PLAXIS

PLAXIS 3D 2024.1

Scientific Manual 3D

Bentley®
Advancing Infrastructure

Last Updated: December 29, 2023

Table of Contents

Chapter 1: Introduction 4

Chapter 2: Deformation theory 5

2.1 Basic equations of continuum deformation 5
2.2 Finite element discretisation 6
2.3 Implicit integration of differential plasticity models 7
2.4 Global iterative procedure 8

Chapter 3: Groundwater flow theory 10

3.1 Basic equations of flow 10
3.1.1 Transient flow 10
3.1.2 Continuity equation 11
3.1.3 Hydraulic gradient 12
3.2 Boundary Conditions 12
3.3 Finite element discretisation 15
3.4 Flow in interface elements 16

Chapter 4: Consolidation theory 17

4.1 Basic equations of consolidation 17
4.2 Finite element discretisation 18
4.3 Elastoplastic consolidation 20
4.4 Critical time step 20

Chapter 5: Dynamics 22

5.1 Basic equation dynamic behaviour 22
5.2 Time integration 23
5.2.1 Implementation of the integration scheme 23
5.2.2 Recommended maximum time step 24
5.2.3 Dynamic integration coefficients 24
5.3 Model Boundaries 25
5.3.1 Viscous boundaries 25
5.3.2 Free-field and compliant base boundaries 25
5.4 Initial stresses and stress increments 27
5.5 Amplification of responses 27
5.6 Pseudo-spectral acceleration response spectrum for a single-degree-of-freedom system 27
5.7 Natural frequency of vibration of a soil deposit 28
5.8 Hydrodynamic pressure 29
5.8.1 Added mass approach 29
5.8.2 Implementation of Zangar's added mass 30

Chapter 6: Element formulations 31

6.1 Interpolation functions of point elements 31
6.1.1 Structural elements 31
6.2 Interpolation functions and numerical integration of line elements 32
6.2.1 Interpolation functions of line elements 32

6.2.2 Structural elements 35
6.2.3 Derivatives of interpolation functions 37
6.2.4 Numerical integration of line elements 39
6.2.5 Calculation of element stiffness matrix 40
6.3 Interpolation functions and numerical integration of area elements 41
6.3.1 Interpolation functions of area elements 41
6.3.2 Structural elements 43
6.3.3 Numerical integration of area elements 44
6.4 Interpolation functions and numerical integration of volume elements 45
6.4.1 10-node tetrahedral element 45
6.4.2 Derivatives of interpolation functions 47
6.4.3 Numerical integration of volume elements 48
6.4.4 Calculation of element stiffness matrix 48
6.5 Special elements (PLAXIS 3D) 49
6.6 Embedded beams (PLAXIS 3D) 49
6.6.1 Finite element discretisation 50
6.6.2 Interaction at the skin 50
6.6.3 Interaction at the foot 51

Chapter 7: Convergence of non-linear calculations in PLAXIS 53

7.1 Convergence criteria for deformation analysis 53
7.1.1 Global error criteria 54
7.1.2 Local error criteria 55
7.1.3 Integrated local error - Accuracy in the plastic zone 60
7.2 Convergence criteria for flow analysis 61
7.2.1 Global error criteria 62
7.2.2 Local error criteria 64
7.2.3 Special case: Steady-state calculation criteria 65

Chapter 8: Reference 68

Appendices 70

Appendix A: Symbols 71
Appendix B: Calculation Process 73
Appendix C: Understanding the iterative convergence process 75
C.1 Quasi-Newton Raphson method 75

1 Introduction

In this part of the manual some scientific background is given of the theories and numerical methods on which the PLAXIS program is based. The manual contains chapters on deformation theory, groundwater flow theory (PLAXIS 2D), consolidation theory, dynamics as well as the corresponding finite element formulations and integration rules for the various types of elements used in PLAXIS. In Calculation Process (on page 73) a global calculation scheme is provided for a plastic deformation analysis.

In addition to the specific information given in this part of the manual, more information on backgrounds of theory and numerical methods can be found in the literature, as amongst others referred to in Reference Manual. For detailed information on stresses, strains, constitutive modelling and the types of soil models used in the PLAXIS program, the reader is referred to the Material Models Manual.

2 Deformation theory

In this chapter the basic equations for the static deformation of a soil body are formulated within the framework of continuum mechanics. A restriction is made in the sense that deformations are considered to be small. This enables a formulation with reference to the original undeformed geometry. The continuum description is discretised according to the finite element method.

2.1 Basic equations of continuum deformation

The static equilibrium of a continuum can be formulated as:

$$\mathbf{L}^T \underline{\sigma} + \underline{b} = \underline{0} \qquad \textit{Eq. [1]}$$

This equation relates the spatial derivatives of the six stress components, assembled in vector $\underline{\sigma}$, to the three components of the body forces, assembled in vector $\underline{b}$. $\mathbf{L}^T$ is the transpose of a differential operator, defined as:

$$\mathbf{L}^T = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial x} & \frac{\partial}{\partial z} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \qquad \textit{Eq. [2]}$$

In addition to the equilibrium equation, the kinematic relation can be formulated as:

$$\underline{\varepsilon} = \mathbf{L}\underline{u} \qquad \textit{Eq. [3]}$$

This equation expresses the six strain components, assembled in vector $\underline{\varepsilon}$, as the spatial derivatives of the three displacement components, assembled in vector $\underline{u}$, using the previously defined differential operator $\mathbf{L}$. The link between Eq. [1] and Eq. [3] is formed by a constitutive relation representing the material behaviour. Constitutive relations, i.e. relations between rates of stress and strain, are extensively discussed in the Reference Manual. The general relation is repeated here for completeness:

$$\dot{\underline{\sigma}} = \mathbf{M}\dot{\underline{\varepsilon}} \qquad \textit{Eq. [4]}$$

The combination of Eq. [1], Eq. [3] and Eq. [4] would lead to a second-order partial differential equation in the displacements $\underline{u}$.

However, instead of a direct combination, the equilibrium equation is reformulated in a weak form according to Galerkin's variation principle:

$$\int \delta \underline{u}^T \left(\mathbf{L}^T \underline{\sigma} + \underline{b} \right) dV = 0 \qquad \textit{Eq. [5]}$$

In this formulation $\delta\underline{u}$ represents a kinematically admissible variation of displacements. Applying Green's theorem for partial integration to the first term in Eq. [5] leads to:

$$\int \delta\underline{\varepsilon}^T \underline{\sigma} dV = \int \delta\underline{u}^T \underline{b} dV + \int \delta\underline{u}^T \underline{t} dS \qquad \textit{Eq. [6]}$$

This introduces a boundary integral in which the boundary traction appears. The three components of the boundary traction are assembled in the vector $\underline{t}$. Eq. [6] is referred to as the virtual work equation.

The development of the stress state $\underline{\sigma}$ can be regarded as an incremental process:

$$\sigma^i = \sigma^{i\text{-}1} + \Delta\sigma \qquad \Delta\sigma = \int \dot{\sigma} dt \qquad \textit{Eq. [7]}$$

In this relation $\underline{\sigma}^i$ represents the actual state of stress which is unknown and $\underline{\sigma}^{i-1}$ represents the previous state of stress which is known. The stress increment $\Delta\underline{\sigma}$ is the stress rate integrated over a small time increment.

If Eq. [6] is considered for the actual state i, the unknown stresses $\underline{\sigma}^i$ can be eliminated using Eq. [7]:

$$\int \delta\underline{\varepsilon}^T \Delta\underline{\sigma} dV = \int \delta\underline{u}^T \underline{b}^i dV + \int \delta\underline{u}^T \underline{t}^i dS - \int \delta\underline{\varepsilon}^T \underline{\sigma}^{i-1} dV \qquad \textit{Eq. [8]}$$

It should be noted that all quantities appearing in Eq. [1] till Eq. [8] are functions of the position in the three-dimensional space.

2.2 Finite element discretisation

According to the finite element method a continuum is divided into a number of (volume) elements. Each element consists of a number of nodes. Each node has a number of degrees of freedom that correspond to discrete values of the unknowns in the boundary value problem to be solved. In the present case of deformation theory the degrees of freedom correspond to the displacement components. Within an element the displacement field $\underline{u}$ is obtained from the discrete nodal values in a vector $\underline{v}$ using interpolation functions assembled in matrix **N** :

$$\underline{u} = \mathbf{N}\underline{v} \qquad \textit{Eq. [9]}$$

The interpolation functions in matrix **N** are often denoted as shape functions. Substitution of Eq. [9] in the kinematic relation Eq. [3] gives:

$$\underline{\varepsilon} = \mathbf{L}\mathbf{N}\underline{v} = \mathbf{B}\underline{v} \qquad \textit{Eq. [10]}$$

In this relation **N** is the strain interpolation matrix, which contains the spatial derivatives of the interpolation functions. Eq. [9] and Eq. [10] can be used in variational, incremental and rate form as well.

