec \times nzRec \times nFreq)\).
- **tx** Resulting tractions in \(x\)-direction \((nWave \times nxRec \times nzRec \times nFreq)\).
- **tz** Resulting tractions in \(z\)-direction \((nWave \times nxRec \times nzRec \times nFreq)\).

**Description**

\[
[ux,uz,tx,tz] = \text{AMP}_P(h,Cs,Cp,Ds,Dp,rho,phi,x,z,omega)
\]
computes the wavenumber domain response due to an incident P-wave in a layered halfspace using the direct stiffness method. The amplitude of the incident wave is equal to 1 at the point \((x = 0, z = z_0)\), where \(z_0\) is the depth of the lowest interface in the halfspace.

The number of elements \(nElt\) is equal to the maximum of the lengths of \(h, Cs, Cp, Ds, Dp,\) and \(rho\). If any of these parameters is defined as a scalar, an identical value is used for all elements.

**Example**

Example 7.1: Amplification of an incident P-wave in a layered halfspace (p. 71).
amp\_sh

Purpose
Amplification of incident SH-waves in a layered halfspace.

Syntax
[uy,ty] = AMP\_SH(h,Cs,Ds,rho,phi,x,z,omega)

Input arguments
- h: Layer thickness, \( \text{INF} \) for a halfspace \((\text{nElt} \times 1) \) or \((1 \times 1) \).
- Cs: Shear wave velocity \((\text{nElt} \times 1) \) or \((1 \times 1) \).
- Ds: Shear damping ratio \((\text{nElt} \times 1) \) or \((1 \times 1) \).
- rho: Density \((\text{nElt} \times 1) \) or \((1 \times 1) \).
- phi: Angle of incidence of the SH-wave w.r.t. the free surface \((\text{nWave} \times 1) \).
- x: Receiver locations (horizontal coordinate) \((\text{nxRec} \times 1) \).
- z: Receiver locations (vertical coordinate) \((\text{nzRec} \times 1) \).
- omega: Circular frequency \((\text{nFreq} \times 1) \).

Output arguments
- uy: Resulting displacements in \( y \)-direction \((\text{nWave} \times \text{nxRec} \times \text{nzRec} \times \text{nFreq}) \).
- ty: Resulting tractions in \( y \)-direction \((\text{nWave} \times \text{nxRec} \times \text{nzRec} \times \text{nFreq}) \).

Description
\([uy,ty] = \text{AMP\_SH}(h,Cs,Ds,rho,phi,x,z,omega)\) computes the wavenumber domain response due to an incident SH-wave in a layered halfspace using the direct stiffness method. The amplitude of the incident wave is equal to 1 at the point \((x = 0, z = z_0)\), where \(z_0\) is the depth of the lowest interface in the halfspace.

The number of elements \(\text{nElt}\) is equal to the maximum of the lengths of \(h, Cs, Ds,\) and \(rho\). If any of these parameters is defined as a scalar, an identical value is used for all elements.
**amp_sv**

**Purpose**
Amplification of incident SV-waves in a layered halfspace.

**Syntax**

\[
[ux,uz,tx,tz] = AMP\_SV(h,Cs,Cp,Ds,Dp,rho,phi,x,z,omega)
\]

**Input arguments**
- **h**  
  Layer thickness, INF for a halfspace \((nElt \times 1)\) or \((1 \times 1)\).
- **Cs**  
  Shear wave velocity \((nElt \times 1)\) or \((1 \times 1)\).
- **Cp**  
  Dilatational wave velocity \((nElt \times 1)\) or \((1 \times 1)\).
- **Ds**  
  Shear damping ratio \((nElt \times 1)\) or \((1 \times 1)\).
- **Dp**  
  Dilatational damping ratio \((nElt \times 1)\) or \((1 \times 1)\).
- **rho**  
  Density \((nElt \times 1)\) or \((1 \times 1)\).
- **phi**  
  Angle of incidence of the SV-wave w.r.t. the free surface \((nWave \times 1)\).
- **x**  
  Receiver locations (horizontal coordinate) \((nxRec \times 1)\).
- **z**  
  Receiver locations (vertical coordinate) \((nzRec \times 1)\).
- **omega**  
  Circular frequency \((nfreq \times 1)\).

**Output arguments**
- **ux**  
  Resulting displacements in \(x\)-direction \((nWave \times nxRec \times nzRec \times nFreq)\).
- **uz**  
  Resulting displacements in \(z\)-direction \((nWave \times nxRec \times nzRec \times nFreq)\).
- **tx**  
  Resulting tractions in \(x\)-direction \((nWave \times nxRec \times nzRec \times nFreq)\).
- **tz**  
  Resulting tractions in \(z\)-direction \((nWave \times nxRec \times nzRec \times nFreq)\).

**Description**

\[[ux,uz,tx,tz] = AMP\_SV(h,Cs,Cp,Ds,Dp,rho,phi,x,z,omega)\] computes the wavenumber domain response due to an incident SV-wave in a layered halfspace using the direct stiffness method. The amplitude of the incident wave is equal to 1 at the point \((x = 0, z = z_0)\), where \(z_0\) is the depth of the lowest interface in the halfspace.

The number of elements \(nElt\) is equal to the maximum of the lengths of \(h, Cs, Cp, Ds, Dp,\) and \(rho\). If any of these parameters is defined as a scalar, an identical value is used for all elements.
arrows

Purpose
Arrow plot of a 2D vector field.

Syntax
ARROWS(x,z,u,w)
ARROWS(...,'DefScale',s)
ARROWS(...,***)
[s,h] = ARROWS(...)

Input arguments
x  Horizontal coordinates (nx × 1).
z  Vertical coordinates (nz × 1).
u  Horizontal vector component (nx × nz).
w  Vertical vector component (nx × nz).

Output arguments
s  Deformation scale used.
h  Handle to the PLOT object.

Description
ARROWS(x,z,u,w) makes an arrow plot in the (x,z)-plane of the vector field (u,w).
ARROWS(...,'DefScale',s) sets the deformation scale to s. By default, the
demotion scale is automatically calculated.
ARROWS(...,***), redirects the parameters *** to the PLOT function.
[s,h] = ARROWS(...) returns the deformation scale s and a handle h to the PLOT
object.
asmk_psv

Purpose
Assemble stiffness matrix (P-SV-waves).

Syntax
K = ASMK_PSV(K1,K2,...)

Input arguments
K1,... Element or substructure stiffness matrices resulting from KE_DSMPSV, 
       KE_TLMPSV, K_DSMPSV, K_TLMPSV, or ASMK_PSV.

Output arguments
K Assembled stiffness matrix.

Description
K = ASMK_PSV(K1,K2,...) assembles the stiffness matrix K for P-SV-waves in 
a layered medium consisting of the elements or substructures with the stiffness 
matrices K1,K2,....

Example
Example 4.3: Stiffness matrices for a layered medium (p. 40).
asmk_sh

Purpose
Assemble stiffness matrix (SH-waves).

Syntax
K = ASMK_SH(K1,K2,...)

Input arguments
K1,... Element or substructure stiffness matrices resulting from KE_DSMSH,
    KE_TLMSH, K_DSMSH, K_TLMSH, or ASMK_SH.

Output arguments
K Assembled stiffness matrix.

Description
K = ASMK_SH(K1,K2,...) assembles the stiffness matrix K for SH-waves in a
layered medium consisting of the elements or substructures with the stiffness
matrices K1,K2,...
be_dsmpsv

Purpose
Element traction shape functions (direct stiffness method, P-SV-waves).

Syntax
[Bxe,Bze] = BE_DSMPSV(h,Cs,Cp,Ds,Dp,rho,k,omega,z,ds,dp)

Input arguments
h  Layer thickness, INF for a halfspace.
Cs  Shear wave velocity.
Cp  Dilatational wave velocity.
Ds  Shear damping ratio.
Dp  Dilatational damping ratio.
rho Density.
k  Horizontal wavenumber.
omega Circular frequency.
z  Receiver depth (nRec x 1).
ds  Shear wave direction. Default: 1.
dp  Dilatational wave direction. Default: equal to ds.

Output arguments
Bxe  Shape functions for horizontal traction (nRec x 4) or (nRec x 2).
Bze  Shape functions for vertical traction (nRec x 4) or (nRec x 2).

Description
[Bxe,Bze] = BE_DSMPSV(h,Cs,Cp,Ds,Dp,rho,k,omega,z,ds,dp) returns the traction shape functions used in the direct stiffness method for P-SV-waves in a layer or a halfspace.

The optional arguments ds and dp are only applicable to halfspace elements, and must be equal to 1 (outgoing waves) or -1 (incoming waves).

For receivers outside the layer or the halfspace, zero is returned.
be_dsmsh

**Purpose**
Element traction shape functions (direct stiffness method, SH-waves).

**Syntax**
```
Bye = BE_DSMSH(h,Cs,Ds,rho,k,omega,z,ds)
```

**Input arguments**
- `h`: Layer thickness, INF for a halfspace.
- `Cs`: Shear wave velocity.
- `Ds`: Shear damping ratio.
- `rho`: Density.
- `k`: Horizontal wavenumber.
- `omega`: Circular frequency.
- `z`: Receiver depth (nRec x 1).

**Output arguments**
- `Bye`: Traction shape functions (nRec x 2) or (nRec x 1).

**Description**
```
Bye = BE_DSMSH(h,Cs,Ds,rho,k,omega,z,ds) returns the traction shape functions used in the direct stiffness method for SH-waves in a layer or a halfspace.
```

The optional argument `ds` is only applicable to halfspace elements, and must be equal to 1 (outgoing waves) or -1 (incoming waves).

For receivers outside the layer or the halfspace, zero is returned.
be_tlmpsv

Purpose
Element traction shape functions (thin layer method, P-SV-waves).

Syntax
[Bxe,Bze] = BE_TLMPSV(h,Cs,Cp,Ds,Dp,rho,z)

Input arguments
h  Layer thickness.
Cs  Shear wave velocity.
Cp  Dilatational wave velocity.
Ds  Shear damping ratio.
Dp  Dilatational damping ratio.
rho Density.
z  Receiver depth (nRec × 1).

Output arguments
Bxe  Shape functions for horizontal traction (nRec × 4).
Bze  Shape functions for vertical traction (nRec × 4).

Description
[Bxe,Bze] = BE_TLMPSV(h,Cs,Cp,Ds,Dp,rho,z) returns the traction shape functions used in the thin layer method for P-SV-waves in a layer.
be_tlmsh

**Purpose**
Element traction shape functions (thin layer method, SH-waves).

**Syntax**

\[
\text{Bye} = \text{BE_TLMSH}(h, Cs, Ds, \rho, z)
\]

**Input arguments**
- \(h\) Layer thickness.
- \(Cs\) Shear wave velocity.
- \(Ds\) Shear damping ratio.
- \(\rho\) Density.
- \(z\) Receiver depth (\(n_{Rec} \times 1\)).

**Output arguments**
- \(\text{Bye}\) Shape functions for tractions (\(n_{Rec} \times 2\)).

**Description**

\(\text{Bye} = \text{BE_TLMSH}(h, Cs, Ds, \rho, z)\) returns the traction shape functions used in the thin layer method for SH-waves in a layer.
b_dsmpsv

Purpose
Global traction shape functions (direct stiffness method, P-SV-waves).

Syntax

\[ [Bx, Bz] = B_DSMPSV(h, Cs, Cp, Ds, Dp, rho, k, omega, z, ds, dp) \]

Input arguments

- **h**  
  Layer thickness, INF for a halfspace \((nElt \times 1)\) or \((1 \times 1)\).
- **Cs**  
  Shear wave velocity \((nElt \times 1)\) or \((1 \times 1)\).
- **Cp**  
  Dilatational wave velocity \((nElt \times 1)\) or \((1 \times 1)\).
- **Ds**  
  Shear damping ratio \((nElt \times 1)\) or \((1 \times 1)\).
- **Dp**  
  Dilatational damping ratio \((nElt \times 1)\) or \((1 \times 1)\).
- **rho**  
  Density \((nElt \times 1)\) or \((1 \times 1)\).
- **k**  
  Horizontal wavenumber.
- **omega**  
  Circular frequency.
- **z**  
  Receiver depth \((nRec \times 1)\).
- **ds**  
  Shear wave direction. Default: 1.
- **dp**  
  Dilatational wave direction. Default: equal to ds.

Output arguments

- **Bx**  
  Shape functions for the horizontal traction \((nRec \times nDOF)\).
- **Bz**  
  Shape functions for the vertical traction \((nRec \times nDOF)\).

Description

\[ [Bx, Bz] = B_DSMPSV(h, Cs, Cp, Ds, Dp, rho, k, omega, z, ds, dp) \]

returns the global traction shape functions used in the direct stiffness method for P-SV-waves in a layered soil.

The number of elements \(nElt\) is equal to the maximum of the lengths of \(h, Cs, Cp, Ds, Dp,\) and \(rho\). If any of these parameters is defined as a scalar, an identical value is used for all elements.

The optional arguments \(ds\) and \(dp\) are only applicable to halfspace elements, and must be equal to 1 (outgoing waves) or -1 (incoming waves).

For receivers outside the soil, zero is returned.
Example

Example 4.5: Vertical transient wave propagation in a layer on a halfspace (p. 47).
**b_dsmsh**

**Purpose**

Global traction shape functions (direct stiffness method, SH-waves).

**Syntax**

\[ \text{By} = \text{B\_DSMSH}(h, \text{Cs}, \text{Ds}, \rho, k, \omega, z, \text{ds}) \]

**Input arguments**

- \( h \): Layer thickness, \( \text{INF} \) for a halfspace \((n_{\text{Elt}} \times 1)\) or \((1 \times 1)\).
- \( \text{Cs} \): Shear wave velocity \((n_{\text{Elt}} \times 1)\) or \((1 \times 1)\).
- \( \text{Ds} \): Shear damping ratio \((n_{\text{Elt}} \times 1)\) or \((1 \times 1)\).
- \( \rho \): Density \((n_{\text{Elt}} \times 1)\) or \((1 \times 1)\).
- \( k \): Horizontal wavenumber.
- \( \omega \): Circular frequency.
- \( z \): Receiver depth \((n_{\text{Rec}} \times 1)\).
- \( \text{ds} \): Shear wave direction. Default: 1.

**Output arguments**

- \( \text{By} \): Traction shape functions \((n_{\text{Rec}} \times n_{\text{DOF}})\).

**Description**

\[ \text{By} = \text{B\_DSMSH}(h, \text{Cs}, \text{Ds}, \rho, k, \omega, z, \text{ds}) \]

returns the traction shape functions used in the direct stiffness method for SH-waves in a layered soil.

The number of elements \( n_{\text{Elt}} \) is equal to the maximum of the lengths of \( h \), \( \text{Cs} \), \( \text{Ds} \), and \( \rho \). If any of these parameters is defined as a scalar, an identical value is used for all elements.

The optional arguments \( \text{ds} \) is only applicable to halfspace elements, and must be equal to 1 (outgoing waves) or -1 (incoming waves).

For receivers outside the soil, zero is returned.
b_tlmsvm

**Purpose**
Global traction shape functions (thin layer method, P-SV-waves).

**Syntax**

```matlab
[Bx,Bz] = B_TLMPSV(h,Cs,Cp,Ds,Dp,rho,z)
```

**Input arguments**
- `h` Layer thickness (nElt x 1) or (1 x 1).
- `Cs` Shear wave velocity (nElt x 1) or (1 x 1).
- `Cp` Dilatational wave velocity (nElt x 1) or (1 x 1).
- `Ds` Shear damping ratio (nElt x 1) or (1 x 1).
- `Dp` Dilatational damping ratio (nElt x 1) or (1 x 1).
- `rho` Density (nElt x 1) or (1 x 1).
- `z` Receiver depth (nRec x 1).

**Output arguments**
- `Bx` Shape functions for horizontal traction (nRec x nDOF).
- `Bz` Shape functions for vertical traction (nRec x nDOF).

**Description**

```
[Bx,Bz] = B_TLMPSV(h,Cs,Cp,Ds,Dp,rho,z) returns the traction shape functions used in the thin layer method for P-SV-waves in a layered soil.
```

The number of elements nElt is equal to the maximum of the lengths of `h`, `Cs`, `Cp`, `Ds`, `Dp`, and `rho`. If any of these parameters is defined as a scalar, an identical value is used for all elements.

For receivers outside the soil, zero is returned.
b_tlmsh

Purpose
Global traction shape functions (thin layer method, SH-waves).

Syntax
By = B_TLMSH(h,Cs,Ds,rho,z)

Input arguments
h   Layer thickness (nElt x 1) or (1 x 1).
Cs  Shear wave velocity (nElt x 1) or (1 x 1).
Ds  Shear damping ratio (nElt x 1) or (1 x 1).
rho Density (nElt x 1) or (1 x 1).
z   Receiver depth (nRec x 1).

Output arguments
By   Shape functions for tractions (nRec x nDOF).

Description
By = B_TLMSH(h,Cs,Ds,rho,z) returns the traction shape functions used in the thin layer method for SH-waves in a layered soil.

The number of elements nElt is equal to the maximum of the lengths of h, Cs, Ds, and rho. If any of these parameters is defined as a scalar, an identical value is used for all elements.

For receivers outside the soil, zero is returned.
disk3d_cyl

**Purpose**

3D disk load solutions in cylindrical coordinates.

**Syntax**

```matlab
[u,s] = DISK3D_CYL(h,Cs,Cp,Ds,Dp,rho,R,zs,p,r,theta,z,omega)
```

**Input arguments**

- `h`: Layer thickness, INF for a halfspace (nElt x 1) or (1 x 1).
  - If `h(end) != INF`, the bottom interface is clamped.
  - If necessary, extra layers are defined so that the source locations `zs` coincide with interfaces between elements.
- `Cs`: Shear wave velocity (nElt x 1) or (1 x 1).
- `Cp`: Dilatational wave velocity (nElt x 1) or (1 x 1).
- `Ds`: Shear damping ratio (nElt x 1) or (1 x 1).
- `Dp`: Dilatational damping ratio (nElt x 1) or (1 x 1).
- `rho`: Density (nElt x 1) or (1 x 1).
- `R`: Radius of the disk load (1 x 1).
- `zs`: Source locations (vertical coordinate) (nzSrc x 1).
- `p`: Slowness, logarithmically sampled (nWave x 1).
  - If `omega != 0`, the wavenumber sampling is given by `k = omega * p`.
  - If `omega = 0`, the wavenumber sampling is given by `k = p`.
- `r`: Receiver locations (radial coordinate) (nrRec x 1).
- `theta`: Receiver locations (circumferential coordinate) (ntRec x 1).
- `z`: Receiver locations (vertical coordinate) (nzRec x 1).
- `omega`: Circular frequency (nFreq x 1).
Output arguments

\[ u \text{ (Displacements) } = u(iCase,1,\ldots) = \hat{u}_r, \]
\[ u(iCase,2,\ldots) = \hat{u}_\theta, \]
\[ u(iCase,3,\ldots) = \hat{u}_z \]

\[ s \text{ (Stresses) } = s(iCase,1,\ldots) = \hat{\sigma}_{rr}, \]
\[ s(iCase,2,\ldots) = \hat{\sigma}_{\theta\theta}, \]
\[ s(iCase,3,\ldots) = \hat{\sigma}_{zz}, \]
\[ s(iCase,4,\ldots) = \hat{\sigma}_{r\theta}, \]
\[ s(iCase,5,\ldots) = \hat{\sigma}_{\theta z}, \]
\[ s(iCase,6,\ldots) = \hat{\sigma}_{zr} \]

For receivers located on interfaces, the stresses in the underlying element are calculated.

Description

\[ [u,s] = \text{DISK3D}_CYL(h,Cs,Cp,Ds,Dp,rho,R,zs,p,r,theta,z,omega) \]
computes the 3D displacements and stresses for disk loads in a layered soil using the direct stiffness method. The displacements and stresses are returned in cylindrical coordinates. Six cases are considered, corresponding to a unit force in the \( x \), \( y \), and \( z \)-direction and a unit moment around the \( x \), \( y \), and \( z \)-axis. The tractions on a horizontal disk with radius \( R \) are defined for each case as:

Case 1 — Horizontal load in \( x \)-direction: \( \hat{p}_x = 1/(\pi R^2), r < R \).
Case 2 — Horizontal load in \( y \)-direction: \( \hat{p}_y = 1/(\pi R^2), r < R \).
Case 3 — Vertical load: \( \hat{p}_z = 1/(\pi R^2), r < R \).
Case 4 — Bending moment around \( x \)-axis: \( \hat{p}_z = 2/(\pi R^4) y, r < R \).
Case 5 — Bending moment around \( y \)-axis: \( \hat{p}_z = 2/(\pi R^4) x, r < R \).
Case 6 — Torsional moment around \( z \)-axis: \( \hat{p}_\theta = 2/(\pi R^4) r/R, r < R \).

The number of elements \( nElt \) is equal to the maximum of the lengths of \( h, Cs, Cp, Ds, Dp, \) and \( rho \). If any of these parameters is defined as a scalar, an identical value is used for all elements.
disk3d_cylz

Purpose
3D vertical disk load solution in cylindrical coordinates.