Eq. [8] can now be reformulated in discretised form as:

$$\int (\mathbf{B}\delta\underline{v})^T \Delta\underline{\sigma}\ d\ V = \int (\mathbf{N}\delta\underline{v})^T \underline{b}^i dV + \int (\mathbf{N}\delta\underline{v})^T \underline{t}^i dS - \int (\mathbf{B}\delta\underline{v})^T \underline{\sigma}^{i-1} dV \qquad \textit{Eq. [11]}$$

The discrete displacements can be placed outside the integral:

$$\delta\underline{v} \int \mathbf{B}^T \Delta\underline{\sigma}\ d\ V = \delta\underline{v} \int \mathbf{N}^T \underline{b}^i dV + \delta\underline{v} \int \mathbf{N}^T \underline{t}^i dS - \delta\underline{v} \int \mathbf{B}^T \underline{\sigma}^{i-1} dV \qquad \textit{Eq. [12]}$$

Provided that Eq. [12] holds for any kinematically admissible displacement variation $\delta\underline{v}^T$, the equation can be written as:

$$\int \mathbf{B}^T \Delta\underline{\sigma}\ d\ V = \int \mathbf{N}^T \underline{b}^i dV + \int \mathbf{N}^T \underline{t}^i dS - \int \mathbf{B}^T \underline{\sigma}^{i-1} dV \qquad \textit{Eq. [13]}$$

The Eq. [13] is the elaborated equilibrium condition in discretised form. The first term on the right-hand side together with the second term represent the current external force vector and the last term represents the internal reaction vector from the previous step. A difference between the external force vector and the internal reaction vector should be balanced by a stress increment $\Delta\underline{\sigma}$.

The relation between stress increments and strain increments is usually non-linear. As a result, strain increments can generally not be calculated directly, and global iterative procedures are required to satisfy the equilibrium condition (Eq. [12]) for all material points. Global iterative procedures are described later in Global iterative procedure (on page 8), but the attention is first focused on the (local) integration of stresses.

2.3 Implicit integration of differential plasticity models

The stress increments $\Delta\underline{\sigma}$ are obtained by integration of the stress rates according to Eq. [7]. For differential plasticity models the stress increments can generally be written as:

$$\Delta\underline{\sigma} = \mathbf{D}^e \left(\Delta\underline{\varepsilon} - \Delta\underline{\varepsilon}^p\right) \qquad \textit{Eq. [14]}$$

In this relation $\mathbf{D}^e$ represents the elastic material matrix for the current stress increment. The strain increments $\Delta\underline{\varepsilon}$ are obtained from the displacement increments $\Delta\underline{u}$ using the strain interpolation matrix $\mathbf{B}$, similar to Eq. [10].

For elastic material behaviour, the plastic strain increment $\Delta\underline{\varepsilon}^p$ is zero. For plastic material behaviour, the plastic strain increment can be written, according to Vermmer (1979) (on page 69), as:

$$\Delta\underline{\varepsilon}^p = \Delta\lambda\left[(1-\omega)\left(\frac{\partial g}{\partial \underline{\sigma}}\right)^{i-1} + \omega\left(\frac{\partial g}{\partial \underline{\sigma}}\right)^{i}\right] \qquad \textit{Eq. [15]}$$

In this equation $\Delta\lambda$ is the increment of the plastic multiplier and ω is a parameter indicating the type of time integration. For $\omega = 0$ the integration is called explicit and for $\omega = 1$ the integration is called implicit.

Vermeer (1979) (on page 69) has shown that the use of implicit integration ($\omega = 1$) has some major advantages, as it overcomes the requirement to update the stress to the yield surface in the case of a transition from elastic to elastoplastic behaviour. Moreover, it can be proven that implicit integration, under certain conditions, leads to a symmetric and positive differential matrix $\partial\underline{\varepsilon} / \partial\underline{u}$, which has a positive influence on iterative procedures. Because of these major advantages, restriction is made here to implicit integration and no attention is given to other types of time integration.

Hence, for $\omega = 1$ Eq. [15] reduces to:

$$\Delta\underline{\varepsilon}^p = \Delta\lambda\left(\frac{\partial g}{\partial \underline{\sigma}}\right)^{i-1} \qquad \textit{Eq. [16]}$$

Substitution of Eq. [16] into Eq. [14] and successively into Eq. [7] gives:

$$\underline{\sigma}^i = \underline{\sigma}^{tr} - \Delta\lambda\mathbf{D}^e\left(\frac{\partial g}{\partial \underline{\sigma}}\right)^i \quad \text{with:} \quad \underline{\sigma}^{tr} = \underline{\sigma}^{i-1} + \mathbf{D}^e\Delta\underline{\varepsilon} \qquad \textit{Eq. [17]}$$

In this relation $\underline{\sigma}^{tr}$ is an auxiliary stress vector, referred to as the elastic stresses or trial stresses, which is the new stress state when considering purely linear elastic material behaviour.

The increment of the plastic multiplier $\Delta\lambda$, as used in Eq. [17], can be solved from the condition that the new stress state has to satisfy the yield condition:

$$f\left(\underline{\sigma}^i\right) = 0 \qquad \textit{Eq. [18]}$$

For perfectly-plastic and linear hardening models the increment of the plastic multiplier can be written as:

$$\Delta\lambda = \frac{f(\underline{\sigma}^{tr})}{d+h} \qquad Eq.\ [19]$$

where:

$$d = \left(\frac{\partial f}{\partial \underline{\sigma}}\right)^{tr} \mathbf{D}^{e} \left(\frac{\partial g}{\partial \underline{\sigma}}\right)^{i} \qquad Eq.\ [20]$$

The symbol h denotes the hardening parameter, PLAXIS which is zero for perfectly-plastic models and constant for linear hardening models. In the latter case the new stress state can be formulated as:

$$\underline{\sigma}^{i} = \underline{\sigma}^{tr} - \frac{\langle f(\underline{\sigma}^{tr})\rangle}{d+h} \mathbf{D}^{e} \left(\frac{\partial g}{\partial \underline{\sigma}}\right)^{i} \qquad Eq.\ [21]$$

The $\langle\rangle$ -brackets are referred to as McCauley brackets, which have the following convention:

$$\langle x\rangle = 0 \quad for \quad x \leq 0 \quad and \quad \langle x\rangle = x \quad for \quad x > 0$$

For non-linear hardening models the increment of the plastic multiplier is obtained using a Newton-type iterative procedure with convergence control.

2.4 Global iterative procedure

Substitution of the relationship between increments of stress and increments of strain, $\Delta\underline{\sigma} = \mathbf{M}\Delta\underline{\varepsilon}$, into the equilibrium equation (Eq. [13]) leads to:

$$\mathbf{K}^{i}\Delta\underline{u}^{i} = \underline{f}_{ex}^{i} - \underline{f}_{in}^{i} \qquad Eq.\ [22]$$

In this equation $\mathbf{K}$ is a stiffness matrix, $\Delta\underline{u}$ is the incremental displacement vector, $\underline{f}_{ex}^{i}$ is the external force vector and $\underline{f}_{in}^{i}$ is the internal reaction vector. The superscript i refers to the step number. However, because the relation between stress increments and strain increments is generally non-linear, the stiffness matrix cannot be formulated exactly beforehand. Hence, a global iterative procedure is required to satisfy both the equilibrium condition and the constitutive relation. The global iteration process can be written as:

$$\mathbf{K}^{i}\delta\underline{u}^{j} = \underline{f}_{ex}^{i} - \underline{f}_{in}^{j-1} \qquad Eq.\ [23]$$

The superscript j refers to the iteration number. $\delta\underline{u}$ is a vector containing sub-incremental displacements, which contribute to the displacement increments of step i :

$$\Delta\underline{u}^{i} = \sum_{j=1}^{n} \delta\underline{u}^{j} \qquad Eq.\ [24]$$

where n is the number of iterations within step i. The stiffness matrix $\mathbf{K}$, as used in Eq. [23], represents the material behaviour in an approximated manner. The more accurate the stiffness matrix, the fewer iterations are required to obtain equilibrium within a certain tolerance.

In its simplest form $\mathbf{K}$ represents a linear-elastic response. In this case the stiffness matrix can be formulated as:

$$\mathbf{K} = \int \mathbf{B}^{T}\mathbf{D}^{e}\mathbf{B}dV \quad (elastic\ stiffness\ matrix) \qquad Eq.\ [25]$$

where $\mathbf{D}^{e}$ is the elastic material matrix according to Hooke's law and $\mathbf{B}$ is the strain interpolation matrix. The use of an elastic stiffness matrix gives a robust iterative procedure as long as the material stiffness does not increase,

even when using non-associated plasticity models. Special techniques such as arc-length control (Riks, 1979) (on page 69), over-relaxation and extrapolation (Vermeer & van Langen, 1989) (on page 69) can be used to improve the iteration process. Moreover, the automatic step size procedure, as introduced by Van Langen & Vermeer (1990) (on page 69), can be used to improve the practical applicability. For material models with linear behaviour in the elastic domain, such as the standard Mohr-Coulomb model, the use of an elastic stiffness matrix is particularly favourable, as the stiffness matrix needs only be formed and decomposed before the first calculation step. This calculation procedure is summarised in Calculation Process (on page 73).

3 Groundwater flow theory

In this chapter we will review the theory of groundwater flow as used in PLAXIS. In addition to a general description of groundwater flow, attention is focused on the finite element formulation.

3.1 Basic equations of flow

3.1.1 *Transient flow*

Flow in a porous medium can be described by Darcy's law which is expressed by the following equation in three dimensions:

$$\underline{q} = \frac{\mathbf{k}}{\rho_w g}\left(\underline{\nabla} p_w + \rho_w \underline{g}\right) \qquad \textit{Eq. [26]}$$

Where

$$\underline{\nabla} = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} \qquad \textit{Eq. [27]}$$

$\underline{q}$, $\mathbf{k}$, $\underline{g}$ and ρ_w are the specific discharge (fluid velocity), the tensor of permeability, the acceleration vector due to gravity Eq. [28] and the density of water, respectively. $\underline{\nabla} p_w$ is the gradient of the water pore pressure which causes groundwater flow. The term $\rho_w \underline{g}$ is used as the flow is not affected by the gradient of the water pore pressure in vertical direction when hydrostatic conditions are assumed.

$$\underline{g} = \begin{bmatrix} 0 \\ -g \\ 0 \end{bmatrix} \qquad \textit{Eq. [28]}$$

In unsaturated soils the coefficient of permeability $\mathbf{k}$ can be related to the soil saturation as:

$$\mathbf{k} = k_{rel}\mathbf{k}^{sat} \qquad \textit{Eq. [29]}$$

where

$$\mathbf{k}^{sat} = \begin{bmatrix} k_x^{sat} & 0 & 0 \\ 0 & k_y^{sat} & 0 \\ 0 & 0 & k_z^{sat} \end{bmatrix} \qquad Eq.\ [30]$$

and k_{rel} is the ratio of the permeability at a given saturation to the permeability in saturated state $\mathbf{k}^{sat}$.

3.1.2 *Continuity equation*

The mass concentration of the water in each elemental volume of the medium is equal to $\rho_w nS$. The parameters n and S are the porosity and the degree of saturation of the soil, respectively. According to the mass conservation, the water outflow from the volume is equal to the changes in the mass concentration. As the water outflow is equal to the divergence of the specific discharge ($\underline{\nabla}^T \cdot (\underline{q})$), the continuity equation has the form (Song, 1990) (on page 69):

$$\underline{\nabla}^T \cdot (\rho_w \underline{q}) = -\frac{\partial}{\partial t}(\rho_w nS) \qquad Eq.\ [31]$$

where the specific discharge $\underline{q}$ is defined as:

$$\underline{q} = \frac{k_{rel}}{\rho_w g}\mathbf{k}^{sat}(\underline{\nabla} p_w + \rho_w \underline{g}) \qquad Eq.\ [32]$$

By neglecting the deformations of solid particles and the gradients of the density of water (Boussinesq's approximation), the continuity equation is simplified to:

$$\underline{\nabla}^T \cdot (\rho_w \underline{q}) + S\underline{m}^T \frac{\partial \underline{\varepsilon}}{\partial t} - n\left(\frac{S}{K_w} - \frac{\partial S}{\partial p_w}\right)\frac{\partial p_w}{\partial t} = 0 \qquad Eq.\ [33]$$

where

$$\underline{m}^T = [1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0] \qquad Eq.\ [34]$$

For transient groundwater flow the displacements of solid particles are neglected. Therefore:

$$\underline{\nabla}^T \cdot (\rho_w \underline{q}) - n\left(\frac{S}{K_w} - \frac{\partial S}{\partial p_w}\right)\frac{\partial p_w}{\partial t} = 0 \qquad Eq.\ [35]$$

For steady state flow ($\partial p_w / \partial t = 0$) the continuity condition applies:

$$\underline{\nabla}^T \cdot (\rho_w \underline{q}) = 0 \qquad Eq.\ [36]$$

Eq. [36] expresses that there is no net inflow or outflow in an elementary area, as illustrated in:

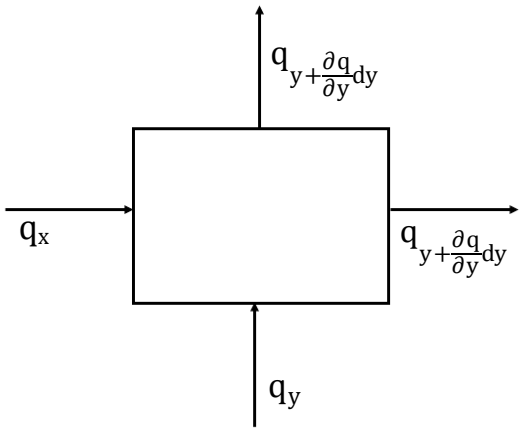


Figure 1: Illustration of continuity condition

3.1.3 *Hydraulic gradient*

Hydraulic gradient or groundwater head gradient, $\underline{i} = \underline{\nabla} h$, is a vectorial variable, such that:

$$\underline{q} = -\mathbf{k}\underline{\nabla} h \qquad \textit{Eq. [37]}$$

$\underline{\nabla} h$ is defined as follows:

$$\underline{\nabla} h = \begin{bmatrix} \frac{\partial h}{\partial x} \\ \frac{\partial h}{\partial y} \\ \frac{\partial h}{\partial z} \end{bmatrix} \qquad \textit{Eq. [38]}$$

For any particular direction:

$$i_x = \frac{dh}{dx} \qquad \textit{Eq. [39]}$$

$$q_x = -k_x \frac{dh}{dx} \qquad \textit{Eq. [40]}$$

Note that the minus sign does not explicitly appear where groundwater flow, q, is directly calculated from pore water pressure, p_w, and gravity acceleration, g, values, since both pore water pressure and the gravity acceleration vector are negative by definition.

The value of hydraulic gradient is taken equal to 0 when the relative permeability, k_{rel}, at that point is lower than 0.99. This pre-condition ensures that the hydraulic gradient is only defined for saturated soil volumes.

3.2 Boundary Conditions

The following boundary conditions are available in PLAXIS:

3.2.1 *Closed*

This type of boundary conditions specifies a zero Darcy flux over the boundary as

$$q_x n_x + q_y n_y = 0 \qquad \textit{Eq. [41]}$$

where n_x and n_y are the outward pointing normal vector components on the boundary.

3.2.2 *Inflow*

A non-zero Darcy flux over a boundary is set by a prescribed recharge value $\bar{q}$ and reads:

$$q_x n_x + q_y n_y = -\bar{q} \qquad \textit{Eq. [42]}$$

This indicates that the Darcy flux vector and the normal vector on the boundary are pointing in opposite directions.

3.2.3 *Outflow*

For outflow boundary conditions the direction of the prescribed Darcy flux, $\bar{q}$, should equal the direction of the normal on the boundary, i.e.:

$$q_x n_x + q_y n_y = \bar{q} \qquad \textit{Eq. [43]}$$

3.2.4 *Head*

For prescribed head boundaries the value of the head $\bar{h}$ (prescribed input value) is imposed as:

$$h = \bar{h} \qquad \textit{Eq. [44]}$$

Alternatively prescribed pressure conditions can be given. Overtopping conditions for example can be formulated as prescribed pressure boundaries

$$p = 0 \qquad \textit{Eq. [45]}$$

These conditions directly relate to a prescribed head boundary condition and are implemented as such.

3.2.5 *Infiltration*

This type of boundary conditions poses a more complex mixed boundary condition. An inflow value $\bar{q}$ may depend on time and as in nature the amount of inflow is limited by the capacity of the soil. If the precipitation rate exceeds this capacity, ponding takes place at a depth $\bar{h}_{p,max}$ and the boundary condition switches from inflow to prescribed head. As soon as the soil capacity meets the infiltration rate the condition switches back.

$$\begin{cases} h = z + \bar{h}_{p,max} & \textit{if ponding} \\ q_x n_x + q_y n_y = -q & \textit{if } h < z + \bar{h}_{p,max} \cap h < z + \bar{h}_{p,\min} \\ h = z + \bar{h}_{p,min} & \textit{if drying} \end{cases} \qquad \textit{Eq. [46]}$$

This boundary condition simulates evaporation for negative values of $\bar{q}$. The outflow boundary condition is limited by a minimum head $\bar{h}_{p,min}$ to ensure numerical stability.

3.2.6 *Seepage*

The water line option generates phreatic/seepage conditions by default. An external head $\bar{h}$ is prescribed on the part of the boundary beneath the water line, seepage or free conditions are applied to the rest of the line. The phreatic/seepage condition reads

$$\begin{cases} h = \bar{h} & if\ \bar{h} \geq z \\ q_x n_x + q_y n_y = 0 & if\ \bar{h} < z \cap h < z \\ h = z & if\ h < z \end{cases} \qquad Eq.\ [47]$$

The seepage condition only allows for outflow of groundwater at atmospheric pressure. For unsaturated conditions at the boundary the boundary is closed.

Alternatively a water line may generate a phreatic/closed condition if the upper part of the line is replaced by closed conditions. This condition is written as

$$\begin{cases} h = \bar{h} & if\ \bar{h} \geq z \\ Q = 0 & if\ \bar{h} < z \end{cases} \qquad Eq.\ [48]$$

The external head $\bar{h}$ may vary in a time dependent way, however the part that remains closed is derived from the initial setting.