Syntax
[u,s] = DISK3D_CYLZ(h,Cs,Cp,Ds,Dp,rho,R,zs,p,r,z,omega)

Input arguments
h  Layer thickness, INF for a halfspace (nElt × 1) or (1 × 1).
   If h(end) ≠ INF, the bottom interface is clamped.
   If necessary, extra layers are defined so that the source locations zs coincide with interfaces between elements.
Cs  Shear wave velocity (nElt × 1) or (1 × 1).
Cp  Dilatational wave velocity (nElt × 1) or (1 × 1).
Ds  Shear damping ratio (nElt × 1) or (1 × 1).
Dp  Dilatational damping ratio (nElt × 1) or (1 × 1).
rho  Density (nElt × 1) or (1 × 1).
R  Radius of the disk load (1 × 1).
zs  Source locations (vertical coordinate) (nzSrc × 1).
p  Slowness, logarithmically sampled (nWave × 1).
   If omega ≠ 0, the wavenumber sampling is given by k = omega × p.
   If omega = 0, the wavenumber sampling is given by k = p.
r  Receiver locations (radial coordinate) (nrRec × 1).
z  Receiver locations (vertical coordinate) (nzRec × 1).
omega  Circular frequency (nFreq × 1).

Output arguments
u  Displacements (2 × nzSrc × nrRec × ntRec × nzRec × nFreq).
   u(1,...) = \hat{u}_r
   u(2,...) = \hat{u}_z
s  Stresses (4 × nzSrc × nrRec × ntRec × nzRec × nFreq).
   s(1,...) = \hat{\sigma}_{rr}
   s(2,...) = \hat{\sigma}_{\theta\theta}
   s(3,...) = \hat{\sigma}_{zz}
   s(4,...) = \hat{\sigma}_{rz}
For receivers located on interfaces, the stresses in the underlying element are calculated.
Description

\[ [u, s] = \text{DISK3D}_\text{CYLZ}(h, Cs, Cp, Ds, Dp, rho, R, zs, p, r, z, \omega) \] computes the 3D displacements and stresses for a vertical disk load in a layered soil using the direct stiffness method. The displacements and stresses are returned in cylindrical coordinates.

The number of elements \( n_{\text{Elt}} \) is equal to the maximum of the lengths of \( h, Cs, Cp, Ds, Dp, \) and \( rho \). If any of these parameters is defined as a scalar, an identical value is used for all elements.
disk3d_rec

Purpose
3D disk load solutions in Cartesian coordinates.

Syntax
[u,s] = DISK3D_REC(h,Cs,Cp,Ds,Dp,rho,R,zs,p,x,y,z,omega)

Input arguments

h    Layer thickness, INF for a halfspace (nElt × 1) or (1 × 1).
     If h(end) ≠ INF, the bottom interface is clamped.
     If necessary, extra layers are defined so that the source locations zs
     coincide with interfaces between elements.
Cs   Shear wave velocity (nElt × 1) or (1 × 1).
Cp   Dilatational wave velocity (nElt × 1) or (1 × 1).
Ds   Shear damping ratio (nElt × 1) or (1 × 1).
Dp   Dilatational damping ratio (nElt × 1) or (1 × 1).
rho  Density (nElt × 1) or (1 × 1).
R    Radius of the disk load (1 × 1).
zs   Source locations (vertical coordinate) (nzSrc × 1).
p    Slowness, logarithmically sampled (nWave × 1).
     If omega ≠ 0, the wavenumber sampling is given by k = omega × p.
     If omega = 0, the wavenumber sampling is given by k = p.
x    Receiver locations (x-coordinate) (nxRec × 1).
y    Receiver locations (y-coordinate) (nyRec × 1).
z    Receiver locations (z-coordinate) (nzRec × 1).
omega  Circular frequency (nFreq × 1).
Output arguments

\[ u(iCase,1,\ldots) = \hat{u}_x \]
\[ u(iCase,2,\ldots) = \hat{u}_y \]
\[ u(iCase,3,\ldots) = \hat{u}_z \]

\[ s(iCase,1,\ldots) = \hat{\sigma}_{xx} \]
\[ s(iCase,2,\ldots) = \hat{\sigma}_{yy} \]
\[ s(iCase,3,\ldots) = \hat{\sigma}_{zz} \]
\[ s(iCase,4,\ldots) = \hat{\sigma}_{xy} \]
\[ s(iCase,5,\ldots) = \hat{\sigma}_{yz} \]
\[ s(iCase,6,\ldots) = \hat{\sigma}_{zx} \]

For receivers located on interfaces, the stresses in the underlying element are calculated.

Description

\[ [u,s] = \text{DISK3D}_\text{REC}(h, Cs, Cp, Ds, Dp, rho, R, zs, p, x, y, z, omega) \]
computes the 3D displacements and stresses for disk loads in a layered soil using the direct stiffness method. The displacements and stresses are returned in Cartesian coordinates. Six cases are considered, corresponding to a unit force in the \( x \), \( y \), and \( z \)-direction and a unit moment around the \( x \), \( y \), and \( z \)-axis. The tractions on a horizontal disk with radius \( R \) are defined for each case as:

- **Case 1** — Horizontal load in \( x \)-direction: \( \hat{p}_x = 1/(\pi R^2) \), \( r < R \).
- **Case 2** — Horizontal load in \( y \)-direction: \( \hat{p}_y = 1/(\pi R^2) \), \( r < R \).
- **Case 3** — Vertical load: \( \hat{p}_z = 1/(\pi R^2) \), \( r < R \).
- **Case 4** — Bending moment around \( x \)-axis: \( \hat{p}_z = 2/(\pi R^5) y \), \( r < R \).
- **Case 5** — Bending moment around \( y \)-axis: \( \hat{p}_z = 2/(\pi R^5) x \), \( r < R \).
- **Case 6** — Torsional moment around \( z \)-axis: \( \hat{p}_\theta = 2/(\pi R^4) r/R \), \( r < R \).

The number of elements \( n\text{Elt} \) is equal to the maximum of the lengths of \( h \), \( Cs \), \( Cp \), \( Ds \), \( Dp \), and \( rho \). If any of these parameters is defined as a scalar, an identical value is used for all elements.
disk3d_recz

Purpose
3D vertical disk load solution in Cartesian coordinates.

Syntax
[u,s] = DISK3D_RECZ(h,Cs,Cp,Ds,Dp,rho,R,zs,p,x,y,z,omega)

Input arguments
h   Layer thickness, INF for a halfspace (nElt x 1) or (1 x 1).
   If h(end) ≠ INF, the bottom interface is clamped.
   If necessary, extra layers are defined so that the source locations zs
   coincide with interfaces between elements.
Cs  Shear wave velocity (nElt x 1) or (1 x 1).
Cp  Dilatational wave velocity (nElt x 1) or (1 x 1).
Ds  Shear damping ratio (nElt x 1) or (1 x 1).
Dp  Dilatational damping ratio (nElt x 1) or (1 x 1).
rho Density (nElt x 1) or (1 x 1).
R   Radius of the disk load (1 x 1).
zs  Source locations (vertical coordinate) (nzSrc x 1).
p   Slowness, logarithmically sampled (nWave x 1).
   If omega ≠ 0, the wavenumber sampling is given by k = omega x p.
   If omega = 0, the wavenumber sampling is given by k = p.
x   Receiver locations (x-coordinate) (nxRec x 1).
y   Receiver locations (y-coordinate) (nyRec x 1).
z   Receiver locations (z-coordinate) (nzRec x 1).
omega Circular frequency (nFreq x 1).
Output arguments

\[ u \quad \text{Displacements } (3 \times \text{nzSrc} \times \text{nrRec} \times \text{ntRec} \times \text{nzRec} \times \text{nFreq}). \]
\[ u(1,\ldots) = \hat{u}_x \]
\[ u(2,\ldots) = \hat{u}_y \]
\[ u(3,\ldots) = \hat{u}_z \]

\[ s \quad \text{Stresses } (6 \times \text{nzSrc} \times \text{nrRec} \times \text{ntRec} \times \text{nzRec} \times \text{nFreq}). \]
\[ s(1,\ldots) = \hat{\sigma}_{xx} \]
\[ s(2,\ldots) = \hat{\sigma}_{yy} \]
\[ s(3,\ldots) = \hat{\sigma}_{zz} \]
\[ s(4,\ldots) = \hat{\sigma}_{xy} \]
\[ s(5,\ldots) = \hat{\sigma}_{yz} \]
\[ s(6,\ldots) = \hat{\sigma}_{zx} \]

For receivers located on interfaces, the stresses in the underlying element are calculated.

Description

\[ [u,s] = \text{DISK3DRECZ}(h, Cs, Cp, Ds, Dp, rho, R, zs, p, x, y, z, omega) \]
computes the 3D displacements and stresses for a vertical disk load in a layered soil using the direct stiffness method. The displacements and stresses are returned in Cartesian coordinates.

The number of elements nElt is equal to the maximum of the lengths of \( h, Cs, Cp, Ds, Dp, \) and \( rho \). If any of these parameters is defined as a scalar, an identical value is used for all elements.
eig_dsmpsv

Purpose
Rayleigh waves in a layered halfspace (direct stiffness method).

Syntax

\[
[f, C, A] = \text{EIG_DSMPSV}(h, Cs, Cp, Ds, Dp, rho, fMax, nMode)
\]

\[
[f, C, A, Ux, Uz, Tx, Tz] = \text{EIG_DSMPSV}(h, Cs, Cp, Ds, Dp, rho, fMax, nMode, fMode, zMode)
\]

\[
[... ] = \text{EIG_DSMPSV}(..., \text{ParamName}, \text{ParamValue})
\]

Input arguments

- \(h\) Element thickness, last term must be \(\text{INF} (n\text{Elt} \times 1)\).
- \(Cs\) Shear wave velocity \((n\text{Elt} \times 1)\) or \((1 \times 1)\).
- \(Cp\) Dilatational wave velocity \((n\text{Elt} \times 1)\) or \((1 \times 1)\).
- \(Ds\) Shear damping ratio \((n\text{Elt} \times 1)\) or \((1 \times 1)\).
- \(Dp\) Dilatational damping ratio \((n\text{Elt} \times 1)\) or \((1 \times 1)\).
- \(rho\) Density \((n\text{Elt} \times 1)\) or \((1 \times 1)\).
- \(fMax\) Maximum frequency.
- \(nMode\) Number of modes. Default: \(\text{INF}\) (all modes up to frequency \(fMax\)).
- \(fMode\) Frequencies where mode shapes are computed \((nf \times 1)\). Default: \([]\).
- \(zMode\) Depths where mode shapes are computed \((nz \times 1)\). Default: \([]\).

Output arguments

- \(f\) Frequency sampling for \(C\) and \(A\) \((M \times nMode)\).
- \(C\) Phase velocity corresponding to \(f\) \((M \times nMode)\).
- \(A\) Attenuation coefficient corresponding to \(f\) \((M \times nMode)\).
- \(Ux\) Horizontal modal displacements \((nz \times nf \times nMode)\).
- \(Uz\) Vertical modal displacements \((nz \times nf \times nMode)\).
- \(Tx\) Horizontal modal tractions \((nz \times nf \times nMode)\).
- \(Tz\) Vertical modal tractions \((nz \times nf \times nMode)\).

Description

\[
[f, C, A, Ux, Uz, Tx, Tz] = \text{EIG_DSMPSV}(h, Cs, Cp, Ds, Dp, rho, fMax, nMode, fMode, zMode)
\]

computes the Rayleigh waves in a layered halfspace with the direct stiffness method. The dispersion curves \(C(f)\), the attenuation curves \(A(f)\), the modal displacements \((Ux, Uz)\), and the modal tractions \((Tx, Tz)\) are calculated up to a frequency \(fMax\) for \(nMode\) modes.

The number of elements \(n\text{Elt}\) is equal to the maximum of the lengths of \(h\), \(Cs\), \(Cp\), \(Ds\), \(Dp\), and \(rho\). If any of these parameters is defined as a scalar, an identical
value is used for all elements. The frequencies $f_{\text{Max}}$, $f_{\text{Mode}}$, and $f$ are in cycles per unit time, not in radians per unit time.

$[\ldots] = \text{EIG_DSMPSV}(\ldots, \text{ParamName}, \text{ParamValue})$ changes the default values of the parameters controlling the algorithm. The dispersion curves are calculated by means of a search algorithm that minimizes the determinant of the stiffness matrix in terms of the complex dimensionless wavenumber $k_d = k \min(Cs)/\omega$. The dispersion curves $C$ and attenuation curves $A$ are obtained afterwards as $C = \omega / \text{Re}(k)$ and $A = -\text{Im}(k)$.

The search algorithm consists of two parts. In the first part, the starting point is identified for each dispersion curve. For the first mode, this point is located at a frequency $f_{\text{Min}}$ (close but not equal to zero) and at the complex wavenumber $k_d$ that minimizes the objective function. For the higher modes, the starting point is located at a wavenumber $k_d$ with a real part $\text{Re}(k_d) = \min(Cs)/Cs(\text{end})$. The imaginary part $\text{Im}(k_d)$ and the frequency $f$ corresponding to the starting point are found by minimization of the objective function. The starting point is first determined for the undamped medium. Next, damping is gradually introduced and the resulting variation of the starting point is traced. During this process, the Modal Assurance Criterion (MAC) value of the mode shapes corresponding to two successive damping values is continuously monitored. The rate to which damping is introduced is adjusted as a function of the MAC value in order to avoid that the algorithm jumps to a different mode. The minimization procedure and the frequency step adjustments are controlled by the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_{\text{Min}}$</td>
<td>$10^{-4} \times f_{\text{Max}}$</td>
<td>Starting frequency for the first mode.</td>
</tr>
<tr>
<td>$f_{\text{Res}}$</td>
<td>$10^{-2} \times df_{\text{Cutoff}}$</td>
<td>Resolution in terms of frequency.</td>
</tr>
<tr>
<td>$k_{\text{DiRes}}$</td>
<td>$10^{-6}$</td>
<td>Resolution in terms of $\text{Im}(k_d)$.</td>
</tr>
<tr>
<td>$f_{\text{Tol}}$</td>
<td>$10^{-4} \times df_{\text{Cutoff}}$</td>
<td>Accuracy in terms of frequency.</td>
</tr>
<tr>
<td>$k_{\text{DiTol}}$</td>
<td>$10^{-10}$</td>
<td>Accuracy in terms of $\text{Im}(k_d)$.</td>
</tr>
<tr>
<td>$\text{macMin}$</td>
<td>0.99</td>
<td>Minimum MAC value.</td>
</tr>
<tr>
<td>$\text{macMax}$</td>
<td>0.999</td>
<td>MAC value required to increase the rate to which damping is introduced.</td>
</tr>
</tbody>
</table>

Herein, $df_{\text{Cutoff}}$ is an estimation of the distance between the cut-off frequencies of the soil if the stiffness of the underlying halfspace tends to infinity.

The second part of the algorithm traces the dispersion curve, starting from the point identified in the first part. To this end, the objective function is minimized at a sequence of frequencies in terms of the real part $\text{Re}(k_d)$ and the imaginary part $\text{Im}(k_d)$ of the dimensionless wavenumber. The frequency step is adjusted as a function of the MAC value of the mode shapes corresponding to two successive points to avoid that the algorithm jumps to a different mode. The minimization procedure and the frequency step adjustments are controlled by the following parameters:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fStepIni</td>
<td>$10^{-3} \times \text{dfCutoff}$</td>
<td>Initial frequency step.</td>
</tr>
<tr>
<td>fStepMin</td>
<td>$10^{-5} \times \text{dfCutoff}$</td>
<td>Minimal frequency step.</td>
</tr>
<tr>
<td>fStepMax</td>
<td>INF</td>
<td>Maximal frequency step.</td>
</tr>
<tr>
<td>macMin</td>
<td>0.99</td>
<td>Minimum MAC value.</td>
</tr>
<tr>
<td>macMax</td>
<td>0.999</td>
<td>MAC value required to enlarge freq. step.</td>
</tr>
<tr>
<td>kdrRes</td>
<td>$10^{-4}$</td>
<td>Resolution in terms of Re($kd$).</td>
</tr>
<tr>
<td>kdiRes</td>
<td>$10^{-6}$</td>
<td>Resolution in terms of Im($kd$).</td>
</tr>
<tr>
<td>kdrTol</td>
<td>$10^{-8}$</td>
<td>Accuracy in terms of Re($kd$).</td>
</tr>
<tr>
<td>kdiTol</td>
<td>$10^{-10}$</td>
<td>Accuracy in terms of Im($kd$).</td>
</tr>
</tbody>
</table>

The algorithm is not always capable to identify all dispersion curves correctly. Controlling the MAC value improves the robustness, but due to the constraint of the minimal frequency step, it may be impossible to comply with the minimal MAC value. If the MAC value is too low, a warning message is shown. The number of warning messages may be large and can be limited by means of the following parameter:

| MaxWarn | INF | Maximum number of warning messages. |

**Examples**

Example 8.1: Surface waves in a layered halfspace (p. 91).
Example 8.3: Dominant surface wave in a layered halfspace (p. 98).
eig_dsmsh

Purpose
Love waves in a layered halfspace (direct stiffness method).

Syntax
\[ [f,C,A] = \text{EIG\_DSMSH}(h,Cs,Ds,rho,fMax,nMode) \]
\[ [f,C,A,Uy,Ty] = \text{EIG\_DSMSH}(h,Cs,Ds,rho,fMax,nMode,fMode,zMode) \]
\[ [...] = \text{EIG\_DSMSH}(...,\text{ParamName},\text{ParamValue}) \]

Input arguments
- \( h \) Element thickness, last term must be INF (\( nElt \times 1 \)).
- \( Cs \) Shear wave velocity (\( nElt \times 1 \)) or (1 \( \times 1 \)).
- \( Ds \) Shear damping ratio (\( nElt \times 1 \)) or (1 \( \times 1 \)).
- \( rho \) Density (\( nElt \times 1 \)) or (1 \( \times 1 \)).
- \( fMax \) Maximum frequency.
- \( nMode \) Number of modes. Default: INF (all modes up to frequency \( fMax \)).
- \( fMode \) Frequencies where mode shapes are computed (\( nf \times 1 \)). Default: [].
- \( zMode \) Depths where mode shapes are computed (\( nz \times 1 \)). Default: [].

Output arguments
- \( f \) Frequency sampling for \( C \) and \( A \) (\( M \times nMode \)).
- \( C \) Phase velocity corresponding to \( f \) (\( M \times nMode \)).
- \( A \) Attenuation coefficient corresponding to \( f \) (\( M \times nMode \)).
- \( Uy \) Modal displacements (\( nz \times nf \times nMode \)).
- \( Ty \) Modal tractions (\( nz \times nf \times nMode \)).

Description
\[ [f,C,A,Uy,Ty] = \text{EIG\_DSMSH}(h,Cs,Ds,rho,fMax,nMode,fMode,zMode) \]
computes the Love waves in a layered halfspace with the direct stiffness method. The dispersion curves \( C(f) \), the attenuation curves \( A(f) \), the modal displacements \( Uy \), and the modal tractions \( Ty \) are calculated up to a frequency \( fMax \) for \( nMode \) modes.

The number of elements \( nElt \) is equal to the maximum of the lengths of \( h \), \( Cs \), \( Ds \), and \( rho \). If any of these parameters is defined as a scalar, an identical value is used for all elements. The frequencies \( fMax \), \( fMode \), and \( f \) are in cycles per unit time, not in radians per unit time.
\[ \text{DSMSH}(\ldots, \text{ParamName}, \text{ParamValue}) \] changes the default values of the parameters controlling the algorithm. The dispersion curves are calculated by means of a search algorithm that minimizes the determinant of the stiffness matrix in terms of the complex dimensionless wavenumber \( kd = \frac{k_{\text{min}}(Cs)}{\omega} \).

The dispersion curves \( C \) and attenuation curves \( A \) are obtained afterwards as 
\[
C = \frac{\omega}{\text{Re}(k)} \quad \text{and} \quad A = -\text{Im}(k).
\]

The search algorithm consists of two parts. In the first part, the starting point is identified for each dispersion curve. For the first mode, this point is located at a frequency \( f_{\text{Min}} \) (close but not equal to zero) and at the complex wavenumber \( kd \) that minimizes the objective function. For the higher modes, the starting point is located at a wavenumber \( kd \) with a real part \( \text{Re}(kd) = \min(Cs)/Cs(\text{end}) \). The imaginary part \( \text{Im}(kd) \) and the frequency \( f \) corresponding to the starting point are found by minimization of the objective function. The starting point is first determined for the undamped medium. Next, damping is gradually introduced and the resulting variation of the starting point is traced. During this process, the Modal Assurance Criterion (MAC) value of the mode shapes corresponding to two successive damping values is continuously monitored. The rate to which damping is introduced is adjusted as a function of the MAC value in order to avoid that the algorithm jumps to a different mode. The minimization procedure and the frequency step adjustments are controlled by the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_{\text{Min}} )</td>
<td>( 10^{-4} \times f_{\text{Max}} )</td>
<td>Starting frequency for the first mode.</td>
</tr>
<tr>
<td>( f_{\text{Res}} )</td>
<td>( 10^{-2} \times \text{dfCutoff} )</td>
<td>Resolution in terms of frequency.</td>
</tr>
<tr>
<td>( k_{\text{diRes}} )</td>
<td>( 10^{-6} )</td>
<td>Resolution in terms of ( \text{Im}(kd) ).</td>
</tr>
<tr>
<td>( f_{\text{Tol}} )</td>
<td>( 10^{-4} \times \text{dfCutoff} )</td>
<td>Accuracy in terms of frequency.</td>
</tr>
<tr>
<td>( k_{\text{diTol}} )</td>
<td>( 10^{-10} )</td>
<td>Accuracy in terms of ( \text{Im}(kd) ).</td>
</tr>
<tr>
<td>macMin</td>
<td>0.99</td>
<td>Minimum MAC value.</td>
</tr>
<tr>
<td>macMax</td>
<td>0.999</td>
<td>MAC value required to increase the rate to which damping is introduced.</td>
</tr>
</tbody>
</table>

Herein, \( \text{dfCutoff} \) is an estimation of the distance between the cut-off frequencies of the soil if the stiffness of the underlying halfspace tends to infinity.

The second part of the algorithm traces the dispersion curve, starting from the point identified in the first part. To this end, the objective function is minimized at a sequence of frequencies in terms of the real part \( \text{Re}(kd) \) and the imaginary part \( \text{Im}(kd) \) of the dimensionless wavenumber. The frequency step is adjusted as a function of the MAC value of the mode shapes corresponding to two successive points to avoid that the algorithm jumps to a different mode. The minimization procedure and the frequency step adjustments are controlled by the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
</table>
The algorithm is not always capable to identify all dispersion curves correctly. Controlling the MAC value improves the robustness, but due to the constraint of the minimal frequency step, it may be impossible to comply with the minimal MAC value. If the MAC value is too low, a warning message is shown. The number of warning messages may be large and can be limited by means of the following parameter:

MaxWarn    INF    Maximum number of warning messages.
eig_tlmpsv

Purpose
Stiffness matrix eigen decomposition (thin layer method, P-SV-waves).

Syntax
[k,phi] = EIG_TLMPSV(A,B,G,M,omega)

Input arguments
A,B,G,M  System matrices obtained with the thin layer method (nDOF x nDOF).
omega  Circular frequency (nFreq x 1).

Output arguments
k  Wavenumbers corresponding to the surface wave modes travelling in positive x-direction (nDOF x nFreq). The modes are sorted on attenuation coefficient in ascending order. Unattenuated modes are sorted on phase velocity in ascending order.
phi  Eigenvectors (nDOF x nDOF x nFreq).

Description
[k,phi] = EIG_TLMPSV(A,B,G,M,omega) calculates the eigenvalues and the eigenvectors of a soil characterized by the matrices A,B,G,M obtained with the thin layer method. Only P-SV-waves are accounted for. See references [11] and [12].

eig_tlms

**Purpose**
Stiffness matrix eigen decomposition (thin layer method, SH-waves).

**Syntax**

```
[k,phi] = EIG_TLMSH(A,G,M,omega)
```

**Input arguments**

- **A, G, M** System matrices obtained with the thin layer method ($nDOF \times nDOF$).
- **omega** Circular frequency ($nFreq \times 1$).

**Output arguments**

- **k** Wavenumbers corresponding to the surface wave modes travelling in positive $x$-direction ($nDOF \times nFreq$). The modes are sorted on attenuation coefficient in ascending order. Unattenuated modes are sorted on phase velocity in ascending order.
- **phi** Eigenvectors ($nDOF \times nDOF \times nFreq$).

**Description**

$[k, phi] = EIG_TLMSH(A, G, M, omega)$ calculates the eigenvalues and the eigenvectors of a soil characterized by the matrices $A, G, M$ obtained with the thin layer method. Only SH-waves are accounted for. See references [11] and [12].

**Example**

Example 8.2: Surface waves in a homogeneous layer on bedrock (p. 95).
**eqlinamp**

**Purpose**
Equivalent linear 1-D site response analysis.

**Syntax**
\[ [a,C,D,g] = \text{EQLINAMP}(h,C0,\rho,gd,Rd,Dd,ai,F,z) \]

**Input arguments**
- \( h \): Layer thickness, \( \text{INF} \) for a halfspace \((n_{\text{Elt}} \times 1)\) or \((1 \times 1)\).
- \( C0 \): Small-strain wave velocity \((n_{\text{Elt}} \times 1)\) or \((1 \times 1)\).
- \( \rho \): Density \((n_{\text{Elt}} \times 1)\) or \((1 \times 1)\).
- \( gd \): Cyclic strain amplitudes for which the modulus reduction and the damping ratio have been measured \((n_{\text{Strain}} \times 1)\).
- \( Rd \): Modulus reduction (i.e., the ratio \( \mu/\mu_0 \) of the secant shear modulus \( \mu \) and the small-strain modulus \( \mu_0 \)) measured at the cyclic strain amplitudes \( gd \) \((n_{\text{Elt}} \times n_{\text{Strain}})\) or \((1 \times n_{\text{Strain}})\).
- \( Dd \): Damping ratio measured at the cyclic strain amplitudes \( gd \) \((n_{\text{Elt}} \times n_{\text{Strain}})\) or \((1 \times n_{\text{Strain}})\).
- \( ai \): Time history of the incident wave acceleration \((N \times 1)\).
- \( F \): Sampling frequency for \( ai \) \((1 \times 1)\).
- \( z \): Receiver depth \((n_{\text{Rec}} \times 1)\). Default: 0.

**Output arguments**
- \( a \): Time history of the acceleration at the receivers \((N \times n_{\text{Rec}})\).
- \( C \): Equivalent wave velocity \((n_{\text{Elt}} \times 1)\).
- \( D \): Equivalent damping ratio \((n_{\text{Elt}} \times 1)\).
- \( g \): Effective strain \((n_{\text{Elt}} \times 1)\).

**Description**
\[ [a,C,D,g] = \text{EQLINAMP}(h,C0,\rho,gd,Rd,Dd,ai,F,z) \] performs a one-dimensional site amplification analysis using an equivalent linear material model.

The number of elements \( n_{\text{Elt}} \) is equal to the maximum of the lengths of \( h, C0, \) and \( \rho \). If any of these parameters is defined as a scalar, an identical value is used for all elements.

\[ [...] = \text{EQLINAMP}(..., \text{ParamName}, \text{ParamValue}) \] sets the value of the specified parameters. The following parameters can be specified:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StrainRatio</td>
<td>0.65</td>
<td>Strain ratio used to compute the effective (cyclic) strain amplitude from the peak strain over the time history of the earthquake.</td>
</tr>
<tr>
<td>Tol</td>
<td>$10^{-3}$</td>
<td>Relative tolerance to terminate updating of the equivalent wave velocity $C$ and damping ratio $D$.</td>
</tr>
</tbody>
</table>

**Example**

Example 7.3: Response of a layered soil due to an earthquake - equivalent linear analysis (p. 78).
**eqlinampf**

**Purpose**
Frequency dependent equivalent linear 1-D site response analysis.

**Syntax**

\[
\text{[a,C,D,G,Gs,f] = EQLINAMPF(h,C0,rho,gdeg,ddeg,ai,F,z)}
\]

**Input arguments**

- **h**: Layer thickness, INF for a halfspace \((nElt \times 1)\) or \((1 \times 1)\).
- **C0**: Small-strain wave velocity \((nElt \times 1)\) or \((1 \times 1)\).
- **rho**: Density \((nElt \times 1)\) or \((1 \times 1)\).
- **gd**: Cyclic strain amplitudes for which the modulus reduction and the damping ratio have been measured \((nStrain \times 1)\).
- **Rd**: Modulus reduction (i.e. the ratio \(\mu/\mu_0\) of the secant shear modulus \(\mu\) and the small-strain modulus \(\mu_0\)) measured at the cyclic strain amplitudes \(gd\) \((nElt \times nStrain)\) or \((1 \times nStrain)\).
- **Dd**: Damping ratio measured at the cyclic strain amplitudes \(gd\) \((nElt \times nStrain)\) or \((1 \times nStrain)\).
- **ai**: Time history of the incident wave acceleration \((N \times 1)\).
- **F**: Sampling frequency for \(ai\) \((1 \times 1)\).
- **z**: Receiver depth \((nRec \times 1)\). Default: 0.

**Output arguments**

- **a**: Time history of the acceleration at the receivers \((N \times nRec)\).
- **C**: Equivalent wave velocity \((nFreq \times nElt)\).
- **D**: Equivalent damping ratio \((nFreq \times nElt)\).
- **G**: Scaled strain amplitude spectrum \((nFreq \times nElt)\).
- **Gs**: Smoothed strain amplitude spectrum \((nFreq \times nElt)\).
- **f**: Frequency sampling for C, D, G, and Gs \((nFreq \times 1)\).

**Description**

\([a,C,D,G,Gs,f] = \text{EQLINAMPF}(h,C0,rho,gdeg,ddeg,ai,F,z)\) performs a one-dimensional site amplification analysis using an equivalent linear material model with frequency dependent moduli and damping ratios, according to Kausel and Assimaki [10].

The number of elements \(nElt\) is equal to the maximum of the lengths of \(h\), \(C0\), and \(rho\). If any of these parameters is defined as a scalar, an identical value is used for all elements.
EQLINAMPF(...,ParamName,ParamValue) sets the value of the specified parameters. The following parameters can be specified:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fMax</td>
<td>F/2</td>
<td>Upper bound of the frequency range where the smoothed strain amplitude spectrum is computed. Above the frequency $f_{\text{Max}}$, small-strain conditions are assumed to apply and a linear material model is used. The frequency $f_{\text{Max}}$ should be smaller than (or equal to) the cut-off frequency of the low-pass filter used to acquire the acceleration $a_i$.</td>
</tr>
<tr>
<td>StrainRatio</td>
<td>1</td>
<td>Strain ratio used to compute the effective (cyclic) strain amplitude from the peak strain over the time history of the earthquake.</td>
</tr>
<tr>
<td>Tol</td>
<td>$10^{-3}$</td>
<td>Relative tolerance to terminate updating of the equivalent wave velocity $C$ and damping ratio $D$.</td>
</tr>
</tbody>
</table>

**Example**

Example 7.4: Response of a layered soil due to an earthquake - frequency dependent equivalent linear analysis (p. 83).
**fht**

**Purpose**
Fast Hankel transform.

**Syntax**
\[ g = \text{FHT}(n,f,k,r,a,dim,accu) \]

**Input arguments**
- \( n \): Order of the Hankel transformation.
- \( f \): Function to transform sampled at \( k \) \((M1 \times M2 \times \cdots \times Nk \times \cdots \times Mn)\).
- \( k \): Log spaced points, must be sorted in ascending order \((Nk \times 1)\).
- \( r \): Sampling in the transformed domain \((Nr \times 1)\).
- \( a \): Bias exponent \(-n < a < 3/2\). Corresponds to \(1 - \mu\) in reference [27]. Default: 1.
- \( \text{dim} \): Operate along dimension \( \text{dim} \). Default: first non-singleton dimension.
- \( \text{accu} \): Kernel accuracy. Determines the amount of padding. Default: \(10^{-6}\).

**Output arguments**
- \( g \): Transformed function sampled at \( r \) \((M1 \times M2 \times \cdots \times Nr \times \cdots \times Mn)\).

**Description**
\[ g(r) = \int_0^\infty f(k)J_n(kr)k \, dk \]
where \(J_n\) is the n-th order Bessel function of the first kind.
fsgreen2d_inplane

Purpose
In-plane 2D fullspace Green’s functions.

Syntax
[ug, sg] = FSGREEN2D_INPLANE(Cs, Cp, Ds, Dp, rho, x, z, omega)

Input arguments
Cs  Shear wave velocity.
Cp  Dilatational wave velocity.
Ds  Shear damping ratio.
Dp  Dilatational damping ratio.
rho Density.
x  Receiver locations (horizontal coordinate) (nxRec × 1).
z  Receiver locations (vertical coordinate) (nzRec × 1).
omega Circular frequency (nFreq × 1).

Output arguments
ug  Green’s displacements (2 × 2 × nxRec × nzRec × nFreq).
ug(1,1,...) = \( \hat{u}_G^{xx} \)  ug(2,1,...) = \( \hat{u}_G^{zx} \)
ug(1,2,...) = \( \hat{u}_G^{xz} \)  ug(2,2,...) = \( \hat{u}_G^{zz} \)
sg  Green’s stresses (2 × 3 × nxRec × nzRec × nFreq).
sg(1,1,...) = \( \hat{\sigma}_G^{xxx} \)  sg(2,1,...) = \( \hat{\sigma}_G^{zxx} \)
sg(1,2,...) = \( \hat{\sigma}_G^{xzx} \)  sg(2,2,...) = \( \hat{\sigma}_G^{zzx} \)
sg(1,3,...) = \( \hat{\sigma}_G^{xzx} \)  sg(2,3,...) = \( \hat{\sigma}_G^{zzx} \)

Description
[ug, sg] = FSGREEN2D_INPLANE(Cs, Cp, Ds, Dp, rho, x, z, omega) computes the in-
plane 2D Green’s functions of a homogeneous fullspace using analytical expres-
sions. The Green’s functions are returned in Cartesian coordinates.
fsgreen2d_outofplane

Purpose
Out-of-plane 2D fullspace Green’s functions.

Syntax
[ug, sg] = FSGREEN2D_OUTOFPLANE(Cs, Ds, rho, x, z, omega)

Input arguments

- Cs  Shear wave velocity.
- Ds  Shear damping ratio.
- rho Density.
- x   Receiver locations (horizontal coordinate) (nxRec × 1).
- z   Receiver locations (vertical coordinate) (nzRec × 1).
- omega Circular frequency (nFreq × 1).

Output arguments

- ug   Green’s displacements (1 × 1 × nxRec × nzRec × nFreq).
  ug(1,1,...) = \hat{u}_G^{yy}
- sg   Green’s stresses (1 × 2 × nxRec × nzRec × nFreq).
  sg(1,1,...) = \hat{\sigma}_G^{yy}
  sg(1,2,...) = \hat{\sigma}_G^{yyz}

Description

[ug, sg] = FSGREEN2D_OUTOFPLANE(Cs, Ds, rho, x, z, omega) computes the out-of-plane 2D Green’s functions of a homogeneous fullspace using analytical expressions. The Green’s functions are returned in Cartesian coordinates.
fsgreen3d_cyl

Purpose
3D fullspace Green’s functions in cylindrical coordinates.

Syntax
[ug,sg] = FSGREEN3D_CYL(Cs,Cp,Ds,Dp,rho,r,theta,z,omega)

Input arguments
Cs Shear wave velocity.
Cp Dilatational wave velocity.
Ds Shear damping ratio.
Dp Dilatational damping ratio.
rho Density.
r Receiver locations (radial coordinate) (nrRec × 1).
theta Receiver locations (circumferential coordinate) (ntRec × 1).
z Receiver locations (vertical coordinate) (nzRec × 1).

Output arguments
ug Green’s displacements (3 × 3 × nrRec × ntRec × nzRec × nFreq).
ug(1,1,...) = ̂u^{G}_r
ug(2,1,...) = ̂u^{G}_θ
ug(3,1,...) = ̂u^{G}_z
ug(1,2,...) = ̂u^{G}_{rr}
ug(2,2,...) = ̂u^{G}_{θθ}
ug(3,2,...) = ̂u^{G}_{zz}
ug(1,3,...) = ̂u^{G}_{rθ}
ug(2,3,...) = ̂u^{G}_{rz}
ug(3,3,...) = ̂u^{G}_{zz}

sg Green’s stresses (3 × 6 × nrRec × ntRec × nzRec × nFreq).
sg(1,1,...) = ̂σ^{G}_{rr}
sg(2,1,...) = ̂σ^{G}_{θθ}
sg(3,1,...) = ̂σ^{G}_{zz}
sg(1,2,...) = ̂σ^{G}_{rθ}
sg(2,2,...) = ̂σ^{G}_{rθ}
sg(3,2,...) = ̂σ^{G}_{zz}
sg(1,3,...) = ̂σ^{G}_{rθ}
sg(2,3,...) = ̂σ^{G}_{rθ}
sg(3,3,...) = ̂σ^{G}_{zz}

Description
[ug,sg] = FSGREEN3D_CYL(Cs,Cp,Ds,Dp,rho,r,theta,z,omega) computes the
3D Green’s displacements and stresses for a homogeneous fullspace using analytical
expressions. The Green’s functions are returned in cylindrical coordinates.
**fsgreenf**

**Purpose**

Fullspace Green’s functions in the $(x,k_y)$-domain.

**Syntax**

```matlab
[ug, sg] = FSGREENF(Cs, Cp, Ds, Dp, rho, x, py, z, omega)
```

**Input arguments**

- **Cs**  
  Shear wave velocity.
- **Cp**  
  Dilatational wave velocity.
- **Ds**  
  Shear damping ratio.
- **Dp**  
  Dilatational damping ratio.
- **rho**  
  Density.
- **x**  
  Receiver locations ($x$-coordinate) ($nxRec \times 1$).
- **py**  
  Slowness ($y$-coordinate) ($nyWave \times 1$).
  
  If $omega \neq 0$, the wavenumber sampling is given by $ky = omega \times py$.
  If $omega = 0$, the wavenumber sampling is given by $ky = py$.
- **z**  
  Receiver locations ($z$-coordinate) ($nzRec \times 1$).
- **omega**  
  Circular frequency ($nFreq \times 1$).

**Output arguments**

- **ug**  
  Green’s displacements ($3 \times 3 \times nxRec \times nyWave \times nzRec \times nFreq$).
  
  - $ug(1,1,\ldots) = \tilde{u}_G^{xx}$
  - $ug(2,1,\ldots) = \tilde{u}_G^{yy}$
  - $ug(3,1,\ldots) = \tilde{u}_G^{zz}$
  - $ug(1,2,\ldots) = \tilde{u}_G^{xy}$
  - $ug(2,2,\ldots) = \tilde{u}_G^{yy}$
  - $ug(3,2,\ldots) = \tilde{u}_G^{yz}$
  - $ug(1,3,\ldots) = \tilde{u}_G^{xz}$
  - $ug(2,3,\ldots) = \tilde{u}_G^{yz}$
  - $ug(3,3,\ldots) = \tilde{u}_G^{zz}$

- **sg**  
  Green’s stresses ($3 \times 6 \times nxRec \times nyWave \times nzRec \times nFreq$).
  
  - $sg(1,1,\ldots) = \tilde{\sigma}_G^{xx}$
  - $sg(2,1,\ldots) = \tilde{\sigma}_G^{yy}$
  - $sg(3,1,\ldots) = \tilde{\sigma}_G^{zz}$
  - $sg(1,2,\ldots) = \tilde{\sigma}_G^{xy}$
  - $sg(2,2,\ldots) = \tilde{\sigma}_G^{yy}$
  - $sg(3,2,\ldots) = \tilde{\sigma}_G^{yz}$
  - $sg(1,3,\ldots) = \tilde{\sigma}_G^{xz}$
  - $sg(2,3,\ldots) = \tilde{\sigma}_G^{yz}$
  - $sg(3,3,\ldots) = \tilde{\sigma}_G^{zz}$

**Description**

$[ug, sg] = FSGREENF(Cs, Cp, Ds, Dp, rho, x, py, z, omega)$ computes the Green’s functions of a homogeneous fullspace in the $(x,k_y)$-domain using analytical solutions. The Green’s functions are returned on a rectangular grid $(x, py)$.
green2d_inplane

Purpose
In-plane 2D Green’s functions.

Syntax
\[ [u_g, s_g] = \text{GREEN2D_INPLANE}(h, C_s, C_p, D_s, D_p, \rho, z_s, p, x, z, \omega) \]

Input arguments
- **h**: Layer thickness, INF for a halfspace \((n\text{Elt} \times 1)\) or \((1 \times 1)\).
- **C_s**: Shear wave velocity \((n\text{Elt} \times 1)\) or \((1 \times 1)\).
- **C_p**: Dilatational wave velocity \((n\text{Elt} \times 1)\) or \((1 \times 1)\).
- **D_s**: Shear damping ratio \((n\text{Elt} \times 1)\) or \((1 \times 1)\).
- **D_p**: Dilatational damping ratio \((n\text{Elt} \times 1)\) or \((1 \times 1)\).
- **\rho**: Density \((n\text{Elt} \times 1)\) or \((1 \times 1)\).
- **z_s**: Source locations (vertical coordinate) \((n\text{zSrc} \times 1)\).
- **p**: Slowness, logarithmically sampled \((n\text{wave} \times 1)\).
- **x**: Receiver locations (horizontal coordinate) \((n\text{xRec} \times 1)\).
- **z**: Receiver locations (vertical coordinate) \((n\text{zRec} \times 1)\).
- **\omega**: Circular frequency \((n\text{Freq} \times 1)\).