3.2.7 *Infiltration well*

Inside the domain wells are modelled as source terms, where $\bar{Q}$ specifies the inflowing flux per meter.

$$Q = \bar{Q} \qquad Eq.\ [49]$$

As the source term in the governing equation simulates water flowing in the system, the source term is positive for a recharge well.

3.2.8 *Extraction well*

A discharge rate $\bar{Q}$ simulates an amount of water leaving the domain

$$Q = -\bar{Q} \qquad Eq.\ [50]$$

The source term in th e governing equation is negative for a discharge well.

3.2.9 *Drain*

Drains are handled as seepage boundaries. However, drains may be located inside the domain as well. The condition is written as

$$\begin{cases} h = z & if\ Q < 0 \\ Q = 0 & if\ h < z \end{cases} \qquad Eq.\ [51]$$

A drain permits water leaving the modelling domain at atmospheric pressure. The drain itself does not generate a resistance against flow.

Initial conditions are generated as a steady state solution for a problem with a given set of boundary conditions.

3.3 Finite element discretisation

The groundwater pore pressure in any position within an element can be expressed in terms of nodal values:

$$p_w = \underline{N}\,\underline{p_w}_n \qquad \textit{Eq. [52]}$$

where $\underline{N}$ is the vector with interpolation functions. For more information on the finite element theory please refer to Bathe & Koshgoftaar (1979) (on page 68), Zienkiewicz (1967) (on page 69). According to Eq. [26], the specific discharge is based on the gradient of the groundwater pore pressure. This gradient can be determined by means of the **L**-matrix, which contains the spatial derivatives of the interpolation functions, see the PLAXIS Scientific Manual. In the numerical formulation the specific discharge, $\underline{q}$, is written as:

$$\underline{q} = \frac{k_{rel}}{\gamma_w}\mathbf{k}^{sat}\left(\mathbf{B}\,\underline{p_w}_n + \rho_w \underline{g}\right) \qquad \textit{Eq. [53]}$$

where:

$$\underline{q} = \begin{bmatrix} q_x \\ q_y \end{bmatrix} \text{ and } \mathbf{k}^{sat} = \begin{bmatrix} k_x^{sat} & 0 \\ 0 & k_y^{sat} \end{bmatrix} \qquad \textit{Eq. [54]}$$

From the specific discharges in the integration points, $\underline{q}$, the nodal discharges $\underline{Q}^e$ can be integrated according to:

$$\underline{Q}^e = -\int \mathbf{k}^{sat}\underline{q}\,dV \qquad \textit{Eq. [55]}$$

in which $\mathbf{B}^T$ is the transpose of the **B**-matrix. The term dV indicates integration over the volume of the body.

Starting from the continuity equation Eq. [35] and applying the Galerkin approach and incorporating prescribed boundary conditions we obtain:

$$-\mathbf{H}\,\underline{p_w}_n - \mathbf{S}\frac{d\,\underline{p_w}_n}{dt} = \underline{q}_p \qquad \textit{Eq. [56]}$$

where **H**, **S** and $\underline{q}_p$ are the permeability matrix, the compressibility matrix and the prescribed recharges that are given by the boundary conditions, respectively:

$$\mathbf{H} = \int (\nabla \mathbf{N})^{\mathrm{T}} \frac{k_{rel}}{\gamma_w}\mathbf{k}^{sat}(\nabla \mathbf{N})\,dV \qquad \textit{Eq. [57]}$$

$$\mathbf{S} = \int \mathbf{N}^T \left(\frac{nS}{K_w} - n\frac{\partial S}{\partial p_w}\right)\mathbf{N}\,dV \qquad \textit{Eq. [58]}$$

$$\underline{q}_p = \int (\nabla \mathbf{N})^{\mathrm{T}} \frac{k_{rel}}{\gamma_w} k^{sat}\rho_w \underline{g}\,dV - \int \mathbf{N}^T \bar{q}\; d\;\Gamma \qquad \textit{Eq. [59]}$$

$\bar{q}$ is the outflow prescribed flux on the boundary. The term $d\Gamma$ indicates a surface integral.

In PlaxFlow the bulk modulus of the pore fluid is taken automatically according to:

$$\frac{K_w}{n} = \frac{3(\nu_u - \nu)}{(1 - 2\nu_u)(1 + \nu)} K_{skeleton} \qquad \textit{Eq. [60]}$$

where ν_u has a default value of 0.495. The value can be modified in the input program on the basis of Skempton's B-parameter. For material just switched on, the bulk modulus of the pore fluid is neglected.

Due to the unsaturated zone the set of equations is highly non-linear and a Picard scheme is used to solve the system of equations iteratively. The linear set is solved in incremental form using an implicit time stepping schema. Application of this procedure to Eq. [56] yields:

$$-(\alpha \Delta t \mathbf{H} + \mathbf{S}) \Delta \underline{p_w}_n = \Delta t \mathbf{H} \underline{p_w}_{n0} + \Delta t q_p \qquad \textit{Eq. [61]}$$

and $\underline{p_w}_{n0}$ denote value of water pore pressure at the beginning of a step. The parameter α is the time integration coefficient. In general the integration coefficient α can take values from 0 to 1. In PlaxFlow the fully implicit scheme of integration is used with $\alpha = 1$.

For steady state flow the governing equation is:

$$-\alpha \mathbf{H} \Delta \underline{p_w}_n = \mathbf{H} \underline{p_w}_{n0} + q_p \qquad \textit{Eq. [62]}$$

3.4 Flow in interface elements

Interface elements are treated specially in groundwater calculations. The interface elements have an active setting for the deformation calculation (soil-structure interaction) and an independent setting for flow calculations. When the interface elements are active in flow, there is a full coupling of the pore pressure degrees of freedom and the interface permeability is taken into account. When the interface elements are inactive in flow, there is no flow from one side of the interface element to the other (impermeable screen). In addition, options are available to make interface elements semi-permeable or to use them as drain elements.

4 Consolidation theory

In this chapter we will review the theory of consolidation as used in PLAXIS. In addition to a general description of Biot's theory for coupled consolidation, attention is focused on the finite element formulation. Moreover, a separate section is devoted to the use of advanced soil models in a consolidation analysis (elastoplastic consolidation).

4.1 Basic equations of consolidation

The governing equations of consolidation as used in PLAXIS follow Biot's theory (Biot, 1956 (on page 68)). Darcy's law for fluid flow and elastic behaviour of the soil skeleton are also assumed. The formulation is based on small strain theory. According to Terzaghi's principle, stresses are divided into effective stresses and pore pressures:

$$\underline{\sigma} = \underline{\sigma}' + \underline{m}(p_{steady} + p_{excess}) \qquad \textit{Eq. [63]}$$

where:

$$\underline{\sigma} = (\sigma_{xx} \quad \sigma_{yy} \quad \sigma_{zz} \quad \sigma_{xy} \quad \sigma_{yz} \quad \sigma_{zx})^T \text{ and } \underline{m} = (1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0)^T \qquad \textit{Eq. [64]}$$

$\underline{\sigma}$ is the vector with total stresses, $\underline{\sigma}'$ contains the effective stresses, p_{excess} is the excess pore pressure and $\underline{m}$ is a vector containing unity terms for normal stress components and zero terms for the shear stress components. The steady state solution at the end of the consolidation process is denoted as p_{steady}. Within PLAXIS p_{steady} is defined as:

$$p_{steady} = p_{input} \qquad \textit{Eq. [65]}$$

where p_{input} is the pore pressure generated in the input program based on phreatic lines after the use of the K0 procedure or Gravity loading.

Note that within PLAXIS compressive stresses are considered to be negative; this applies to effective stresses as well as to pore pressures. In fact it would be more appropriate to refer to p_{excess} and p_{steady} as pore stresses, rather than pressures. However, the term pore pressure is retained, although it is positive for tension.

The constitutive equation is written in incremental form. Denoting an effective stress increment as $\dot{\underline{\sigma}}'$ and a strain increment as $\dot{\underline{\varepsilon}}$, the constitutive equation is:

$$\dot{\underline{\sigma}}' = \mathbf{M}\dot{\underline{\varepsilon}} \qquad \textit{Eq. [66]}$$

where:

$$\dot{\underline{\varepsilon}} = (\dot{\varepsilon}_{xx} \quad \dot{\varepsilon}_{yy} \quad \dot{\varepsilon}_{zz} \quad \dot{\gamma}_{xy} \quad \dot{\gamma}_{yz} \quad \dot{\gamma}_{zx})^T \qquad \textit{Eq. [67]}$$

and **M** represents the material stiffness matrix. For details on constitutive relations, see the Material Models Manual.

4.2 Finite element discretisation

To apply a finite element approximation we use the standard notation:

$$\underline{u} = \mathbf{N}\underline{v} \quad p = \mathbf{N}p_n \quad \underline{\varepsilon} = \mathbf{B}\underline{v} \qquad \textit{Eq. [68]}$$

where $\underline{v}$ is the nodal displacement vector, $\underline{p}_n$ is the nodal excess pore pressure vector, $\underline{u}$ is the continuous displacement vector within an element and $\underline{p}$ is the (excess) pore pressure. The matrix **N** contains the interpolation functions and **B** is the strain interpolation matrix.