Output arguments
- **u_g**: Green’s displacements \((2 \times 2 \times n\text{zSrc} \times n\text{xRec} \times n\text{zRec} \times n\text{Freq})\).
  - \(u_{11} \ldots = \hat{u}_G^{\alpha \alpha} \)
  - \(u_{12} \ldots = \hat{u}_G^{\alpha \beta} \)
  - \(u_{21} \ldots = \hat{u}_G^{\beta \alpha} \)
  - \(u_{22} \ldots = \hat{u}_G^{\beta \beta} \)
- **s_g**: Green’s stresses \((2 \times 3 \times n\text{zSrc} \times n\text{xRec} \times n\text{zRec} \times n\text{Freq})\).
  - \(s_{11} \ldots = \hat{\sigma}_G^{\alpha \alpha \alpha} \)
  - \(s_{12} \ldots = \hat{\sigma}_G^{\alpha \alpha \beta} \)
  - \(s_{13} \ldots = \hat{\sigma}_G^{\alpha \beta \alpha} \)
  - \(s_{21} \ldots = \hat{\sigma}_G^{\alpha \beta \beta} \)
  - \(s_{22} \ldots = \hat{\sigma}_G^{\beta \beta \beta} \)
  - \(s_{23} \ldots = \hat{\sigma}_G^{\beta \beta \alpha} \)

For receivers located on interfaces, the stresses in the underlying element are calculated.
**Description**

\[ u_g, u_s \] = GREEN2D_INPLANE(\( h, C_s, C_p, D_s, D_p, \rho, z_s, p, x, z, \omega \)) computes the in-plane 2D Green's functions of a layered soil using the direct stiffness method. The Green's functions are returned in Cartesian coordinates.

The number of elements \( n_{El} \) is equal to the maximum of the lengths of \( h, C_s, C_p, D_s, D_p, \text{ and } \rho \). If any of these parameters is defined as a scalar, an identical value is used for all elements.
**green2d_outofplane**

**Purpose**
Out-of-plane 2D Green’s functions.

**Syntax**

\[
[u_g, s_g] = \text{GREEN2D\_OUTOFPLANE}(h, C_s, D_s, \rho, z_s, p, x, z, \omega)
\]

**Input arguments**

- **h**: Layer thickness, **INF** for a halfspace \((n\text{Elt} \times 1)\) or \((1 \times 1)\).
  - If \(h(\text{end}) \neq \text{INF}\), the bottom interface is clamped.
  - If necessary, extra layers are defined so that the source locations \(z_s\) coincide with interfaces between elements.

- **Cs**: Shear wave velocity \((n\text{Elt} \times 1)\) or \((1 \times 1)\).

- **Ds**: Shear damping ratio \((n\text{Elt} \times 1)\) or \((1 \times 1)\).

- **rho**: Density \((n\text{Elt} \times 1)\) or \((1 \times 1)\).

- **zs**: Source locations (vertical coordinate) \((nz\text{Src} \times 1)\).

- **p**: Slowness, logarithmically sampled \((n\text{Wave} \times 1)\).
  - If \(\omega \neq 0\), the wavenumber sampling is given by \(k = \omega \times p\).
  - If \(\omega = 0\), the wavenumber sampling is given by \(k = p\).

- **x**: Receiver locations (horizontal coordinate) \((nx\text{Rec} \times 1)\).

- **z**: Receiver locations (vertical coordinate) \((nz\text{Rec} \times 1)\).

- **omega**: Circular frequency \((n\text{Freq} \times 1)\).

**Output arguments**

- **u_g**: Green’s displacements \((1 \times 1 \times nz\text{Src} \times nx\text{Rec} \times nz\text{Rec} \times n\text{Freq})\).
  - \(u_g(1,1,...) = \hat{u}_{yy}^G\)

- **s_g**: Green’s stresses \((1 \times 2 \times nz\text{Src} \times nx\text{Rec} \times nz\text{Rec} \times n\text{Freq})\).
  - \(s_g(1,1,...) = \hat{\sigma}_{yy}^G\)
  - \(s_g(1,2,...) = \hat{\sigma}_{yyz}^G\)
  - For receivers located on interfaces, the stresses in the underlying element are calculated.

**Description**

\([u_g, s_g] = \text{GREEN2D\_OUTOFPLANE}(h, C_s, D_s, \rho, z_s, p, x, z, \omega)\) computes the 2D Green’s functions of a layered soil using the direct stiffness method. The Green’s functions are returned in Cartesian coordinates.
The number of elements \( n_{\text{Elt}} \) is equal to the maximum of the lengths of \( h \), \( Cs \), \( Ds \), and \( \rho \). If any of these parameters is defined as a scalar, an identical value is used for all elements.
green2d_z

Purpose

2D Green’s functions $\hat{u}^G_{zj}$ and $\hat{\sigma}^G_{zjk}$.

Syntax

$$\left[u_g, s_g\right] = \text{GREEN2D}_Z(h, C_s, C_p, D_s, D_p, \rho, z_s, p, x, z, \omega)$$

Input arguments

- $h$: Layer thickness, INF for a halfspace ($n_{\text{Elt}} \times 1$) or ($1 \times 1$).
  - If $h(\text{end}) \neq \text{INF}$, the bottom interface is clamped.
  - If necessary, extra layers are defined so that the source locations $z_s$ coincide with interfaces between elements.
- $C_s$: Shear wave velocity ($n_{\text{Elt}} \times 1$) or ($1 \times 1$).
- $C_p$: Dilatational wave velocity ($n_{\text{Elt}} \times 1$) or ($1 \times 1$).
- $D_s$: Shear damping ratio ($n_{\text{Elt}} \times 1$) or ($1 \times 1$).
- $D_p$: Dilatational damping ratio ($n_{\text{Elt}} \times 1$) or ($1 \times 1$).
- $\rho$: Density ($n_{\text{Elt}} \times 1$) or ($1 \times 1$).
- $z_s$: Source locations (vertical coordinate) ($n_{\text{Src}} \times 1$).
- $p$: Slowness, logarithmically sampled ($n_{\text{Wave}} \times 1$).
  - If $\omega \neq 0$, the wavenumber sampling is given by $k = \omega \times p$.
  - If $\omega = 0$, the wavenumber sampling is given by $k = p$.
- $x$: Receiver locations (horizontal coordinate) ($n_{\text{Rec}} \times 1$).
- $z$: Receiver locations (vertical coordinate) ($n_{\text{Rec}} \times 1$).
- $\omega$: Circular frequency ($n_{\text{Freq}} \times 1$).

Output arguments

- $u_g$: Green’s displacements ($2 \times n_{\text{Src}} \times n_{\text{Rec}} \times n_{\text{Rec}} \times n_{\text{Freq}}$).
  - $u_g(1, \ldots) = \hat{u}^G_{zx}$
  - $u_g(2, \ldots) = \hat{u}^G_{zz}$
- $s_g$: Green’s stresses ($3 \times n_{\text{Src}} \times n_{\text{Rec}} \times n_{\text{Rec}} \times n_{\text{Freq}}$).
  - $s_g(1, \ldots) = \hat{\sigma}^G_{zxx}$
  - $s_g(2, \ldots) = \hat{\sigma}^G_{zzz}$
  - $s_g(3, \ldots) = \hat{\sigma}^G_{zzx}$
  - For receivers located on interfaces, the stresses in the underlying element are calculated.
Description

\[ [u_g, s_g] = \text{GREEN2D}_Z(h, C_s, C_p, D_s, D_p, \rho, z_s, p, x, z, \omega) \]
computes the 2D Green’s functions \( u_{Gj} \) and \( \sigma_{Gjk} \) of a layered soil using the direct stiffness method. The Green’s functions are returned in Cartesian coordinates.

The number of elements \( n_{Elt} \) is equal to the maximum of the lengths of \( h, C_s, C_p, D_s, D_p, \) and \( \rho \). If any of these parameters is defined as a scalar, an identical value is used for all elements.
green2d_zz

Purpose

2D Green’s function $\hat{u}_{zz}^G$.

Syntax

$ugzz = \text{GREEN2D\_ZZ}(h,Cs,Cp,Ds,Dp,rho,zs,p,x,z,omega)$

Input arguments

- $h$: Layer thickness, INF for a halfspace ($nElt \times 1$) or $(1 \times 1)$.
  - If $h(\text{end}) \neq \text{INF}$, the bottom interface is clamped.
  - If necessary, extra layers are defined so that the source locations $zs$ coincide with interfaces between elements.
- $Cs$: Shear wave velocity ($nElt \times 1$) or $(1 \times 1)$.
- $Cp$: Dilatational wave velocity ($nElt \times 1$) or $(1 \times 1)$.
- $Ds$: Shear damping ratio ($nElt \times 1$) or $(1 \times 1)$.
- $Dp$: Dilatational damping ratio ($nElt \times 1$) or $(1 \times 1)$.
- $rho$: Density ($nElt \times 1$) or $(1 \times 1)$.
- $zs$: Source locations (vertical coordinate) ($nzSrc \times 1$).
- $p$: Slowness, logarithmically sampled ($nWave \times 1$).
  - If $omega \neq 0$, the wavenumber sampling is given by $k = omega \times p$.
  - If $omega = 0$, the wavenumber sampling is given by $k = p$.
- $x$: Receiver locations (horizontal coordinate) ($nxRec \times 1$).
- $z$: Receiver locations (vertical coordinate) ($nzRec \times 1$).
- $omega$: Circular frequency ($nFreq \times 1$).

Output arguments

- $ugzz$: Green’s displacements ($nzSrc \times nxRec \times nzRec \times nFreq$).

Description

$ugzz = \text{GREEN2D\_ZZ}(h,Cs,Cp,Ds,Dp,rho,zs,p,x,z,omega)$ computes the 2D Green’s function $u_{zz}^G$ of a layered soil using the direct stiffness method. The Green’s function is returned in Cartesian coordinates.

The number of elements $nElt$ is equal to the maximum of the lengths of $h$, $Cs$, $Cp$, $Ds$, $Dp$, and $rho$. If any of these parameters is defined as a scalar, an identical value is used for all elements.
green3d_bem

Purpose

3D Green’s functions for boundary element calculations.

Syntax

\[[\text{ug}, \text{sg}] = \text{GREEN3D\_BEM}(h, \text{Cs}, \text{Cp}, \text{Ds}, \text{Dp}, \text{rho}, \text{zs}, p, r, z, \omega)\]

Input arguments

\(h\)  
Layer thickness, INF for a halfspace (\(nElt \times 1\)) or (1 x 1).  
If h(end) \neq INF, the bottom interface is clamped.  
If necessary, extra layers are defined so that the source locations \(zs\) coincide with interfaces between elements.

\(Cs\)  
Shear wave velocity (\(nElt \times 1\)) or (1 x 1).

\(Cp\)  
Dilatational wave velocity (\(nElt \times 1\)) or (1 x 1).

\(Ds\)  
Shear damping ratio (\(nElt \times 1\)) or (1 x 1).

\(Dp\)  
Dilatational damping ratio (\(nElt \times 1\)) or (1 x 1).

\(rho\)  
Density (\(nElt \times 1\)) or (1 x 1).

\(zs\)  
Source locations (vertical coordinate) (\(nzSrc \times 1\)).

\(p\)  
Slowness, logarithmically sampled (\(nWave \times 1\)).  
If \(\omega\) \neq 0, the wavenumber sampling is given by \(k = \omega \times p\).  
If \(\omega\) = 0, the wavenumber sampling is given by \(k = p\).

\(r\)  
Receiver locations (radial coordinate) (\(nrRec \times 1\)).

\(z\)  
Receiver locations (vertical coordinate) (\(nzRec \times 1\)).

\(\omega\)  
Circular frequency (\(nFreq \times 1\)).

Output arguments

\(ug\)  
Green’s displacements (5 x nzSrc x nrRec x nzRec x nFreq).

\(ug(1,\ldots) = \hat{u}_{x}^{G}\)  
\(ug(3,\ldots) = \hat{u}_{y}^{G}\)  
\(ug(4,\ldots) = \hat{u}_{z}^{G}\)

\(ug(2,\ldots) = \hat{u}_{x}^{G}\)

\(sg\)  
Green’s stresses (10 x nzSrc x nrRec x nzRec x nFreq).

\(sg(1,\ldots) = \hat{\sigma}_{xxr}^{G}\)  
\(sg(5,\ldots) = \hat{\sigma}_{yrr}^{G}\)  
\(sg(7,\ldots) = \hat{\sigma}_{zzr}^{G}\)

\(sg(2,\ldots) = \hat{\sigma}_{xxy}^{G}\)

\(sg(6,\ldots) = \hat{\sigma}_{yxy}^{G}\)

\(sg(8,\ldots) = \hat{\sigma}_{xxx}^{G}\)

\(sg(3,\ldots) = \hat{\sigma}_{zzy}^{G}\)

\(sg(9,\ldots) = \hat{\sigma}_{zzz}^{G}\)

\(sg(4,\ldots) = \hat{\sigma}_{xzx}^{G}\)

\(sg(10,\ldots) = \hat{\sigma}_{zzr}^{G}\)

For receivers located on interfaces, the stresses in the underlying element are calculated.
Description

\([ug, sg] = \text{GREEN3D\_BEM}(h, Cs, Cp, Ds, Dp, rho, zs, p, r, z, omega)\)

computes the 3D Green’s functions of a layered soil using the direct stiffness method. The Green’s functions are returned in cylindrical coordinates. They are only calculated for receiver locations with a circumferential coordinate \(\theta = 0\). For other receiver locations, the response can be calculated as a linear combination of the Green’s functions at \(\theta = 0\). This approach is typically followed in boundary element calculations in order to save computer memory. For \(\theta = 0\), four components of the Green’s displacement tensor and eight components of the Green’s stress tensor are equal to zero. In order to save computer memory, only the nonzero components are returned by this function.

The number of elements \(nElt\) is equal to the maximum of the lengths of \(h\), \(Cs\), \(Cp\), \(Ds\), \(Dp\), and \(rho\). If any of these parameters is defined as a scalar, an identical value is used for all elements.
green3d_cyl

Purpose

3D Green’s functions in cylindrical coordinates.

Syntax

\[ [\text{ug}, \text{sg}] = \text{GREEN3D_CYL}(h, Cs, Cp, Ds, Dp, rho, zs, p, r, theta, z, omega) \]

Input arguments

- **h**  
  Layer thickness, \( \text{INF} \) for a halfspace \((n\text{Elt} \times 1)\) or \((1 \times 1)\).  
  If \( h(\text{end}) \neq \text{INF} \), the bottom interface is clamped.  
  If necessary, extra layers are defined so that the source locations \( zs \) coincide with interfaces between elements.
- **Cs**  
  Shear wave velocity \((n\text{Elt} \times 1)\) or \((1 \times 1)\).
- **Cp**  
  Dilatational wave velocity \((n\text{Elt} \times 1)\) or \((1 \times 1)\).
- **Ds**  
  Shear damping ratio \((n\text{Elt} \times 1)\) or \((1 \times 1)\).
- **Dp**  
  Dilatational damping ratio \((n\text{Elt} \times 1)\) or \((1 \times 1)\).
- **rho**  
  Density \((n\text{Elt} \times 1)\) or \((1 \times 1)\).
- **zs**  
  Source locations (vertical coordinate) \((n\text{Src} \times 1)\).
- **p**  
  Slowness, logarithmically sampled \((n\text{Wave} \times 1)\).  
  If \( \omega \neq 0 \), the wavenumber sampling is given by \( k = \omega \times p \).  
  If \( \omega = 0 \), the wavenumber sampling is given by \( k = p \).
- **r**  
  Receiver locations (radial coordinate) \((n\text{Rec} \times 1)\).
- **theta**  
  Receiver locations (circumferential coordinate) \((n\text{Rec} \times 1)\).
- **z**  
  Receiver locations (vertical coordinate) \((n\text{Rec} \times 1)\).
- **omega**  
  Circular frequency \((n\text{Freq} \times 1)\).
Output arguments

\[ \text{ug} \] Green’s displacements \((3 \times 3 \times \text{nzSrc} \times \text{nrRec} \times \text{ntRec} \times \text{nzRec} \times \text{nFreq})\).

\[ \text{ug}(1,1,\ldots) = \hat{u}_G^{x,rr} \quad \text{ug}(2,1,\ldots) = \hat{u}_G^{y,rr} \quad \text{ug}(3,1,\ldots) = \hat{u}_G^{z,rr} \]

\[ \text{ug}(1,2,\ldots) = \hat{u}_G^{x,θθ} \quad \text{ug}(2,2,\ldots) = \hat{u}_G^{y,θθ} \quad \text{ug}(3,2,\ldots) = \hat{u}_G^{z,θθ} \]

\[ \text{ug}(1,3,\ldots) = \hat{u}_G^{xz} \quad \text{ug}(2,3,\ldots) = \hat{u}_G^{yz} \quad \text{ug}(3,3,\ldots) = \hat{u}_G^{zz} \]

\[ \text{sg} \] Green’s stresses \((3 \times 6 \times \text{nzSrc} \times \text{nrRec} \times \text{ntRec} \times \text{nzRec} \times \text{nFreq})\).

\[ \text{sg}(1,1,\ldots) = \hat{σ}_G^{x,rr} \quad \text{sg}(2,1,\ldots) = \hat{σ}_G^{y,rr} \quad \text{sg}(3,1,\ldots) = \hat{σ}_G^{z,rr} \]

\[ \text{sg}(1,2,\ldots) = \hat{σ}_G^{x,θθ} \quad \text{sg}(2,2,\ldots) = \hat{σ}_G^{y,θθ} \quad \text{sg}(3,2,\ldots) = \hat{σ}_G^{z,θθ} \]

\[ \text{sg}(1,3,\ldots) = \hat{σ}_G^{xz} \quad \text{sg}(2,3,\ldots) = \hat{σ}_G^{yz} \quad \text{sg}(3,3,\ldots) = \hat{σ}_G^{zz} \]

\[ \text{sg}(1,4,\ldots) = \hat{σ}_G^{xθr} \quad \text{sg}(2,4,\ldots) = \hat{σ}_G^{yθr} \quad \text{sg}(3,4,\ldots) = \hat{σ}_G^{zθr} \]

\[ \text{sg}(1,5,\ldots) = \hat{σ}_G^{xθz} \quad \text{sg}(2,5,\ldots) = \hat{σ}_G^{yθz} \quad \text{sg}(3,5,\ldots) = \hat{σ}_G^{zθz} \]

\[ \text{sg}(1,6,\ldots) = \hat{σ}_G^{xzr} \quad \text{sg}(2,6,\ldots) = \hat{σ}_G^{yxr} \quad \text{sg}(3,6,\ldots) = \hat{σ}_G^{zzr} \]

For receivers located on interfaces, the stresses in the underlying element are calculated.

Description

\([\text{ug}, \text{sg}] = \text{GREEN3D}_\text{CYL}(h, \text{Cs}, \text{Cp}, \text{Ds}, \text{Dp}, \rho, z, p, r, \theta, z, \omega)\)

computes the 3D Green’s functions of a layered soil using the direct stiffness method. The Green’s functions are returned in cylindrical coordinates.

The number of elements \(\text{nElt}\) is equal to the maximum of the lengths of \(h, \text{Cs}, \text{Cp}, \text{Ds}, \text{Dp},\) and \(\rho\). If any of these parameters is defined as a scalar, an identical value is used for all elements.
green3d_cylz

Purpose

3D Green’s functions \( \hat{u}_{zj}^G \) and \( \hat{\sigma}_{zjk}^G \) in cylindrical coordinates.

Syntax

\[
[ug, sg] = \text{GREEN3D_CYLZ}(h, C_s, C_p, D_s, D_p, \rho, z_s, p, r, z, \omega)
\]

Input arguments

- \( h \) Layer thickness, \( \text{INF} \) for a halfspace \((\text{nElt} \times 1) \) or \((1 \times 1)\).
  - If \( h(\text{end}) \neq \text{INF} \), the bottom interface is clamped.
  - If necessary, extra layers are defined so that the source locations \( z_s \) coincide with interfaces between elements.
- \( C_s \) Shear wave velocity \((\text{nElt} \times 1) \) or \((1 \times 1)\).
- \( C_p \) Dilatational wave velocity \((\text{nElt} \times 1) \) or \((1 \times 1)\).
- \( D_s \) Shear damping ratio \((\text{nElt} \times 1) \) or \((1 \times 1)\).
- \( D_p \) Dilatational damping ratio \((\text{nElt} \times 1) \) or \((1 \times 1)\).
- \( \rho \) Density \((\text{nElt} \times 1) \) or \((1 \times 1)\).
- \( z_s \) Source locations (vertical coordinate) \((\text{nzSrc} \times 1)\).
- \( p \) Slowness, logarithmically sampled \((\text{nWave} \times 1)\).
  - If \( \omega \neq 0 \), the wavenumber sampling is given by \( k = \omega \times p \).
  - If \( \omega = 0 \), the wavenumber sampling is given by \( k = p \).
- \( r \) Receiver locations (radial coordinate) \((\text{nrRec} \times 1)\).
- \( z \) Receiver locations (vertical coordinate) \((\text{nzRec} \times 1)\).
- \( \omega \) Circular frequency \((\text{nFreq} \times 1)\).

Output arguments

- \( u_g \) Green’s displacements \((2 \times \text{nzSrc} \times \text{nrRec} \times \text{nzRec} \times \text{nFreq})\).
  - \( \text{ug}(1,...) = \hat{u}_{zr}^G \)
  - \( \text{ug}(2,...) = \hat{u}_{zz}^G \)
- \( s_g \) Green’s stresses \((4 \times \text{nzSrc} \times \text{nrRec} \times \text{nzRec} \times \text{nFreq})\).
  - \( \text{sg}(1,...) = \hat{\sigma}_{zrr}^G \)
  - \( \text{sg}(2,...) = \hat{\sigma}_{z\theta\theta}^G \)
  - \( \text{sg}(3,...) = \hat{\sigma}_{zzz}^G \)
  - \( \text{sg}(4,...) = \hat{\sigma}_{zrz}^G \)

For receivers located on interfaces, the stresses in the underlying element are calculated.
Description

\[[u_g, s_g] = \text{GREEN3D}\_\text{CYLZ}(h, C_s, C_p, D_s, D_p, \rho, z_s, p, r, z, \omega)\]

computes the 3D Green's functions \( \hat{u}_{Gj} \) and \( \hat{\sigma}_{Gjk} \) of a layered soil using the direct stiffness method. The Green's functions are returned in cylindrical coordinates.

The number of elements \( n\text{Elt} \) is equal to the maximum of the lengths of \( h, C_s, C_p, D_s, D_p, \) and \( \rho \). If any of these parameters is defined as a scalar, an identical value is used for all elements.
green3d_cylzz

Purpose
3D Green’s function $\hat{u}_G^{zz}$ in cylindrical coordinates.

Syntax
ugzz = GREEN3D_CYLZZ(h,Cs,Cp,Ds,Dp,rho,zs,p,r,z,omega)

Input arguments
- **h**: Layer thickness, INF for a halfspace ($nElt \times 1$) or $(1 \times 1)$. If $h$ is not INF, the bottom interface is clamped. If necessary, extra layers are defined so that the source locations $zs$ coincide with interfaces between elements.
- **Cs**: Shear wave velocity ($nElt \times 1$) or $(1 \times 1)$.
- **Cp**: Dilatational wave velocity ($nElt \times 1$) or $(1 \times 1)$.
- **Ds**: Shear damping ratio ($nElt \times 1$) or $(1 \times 1)$.
- **Dp**: Dilatational damping ratio ($nElt \times 1$) or $(1 \times 1)$.
- **rho**: Density ($nElt \times 1$) or $(1 \times 1)$.
- **zs**: Source locations (vertical coordinate) ($nzSrc \times 1$).
- **p**: Slowness, logarithmically sampled ($nWave \times 1$). If $omega \neq 0$, the wavenumber sampling is given by $k = omega \times p$. If $omega = 0$, the wavenumber sampling is given by $k = p$.
- **r**: Receiver locations (radial coordinate) ($nrRec \times 1$).
- **z**: Receiver locations (vertical coordinate) ($nzRec \times 1$).
- **omega**: Circular frequency ($nFreq \times 1$).

Output arguments
- **ugzz**: Green’s displacements ($nzSrc \times nrRec \times nzRec \times nFreq$).

Description
$ugzz = GREEN3D_CYLZZ(h,Cs,Cp,Ds,Dp,rho,zs,p,r,z,omega)$ computes the 3D Green’s function $\hat{u}_G^{zz}$ of a layered soil using the direct stiffness method. The Green’s function is returned in cylindrical coordinates.

The number of elements $nElt$ is equal to the maximum of the lengths of $h$, Cs, Cp, Ds, Dp, and rho. If any of these parameters is defined as a scalar, an identical value is used for all elements.
green3d_rec

Purpose
3D Green’s functions in Cartesian coordinates.

Syntax
[ug, sg] = GREEN3D_REC(h, Cs, Cp, Ds, Dp, rho, zs, p, x, y, z, omega)

Input arguments
h Layer thickness, INF for a halfspace (nElt x 1) or (1 x 1).
If h(end) ≠ INF, the bottom interface is clamped.
If necessary, extra layers are defined so that the source locations zs coincide with interfaces between elements.
Cs Shear wave velocity (nElt x 1) or (1 x 1).
Cp Dilatational wave velocity (nElt x 1) or (1 x 1).
Ds Shear damping ratio (nElt x 1) or (1 x 1).
Dp Dilatational damping ratio (nElt x 1) or (1 x 1).
rho Density (nElt x 1) or (1 x 1).
zs Source locations (vertical coordinate) (nzSrc x 1).
p Slowness, logarithmically sampled (nWave x 1).
If omega ≠ 0, the wavenumber sampling is given by k = omega × p.
If omega = 0, the wavenumber sampling is given by k = p.
x Receiver locations (x-coordinate) (nxRec x 1).
y Receiver locations (y-coordinate) (nyRec x 1).
z Receiver locations (z-coordinate) (nzRec x 1).
omega Circular frequency (nFreq x 1).
Output arguments

\[\text{ug} \text{ Green’s displacements } (3 \times 3 \times \text{nzSrc} \times \text{nxRec} \times \text{nyRec} \times \text{nzRec} \times \text{nFreq}) \]
\[\text{ug}(1,1,\ldots) = \hat{u}_G^{xx} \text{ ug}(2,1,\ldots) = \hat{u}_G^{yx} \text{ ug}(3,1,\ldots) = \hat{u}_G^{zx} \]
\[\text{ug}(1,2,\ldots) = \hat{u}_G^{xy} \text{ ug}(2,2,\ldots) = \hat{u}_G^{yy} \text{ ug}(3,2,\ldots) = \hat{u}_G^{zy} \]
\[\text{ug}(1,3,\ldots) = \hat{u}_G^{xz} \text{ ug}(2,3,\ldots) = \hat{u}_G^{yz} \text{ ug}(3,3,\ldots) = \hat{u}_G^{zz} \]

\[\text{sg} \text{ Green’s stresses } (3 \times 6 \times \text{nzSrc} \times \text{nxRec} \times \text{nyRec} \times \text{nzRec} \times \text{nFreq}) \]
\[\text{sg}(1,1,\ldots) = \hat{\sigma}_G^{xxx} \text{ sg}(2,1,\ldots) = \hat{\sigma}_G^{yxx} \text{ sg}(3,1,\ldots) = \hat{\sigma}_G^{zxx} \]
\[\text{sg}(1,2,\ldots) = \hat{\sigma}_G^{xyy} \text{ sg}(2,2,\ldots) = \hat{\sigma}_G^{yyy} \text{ sg}(3,2,\ldots) = \hat{\sigma}_G^{zyy} \]
\[\text{sg}(1,3,\ldots) = \hat{\sigma}_G^{xzz} \text{ sg}(2,3,\ldots) = \hat{\sigma}_G^{yzx} \text{ sg}(3,3,\ldots) = \hat{\sigma}_G^{zzx} \]
\[\text{sg}(1,4,\ldots) = \hat{\sigma}_G^{xxy} \text{ sg}(2,4,\ldots) = \hat{\sigma}_G^{yxy} \text{ sg}(3,4,\ldots) = \hat{\sigma}_G^{zyx} \]
\[\text{sg}(1,5,\ldots) = \hat{\sigma}_G^{xyz} \text{ sg}(2,5,\ldots) = \hat{\sigma}_G^{yxy} \text{ sg}(3,5,\ldots) = \hat{\sigma}_G^{zyz} \]
\[\text{sg}(1,6,\ldots) = \hat{\sigma}_G^{yxx} \text{ sg}(2,6,\ldots) = \hat{\sigma}_G^{zyx} \text{ sg}(3,6,\ldots) = \hat{\sigma}_G^{zzx} \]

For receivers located on interfaces, the stresses in the underlying element are calculated.

Description

\[\text{[ug, sg]} = \text{GREEN3D_REC(h, Cs, Cp, Ds, Dp, rho, zs, p, x, y, z, omega)}\] computes the 3D Green’s functions of a layered soil using the direct stiffness method. The Green’s functions are returned in Cartesian coordinates.

The number of elements \text{nElt} is equal to the maximum of the lengths of \text{h, Cs, Cp, Ds, Dp, and rho}. If any of these parameters is defined as a scalar, an identical value is used for all elements.
green3d_recz

Purpose
3D Green’s functions $\hat{u}^G_{zj}$ and $\hat{\sigma}^G_{zjk}$ in Cartesian coordinates.

Syntax
$$[u_g, s_g] = \text{GREEN3D\_REcz}(h, C_s, C_p, D_s, D_p, \rho, z_s, p, x, y, z, \omega)$$

Input arguments
- $h$: Layer thickness, INF for a halfspace ($n_{Elt} \times 1$) or ($1 \times 1$).
  - If $h(\text{end}) \neq \text{INF}$, the bottom interface is clamped.
  - If necessary, extra layers are defined so that the source locations $z_s$
    coincide with interfaces between elements.
- $C_s$: Shear wave velocity ($n_{Elt} \times 1$) or ($1 \times 1$).
- $C_p$: Dilatational wave velocity ($n_{Elt} \times 1$) or ($1 \times 1$).
- $D_s$: Shear damping ratio ($n_{Elt} \times 1$) or ($1 \times 1$).
- $D_p$: Dilatational damping ratio ($n_{Elt} \times 1$) or ($1 \times 1$).
- $\rho$: Density ($n_{Elt} \times 1$) or ($1 \times 1$).
- $z_s$: Source locations (vertical coordinate) ($n_{Src} \times 1$).
- $p$: Slowness, logarithmically sampled ($n_{Wave} \times 1$).
  - If $\omega \neq 0$, the wavenumber sampling is given by $k = \omega \times p$.
  - If $\omega = 0$, the wavenumber sampling is given by $k = p$.
- $x$: Receiver locations ($x$-coordinate) ($n_{Rec} \times 1$).
- $y$: Receiver locations ($y$-coordinate) ($n_{Rec} \times 1$).
- $z$: Receiver locations ($z$-coordinate) ($n_{Rec} \times 1$).
- $\omega$: Circular frequency ($n_{Freq} \times 1$).
Output arguments

\[ \text{ug} \]
Green’s displacements \((3 \times \text{nzSrc} \times \text{nxRec} \times \text{nyRec} \times \text{nzRec} \times \text{nFreq})\).
- \(\text{ug}(1,\ldots) = \hat{u}_x^G\)
- \(\text{ug}(2,\ldots) = \hat{u}_y^G\)
- \(\text{ug}(3,\ldots) = \hat{u}_z^G\)

\[ \text{sg} \]
Green’s stresses \((6 \times \text{nzSrc} \times \text{nxRec} \times \text{nyRec} \times \text{nzRec} \times \text{nFreq})\).
- \(\text{sg}(1,\ldots) = \hat{\sigma}_{xx}^G\)
- \(\text{sg}(2,\ldots) = \hat{\sigma}_{yy}^G\)
- \(\text{sg}(3,\ldots) = \hat{\sigma}_{zz}^G\)
- \(\text{sg}(4,\ldots) = \hat{\sigma}_{xy}^G\)
- \(\text{sg}(5,\ldots) = \hat{\sigma}_{yz}^G\)
- \(\text{sg}(6,\ldots) = \hat{\sigma}_{zx}^G\)

For receivers located on interfaces, the stresses in the underlying element are calculated.

Description

\[ [\text{ug}, \text{sg}] = \text{GREEN3D} \_\text{RECZ}(h,Cs,Cp,Ds,Dp,rho,zs,p,x,y,z,omega) \]
computes the 3D Green’s functions \(\hat{u}_g^{ij}\) and \(\hat{\sigma}_g^{ijk}\) of a layered soil using the direct stiffness method. The Green’s functions are returned in Cartesian coordinates.

The number of elements \(nElt\) is equal to the maximum of the lengths of \(h, Cs, Cp, Ds, Dp,\) and \(rho\). If any of these parameters is defined as a scalar, an identical value is used for all elements.

Examples

Example 9.1: The response of a halfspace due to a harmonic point load (p. 104).
Example 9.2: The response of a halfspace due to an impulsive point load (p. 105).
green3d_reczz

Purpose

3D Green’s function $\hat{u}_G^{zz}$ in Cartesian coordinates.

Syntax

$ugzz = \text{GREEN3D	extunderscore RECZZ}(h, Cs, Cp, Ds, Dp, rho, zs, p, x, y, z, \omega)$$

Input arguments

$h$  Layer thickness, INF for a halfspace ($nElt \times 1$) or ($1 \times 1$).
  If $h(\text{end}) \neq$ INF, the bottom interface is clamped.
  If necessary, extra layers are defined so that the source locations $zs$
  coincide with interfaces between elements.
$Cs$  Shear wave velocity ($nElt \times 1$) or ($1 \times 1$).
$Cp$  Dilatational wave velocity ($nElt \times 1$) or ($1 \times 1$).
$Ds$  Shear damping ratio ($nElt \times 1$) or ($1 \times 1$).
$Dp$  Dilatational damping ratio ($nElt \times 1$) or ($1 \times 1$).
$rho$  Density ($nElt \times 1$) or ($1 \times 1$).
$zs$  Source locations (vertical coordinate) ($nzSrc \times 1$).
$p$   Slowness, logarithmically sampled ($nWave \times 1$).
  If $\omega \neq 0$, the wavenumber sampling is given by $k = \omega \times p$.
  If $\omega = 0$, the wavenumber sampling is given by $k = p$.
$x$  Receiver locations ($x$-coordinate) ($nxRec \times 1$).
$y$  Receiver locations ($y$-coordinate) ($nyRec \times 1$).
$z$  Receiver locations ($z$-coordinate) ($nzRec \times 1$).
$\omega$  Circular frequency ($nFreq \times 1$).

Output arguments

$ugzz$  Green’s displacements ($nzSrc \times nxRec \times nyRec \times nzRec \times nFreq$).

Description

$ugzz = \text{GREEN3D	extunderscore RECZZ}(h, Cs, Cp, Ds, Dp, rho, zs, p, x, y, z, \omega)$ computes the
3D Green’s function $\hat{u}_G^{zz}$ of a layered soil using the direct stiffness method. The
Green’s function is returned in Cartesian coordinates.

The number of elements $nElt$ is equal to the maximum of the lengths of $h$, $Cs$, $Cp$, $Ds$, $Dp$, and $rho$. If any of these parameters is defined as a scalar, an identical
value is used for all elements.
green_psv

Purpose
Wavenumber domain P-SV Green’s functions.

Syntax
[Ug,Tg] = GREEN_PSV(h,Cs,Cp,Ds,Dp,rho,zs,p,z,omega)

Input arguments
h Layer thickness, INF for a halfspace (nElt x 1) or (1 x 1).
   If h(end) ≠ INF, the bottom interface is clamped.
   If necessary, extra layers are defined so that the source locations zs coincide with interfaces between elements.
Cs Shear wave velocity (nElt x 1) or (1 x 1).
Cp Dilatational wave velocity (nElt x 1) or (1 x 1).
Ds Shear damping ratio (nElt x 1) or (1 x 1).
Dp Dilatational damping ratio (nElt x 1) or (1 x 1).
rho Density (nElt x 1) or (1 x 1).
zs Source locations (vertical coordinate) (nSrc x 1).
p Slowness (nWave x 1).
   If omega ≠ 0, the wavenumber sampling is given by k = omega x p.
   If omega = 0, the wavenumber sampling is given by k = p.
z Receiver locations (vertical coordinate) (nRec x 1).
omega Circular frequency (nFreq x 1).

Output arguments
Ug Green’s displacements (2 x 2 x nSrc x nWave x nRec x nFreq).
   Ug(1,1,...) = \tilde{u}_{Gxx}^{G}
   Ug(1,2,...) = \tilde{u}_{Gxz}^{G}
   Ug(2,1,...) = \tilde{u}_{Gzx}^{G}
   Ug(2,2,...) = \tilde{u}_{Gzz}^{G}
Tg Green’s tractions (2 x 2 x nSrc x nWave x nRec x nFreq).
   Tg(1,1,...) = \tilde{t}_{Gxx}^{G}
   Tg(1,2,...) = \tilde{t}_{Gxz}^{G}
   Tg(2,1,...) = \tilde{t}_{Gzx}^{G}
   Tg(2,2,...) = \tilde{t}_{Gzz}^{G}

Description
[Ug,Tg] = GREEN_PSV(h,Cs,Cp,Ds,Dp,rho,zs,p,z,omega) computes the wavenumber domain P-SV Green’s functions of a layered soil using the direct stiffness method.
The number of elements $n\text{Elt}$ is equal to the maximum of the lengths of $h$, $Cs$, $Cp$, $Ds$, $Dp$, and $\rho$. If any of these parameters is defined as a scalar, an identical value is used for all elements.
green_sh

Purpose
Wavenumber domain SH Green’s functions.

Syntax
[Ug,Tg] = GREEN_SH(h,Cs,Ds,rho,zs,p,z,omega)

Input arguments
h    Layer thickness, INF for a halfspace \((nElt \times 1)\) or \((1 \times 1)\).
     If \(h(\text{end}) \neq \text{INF}\), the bottom interface is clamped.
     If necessary, extra layers are defined so that the source locations \(zs\)
     coincide with interfaces between elements.
Cs   Shear wave velocity \((nElt \times 1)\) or \((1 \times 1)\).
Ds   Shear damping ratio \((nElt \times 1)\) or \((1 \times 1)\).
rho  Density \((nElt \times 1)\) or \((1 \times 1)\).
zs   Source locations (vertical coordinate) \((nSrc \times 1)\).
P    Slowness \((nWave \times 1)\).
     If \(omega \neq 0\), the wavenumber sampling is given by \(k = omega \times p\).
     If \(omega = 0\), the wavenumber sampling is given by \(k = p\).
z    Receiver locations (vertical coordinate) \((nRec \times 1)\).
omega Circular frequency \((nFreq \times 1)\).

Output arguments
Ug   Green’s displacements \((1 \times 1 \times nSrc \times nWave \times nRec \times nFreq)\).
     \(Ug(1,1,...) = \tilde{u}_G^y\)
Tg   Green’s tractions \((1 \times 1 \times nSrc \times nWave \times nRec \times nFreq)\).
     \(Tg(1,1,...) = \tilde{t}_G^y\)

Description
[Ug,Tg] = GREEN_SH(h,Cs,Ds,rho,zs,p,z,omega) computes the wavenumber
domain SH Green’s functions of a layered soil using the direct stiffness method.

The number of elements \(nElt\) is equal to the maximum of the lengths of \(h\), \(Cs\),
\(Ds\), and \(rho\). If any of these parameters is defined as a scalar, an identical value is
used for all elements.
green_z

Purpose
Wavenumber domain Green’s functions $\tilde{u}_z^G$ and $\tilde{t}_z^G$.

Syntax

$$[Ug,Tg] = \text{GREEN}_Z(h,Cs,Cp,Ds,Dp,rho,zs,p,z,omega)$$

Input arguments

- $h$  Layer thickness, INF for a halfspace ($n\text{Elt} \times 1$) or $(1 \times 1)$.
  - If $h(\text{end}) \neq INF$, the bottom interface is clamped.
  - If necessary, extra layers are defined so that the source locations $zs$
    coincide with interfaces between elements.
- $Cs$  Shear wave velocity ($n\text{Elt} \times 1$) or $(1 \times 1)$.
- $Cp$  Dilatational wave velocity ($n\text{Elt} \times 1$) or $(1 \times 1)$.
- $Ds$  Shear damping ratio ($n\text{Elt} \times 1$) or $(1 \times 1)$.
- $Dp$  Dilatational damping ratio ($n\text{Elt} \times 1$) or $(1 \times 1)$.
- $rho$  Density ($n\text{Elt} \times 1$) or $(1 \times 1)$.
- $zs$  Source locations (vertical coordinate) ($n\text{Src} \times 1$).
- $p$  Slowness ($n\text{Wave} \times 1$).
  - If $omega \neq 0$, the wavenumber sampling is given by $k = omega \times p$.
  - If $omega = 0$, the wavenumber sampling is given by $k = p$.
- $z$  Receiver locations (vertical coordinate) ($n\text{Rec} \times 1$).
- $omega$  Circular frequency ($n\text{Freq} \times 1$).

Output arguments

- $Ug$  Green’s displacements ($2 \times n\text{Src} \times n\text{Wave} \times n\text{Rec} \times n\text{Freq}$).
  - $Ug(1,\ldots) = \tilde{u}_z^G$
  - $Ug(2,\ldots) = \tilde{u}_z^G$
- $Tg$  Green’s tractions ($2 \times n\text{Src} \times n\text{Wave} \times n\text{Rec} \times n\text{Freq}$).
  - $Tg(1,\ldots) = \tilde{t}_z^G$
  - $Tg(2,\ldots) = \tilde{t}_z^G$

Description

$$[Ug,Tg] = \text{GREEN}_Z(h,Cs,Cp,Ds,Dp,rho,zs,p,z,omega)$$ computes the
wavenumber domain Green’s functions $\tilde{u}_z^G$ and $\tilde{t}_z^G$ of a layered soil using
the direct stiffness method.
The number of elements \( n_{El} \) is equal to the maximum of the lengths of \( h \), \( C_s \), \( C_p \), \( D_s \), \( D_p \), and \( \rho \). If any of these parameters is defined as a scalar, an identical value is used for all elements.
green_zz

Purpose
Wavenumber domain Green's functions $\tilde{u}_{zz}^G$ and $\tilde{t}_{zz}^G$.