In general the interpolation functions for the displacements may be different from the interpolation functions for the pore pressure. In PLAXIS, however, the same functions are used for displacements and pore pressures.

Starting from the incremental equilibrium equation and applying the above finite element approximation we obtain:

$$\int \mathbf{B}^T \Delta\underline{\sigma}\, dV = \int \mathbf{N}^T \Delta\underline{b}\, dV + \int \mathbf{N}^T \Delta\underline{t}\, dS + \underline{r}_0 \qquad \textit{Eq. [69]}$$

with:

$$\underline{r}_0 = \int \mathbf{N}^T \underline{b}_0\, dV + \int \mathbf{N}^T \underline{t}_0\, dS - \int \mathbf{B}^T \underline{\sigma}_0\, dV \qquad \textit{Eq. [70]}$$

where $\underline{b}$ is a body force due to self-weight and $\underline{t}$ represents the surface tractions. In general the residual force vector $\underline{r}_0$ will be equal to zero, but solutions of previous load steps may have been inaccurate. By adding the residual force vector the computational procedure becomes self-correcting. The term *dV* indicates integration over the volume of the body considered and *dS* indicates a surface integral.

Dividing the total stresses into pore pressure and effective stresses and introducing the constitutive relationship gives the nodal equilibrium equation:

$$\mathbf{K}\Delta\underline{v} + \mathbf{L}\Delta\underline{p}_n = \Delta\underline{f}_n \qquad \textit{Eq. [71]}$$

where **K** is the stiffness matrix, **L** is the coupling matrix and $\underline{f}_n$ is the incremental load vector:

$$\begin{aligned} \mathbf{K} &= \int \mathbf{B}^T \mathbf{M} \mathbf{B}\, d\ V \\ \mathbf{L} &= \int \mathbf{B}^T \underline{m} \mathbf{N}\, d\ V \\ \Delta\underline{f}_n &= \int \mathbf{N}^T \Delta\underline{b}\, d\ V + \int \mathbf{N}^T \Delta\underline{t}\, d\ S \end{aligned} \qquad \textit{Eq. [72]}$$

To formulate the flow problem, the continuity equation is adopted in the following form:

$$\nabla^{\mathrm{T}} \cdot \left(\mathbf{k} \nabla \left(\gamma_{\mathrm{w}} y - p_{\mathrm{steady}} - p \right) \Big/ \gamma_{\mathrm{w}} \right) - \underline{\mathrm{m}}^{\mathrm{T}} \frac{\partial \underline{\varepsilon}}{\partial t} + \frac{n}{K_w} \frac{\partial p}{\partial t} = 0 \qquad \textit{Eq. [73]}$$

where **k** is the permeability matrix:

$$\mathbf{k} = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{bmatrix} \qquad \textit{Eq. [74]}$$

n is the porosity, K_w is the bulk modulus of the pore fluid and γ_w is the unit weight of the pore fluid. This continuity equation includes the sign convention that p_{steady} and p are considered positive for tension.

As the steady state solution is defined by the equation:

$$\nabla^{\mathrm{T}} \cdot \left(\mathbf{k}\nabla\left(\gamma_{\mathrm{w}} y - p_{\mathrm{steady}}\right)/\gamma_{\mathrm{w}}\right) = 0 \qquad \textit{Eq. [75]}$$

the continuity equation takes the following form:

$$\nabla^{\mathrm{T}} \cdot \left(\mathbf{k}\nabla p \Big/ \gamma_{\mathrm{w}}\right) + \underline{\mathrm{m}}^{\mathrm{T}} \frac{\partial \varepsilon}{\partial t} - \frac{n}{K_w}\frac{\partial p}{\partial t} = 0 \qquad \textit{Eq. [76]}$$

Applying finite element discretisation using a Galerkin procedure and incorporating prescribed boundary conditions we obtain:

$$-\mathbf{H}\underline{p}_n + \mathbf{L}^T \frac{\partial \underline{v}}{\partial t} - \mathbf{S}\frac{\partial \underline{p}_n}{\partial t} = \underline{q}_n \qquad \textit{Eq. [77]}$$

where:

$$\mathbf{H} = \int (\nabla \cdot \mathbf{N})^T \mathbf{k} (\nabla \cdot \mathbf{N})\, d\ V, \qquad \mathbf{S} = \int \frac{n}{K_w} \mathbf{N}^T \mathbf{N}\, d\ V \qquad \textit{Eq. [78]}$$

and $\underline{q}_n$ is a vector due to prescribed outflow at the boundary. However within PLAXIS it is not possible to have boundaries with non-zero prescribed outflow. The boundary is either closed (zero flux) or open (zero excess pore pressure). In reality the bulk modulus of water is very high and so the compressibility of water can be neglected in comparison to the compressibility of the soil skeleton.

In PLAXIS the bulk modulus of the pore fluid is taken automatically according to (also see Reference Manual):

$$\frac{K_w}{n} = \frac{3\left(\nu_u - \nu\right)}{\left(1 - 2\nu_u\right)\left(1 + \nu\right)} K_{skeleton} \qquad \textit{Eq. [79]}$$

Where ν_u has a default value of 0.495. The value can be modified in the input program on the basis of Skempton's B-parameter. For drained material and material just switched on, the bulk modulus of the pore fluid is neglected.

The equilibrium and continuity equations may be compressed into a block matrix equation:

$$\begin{bmatrix} \mathbf{K} & \mathbf{L} \\ \mathbf{L}^T & -\mathbf{S} \end{bmatrix} \begin{bmatrix} \frac{\partial \underline{v}}{\partial t} \\ \frac{\partial \underline{p}_n}{\partial t} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{H} \end{bmatrix} \begin{bmatrix} \underline{v} \\ \underline{p}_n \end{bmatrix} + \begin{bmatrix} \frac{\partial \underline{f}_n}{\partial t} \\ \underline{q}_n \end{bmatrix} \qquad \textit{Eq. [80]}$$

A simple step-by-step integration procedure is used to solve this equation. Using the symbol Δ to denote finite increments, the integration gives:

$$\begin{bmatrix} \mathbf{K} & \mathbf{L} \\ \mathbf{L}^T & -\mathbf{S}^* \end{bmatrix} \begin{bmatrix} \Delta \underline{v} \\ \Delta \underline{p}_n \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & \Delta t \mathbf{H}^* \end{bmatrix} \begin{bmatrix} \underline{v}_0 \\ \underline{p}_{n0} \end{bmatrix} + \begin{bmatrix} \Delta \underline{f}_n \\ \Delta t \underline{q}_n \end{bmatrix} \qquad \textit{Eq. [81]}$$

where:

$$\mathbf{S} = \alpha \Delta t \mathbf{H} + \mathbf{S} \quad \underline{q}_n^* = \underline{q}_{n0} + \alpha \Delta \underline{q}_n \qquad \textit{Eq. [82]}$$

and $\underline{v}_0$ and $\underline{p}_0$ denote values at the beginning of a time step. The parameter α is the time integration coefficient. In general the integration coefficient α can take values from 0 to 1. In PLAXIS the fully implicit scheme of integration is used with $\alpha=1$.

4.3 Elastoplastic consolidation

In general, when a non-linear material model is used, iterations are needed to arrive at the correct solution. Due to plasticity or stress-dependent stiffness behaviour the equilibrium equations are not necessarily satisfied using the technique described above. Therefore the equilibrium equation is inspected here. Instead of Eq. [71] the equilibrium equation is written in sub-incremental form:

$$\mathbf{K}\delta\underline{v} + \mathbf{L}\delta\underline{p}_n = \delta\underline{f}_n \qquad \textit{Eq. [83]}$$

where $\underline{r}_n$ is the global residual force vector. The total displacement increment $\Delta\underline{v}$ is the summation of sub-increments $\delta\underline{v}$ from all iterations in the current step:

$$\underline{r}_n = \int \mathbf{N}^T \underline{b} dV + \int \mathbf{N}^T \underline{t} dS - \int \mathbf{B}^T \underline{\sigma} dV \qquad \textit{Eq. [84]}$$

with:

$$\underline{b} = \underline{b}_0 + \Delta\underline{b} \qquad \text{and} \qquad \underline{t} = \underline{t}_0 + \Delta\underline{t} \qquad \textit{Eq. [85]}$$

In the first iteration we consider $\underline{\sigma} = \underline{\sigma}_0$, i.e. the stress at the beginning of the step. Successive iterations are used on the current stresses that are computed from the appropriate constitutive model.