Syntax

$$[U_{gzz}, T_{gzz}] = \textsc{green}_\text{zz}(h, C_s, C_p, D_s, D_p, \rho, z, \omega)$$

Input arguments

- $h$: Layer thickness, INF for a halfspace ($n_{\text{Elt}} \times 1$) or ($1 \times 1$).
  - If $h(\text{end}) \neq \text{INF}$, the bottom interface is clamped.
  - If necessary, extra layers are defined so that the source locations $z_s$ coincide with interfaces between elements.

- $C_s$: Shear wave velocity ($n_{\text{Elt}} \times 1$) or ($1 \times 1$).

- $C_p$: Dilatational wave velocity ($n_{\text{Elt}} \times 1$) or ($1 \times 1$).

- $D_s$: Shear damping ratio ($n_{\text{Elt}} \times 1$) or ($1 \times 1$).

- $D_p$: Dilatational damping ratio ($n_{\text{Elt}} \times 1$) or ($1 \times 1$).

- $\rho$: Density ($n_{\text{Elt}} \times 1$) or ($1 \times 1$).

- $z_s$: Source locations (vertical coordinate) ($n_{\text{Src}} \times 1$).

- $p$: Slowness ($n_{\text{Wave}} \times 1$).
  - If $\omega \neq 0$, the wavenumber sampling is given by $k = \omega \times p$.
  - If $\omega = 0$, the wavenumber sampling is given by $k = p$.

- $z$: Receiver locations (vertical coordinate) ($n_{\text{Rec}} \times 1$).

- $\omega$: Circular frequency ($n_{\text{Freq}} \times 1$).

Output arguments

- $U_{gzz}$: Green's displacements ($n_{\text{Src}} \times n_{\text{Wave}} \times n_{\text{Rec}} \times n_{\text{Freq}}$).

- $T_{gzz}$: Green's tractions ($n_{\text{Src}} \times n_{\text{Wave}} \times n_{\text{Rec}} \times n_{\text{Freq}}$).

Description

$$[U_{gzz}, T_{gzz}] = \textsc{green}_\text{zz}(h, C_s, C_p, D_s, D_p, \rho, z_s, p, z, \omega)$$ computes the wavenumber domain Green's functions $\tilde{u}_{zz}^G$ and $\tilde{t}_{zz}^G$ of a layered soil using the direct stiffness method.

The number of elements $n_{\text{Elt}}$ is equal to the maximum of the lengths of $h$, $C_s$, $C_p$, $D_s$, $D_p$, and $\rho$. If any of these parameters is defined as a scalar, an identical value is used for all elements.
**greenf**

**Purpose**
Green's functions in the \((x,k_y)\)-domain.

**Syntax**

\[
\begin{align*}
[u_g, s_g] &= \text{GREENF}(h, C_s, C_p, D_s, D_p, \rho, z_s, \rho, x, py, z, \omega)
\end{align*}
\]

**Input arguments**

- **h** Layer thickness, \(\text{INF}\) for a halfspace \((nElt \times 1)\) or \((1 \times 1)\). 
  - If \(h(\text{end}) \neq \text{INF}\), the bottom interface is clamped. 
  - If necessary, extra layers are defined so that the source locations \(z_s\) coincide with interfaces between elements.
- **Cs** Shear wave velocity \((nElt \times 1)\) or \((1 \times 1)\).
- **Cp** Dilatational wave velocity \((nElt \times 1)\) or \((1 \times 1)\).
- **Ds** Shear damping ratio \((nElt \times 1)\) or \((1 \times 1)\).
- **Dp** Dilatational damping ratio \((nElt \times 1)\) or \((1 \times 1)\).
- **rho** Density \((nElt \times 1)\) or \((1 \times 1)\).
- **zs** Source locations (vertical coordinate) \((nzSrc \times 1)\).
- **px** Slowness in \(x\)-direction, logarithmically sampled \((nxWave \times 1)\). 
  - If \(\omega \neq 0\), the wavenumber sampling is given by \(k_x = \omega \times px\). 
  - If \(\omega = 0\), the wavenumber sampling is given by \(k_x = px\).
- **x** Receiver locations (\(x\)-coordinate) \((nxRec \times 1)\).
- **py** Slowness in \(y\)-direction \((nyWave \times 1)\) 
  - If \(\omega \neq 0\), the wavenumber sampling is given by \(k_y = \omega \times py\). 
  - If \(\omega = 0\), the wavenumber sampling is given by \(k_y = py\).
- **z** Receiver locations (\(z\)-coordinate) \((nzRec \times 1)\).
- **omega** Circular frequency \((nFreq \times 1)\).
Output arguments

\[ \text{ug} \]
Green's displacements \((3 \times 3 \times \text{nzSrc} \times \text{nxRec} \times \text{nyWave} \times \text{nzRec} \times \text{nFreq})\).

\[ \text{ug}(1,1,...) = \tilde{u}_G^{xx} \]
\[ \text{ug}(2,1,...) = \tilde{u}_G^{yx} \]
\[ \text{ug}(3,1,...) = \tilde{u}_G^{zx} \]
\[ \text{ug}(1,2,...) = \tilde{u}_G^{xy} \]
\[ \text{ug}(2,2,...) = \tilde{u}_G^{yy} \]
\[ \text{ug}(3,2,...) = \tilde{u}_G^{zy} \]
\[ \text{ug}(1,3,...) = \tilde{u}_G^{xz} \]
\[ \text{ug}(2,3,...) = \tilde{u}_G^{yz} \]
\[ \text{ug}(3,3,...) = \tilde{u}_G^{zz} \]

\[ \text{sg} \]
Green’s stresses \((3 \times 6 \times \text{nzSrc} \times \text{nxRec} \times \text{nyWave} \times \text{nzRec} \times \text{nFreq})\).

\[ \text{sg}(1,1,...) = \tilde{\sigma}_G^{xxx} \]
\[ \text{sg}(2,1,...) = \tilde{\sigma}_G^{yxx} \]
\[ \text{sg}(3,1,...) = \tilde{\sigma}_G^{zxx} \]
\[ \text{sg}(1,2,...) = \tilde{\sigma}_G^{xyy} \]
\[ \text{sg}(2,2,...) = \tilde{\sigma}_G^{yyy} \]
\[ \text{sg}(3,2,...) = \tilde{\sigma}_G^{zyy} \]
\[ \text{sg}(1,3,...) = \tilde{\sigma}_G^{xzz} \]
\[ \text{sg}(2,3,...) = \tilde{\sigma}_G^{yzx} \]
\[ \text{sg}(3,3,...) = \tilde{\sigma}_G^{zzx} \]

For receivers located on interfaces, the stresses in the underlying element are calculated.

Description

\[ [\text{ug}, \text{sg}] = \text{GREENF}(h, \text{Cs}, \text{Cp}, \text{Ds}, \text{Dp}, \text{rho}, \text{zs}, \text{px}, \text{py}, \text{z}, \omega) \]
computes the Green's functions of a layered soil in the \((x,k_y)\)-domain using the direct stiffness method. The Green's functions are returned on a rectangular grid \((x,py)\).

The number of elements \text{nElt} is equal to the maximum of the lengths of \(h, \text{Cs}, \text{Cp}, \text{Ds}, \text{Dp}, \) and \text{rho}. If any of these parameters is defined as a scalar, an identical value is used for all elements.
greenff

Purpose
Green's functions in the \((k_x, k_y)\)-domain.

Syntax

\[
[Ug, Sg] = \text{GREENFF}(h, Cs, Cp, Ds, Dp, rho, zs, p, px, py, z, omega)
\]

Input arguments

- \(h\): Layer thickness, \text{INF} for a halfspace \((nElt \times 1)\) or \((1 \times 1)\).
  If \(h(\text{end}) \neq \text{INF}\), the bottom interface is clamped.
  If necessary, extra layers are defined so that the source locations \(zs\) coincide with interfaces between elements.
- \(Cs\): Shear wave velocity \((nElt \times 1)\) or \((1 \times 1)\).
- \(Cp\): Dilatational wave velocity \((nElt \times 1)\) or \((1 \times 1)\).
- \(Ds\): Shear damping ratio \((nElt \times 1)\) or \((1 \times 1)\).
- \(Dp\): Dilatational damping ratio \((nElt \times 1)\) or \((1 \times 1)\).
- \(rho\): Density \((nElt \times 1)\) or \((1 \times 1)\).
- \(zs\): Source locations (vertical coordinate) \((nzSrc \times 1)\).
- \(p\): Radial slowness where the Green's functions are calculated. The result is used to compute the values on the \((px, py)\)-grid by interpolation. If \(p\) is an empty matrix, the Green's functions are calculated on each point of the \((px, py)\)-grid and no interpolation is performed.
  If \(omega \neq 0\), the wavenumber sampling is given by \(k = omega \times p\).
  If \(omega = 0\), the wavenumber sampling is given by \(k = p\).
- \(px\): Slowness in \(x\)-direction \((nxWave \times 1)\).
- \(py\): Slowness in \(y\)-direction \((nyWave \times 1)\).
- \(z\): Receiver locations (vertical coordinate) \((nzRec \times 1)\).
- \(omega\): Circular frequency \((nFreq \times 1)\).
Green's displacements \((3 \times 3 \times \text{nzSrc} \times \text{nxWave} \times \text{nyWave} \times \text{nzRec} \times \text{nFreq})\).

- \(U_g(1,1,...) = \tilde{u}_\text{xx}^G\)
- \(U_g(2,1,...) = \tilde{u}_\text{yx}^G\)
- \(U_g(3,1,...) = \tilde{u}_\text{zx}^G\)
- \(U_g(1,2,...) = \tilde{u}_\text{xy}^G\)
- \(U_g(2,2,...) = \tilde{u}_\text{yy}^G\)
- \(U_g(3,2,...) = \tilde{u}_\text{zy}^G\)
- \(U_g(1,3,...) = \tilde{u}_\text{xz}^G\)
- \(U_g(2,3,...) = \tilde{u}_\text{yz}^G\)
- \(U_g(3,3,...) = \tilde{u}_\text{zz}^G\)

Green's stresses \((3 \times 6 \times \text{nzSrc} \times \text{nxWave} \times \text{nyWave} \times \text{nzRec} \times \text{nFreq})\).

- \(S_g(1,1,...) = \tilde{\sigma}_\text{xxx}^G\)
- \(S_g(2,1,...) = \tilde{\sigma}_\text{yxx}^G\)
- \(S_g(3,1,...) = \tilde{\sigma}_\text{zxx}^G\)
- \(S_g(1,2,...) = \tilde{\sigma}_\text{xyy}^G\)
- \(S_g(2,2,...) = \tilde{\sigma}_\text{yyy}^G\)
- \(S_g(3,2,...) = \tilde{\sigma}_\text{zyy}^G\)
- \(S_g(1,3,...) = \tilde{\sigma}_\text{xz}^G\)
- \(S_g(2,3,...) = \tilde{\sigma}_\text{yz}^G\)
- \(S_g(3,3,...) = \tilde{\sigma}_\text{zz}^G\)
- \(S_g(1,4,...) = \tilde{\sigma}_\text{xxy}^G\)
- \(S_g(2,4,...) = \tilde{\sigma}_\text{yxy}^G\)
- \(S_g(3,4,...) = \tilde{\sigma}_\text{zz}^G\)
- \(S_g(1,5,...) = \tilde{\sigma}_\text{xzy}^G\)
- \(S_g(2,5,...) = \tilde{\sigma}_\text{yyz}^G\)
- \(S_g(3,5,...) = \tilde{\sigma}_\text{zz}^G\)
- \(S_g(1,6,...) = \tilde{\sigma}_\text{xzx}^G\)
- \(S_g(2,6,...) = \tilde{\sigma}_\text{yzy}^G\)
- \(S_g(3,6,...) = \tilde{\sigma}_\text{zz}^G\)

For receivers located on interfaces, the stresses in the underlying element are calculated.

**Description**

\([U_g,S_g] = \text{GREENFF}(h,\text{Cs},\text{Cp},\text{Ds},\text{Dp},\text{rho},\text{zs},\text{p},\text{px},\text{py},\text{z},\text{omega})\)

computes the Green's functions of a layered soil in the \((k_x,k_y)\)-domain using the direct stiffness method. The Green's functions are returned on a rectangular grid \((p_x,p_y)\).

The number of elements \(\text{nElt}\) is equal to the maximum of the lengths of \(h, \text{Cs}, \text{Cp}, \text{Ds}, \text{Dp},\) and \(\text{rho}\). If any of these parameters is defined as a scalar, an identical value is used for all elements.
greenff_z

Purpose
Green’s functions $\tilde{u}^G_{zj}$ and $\tilde{\sigma}^G_{zjk}$ in the $(k_x,k_y)$-domain.

Syntax

$$[Ug,Sg] = \text{GREENFF}_Z(h,Cs,Cp,Ds,Dp,rho,zs,p,px,py,z,\omega)$$

Input arguments

- $h$ Layer thickness, INF for a halfspace ($nElt \times 1$) or $(1 \times 1)$.
  - If $h(\text{end}) \neq \text{INF}$, the bottom interface is clamped.
  - If necessary, extra layers are defined so that the source locations $zs$ coincide with interfaces between elements.
- $Cs$ Shear wave velocity ($nElt \times 1$) or $(1 \times 1)$.
- $Cp$ Dilatational wave velocity ($nElt \times 1$) or $(1 \times 1)$.
- $Ds$ Shear damping ratio ($nElt \times 1$) or $(1 \times 1)$.
- $Dp$ Dilatational damping ratio ($nElt \times 1$) or $(1 \times 1)$.
- $rho$ Density ($nElt \times 1$) or $(1 \times 1)$.
- $zs$ Source locations (vertical coordinate) ($nzSrc \times 1$).
- $p$ Radial slowness where the Green’s functions are calculated. The result is used to compute the values on the $(px,py)$-grid by interpolation. If $p$ is an empty matrix, the Green’s functions are calculated on each point of the $(px,py)$-grid and no interpolation is performed.
  - If $\omega \neq 0$, the wavenumber sampling is given by $k = \omega \times p$.
  - If $\omega = 0$, the wavenumber sampling is given by $k = p$.
- $px$ Slowness in $x$-direction ($nxWave \times 1$).
- $py$ Slowness in $y$-direction ($nyWave \times 1$).
- $z$ Receiver locations (vertical coordinate) ($nzRec \times 1$).
- $\omega$ Circular frequency ($nFreq \times 1$).
Output arguments

\[ \text{Ug} \]
Green's displacements \((3 \times \text{nzSrc} \times \text{nxWave} \times \text{nyWave} \times \text{nzRec} \times \text{nFreq})\).

\[ \text{Ug}(1,...) = \tilde{u}_G^{zx} \]
\[ \text{Ug}(2,...) = \tilde{u}_G^{zy} \]
\[ \text{Ug}(3,...) = \tilde{u}_G^{zz} \]

\[ \text{Sg} \]
Green's stresses \((6 \times \text{nzSrc} \times \text{nxWave} \times \text{nyWave} \times \text{nzRec} \times \text{nFreq})\).

\[ \text{Sg}(1,...) = \tilde{\sigma}_G^{zx} \]
\[ \text{Sg}(2,...) = \tilde{\sigma}_G^{zy} \]
\[ \text{Sg}(3,...) = \tilde{\sigma}_G^{zz} \]
\[ \text{Sg}(4,...) = \tilde{\sigma}_G^{zxy} \]
\[ \text{Sg}(5,...) = \tilde{\sigma}_G^{zyz} \]
\[ \text{Sg}(6,...) = \tilde{\sigma}_G^{zzx} \]

For receivers located on interfaces, the stresses in the underlying element are calculated.

Description

\([\text{Ug},\text{Sg}] = \text{GREENFF}_Z(h,Cs,Cp,Ds,Dp,rho,zs,p,px,py,z,omega)\)
computes the Green's functions \(\tilde{u}_G^{zj}\) and \(\tilde{\sigma}_G^{zjk}\) of a layered soil in the \((k_x,k_y)\)-domain using the direct stiffness method. The Green's functions are returned on a rectangular grid \((px,py)\).

The number of elements \(\text{nElt}\) is equal to the maximum of the lengths of \(h, Cs, Cp, Ds, Dp,\) and \(rho\). If any of these parameters is defined as a scalar, an identical value is used for all elements.
greenff_zz

Purpose
Green’s function \( \tilde{u}_{zz}^G \) in the \((k_x, k_y)\)-domain.

Syntax

Input arguments

- \( h \) Layer thickness, \( \text{INF} \) for a halfspace \((\text{nElt} \times 1) \) or \((1 \times 1)\).
  - If \( h(\text{end}) \neq \text{INF} \), the bottom interface is clamped.
  - If necessary, extra layers are defined so that the source locations \( z_s \)
    coincide with interfaces between elements.
- \( C_s \) Shear wave velocity \((\text{nElt} \times 1) \) or \((1 \times 1)\).
- \( C_p \) Dilatational wave velocity \((\text{nElt} \times 1) \) or \((1 \times 1)\).
- \( D_s \) Shear damping ratio \((\text{nElt} \times 1) \) or \((1 \times 1)\).
- \( D_p \) Dilatational damping ratio \((\text{nElt} \times 1) \) or \((1 \times 1)\).
- \( \rho \) Density \((\text{nElt} \times 1) \) or \((1 \times 1)\).
- \( z_s \) Source locations (vertical coordinate) \((\text{nzSrc} \times 1)\).
- \( p \) Radial slowness where the Green’s functions are calculated. The result is
  used to compute the values on the \((px, py)\)-grid by interpolation. If \( p \) is an empty matrix, the Green’s functions are calculated on each point of the \((px, py)\)-grid and no interpolation is performed.
  - If \( \omega \neq 0 \), the wavenumber sampling is given by \( k = \omega \times p \).
  - If \( \omega = 0 \), the wavenumber sampling is given by \( k = p \).
- \( px \) Slowness in \( x \)-direction \((\text{nxWave} \times 1)\).
- \( py \) Slowness in \( y \)-direction \((\text{nyWave} \times 1)\).
- \( z \) Receiver locations (vertical coordinate) \((\text{nzRec} \times 1)\).
- \( \omega \) Circular frequency \((\text{nFreq} \times 1)\).

Output arguments

- \( U_{gzz} \) Green’s displacements \((\text{nzSrc} \times \text{nxWave} \times \text{nyWave} \times \text{nzRec} \times \text{nFreq})\).

Description

\( U_{gzz} = \text{GREENFF}_ZZ(h, C_s, C_p, D_s, D_p, \rho, z_s, p, px, py, z, \omega) \) computes the Green’s function \( \tilde{u}_{zz}^G \) of a layered soil in the \((k_x, k_y)\)-domain using the direct stiffness method. The Green’s function is returned on a rectangular grid \((px, py)\).
The number of elements nElt is equal to the maximum of the lengths of h, Cs, Cp, Ds, Dp, and rho. If any of these parameters is defined as a scalar, an identical value is used for all elements.
ke_dsmpsv

Purpose
Element stiffness matrix (direct stiffness method, P-SV-waves).

Syntax
Ke = KE_DSMPSV(h,Cs,Cp,Ds,Dp,rho,k,omega,ds,dp)

Input arguments
- h: Layer thickness, INF for a halfspace.
- Cs: Shear wave velocity.
- Cp: Dilatational wave velocity.
- Ds: Shear damping ratio.
- Dp: Dilatational damping ratio.
- rho: Density.
- k: Horizontal wavenumber.
- omega: Circular frequency.
- dp: Dilatational wave direction. Default: equal to ds.

Output arguments
- Ke: Element stiffness matrix (4 × 4) or (2 × 2).

Description
Ke = KE_DSMPSV(h,Cs,Cp,Ds,Dp,rho,k,omega,ds,dp) computes the element stiffness matrix Ke for P-SV-waves in a layer or a halfspace using the direct stiffness method.

The optional arguments ds and dp are only applicable to halfspace elements, and must be equal to 1 (outgoing waves) or -1 (incoming waves).

Examples
Example 4.1: The stiffness matrix of a halfspace (p. 35).
Example 4.2: Vertical harmonic wave propagation in a layer on bedrock (p. 37).
Example 4.3: Stiffness matrices for a layered medium (p. 40).
ke_dsmsh

Purpose
Element stiffness matrix (direct stiffness method, SH-waves).

Syntax
Ke = KE_DSMSH(h,Cs,Ds,rho,k,omega,ds)

Input arguments
h    Layer thickness, INF for a halfspace.
Cs   Shear wave velocity.
Ds   Shear damping ratio.
rho  Density.
k    Horizontal wavenumber.
omega Circular frequency.
ds   Shear wave direction. Default: 1.

Output arguments
Ke   Element stiffness matrix (2 × 2) or (1 × 1).

Description
Ke = KE_DSMSH(h,Cs,Ds,rho,k,omega,ds) computes the element stiffness matrix Ke for SH-waves in a layer or a halfspace using the direct stiffness method.

The optional argument ds is only applicable to halfspace elements, and must be equal to 1 (outgoing waves) or -1 (incoming waves).
ke_tlmmps

Purpose
Element stiffness matrix (thin layer method, P-SV-waves).

Syntax

\[
[Ae, Be, Ge, Me] = \text{KE\_TLMPSV}(h, Cs, Cp, Ds, Dp, rho)
\]

Input arguments
- \(h\) Layer thickness.
- \(Cs\) Shear wave velocity.
- \(Cp\) Dilatational wave velocity.
- \(Ds\) Shear damping ratio.
- \(Dp\) Dilatational damping ratio.
- \(rho\) Density.

Output arguments
- \(Ae\) Layer stiffness matrix component proportional to \(k^2\) (4 x 4).
- \(Be\) Layer stiffness matrix component proportional to \(k\) (4 x 4).
- \(Ge\) Constant component of layer stiffness matrix (4 x 4).
- \(Me\) Layer stiffness matrix component proportional to \(-\omega^2\) (4 x 4).

Description

\([Ae, Be, Ge, Me] = \text{KE\_TLMPSV}(h, Cs, Cp, Ds, Dp, rho)\) returns the element stiffness matrices for P-SV-waves in a layer using the thin layer method. The excitation frequency is assumed to be positive. For a zero or negative excitation frequency, use respectively the real part or the complex conjugate of the matrices \(Ae, Be, Ge,\) and \(Me.\)
ke_tlmsh

Purpose
Element stiffness matrix (thin layer method, SH-waves).

Syntax

\[ [A_e, B_e, G_e, M_e] = KE_TLMSH(h, C_s, D_s, \rho) \]

Input arguments
- \( h \) Layer thickness.
- \( C_s \) Shear wave velocity.
- \( D_s \) Shear damping ratio.
- \( \rho \) Density.

Output arguments
- \( A_e \) Layer stiffness matrix component proportional to \( k^2 \) \((4 \times 4)\).
- \( B_e \) Layer stiffness matrix component proportional to \( k \) \((4 \times 4)\).
- \( G_e \) Constant component of layer stiffness matrix \((4 \times 4)\).
- \( M_e \) Layer stiffness matrix component proportional to \(-\omega^2 \) \((4 \times 4)\).

Description

\[ [A_e, B_e, G_e, M_e] = KE_TLMSH(h, C_s, D_s, \rho) \] returns the element stiffness matrix for SH-waves in a layer using the thin layer method. The excitation frequency is assumed to be positive. For a zero or negative excitation frequency, use respectively the real part or the complex conjugate of the matrices \( A_e \), \( B_e \), \( G_e \), and \( M_e \).
**k_dsmpsv**

**Purpose**
Global stiffness matrix (direct stiffness method, P-SV-waves).

**Syntax**
\[ K = K_{DSMPSV}(h, Cs, Cp, Ds, Dp, rho, k, omega, ds, dp) \]

**Input arguments**
- **h** Layer thickness, INF for a halfspace \((nElts \times 1)\) or \((1 \times 1)\).
- **Cs** Shear wave velocity \((nElts \times 1)\) or \((1 \times 1)\).
- **Cp** Dilatational wave velocity \((nElts \times 1)\) or \((1 \times 1)\).
- **Ds** Shear damping ratio \((nElts \times 1)\) or \((1 \times 1)\).
- **Dp** Dilatational damping ratio \((nElts \times 1)\) or \((1 \times 1)\).
- **rho** Density \((nElts \times 1)\) or \((1 \times 1)\).
- **k** Horizontal wavenumber.
- **omega** Circular frequency.
- **ds** Shear wave direction. Default: 1.
- **dp** Dilatational wave direction. Default: equal to **ds**.

**Output arguments**
- **K** Stiffness matrix \((nDOFs \times nDOFs)\).

**Description**
\[ K = K_{DSMPSV}(h, Cs, Cp, Ds, Dp, rho, k, omega, ds, dp) \] assembles the stiffness matrix \(K\) for P-SV-waves in a layered soil by means of the direct stiffness method.

The number of elements \(nElts\) is equal to the maximum of the lengths of \(h, Cs, Cp, Ds, Dp,\) and \(rho\). If any of these parameters is defined as a scalar, an identical value is used for all elements.

The optional arguments **ds** and **dp** are only applicable to halfspace elements, and must be equal to 1 (outgoing waves) or -1 (incoming waves).

**Examples**
Example 4.3: Stiffness matrices for a layered medium (p. 40).
Example 4.5: Vertical transient wave propagation in a layer on a halfspace (p. 47).
**k_dsmsh**

**Purpose**
Global stiffness matrix (direct stiffness method, SH-waves).

**Syntax**
\[
K = K\_DSMSH(h,Cs,Ds,rho,k,omega,ds)
\]

**Input arguments**
- \( h \)
  - Layer thickness, INF for a halfspace (\( nElt \times 1 \)) or (1 \times 1).
- \( Cs \)
  - Shear wave velocity (\( nElt \times 1 \)) or (1 \times 1).
- \( Ds \)
  - Shear damping ratio (\( nElt \times 1 \)) or (1 \times 1).
- \( rho \)
  - Density (\( nElt \times 1 \)) or (1 \times 1).
- \( k \)
  - Horizontal wavenumber.
- \( omega \)
  - Circular frequency.
- \( ds \)
  - Shear wave direction. Default: 1.

**Output arguments**
- \( K \)
  - Stiffness matrix (\( nDOF \times nDOF \)).

**Description**
\( K = K\_DSMSH(h,Cs,Ds,rho,k,omega,ds) \) assembles the stiffness matrix \( K \) for SH-waves in a layered soil by means of the direct stiffness method.

The number of elements \( nElt \) is equal to the maximum of the lengths of \( h \), \( Cs \), \( Ds \), and \( rho \). If any of these parameters is defined as a scalar, an identical value is used for all elements.

The optional argument \( ds \) is only applicable to halfspace elements, and must be equal to 1 (outgoing waves) or -1 (incoming waves).
k_tlmpsv

Purpose
Global stiffness matrix (thin layer method, P-SV-waves).

Syntax
[A,B,G,M] = K_TLMPSV(h,Cs,Cp,Ds,Dp,rho)

Input arguments
- h  Layer thickness (nElt x 1) or (1 x 1).
- Cs Shear wave velocity (nElt x 1) or (1 x 1).
- Cp Dilatational wave velocity (nElt x 1) or (1 x 1).
- Ds Shear damping ratio (nElt x 1) or (1 x 1).
- Dp Dilatational damping ratio (nElt x 1) or (1 x 1).
- rho Density (nElt x 1) or (1 x 1).

Output arguments
- A Stiffness matrix component proportional to $k^2$ (nDOF x nDOF).
- B Stiffness matrix component proportional to $k$ (nDOF x nDOF).
- G Constant component of stiffness matrix (nDOF x nDOF).
- M Stiffness matrix component proportional to $-\omega^2$ (nDOF x nDOF).

Description
[A,B,G,M] = K_TLMPSV(h,Cs,Cp,Ds,Dp,rho) assembles the stiffness matrix for P-SV-waves in a layered soil by means of the thin layer method. The excitation frequency is assumed to be positive. For a zero or negative excitation frequency, use respectively the real part or the complex conjugate of the matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{G}$, and $\mathbf{M}$.

The number of elements nElt is equal to the maximum of the lengths of h, Cs, Cp, Ds, Dp, and rho. If any of these parameters is defined as a scalar, an identical value is used for all elements.

Example
Example 5.1: Vertical harmonic wave propagation in a layer on bedrock (p. 54).
k_tlmsh

Purpose
Global stiffness matrix (thin layer method, SH-waves).

Syntax
\[ [A, B, G, M] = \text{K_TLMSH}(h, Cs, Ds, rho) \]

Input arguments
- \( h \) Layer thickness (\( n_{\text{Elt}} \times 1 \)) or (1 x 1).
- \( Cs \) Shear wave velocity (\( n_{\text{Elt}} \times 1 \)) or (1 x 1).
- \( Ds \) Shear damping ratio (\( n_{\text{Elt}} \times 1 \)) or (1 x 1).
- \( rho \) Density (\( n_{\text{Elt}} \times 1 \)) or (1 x 1).

Output arguments
- \( A \) Stiffness matrix component proportional to \( k^2 \) (\( n_{\text{DOF}} \times n_{\text{DOF}} \)).
- \( B \) Stiffness matrix component proportional to \( k \) (\( n_{\text{DOF}} \times n_{\text{DOF}} \)).
- \( G \) Constant component of stiffness matrix (\( n_{\text{DOF}} \times n_{\text{DOF}} \)).
- \( M \) Stiffness matrix component proportional to \( -\omega^2 \) (\( n_{\text{DOF}} \times n_{\text{DOF}} \)).

Description
\[ [A, B, G, M] = \text{K_TLMSH}(h, Cs, Ds, rho) \] assembles the stiffness matrix for SH waves in a layered soil by means of the thin layer method. The excitation frequency is assumed to be positive. For a zero or negative excitation frequency, use respectively the real part or the complex conjugate of the matrices \( A, B, G, \) and \( M \).

The number of elements \( n_{\text{Elt}} \) is equal to the maximum of the lengths of \( h, Cs, Ds, \) and \( rho \). If any of these parameters is defined as a scalar, an identical value is used for all elements.

Example
Example 8.2: Surface waves in a homogeneous layer on bedrock (p. 95).
linamp

Purpose
Linear 1-D site response analysis.

Syntax
\[ a = \text{LINAMP}(h, C, D, \rho, a_i, F, z) \]

Input arguments
- \( h \)  Layer thickness, \( \text{INF} \) for a halfspace (\( n_{\text{Elt}} \times 1 \)) or (1 \times 1).
- \( C \)  Wave velocity (\( n_{\text{Elt}} \times 1 \)) or (1 \times 1).
- \( D \)  Damping ratio (\( n_{\text{Elt}} \times 1 \)) or (1 \times 1).
- \( \rho \)  Density (\( n_{\text{Elt}} \times 1 \)) or (1 \times 1).
- \( a_i \)  Time history of the incident wave acceleration (\( N \times 1 \)).
- \( F \)  Sampling frequency for \( a_i \) (1 \times 1).
- \( z \)  Receiver depth (\( n_{\text{Rec}} \times 1 \)). Default: 0.

Output arguments
- \( a \)  Time history of the acceleration at the receivers (\( N \times n_{\text{Rec}} \)).

Description
\[ a = \text{LINAMP}(h, C, D, \rho, a_i, F, z) \] performs a one-dimensional site amplification analysis using a linear material model.

The number of elements \( n_{\text{Elt}} \) is equal to the maximum of the lengths of \( h, C, D, \) and \( \rho \). If any of these parameters is defined as a scalar, an identical value is used for all elements.

Example
Example 7.2: Response of a layered soil due to an earthquake - linear analysis (p. 73).
logffcos

Purpose
Logarithmic fast Fourier cosine transform.

Syntax
\[
g = \text{LOGFFCOS}(f, k, r, \text{dim}, \text{accu})
\]

Input arguments
- \( f \): Function to transform sampled at \( k \) \((M_1 \times M_2 \times \ldots \times M_k \times \ldots \times M_n)\).
- \( k \): Log spaced points, must be sorted in ascending order \((N_k \times 1)\).
- \( r \): Sampling in the transformed domain \((N_r \times 1)\).
- \( \text{dim} \): Operate along dimension \( \text{dim} \). Default: first non-singleton dimension.
- \( \text{accu} \): Kernel accuracy. Determines amount of padding. Default: \(10^{-6}\).

Output arguments
- \( g \): Transformed function sampled at \( r \) \((M_1 \times M_2 \times \ldots \times N_r \times \ldots \times M_n)\).

Description
\[
g = \text{LOGFFCOS}(f, k, r, \text{dim}, \text{accu}) \text{ calculates the Fourier cosine transform } g(r) \text{ of the function } f(k) \text{ where the } k-\text{axis is logarithmically sampled [27]. The Fourier cosine transform } g(r) \text{ of the function } f(k) \text{ is defined as:}
\]
\[
g(r) = \int_0^{\infty} f(k) \cos(kr) \, dk
\]

Example
Example 6.1: The response of a halfspace due to a harmonic line load (p. 62).
**logffsin**

**Purpose**
Logarithmic fast Fourier sine transform.

**Syntax**
\[
g = \text{LOGFFSIN}(f, k, r, \text{dim}, \text{accu})
\]

**Input arguments**
- **f**: Function to transform sampled at \( k \) \((M_1 \times M_2 \times \ldots \times N_k \times \ldots \times M_n)\).
- **k**: Log spaced points, must be sorted in ascending order \((N_k \times 1)\).
- **r**: Sampling in the transformed domain \((N_r \times 1)\).
- **dim**: Operate along dimension \text{dim}. Default: first non-singleton dimension.
- **accu**: Kernel accuracy. Determines amount of padding. Default: \(10^{-6}\).

**Output arguments**
- **g**: Transformed function sampled at \( r \) \((M_1 \times M_2 \times \ldots \times N_r \times \ldots \times M_n)\).

**Description**
\[
g(\mathbf{r}) = \int_0^\infty f(\mathbf{k}) \sin(\mathbf{k} \cdot \mathbf{r}) d\mathbf{k}
\]

\text{g} = \text{LOGFFSIN}(\mathbf{f}, \mathbf{k}, \mathbf{r}, \text{dim}, \text{accu}) \text{ calculates the Fourier sine transform } \text{g}(\mathbf{r}) \text{ of the function } f(\mathbf{k}) \text{ where the k-axis is logarithmically sampled} [27]. \text{ The Fourier sine transform } g(\mathbf{r}) \text{ of the function } f(\mathbf{k}) \text{ is defined as:}
**logfft**

**Purpose**
Logarithmic fast forward Fourier transform of causal signals.

**Syntax**

\[ g = \text{LOGFFT}(f,k,r,\text{dim},\text{accu}) \]

**Input arguments**
- **f**: Function to transform sampled at \( k (M_1 \times M_2 \times \ldots \times N_k \times \ldots \times M_n) \).
- **k**: Log spaced points, must be sorted in ascending order \( (N_k \times 1) \).
- **r**: Sampling in the transformed domain \( (N_r \times 1) \).
- **dim**: Operate along dimension \( \text{dim} \). Default: first non-singleton dimension.
- **accu**: Kernel accuracy. Determines amount of padding. Default: \( 10^{-6} \).

**Output arguments**
- **g**: Transformed function sampled at \( r (M_1 \times M_2 \times \ldots \times N_r \times \ldots \times M_n) \).

**Description**

\( g = \text{LOGFFT}(f,k,r,\text{dim},\text{accu}) \) calculates the forward Fourier transform \( g(r) \) of the causal function \( f(k) \) where the \( k \)-axis is logarithmically sampled [27]. The forward Fourier transform \( g(r) \) of the function \( f(k) \) is defined as:

\[ g(r) = \int_{-\infty}^{\infty} f(k)e^{-ikr} \, dk \]
logifft

Purpose
Logarithmic fast inverse Fourier transform of real signals.

Syntax
\[ g = \text{LOGIFFT}(f,k,r,\text{dim},\text{accu}) \]

Input arguments
- \( f \): Function to transform sampled at \( k \) (\( M_1 \times M_2 \times \cdots \times N_k \times \cdots \times M_n \)).
- \( k \): Log spaced points, must be sorted in ascending order (\( N_k \times 1 \)).
- \( r \): Sampling in the transformed domain (\( N_r \times 1 \)).
- \( \text{dim} \): Operate along dimension \( \text{dim} \). Default: first non-singleton dimension.
- \( \text{accu} \): Kernel accuracy. Determines amount of padding. Default: \( 10^{-6} \).

Output arguments
- \( g \): Transformed function sampled at \( r \) (\( M_1 \times M_2 \times \cdots \times N_r \times \cdots \times M_n \)).

Description
\( g = \text{LOGIFFT}(f,k,r,\text{dim},\text{accu}) \) calculates the inverse Fourier transform \( g(r) \) of the function \( f(k) \) where the \( k \)-axis is logarithmically sampled [27]. The function \( g(r) \) is assumed to be a real signal. The inverse Fourier transform \( g(r) \) of the function \( f(k) \) is defined as:
\[
g(r) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(k) e^{ikr} \, dk
\]

Example
Example 9.2: The response of a halfspace due to an impulsive point load (p. 105).
ne_dsmpsv

Purpose
Element displacement shape functions (direct stiffness method, P-SV-waves).

Syntax
[Nxe,Nze] = NE_DSMPSV(h,Cs,Cp,Ds,Dp,rho,k,omega,z,ds,dp)

Input arguments
- **h**: Layer thickness, INF for a halfspace.
- **Cs**: Shear wave velocity.
- **Cp**: Dilatational wave velocity.
- **Ds**: Shear damping ratio.
- **Dp**: Dilatational damping ratio.
- **rho**: Density.
- **k**: Horizontal wavenumber.
- **omega**: Circular frequency.
- **z**: Receiver depth (nRec × 1).
- **ds**: Shear wave direction. Default: 1.
- **dp**: Dilatational wave direction. Default: equal to ds.

Output arguments
- **Nxe**: Shape functions for horizontal displacement (nRec × 4) or (nRec × 2).
- **Nze**: Shape functions for vertical displacement (nRec × 4) or (nRec × 2).

Description
[Nxe,Nze] = NE_DSMPSV(h,Cs,Cp,Ds,Dp,rho,k,omega,z,ds,dp) returns the displacement shape functions Nxe and Nze used in the direct stiffness method for P-SV waves in a layer or a halfspace. The optional arguments ds and dp are only applicable to halfspace elements, and must be equal to 1 (outgoing waves) or -1 (incoming waves). For receivers outside the layer or the halfspace, zero is returned.

Example
Example 4.4: The shape functions of a layer element (p. 43).


**ne_dsmsh**

**Purpose**
Element displacement shape functions (direct stiffness method, SH-waves).

**Syntax**

\[ Nye = \text{NE}_D\text{SMSH}(h, Cs, Ds, rho, k, omega, z, ds) \]

**Input arguments**
- \( h \): Layer thickness, \( \text{INF} \) for a halfspace.
- \( Cs \): Shear wave velocity.
- \( Ds \): Shear damping ratio.
- \( rho \): Density.
- \( k \): Horizontal wavenumber.
- \( omega \): Circular frequency.
- \( z \): Receiver depth (\( nRec \times 1 \)).
- \( ds \): Shear wave direction. Default: 1.

**Output arguments**
- \( Nye \): Shape functions (\( nRec \times 2 \)) or (\( nRec \times 1 \)).

**Description**

\[ Nye = \text{NE}_D\text{SMSH}(h, Cs, Ds, rho, k, omega, z, ds) \] returns the displacement shape functions \( Nye \) used in the direct stiffness method for SH-waves in a layer or a halfspace.

The optional argument \( ds \) is only applicable to halfspace elements, and must be equal to 1 (outgoing waves) or -1 (incoming waves).

For receivers outside the layer or the halfspace, zero is returned.
ne_tlmpsv

Purpose
Element displacement shape functions (thin layer method, P-SV-waves).

Syntax

\[
\text{[Nxe,Nze]} = \text{NE_TLMPSV}(h,z)
\]

Input arguments

h    Layer thickness.
\(z\) Receiver depth (\(n_{\text{Rec}} \times 1\)).

Output arguments

Nxe  Shape functions for horizontal displacement (\(n_{\text{Rec}} \times 4\)).
Nze  Shape functions for vertical displacement (\(n_{\text{Rec}} \times 4\)).

Description

\[
\text{[Nxe,Nze]} = \text{NE_TLMPSV}(h,z)
\]
returns the displacement shape functions used in the thin layer method for P-SV-waves in a layer.
ne_tlmsh

Purpose
Element displacement shape functions (thin layer method, SH-waves).

Syntax
Nye = NE_TLMSH(h,z)

Input arguments
h    Layer thickness.
z    Receiver depth (nRec × 1).

Output arguments
Nye    Shape functions (nRec × 2).

Description
Nye = NE_TLMSH(h,z) returns the displacement shape functions used in the thin layer method for SH-waves in a layer.
n_dsmpsv

**Purpose**
Global displacement shape functions (direct stiffness method, P-SV-waves).

**Syntax**

\[
[Nx, Nz] = \text{N\_DSMPSV}(h, Cs, Cp, Ds, Dp, rho, k, omega, z, ds, dp)
\]

**Input arguments**
- \(h\): Layer thickness, \(\text{INF}\) for a halfspace (\(n\text{Elt} \times 1\) or \(1 \times 1\)).
- \(Cs\): Shear wave velocity (\(n\text{Elt} \times 1\) or \(1 \times 1\)).
- \(Cp\): Dilatational wave velocity (\(n\text{Elt} \times 1\) or \(1 \times 1\)).
- \(Ds\): Shear damping ratio (\(n\text{Elt} \times 1\) or \(1 \times 1\)).
- \(Dp\): Dilatational damping ratio (\(n\text{Elt} \times 1\) or \(1 \times 1\)).
- \(rho\): Density (\(n\text{Elt} \times 1\) or \(1 \times 1\)).
- \(k\): Horizontal wavenumber.
- \(omega\): Circular frequency.
- \(z\): Receiver depth (\(n\text{Rec} \times 1\)).
- \(ds\): Shear wave direction. Default: 1.
- \(dp\): Dilatational wave direction. Default: equal to \(ds\).

**Output arguments**
- \(Nx\): Shape functions for horizontal displacement (\(n\text{Rec} \times n\text{DOF}\)).
- \(Nz\): Shape functions for vertical displacement (\(n\text{Rec} \times n\text{DOF}\)).