4.4 Critical time step

For most numerical integration procedures, accuracy increases when the time step is reduced, but for consolidation there is a threshold value. Below a particular time increment (*critical time step*) the accuracy rapidly decreases. Care should be taken with time steps that are smaller than the advised minimum time step. The critical time step is calculated as:

$$\Delta t_{critical} = \frac{H^2}{\eta \alpha c_v} \qquad \textit{Eq. [86]}$$

where α is the time integration coefficient which is equal to 1 for fully implicit integration scheme, η is a constant parameter which is determined for each types of element and H is the height of the element used. c_v is the consolidation coefficient and is calculated as:

$$c_v = \frac{k / \gamma_w}{1/K' + Q} \qquad \textit{Eq. [87]}$$

where γ_w is the unit weight of the pore fluid, k is the coefficient of permeability, K' is the drained bulk modulus of soil skeleton and Q represents the compressibility of the fluid which is defined as:

$$Q = n\left(\frac{S}{K_w} - \frac{\partial S}{\partial p_w}\right) \qquad \textit{Eq. [88]}$$

where n is the porosity, S is the degree of saturation, p_w is the suction pore pressure and K_w is the elastic bulk modulus of water. Therefore the critical time step can be derived as:

$$\Delta t_{critical} = \frac{H^2 \gamma_w}{\eta k}\left(\frac{1}{K'} + Q\right) \qquad Eq.\ [89]$$

For one dimensional consolidation (vertical flow) in fully saturated soil, the critical time step can be simplified as:

$$\Delta t_{critical} = \frac{H^2 \gamma_w}{\eta k_y}\left(\frac{1}{E_{oed}} + \frac{n}{K_w}\right) \qquad Eq.\ [90]$$

in which E_{oed} is the oedometer modulus:

$$E_{oed} = \frac{E(1 - \nu)}{(1 - 2\nu)(1 + \nu)} \qquad Eq.\ [91]$$

ν is Poisson's ratio and E is the elastic Young's modulus.

For two dimensional elements as used in PLAXIS 2D, $\eta = 80$ and $\eta = 40$ for 15-node triangle and 6-node triangle elements, respectively. Therefore, the critical time step for fully saturated soils can be calculated by:

$$\Delta t_{critical} = \frac{H^2 \gamma_w}{80 k_y}\left(\frac{1}{E_{oed}} + \frac{n}{K_w}\right) \quad (15 - node\ triangles) \qquad Eq.\ [92]$$

$$\Delta t_{critical} = \frac{H^2 \gamma_w}{40 k_y}\left(\frac{1}{E_{oed}} + \frac{n}{K_w}\right) \quad (6 - node\ triangles) \qquad Eq.\ [93]$$

For three dimensional elements as used in PLAXIS 3D $\eta = 3$. Therefore, the critical time step for fully saturated soils can be calculated by:

$$\Delta t_{critical} = \frac{H^2 \gamma_w}{3 k_y}\left(\frac{1}{E_{oed}} + \frac{n}{K_w}\right) \qquad Eq.\ [94]$$

Fine meshes allow for smaller time steps than coarse meshes. For unstructured meshes with different element sizes or when dealing with different soil layers and thus different values of k, E and ν, the above formula yields different values for the critical time step. To be on the safe side, the time step should not be smaller than the maximum value of the critical time steps of all individual elements. This overall critical time step is automatically adopted as the *First time step* in a consolidation analysis. For an introduction to the critical time step concept, the reader is referred to Vermmer & Verruijt (1981) (on page 69). Detailed information for various types of finite elements is given by Song (1990) (on page 69).

5 Dynamics

This chapter highlights some of the theoretical backgrounds of the dynamic module. The chapter does not give a full theoretical description of the dynamic modelling. For a more detailed description you are referred to the literature Zienkiewicz & Taylor (1991) (on page 69), Hughes (1987) (on page 68), Das (1995) (on page 68), Kramer (1996) (on page 68), Haigh et al. (2005) (on page 68), Basabe & Sen (2007) (on page 68), Kelly et al. (1976) (on page 68) and Pradhan et al (2004) (on page 68).

5.1 Basic equation dynamic behaviour

The basic equation for the time-dependent movement of a volume under the influence of a (dynamic) load is:

$$\mathbf{M}\ddot{\underline{u}} + \mathbf{C}\dot{\underline{u}} + \mathbf{K}\underline{u} = \underline{F} \qquad Eq.\ [95]$$

Here, **M** is the mass matrix, $\underline{u}$ is the displacement vector, **C** is the damping matrix, **K** is the stiffness matrix and $\underline{F}$ is the load vector. The displacement, $\underline{u}$, the velocity, $\dot{\underline{u}}$, and the acceleration, $\ddot{\underline{u}}$, can vary with time. The last two terms in the Eq. [95] ($\mathbf{K}\underline{u} = \underline{F}$) correspond to the static deformation.

Here the theory is described on the bases of linear elasticity. However, in principle, all models in PLAXIS can be used for dynamics analisis. The soil behaviour can be both drained and undrained. In the latter case, the bulk stiffness of the groundwater is added to the stiffness matrix **K**, as is the case for the static calculation.

In the matrix **M**, the mass of the materials (soil + water + any constructions) is taken into account. In PLAXIS the mass matrix is implemented as a lumped matrix.

The matrix **C** represents the material damping of the materials. In reality, material damping is caused by friction or by irreversible deformations (plasticity or viscosity). With more viscosity or more plasticity, more vibration energy can be dissipated. If elasticity is assumed, damping can still be taken into account using the matrix **C**. To determine the damping matrix, extra parameters are required, which are difficult to determine from tests. In finite element formulations, **C** is often formulated as a function of the mass and stiffness matrices (Rayleigh damping) (Zienkiewicz & Taylor, 1991 (on page 69); Hughes, 1987) (on page 68)) as:

$$\mathbf{C} = \alpha_R\mathbf{M} + \beta_R\mathbf{K} \qquad Eq.\ [96]$$

This limits the determination of the damping matrix to the Rayleigh coefficients α_R and β_R. Here, when the contribution of **M** is dominant (for example, $\alpha_R = 10^{-2}$ and $\beta_R = 10^{-3}$) more of the low frequency vibrations are damped, and when the contribution of **K** is dominant (for example, $\alpha_R = 10^{-3}$ and $\beta_R = 10^{-2}$) more of the high-frequency vibrations are damped. In the standard setting of PLAXIS, $\alpha_R = \beta_R = 0$.

5.2 Time integration

In the numerical implementation of dynamics, the formulation of the time integration constitutes an important factor for the stability and accuracy of the calculation process. Explicit and implicit integration are the two commonly used time integration schemes. The advantage of explicit integration is that it is relatively simple to formulate. However, the disadvantage is that the calculation process is not as robust and it imposes serious limitations on the time step. The implicit method is more complicated, but it produces a more reliable (more stable) calculation process and usually a more accurate solution (Sluys, 1992 (on page 69)).

The implicit time integration scheme of Newmark is a frequently used method. With this method, the displacement and the velocity at the point in time $t+\Delta t$ are expressed respectively as:

$$u^{t+\Delta t} = u^{t} + \dot{u}^{t}\,\Delta t + \left(\left(\frac{1}{2} - \alpha\right)\ddot{u}^{t} + \alpha\ddot{u}^{t+\Delta t}\right)\Delta t^{2} \qquad \textit{Eq. [97]}$$

$$\dot{u}^{t+\Delta t} = \dot{u}^{t} + \left((1-\beta)\ddot{u}^{t} + \beta\ddot{u}^{t+\Delta t}\right)\Delta t \qquad \textit{Eq. [98]}$$

In the above equations, Δt is the time step. The coefficients α and β determine the accuracy of the numerical time integration. They are not equal to the α and β for the Rayleigh damping. In order to obtain a stable solution, the following condition must apply:

$$\beta \geq \frac{1}{2}, \quad \alpha \geq \frac{1}{4}\left(\frac{1}{2} + \beta\right)^{2} \qquad \textit{Eq. [99]}$$

The user is advised to use the standard setting of PLAXIS, in which the Newmark scheme with $\alpha=0.25$ and $\beta=0.5$ (average acceleration method) is utilised. Other combinations are also possible, however.

5.2.1 *Implementation of the integration scheme*

Eq. [97] can also be written as:

$$\begin{aligned} \ddot{u}^{t+\Delta t} &= c_0\Delta u - c_2\dot{u}^{t} - c_3\ddot{u}^{t} & & \ddot{u}^{t+\Delta t} = c_0\Delta u - c_2\dot{u}^{t} - c_3\ddot{u}^{t} \\ \dot{u}^{t+\Delta t} &= \dot{u}^{t} + c_6\ddot{u}^{t} + c_7\ddot{u}^{t+\Delta t} & \text{or as:} \quad & u^{t+\Delta t} = c_1\Delta u - c_4\dot{u}^{t} - c_5\ddot{u}^{t} \\ u^{t+\Delta t} &= u^{t} + \Delta u & & u^{t+\Delta t} = u^{t} + \Delta u \end{aligned} \qquad \textit{Eq. [100]}$$

where the coefficients $c_0, \ldots, c_7$, can be expressed in the time step and in the integration parameters α and β. In this way, the displacement, the velocity and the acceleration at the end of the time step are expressed by those at the start of the time step and the displacement increment. With implicit time integration, Eq. [95] must be obtained at the end of a time step ($t+\Delta t$):

$$\mathbf{M}\ddot{\underline{u}}^{t+\Delta t} + \mathbf{C}\dot{\underline{u}}^{t+\Delta t} + \mathbf{K}\underline{u}^{t+\Delta t} = \underline{F}^{t+\Delta t} \qquad \textit{Eq. [101]}$$

This equation, combined with the expressions Eq. [100] for the displacements, velocities and accelerations at the end of the time step, produce:

$$\begin{aligned} (c_0\mathbf{M} + c_1\mathbf{C} + \mathbf{K})\Delta\underline{u} = \; & \underline{F}_{ext}{}^{t+\Delta t} + \mathbf{M}\left(c_2\dot{\underline{u}}^{t} + c_3\ddot{\underline{u}}^{t}\right) + \mathbf{C}\left(c_4\dot{\underline{u}}^{t} + c_5\ddot{\underline{u}}^{t}\right) \\ & - \underline{F}_{\mathrm{int}}{}^{t+\Delta t} \end{aligned} \qquad \textit{Eq. [102]}$$

In this form, the system of equations for a dynamic analysis reasonably matches that of a static analysis. The difference is that the 'stiffness matrix' contains extra terms for mass and damping and that the right-hand term contains extra terms specifying the velocity and acceleration at the start of the time step (time t).

5.2.2 *Recommended maximum time step*

Despite the advantages of the implicit integration, the time step used in the calculation is subject to some limitations. If the time step is too large, the solution will display substantial deviations and the calculated response will be unreliable. The maximum recommended time step depends on the maximum frequency and the coarseness (fineness) of the finite element mesh. The equation used for a single element is:

$$\Delta t_{\max,recommended} = \frac{l_{\min}}{V_s} \qquad \textit{Eq. [103]}$$

where l_{min} is the minimum length between three nodes of an element and V_s is the shear wave velocity of an element. In a finite element model, the recommended maximum time step is equal to the minimum value of Δt according toEq. [103] over all elements. In this way, the time step is chosen to ensure that a wave during a single step does not move a distance larger than the minimum dimension of one element in case of 2nd order elements and half an element in case of 4th order elements.

5.2.3 *Dynamic integration coefficients*

The Newmark implicit time history integration scheme has been used in PLAXIS code to solve the equilibrium equation (dynamics) of the system. This method requires the calculation of integration constants or coefficients. The time step, Δt is selected on the basis of the sampling time of the input signal and the number of dynamic sub-steps necessary for the analysis. Once is fixed, the dynamic integration coefficients (c_0, c_1, c_2, c_3, c_4, c_5, c_6 and c_7) required for the numerical evaluation of the effective or pseudo-stiffness matrix and subsequent computation of the displacements, velocities and accelerations at the end of each time step may be calculated as follows:

$$\begin{aligned}
c_0 &= \frac{1}{\alpha \Delta t^2} \\
c_1 &= \frac{\beta}{\alpha \Delta t} \\
c_2 &= \frac{1}{\alpha \Delta t} \\
c_3 &= \frac{1}{2\alpha} - 1 \\
c_4 &= \frac{\beta}{\alpha} - 1 \\
c_5 &= \frac{\beta}{\alpha - 2}\frac{\Delta t}{2} \\
c_6 &= (1 - \beta)\Delta t \\
c_7 &= \beta \Delta t
\end{aligned} \qquad \textit{Eq. [104]}$$

where, α and β are the Newmark parameters that can be determined to obtain the integration accuracy and stability.

5.3 Model Boundaries

In the case of a static deformation analysis, prescribed boundary displacements are introduced at the boundaries of a finite element model. The boundaries can be completely free, or fixities can be applied in one or two directions. Particularly the vertical boundaries of a mesh are often non-physical (synthetic) boundaries that have been chosen so that they do not influence the deformation behaviour of the construction to be modelled. In other words: the boundaries are 'far away'. For dynamics calculations, the boundaries should in principle be much further away than those for static calculations, because, otherwise, stress waves will be reflected leading to distortions in the computed results. However, locating the boundaries far away requires many extra elements and therefore a lot of extra memory and calculating time.

To counteract reflections and avoid spurious waves, various methods are used at the boundaries, which include:

- Use of half-infinite elements (boundary elements).
- Adaptation of the material properties of elements at the boundary (low stiffness, high viscosity).
- Use of viscous boundaries (dampers).
- Use of free-field and compliant base boundaries (boundary elements).

All of these methods have their advantages and disadvantages and are problem dependent. For the implementation of dynamic effects in PLAXIS, the viscous boundaries are used for problems where the dynamic source is inside the mesh and the free-field boundaries when the dynamic source is applied as a boundary condition (e.g. earthquake motions).

5.3.1 *Viscous boundaries*

In opting for viscous boundaries, a damper is used instead of applying fixities in a certain direction. The damper ensures that an increase in stress on the boundary is absorbed without rebounding. The boundary then starts to move. The use of viscous boundaries in PLAXIS is based on the method described by Lysmer & Kuhlmeyer, (1969) (on page 68). The normal and shear stress components absorbed by a damper in x-direction are:

$$\sigma_n = -C_1 \rho V_p \dot{u}_x$$
$$\tau = -C_2 \rho V_s \dot{u}_y \qquad \textit{Eq. [105]}$$

where ρ is the density of the materials, V_p and V_s are the pressure wave velocity and the shear wave velocity respectively, $\dot{u}_x$ and $\dot{u}_y$ are the normal and shear particle velocities derived by time integration, C_1 and C_2 are relaxation coefficients to modify the effect of the absorption. When pressure waves only strike the boundary perpendicular, relaxation is redundant ($C_1 = C_2 = 1$).

In the presence of shear waves, the damping effect of the viscous boundaries is perfect. The effect can be modified by adapting the second coefficient in particular. The experience gained until now shows that the use of $C_1 = 1$ and $C_2 = 1$ results in a reasonable absorption of any waves reaching the boundary, which is sufficient for practical applications.

5.3.2 *Free-field and compliant base boundaries*

Using free-field boundaries, the domain is reduced to the area of interest and the free field motion is applied to the boundaries employing free-field elements. A free-field element consists of a one-dimensional element (in 2D problems) coupled to the main grid by viscous dashpots (Figure 2 (on page 26)). To describe the propagation of waves inside the free-field elements, the same mechanical behaviour as the adjacent soil element in the main domain is used.

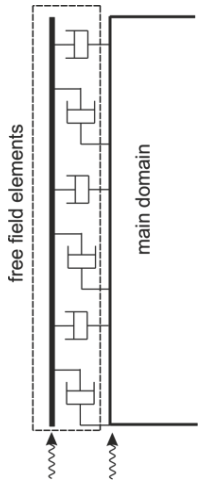


Figure 2: Free field elements

The free field motion is transferred from free-field elements to the main domain by applying the equivalent forces according to Eq. [106]. In these equations, the effect of a viscous boundary condition is also considered at the boundary of the main domain to absorb the outgoing waves from the internal structures. The normal and shear stress components transferred from the free-field element to the main domain, for a damper in *x*-direction, are:

$$\sigma_n = -C_1 \rho V_p \left(\dot{u}_x^m - \dot{u}_x^{ff} \right)$$
$$\tau = -C_2 \rho V_s \left(\dot{u}_y^m - \dot{u}_y^{ff} \right) \qquad \textit{Eq. [106]}$$

where ρ is the density of the materials, V_p and V_s are the pressure wave velocity and the shear wave velocity respectively (**Materials Manual - Chapter 3 - Basic Parameters of the Mohr-Coulomb model**), $\dot{u}^m$ and $\dot{u}^{ff}$ are the particle velocities in the main grid and in the free-field element respectively, C_1 and C_2 are relaxation coefficients to modify the effect of the absorption. When pressure waves only strike the boundary perpendicular, relaxation is redundant ($C_1 = C_2 = 1$).

Free-field elements can be attached to the lateral boundaries of the main domain. If the base cluster is considered, absorption and application of dynamic input can be done at the same place at the bottom of the model with the compliant base boundaries (Joyner & Chen, 1975 (on page 68)). The equivalent stresses in an compliant base are given by:

$$\sigma_n = -C_1 \rho V_p \left(\dot{u}_x^d - 2\dot{u}_x^u \right)$$
$$\tau = -C_2 \rho V_s \left(\dot{u}_y^d - 2\dot{u}_y^u \right) \qquad \textit{Eq. [107]}$$

where $\dot{u}^d$ and $\dot{u}^u$ are the upward and downward particle velocities, which can be considered as displacement in the compliant base element and the main domain, respectively. The compliant base works correctly if the relaxation coefficients C_1 and C_2 are equal to 1. The reaction of the dashpots is multiplied by a factor 2 since half of the input is absorbed by the viscous dashpots and half is transferred to the main domain. This is the difference between the compliant base and the free field boundary conditions.

5.4 Initial stresses and stress increments

By removing the boundary fixities during the transition from a static analysis to a dynamics analysis, the boundary stresses also cease. This means that the boundary will start to move as a result of initial stresses. To prevent this, the original boundary stress will be converted to an initial (virtual) boundary velocity. When calculating the stress, the initial boundary velocity must be subtracted from the real velocity:

$$\sigma_n = -c_1 \rho V_p \dot{u}_n + \sigma_n^0 = -c_1 \rho V_p \left(\dot{u}_n - \dot{u}_n^0\right) \qquad \textit{Eq. [108]}$$

This initial velocity is calculated at the start of the dynamics analisis and is therefore based purely on the original boundary stress (preceding calculation or initial stress state).