**Description**

\([Nx, Nz] = \text{N\_DSMPSV}(h, Cs, Cp, Ds, Dp, rho, k, omega, z, ds, dp)\) computes the displacement shape functions \(Nx\) and \(Nz\) used in the direct stiffness method for P-SV waves in a layered soil.

The number of elements \(n\text{Elt}\) is equal to the maximum of the lengths of \(h\), \(Cs\), \(Cp\), \(Ds\), \(Dp\), and \(rho\). If any of these parameters is defined as a scalar, an identical value is used for all elements.

The optional arguments \(ds\) and \(dp\) are only applicable to halfspace elements, and must be equal to 1 (outgoing waves) or -1 (incoming waves).

For receivers outside the soil, zero is returned.
Example

Example 4.5: Vertical transient wave propagation in a layer on a halfspace (p. 47).
n_dsmsh

Purpose
Global displacement shape functions (direct stiffness method, SH-waves).

Syntax
\( Ny = \text{N_DSMSH}(h, Cs, Ds, rho, k, omega, z, ds) \)

Input arguments
- \( h \): Layer thickness, \text{INF} for a halfspace \((nElt \times 1)\) or \((1 \times 1)\).
- \( Cs \): Shear wave velocity \((nElt \times 1)\) or \((1 \times 1)\).
- \( Ds \): Shear damping ratio \((nElt \times 1)\) or \((1 \times 1)\).
- \( rho \): Density \((nElt \times 1)\) or \((1 \times 1)\).
- \( k \): Horizontal wavenumber.
- \( omega \): Circular frequency.
- \( z \): Receiver depth \((nRec \times 1)\).
- \( ds \): Shear wave direction. Default: 1.

Output arguments
- \( Ny \): Shape functions \((nRec \times nDOF)\).

Description
\( Ny = \text{N_DSMSH}(h, Cs, Ds, rho, k, omega, z, ds) \) returns the displacement shape functions \( Ny \) used in the direct stiffness method for SH-waves in a layered soil.

The number of elements \( nElt \) is equal to the maximum of the lengths of \( h, Cs, Ds, \) and \( rho \). If any of these parameters is defined as a scalar, an identical value is used for all elements.

The optional argument \( ds \) is only applicable to halfspace elements, and must be equal to 1 (outgoing waves) or -1 (incoming waves).

For receivers outside the soil, zero is returned.
n_tlmpsv

Purpose
Global displacement shape functions (thin layer method, P-SV-waves).

Syntax
[Nx,Nz] = n_TLMPSV(h,z)

Input arguments
h   Layer thickness (nElt × 1).
z   Receiver depth (nRec × 1).

Output arguments
Nx   Shape functions for horizontal displacement (nRec × nDOF).
Nz   Shape functions for vertical displacement (nRec × nDOF).

Description
[Nx,Nz] = n_TLMPSV(h,z) returns the displacement shape functions used in the thin layer method for P-SV-waves in a layered soil.
n_tlmsh

**Purpose**
Global displacement shape functions (thin layer method, SH-waves).

**Syntax**

\[ Ny = N_{\text{TLM SH}}(h,z) \]

**Input arguments**
- \( h \)  Layer thickness \((n_{\text{Elt}} \times 1)\) or \((1 \times 1)\).
- \( z \)  Receiver depth \((n_{\text{Rec}} \times 1)\).

**Output arguments**
- \( Ny \)  Shape functions \((n_{\text{Rec}} \times n_{\text{DOF}})\).

**Description**

\[ Ny = N_{\text{TLM SH}}(h,z) \] returns the displacement shape functions used in the thin layer method for SH-waves in a layered soil.
plotprofile

Purpose
Plot a property of a layered medium as a function of depth.

Syntax
PLOTPROFILE(h,c)
PLOTPTOFIL(h,c,...)
H = PLOTPROFILE(...)  

Input arguments
h    Layer thickness, INF for a halfspace (nElt × 1) or (1 × 1).
c    Layer properties (nElt × 1) or (1 × 1).

Output arguments
H    Handle to the plotted line.

Description
PLOTPROFILE(h,c) plots the property c of a layered medium as a function of depth. The layer thicknesses are given by h.
PLOTPROFILE(h,c,...) redirects the additional arguments to the PLOT function.
H = PLOTPROFILE(...) returns a handle to the plotted line.
rayleigh_domk

Purpose
Dominant Rayleigh wave velocity and attenuation.

Syntax

[C,A] = RAYLEIGH_DOMK(h,Cs,Cp,Ds,Dp,rho,f)
[...] = RAYLEIGH_DOMK(...,ParamName,ParamValue)

Input arguments

h   Layer thickness, INF for a halfspace (nElt × 1) or (1 × 1).
    If h(end) ≠ INF, the bottom interface is clamped.
Cs  Shear wave velocity (nElt × 1) or (1 × 1).
Cp  Dilatational wave velocity (nElt × 1) or (1 × 1).
Ds  Shear damping ratio (nElt × 1) or (1 × 1).
Dp  Dilatational damping ratio (nElt × 1) or (1 × 1).
rho Density (nElt × 1) or (1 × 1).
f   Frequency (nFreq × 1).

Output arguments

C   Phase velocity of the dominant Rayleigh wave (nFreq × 1).
A   Attenuation coefficient of the dominant Rayleigh wave (nFreq × 1).

Description

[C,A] = RAYLEIGH_DOMK(h,Cs,Cp,Ds,Dp,rho,f) estimates the phase velocity C
and the attenuation coefficient A of the dominant Rayleigh wave in a layered soil
excited by a vertical force at the surface. Both C and A are derived from the
modulus of the Green’s function \(\tilde{u}_{zz}^G\). The phase velocity C is determined from
the dominant wavenumber, i.e. the wavenumber \(k\) where the modulus of the
Green’s function \(\tilde{u}_{zz}^G\) reaches its global maximum. The attenuation coefficient A is
determined by means of the half power bandwidth method.

The number of elements nElt is equal to the maximum of the lengths of h, Cs,
Cp, Ds, Dp, and rho. If any of these parameters is defined as a scalar, an identical
value is used for all elements. The frequencies f and fStep (defined below) are in
cycles per unit time, not in radians per unit time.

The dominant wavenumber is identified by means of a global optimization
procedure that maximizes \(|\tilde{u}_{zz}^G(f,kd)|\), where \(kd = k/\omega \times \min(Cs)\) is the
dimensionless wavenumber. At specific frequencies, a full scan of the objective
function is performed in the dimensionless wavenumber interval \([0, 1.5]\) to identify the global maximum. Starting from these maxima, the global maxima at intermediate frequencies are identified by means of a local optimization scheme that searches along the \(k_d\)-axis.

\[
[... \text{] = RAYLEIGH\_DOMK(...,ParamName,ParamValue)}\]

changes the default values of the following parameters that control the optimization procedure:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k_d_\text{Tol})</td>
<td>(10^{-4})</td>
<td>Accuracy in terms of (k_d).</td>
</tr>
<tr>
<td>(k_d_\text{Res})</td>
<td>(10^{-2})</td>
<td>Resolution of the full scan for the global maximum.</td>
</tr>
<tr>
<td>(f_\text{Step})</td>
<td>5</td>
<td>Interval between freqs where a full scan is performed.</td>
</tr>
<tr>
<td>(h_\text{pLevel})</td>
<td>0.99</td>
<td>Relative amplitude for half power bandwidth method.</td>
</tr>
</tbody>
</table>

**Example**

Example 8.3: Dominant surface wave in a layered halfspace (p. 98).
wavemovie_cyl

Purpose
Animate a wave field in cylindrical coordinates.

Syntax
WAVEMOVIE_CYL(r,theta,z,ur,ut,uz,c)
WAVEMOVIE_CYL(r,theta,z,u,c)
WAVEMOVIE_CYL(...,ParamName,ParamValue)
[ds,cs] = WAVEMOVIE_CYL(...)

Input arguments
r   Vertex r-coordinates (n1 x n2 x n3) or (n1 x 1).
theta Vertex θ-coordinates (n1 x n2 x n3) or (n2 x 1).
z   Vertex z-coordinates (n1 x n2 x n3) or (n3 x 1).
ur   Mesh displacements in r-direction (n1 x n2 x n3 x nt).
ut   Mesh displacements in θ-direction (n1 x n2 x n3 x nt).
uz   Mesh displacements in z-direction (n1 x n2 x n3 x nt).
u   Mesh displacements (2 x n1 x n2 x n3 x nt) or (3 x n1 x n2 x n3 x nt).
c   Scalar wave field (n1 x n2 x n3 x nt). Default: √(ur² + ut² + uz²).

Output arguments
ds   Deformation scale used.
cs   Color scale used.

Description
WAVEMOVIE_CYL(r,theta,z,ur,ut,uz,c) creates a color movie of a harmonic or transient scalar wave field c(r,theta,z) on a deforming mesh. The mesh displacements are given by the vector field (ur,ut,uz). The mesh and the deformation are defined in cylindrical coordinates.

WAVEMOVIE_CYL(r,theta,z,u,c) is an alternative syntax for a 2-D wave field u = (ur,uz) or 3-D wave field u = (ur,ut,uz).

For a transient wave field, ur, ut, uz, and c must be real-valued matrices, and nt is the number of time steps.

For a harmonic wave field, ur, ut, uz, and c may be complex-valued, and nt must be equal to 1.
If one or more of the dimensions of $ur$, $ut$, $uz$, or $c$ is equal to 1 instead of $n1$, $n2$, $n3$, or $nt$, this function attempts to apply `REPMAT` to $ur$, $ut$, $uz$, or $c$ in order to generate an $(n1 \times n2 \times n3 \times nt)$ matrix.

`WAVEMOVIE_CYL(...,ParamName,ParamValue)` sets the value of the specified parameters. The following parameters can be specified:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AviFile</td>
<td>-</td>
<td>Save animation as an AVI file with the specified name.</td>
</tr>
<tr>
<td>AnimType</td>
<td>'auto'</td>
<td>Animation type, 'transient' or 'harmonic'.</td>
</tr>
<tr>
<td>nCycle</td>
<td>1</td>
<td>Number of cycles for harmonic animations.</td>
</tr>
<tr>
<td>nAngle</td>
<td>36</td>
<td>Number of frames per cycle for harmonic animations.</td>
</tr>
<tr>
<td>DefScale</td>
<td>'auto'</td>
<td>Deformation scale.</td>
</tr>
<tr>
<td>ColorScale</td>
<td>'auto'</td>
<td>Two-element vector specifying the color scale.</td>
</tr>
</tbody>
</table>

Additional parameters are redirected to either the `SURF` function or the `AVIFILE` function, as appropriate.

`[ds,cs] = WAVEMOVIE_CYL(...)` returns the deformation scale $ds$ and the color scale $cs$. 
**wavemovie_rec**

**Purpose**
Animate a wave field in Cartesian coordinates.

**Syntax**

```
WAVEMOVIE_REC(x,y,z,ux,uy,uz,c)
WAVEMOVIE_REC(x,y,z,u,c)
WAVEMOVIE_REC(...,ParamName,ParamValue)
[ds,cs] = WAVEMOVIE_REC(...)
```

**Input arguments**

- **x**: Vertex x-coordinates \((n1 \times n2 \times n3)\) or \((n1 \times 1)\).
- **y**: Vertex y-coordinates \((n1 \times n2 \times n3)\) or \((n2 \times 1)\).
- **z**: Vertex z-coordinates \((n1 \times n2 \times n3)\) or \((n3 \times 1)\).
- **ux**: Mesh displacements in x-direction \((n1 \times n2 \times n3 \times nt)\).
- **uy**: Mesh displacements in y-direction \((n1 \times n2 \times n3 \times nt)\).
- **uz**: Mesh displacements in z-direction \((n1 \times n2 \times n3 \times nt)\).
- **u**: Mesh displacements \((2 \times n1 \times n2 \times n3 \times nt)\) or \((3 \times n1 \times n2 \times n3 \times nt)\).
- **c**: Scalar wave field \((n1 \times n2 \times n3 \times nt)\). Default: \(\sqrt{ux^2 + uy^2 + uz^2}\).

**Output arguments**

- **ds**: Deformation scale used.
- **cs**: Color scale used.

**Description**

`WAVEMOVIE_REC(x,y,z,ux,uy,uz,c)` creates a color movie of a harmonic or transient scalar wave field \(c(x,y,z)\) on a deforming mesh. The mesh displacements are given by the vector field \((ux, uy, uz)\). The mesh and the deformation are defined in Cartesian coordinates.

`WAVEMOVIE_REC(x,y,z,u,c)` is an alternative syntax for a 2-D wave field \(u = (ux, uz)\) or 3-D wave field \(u = (ux, uy, uz)\).

For a transient wave field, \(ux, uy, uz,\) and \(c\) must be real-valued matrices, and \(nt\) is the number of time steps.

For a harmonic wave field, \(ux, uy, uz,\) and \(c\) may be complex-valued, and \(nt\) must be equal to 1.
If one or more of the dimensions of \( ux, uy, uz, \) or \( c \) is equal to 1 instead of \( n1, n2, n3, \) or \( nt \), this function attempts to apply \texttt{REPMAT} to \( ux, uy, uz, \) or \( c \) in order to generate an \((n1 \times n2 \times n3 \times nt)\) matrix.

\texttt{WAVEMOVIE\_REC(...,ParamName,ParamValue)} sets the value of the specified parameters. The following parameters can be specified:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AviFile</td>
<td>-</td>
<td>Save animation as an AVI file with the specified name.</td>
</tr>
<tr>
<td>AnimType</td>
<td>'auto'</td>
<td>Animation type, 'transient' or 'harmonic'.</td>
</tr>
<tr>
<td>nCycle</td>
<td>1</td>
<td>Number of cycles for harmonic animations.</td>
</tr>
<tr>
<td>nAngle</td>
<td>36</td>
<td>Number of frames per cycle for harmonic animations.</td>
</tr>
<tr>
<td>DefScale</td>
<td>'auto'</td>
<td>Deformation scale.</td>
</tr>
<tr>
<td>ColorScale</td>
<td>'auto'</td>
<td>Two-element vector specifying the color scale.</td>
</tr>
</tbody>
</table>

Additional parameters are redirected to either the \texttt{SURF} function or the \texttt{AVIFILE} function, as appropriate.

\([ds, cs] = \texttt{WAVEMOVIE\_REC(...)}\) returns the deformation scale \( ds \) and the color scale \( cs \).
## waveplot_cyl

**Purpose**
Plot a wave field in cylindrical coordinates.

**Syntax**

```
WAVEPLOT_CYL(r, theta, z, ur, ut, uz, c)
WAVEPLOT_CYL(r, theta, z, u, c)
WAVEPLOT_CYL(..., ParamName, ParamValue)
[ds, cs, h] = WAVEPLOT_CYL(...)
```

**Input arguments**

- `r` Vertex `r`-coordinates (`n1 × n2 × n3`) or (`n1 × 1`).
- `theta` Vertex `θ`-coordinates (`n1 × n2 × n3`) or (`n2 × 1`).
- `z` Vertex `z`-coordinates (`n1 × n2 × n3`) or (`n3 × 1`).
- `ur` Mesh displacements in `r`-direction (`n1 × n2 × n3`).
- `ut` Mesh displacements in `θ`-direction (`n1 × n2 × n3`).
- `uz` Mesh displacements in `z`-direction (`n1 × n2 × n3`).
- `u` Mesh displacements (`2 × n1 × n2 × n3`) or (`3 × n1 × n2 × n3`).
- `c` Scalar wave field (`n1 × n2 × n3`). Default: $\sqrt{ur^2 + ut^2 + uz^2}$.

**Output arguments**

- `ds` Deformation scale used.
- `cs` Color scale used.
- `h` Handle to the `SURF` object(s).

**Description**

`WAVEPLOT_CYL(r, theta, z, ur, ut, uz, c)` creates a color plot of a scalar wave field `c(r, theta, z)` on a deformed mesh. The mesh displacements are given by the vector field `(ur, ut, uz)`. The mesh and the deformation are defined in cylindrical coordinates.

`WAVEPLOT_CYL(r, theta, z, u, c)` is an alternative syntax for a 2-D wave field `u = (ur, uz)` or 3-D wave field `u = (ur, ut, uz)`. If one or more of the dimensions of `ur`, `ut`, `uz`, or `c` is equal to 1 instead of `n1`, `n2`, or `n3`, this function attempts to apply `REPMAT` to `ur`, `ut`, `uz`, or `c` in order to generate an `(n1 × n2 × n3)` matrix.

`WAVEPLOT_CYL(..., ParamName, ParamValue)` sets the value of the specified parameters. The following parameters can be specified:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DefScale</td>
<td>'auto'</td>
<td>Deformation scale.</td>
</tr>
<tr>
<td>ColorScale</td>
<td>'auto'</td>
<td>Two-element vector specifying the color scale.</td>
</tr>
</tbody>
</table>

Additional parameters are redirected to the `SURF` function.

```
[ds,cs,h] = WAVEPLOT_CYL(...) returns the deformation scale ds, the color scale cs, and a handle h to the SURF object(s).
```
waveplot_rec

Purpose
Plot a wave field in Cartesian coordinates.

Syntax
WAVEPLOT_REC(x,y,z,ux,uy,uz,c)
WAVEPLOT_REC(x,y,z,u,c)
WAVEPLOT_REC(...,ParamName,ParamValue)
[ds,cs,h] = WAVEPLOT_REC(...)

Input arguments
x  Vertex x-coordinates (n1 x n2 x n3) or (n1 x 1).
y  Vertex y-coordinates (n1 x n2 x n3) or (n2 x 1).
z  Vertex z-coordinates (n1 x n2 x n3) or (n3 x 1).
ux Mesh displacements in x-direction (n1 x n2 x n3).
uy Mesh displacements in y-direction (n1 x n2 x n3).
uz Mesh displacements in z-direction (n1 x n2 x n3).
u  Mesh displacements (2 x n1 x n2 x n3) or (3 x n1 x n2 x n3).
c  Scalar wave field (n1 x n2 x n3). Default: \sqrt{ux^2 + uy^2 + uz^2}.

Output arguments
ds  Deformation scale used.
cs  Color scale used.
h  Handle to the SURF object(s).

Description
WAVEPLOT_REC(x,y,z,ux,uy,uz,c) creates a color plot of a scalar wave field c(x, y, z) on a deformed mesh. The mesh displacements are given by the vector field (ux, uy, uz). The mesh and the deformation are defined in Cartesian coordinates.

WAVEPLOT_REC(x,y,z,u,c) is an alternative syntax for a 2-D wave field u = (ux, uz) or 3-D wave field u = (ux, uy, uz).

If one or more of the dimensions of ux, uy, uz, or c is equal to 1 instead of n1, n2, or n3, this function attempts to apply REPMAT to ux, uy, uz, or c in order to generate an (n1 x n2 x n3) matrix.

WAVEPLOT_REC(...,ParamName,ParamValue) sets the value of the specified parameters. The following parameters can be specified:
Parameter | Default | Description
--- | --- | ---
DefScale | 'auto' | Deformation scale.
ColorScale | 'auto' | Two-element vector specifying the color scale.

Additional parameters are redirected to the SURF function.

\[
\text{[ds,cs,h]} = \text{WAVEPLOT,REC(...)} \text{ returns the deformation scale } \text{ds}, \text{ the color scale } \text{cs}, \text{ and a handle } \text{h to the SURF object(s)}.\]

**Examples**

Example 9.1: The response of a halfspace due to a harmonic point load (p. 104).
Example 9.2: The response of a halfspace due to an impulsive point load (p. 105).
wiggle

Purpose
Plot wiggle traces.

Syntax
WIGGLE(x,t,u)
WIGGLE(...,ParamName,ParamValue)
[s,h] = WIGGLE(...)

Input arguments
x    Spatial coordinates for all channels (M × 1).
t    Time axis (N × 1).
u    Data to plot (N × M).

Output arguments
s    Deformation scale used for all traces.
h    Handles to the plotted lines (row 1) and patches (rows 2 and 3).

Description
WIGGLE(x,t,u) plots the signals in the columns of u, corresponding to the spatial coordinates x, as wiggle traces centered around their x-coordinates.
WIGGLE(...,ParamName,ParamValue) sets the value of the specified parameters.
The following parameters can be specified:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defscale</td>
<td>1.5/MAX(MAX(ABS(u)))</td>
<td>Deformation scale.</td>
</tr>
<tr>
<td>PosColor</td>
<td>[0 0 0]</td>
<td>Color of the positive part of the traces.</td>
</tr>
<tr>
<td>NegColor</td>
<td>'none'</td>
<td>Color of the negative part of the traces.</td>
</tr>
</tbody>
</table>

Additional parameters are redirected to the PLOT function.

If the deformation scale is a scalar, all channels are plotted on the same scale. By default, the scalar value 1.5/MAX(MAX(ABS(u))) is used. Specify e.g. 1.5./MAX(ABS(u)) to normalize all channels individually.

[s,h] = WIGGLE(...) returns the deformation scale s for all traces and a matrix h with handles to the plotted lines (row 1) and patches (rows 2 and 3).
Example
Example 4.5: Vertical transient wave propagation in a layer on a halfspace (p. 47).
References


REFERENCES