At present, situations can arise where a new load is applied at a certain location on the model and is continuously present from that moment onward. Such a load should result in an increase in the average boundary stress. If it involves a viscous boundary, the average incremental stress cannot be absorbed. Instead, the boundary will start to move. In most situations, however, there are also fixed (non-absorbent) boundaries elsewhere in the mesh – for example, on the bottom. The bottom of the mesh, at the location of the transition from a non-rigid to a hard (stiff) soil layer, is often chosen for this. Here, reflections also occur in reality, so that such a bottom boundary in a dynamics analisis can simply be provided with standard (fixed) peripheral conditions. In the above-mentioned case of an increased load on the model, that increase will eventually have to be absorbed by the (fixed) bottom boundary – if necessary, after redistributing the stresses.

5.5 Amplification of responses

Let there be an acceleration time history (of size N) defined by a set of accelerations (may be other responses in form of velocities or displacements), [$a_1, a_2, a_3, \ldots, a_N$] recorded at time steps [$t_1, t_2, t_3, \ldots, t_N$] with uniform sampling rate. On performing Fourier transform on the given series, the time signature can be converted to frequency dependent Fourier spectra like [A_1+iB_1, A_2+iB_2, A_3+i$B_3, \ldots, A_M$+iB_M] against the frequency set of [$f_1, f_2, f_3, \ldots, f_N$], where M is defined as follows:

$$M = \left.\begin{array}{ll} \frac{N}{2} & : \quad N \subset \text{even integers} \\ \frac{N+1}{2} & : \quad N \subset \text{odd number} \end{array}\right\} \qquad \textit{Eq. [109]}$$

The power spectra of the response may subsequently be obtained as a set of [$\frac{1}{2}(A_1^2 + B_1^2)$, $\frac{1}{2}(A_2^2 + B_2^2)$, ..., $\frac{1}{2}(A_M^2 + B_M^2)$] against the frequency set of [$f_1, f_2, f_3, \ldots, f_M$].

5.6 Pseudo-spectral acceleration response spectrum for a single-degree-of-freedom system

Let a structure be idealized as a single-degree-of-freedom (SDOF) system. This SDOF structure may be physically modelled as a combination of mass-spring-dashpot system attached to the ground surface. The equation for this SDOF system may be written as:

$$m\ddot{x} + c\dot{x} + kx = -m\ddot{x}_{gs} \qquad Eq.\ [110]$$

where, m is the mass of the structure, x is the lateral displacement of the structure, c represents the viscous damping coefficient of the structure, k is the stiffness of the structure and $\ddot{x}_{gs}$ is the horizontal acceleration time history at the ground surface at the base of the structure. The expressions for the damping coefficient and the structural stiffness are given by:

$$c = 2m\zeta_s\omega_n \qquad \text{and} \qquad k = m{\omega_n}^2 \qquad Eq.\ [111]$$

respectively. ζ_s and ω denote respectively the damping ratio and natural frequency of the structure. The natural frequency is the inverse of the natural time period. The pseudo-acceleration, a is defined by the following equation.

$$a = \left| x_{\max} \right| {\omega_n}^2 \qquad Eq.\ [112]$$

where

$|x_{max}|$ = absolute peak response of a structure during the whole period of dynamic loading

The acceleration time history obtained at the soil-structure interface (i.e. at soil surface as obtained from PLAXIS is used as an input excitation to the structure. The above equation may now be solved in time domain for a particular time period of a SDOF structure to obtain the displacements of the structure at every time point and subsequently the absolute maximum displacement response (i.e. $|x_{max}|$) of the structure can be found out from this displacement time history (and hence its pseudo-spectral acceleration from Eq. [112]) for the whole duration of the time history. Thus, this equation may be repeatedly solved for different natural time periods of the structure to plot its pseudo-acceleration response versus time period giving rise to PSA plot. This would enable the users to perform seismic soil-structure interaction analysis or seismic analysis or structures.

The stiffness ratio, $\bar{s}$ is the ratio of structural stiffness to soil stiffness defined by the following equation (page 261 of Kramer, 1996 (on page 68))

$$\bar{s} = \frac{\omega_n L}{V_s} \qquad Eq.\ [113]$$

in which V_s is the shear wave velocity of the supporting soil medium and L is the height of structure above the foundation.

5.7 Natural frequency of vibration of a soil deposit

The natural frequency of vibration of a soil deposit may be calculated from the following equation (page 261 of Kramer, 1996 (on page 68)):

$$f_n = \frac{V_s}{4H}(1 + 2n) \qquad Eq.\ [114]$$

where, f_n is the n^{th} natural frequency of the soil deposit in Hz and n = 0,1,2,....

For $n = 0$, the first natural frequency, f_0 (i.e. the fundamental frequency) of vibration of the soil deposit of thickness H is given by

$$f_n = \frac{V_s}{4H} \qquad Eq.\ [115]$$

5.8 Hydrodynamic pressure

The hydrodynamic pressures in a dam-reservoir system can be dealt with using the added mass concept. The added mass approach introduced by Westergaard is widely used in practice and simplifies the analysis procedure of the response of an incompressible dam-reservoir during earthquakes. Westergaard's analytical solution was obtained for a rigid dam with a vertical upstream face under a horizontal harmonic ground motion, where the fluid is treated as an added mass to the body of the dam. Zangar extended Westergaard's work by considering the sloping upstream face of the dam.

5.8.1 Added mass approach

The added mass is generally described as a matrix which models the interaction between water pressure and the structure. It allows to investigate the dynamical response of the structure without explicitly modelling the fluid motion and consequently reducing the modelling efforts.

Zangar derived experimentally an equation for the pressure distribution over the height of the dam for different inclinations of the dam. Based on the assumptions of water incompressibility and rigid structure, the expression for the hydrodynamic pressure distribution is given as:

$$p(y) = Ca\rho_w h \qquad Eq.\ [116]$$

where

C	=	Pressure coefficient factor
a	=	Horizontal acceleration
ρ_w	=	Density of water
h	=	Reservoir height
y	=	Distance of the considered point above the bottom of the reservoir

For different inclination angles of the upstream surface, Zangar derived a parabolic shape of the mass or pressure distribution based on the experimental results. Accordingly, he obtained the following expression for the pressure coefficient factor:

$$C = \frac{C_m}{2}\left[\frac{y}{h}\left(2 - \frac{y}{h}\right) + \sqrt{\frac{y}{h}\left(2 - \frac{y}{h}\right)}\right] \qquad Eq.\ [117]$$

The constant factor C_m is defined as the maximum occuring pressure coefficient for a specific inclination. By considering water incompressible, the values of C_m can be calculated from

$$C_m = -0.0073\theta + 0.7412 \qquad Eq.\ [118]$$

where the angle of inclination θ is measured in degrees.

5.8.2 Implementation of Zangar's added mass

The added mass matrix of a finite element corresponding to the Zangar's approach is obtained as follows. For an arbitrary segment of the face of the dam with an area of $1 \times dl$, the corresponding normal inertia force due to the hydrodynamic pressure is written as:

$$dF_N = pdl = \lambda a_x dl \qquad Eq.\ [119]$$

where:

$$\lambda = \frac{C_m \rho_w}{2}\left[\frac{u}{h}\left(2 - \frac{y}{h} + \sqrt{\frac{y}{h}\left(2 - \frac{y}{h}\right)}\right)\right] \qquad Eq.\ [120]$$

Notice here that the y parameter is measured from the bottom of the reservoir. By assuming θ to be the angle of the upstream face of the dam with the global x-axis, the inertia force components in the x and y directions are obtained using the following transformation

$$dF_N = \begin{bmatrix} dF_x \\ dF_y \end{bmatrix} = \lambda \begin{bmatrix} \sin\theta & 0 \\ \cos\theta & 0 \end{bmatrix} \begin{bmatrix} a_x \\ a_y \end{bmatrix} dl \qquad Eq.\ [121]$$

The total nodal inertia forces are computed by integrating over the upstream face as:

$$F = \int dF\ d\ l = \int \lambda N^T TN\ dl \qquad Eq.\ [122]$$

where N is the matrix of shape functions and T the transformation matrix.

By considering the inertia force as the product of mass and acceleration, the added mass matrix is obtained as:

$$M_{added} = \int \lambda N^T TN\, dl \qquad Eq.\ [123]$$

The computed added mass matrix in PLAXIS is always a consistent matrix regardless of the user request for lumped mass matrix for soil and structural elements. For an inclined face of the dam, applying horizontal acceleration generates vertical hydrodynamic forces but lumping the mass matrix eliminates these vertical forces and therefore should be avoided.

6 Element formulations

In this chapter the interpolation functions of the finite elements used in the PLAXIS program are described. Each element consists of a number of nodes. Each node has a number of degrees of freedom that correspond to discrete values of the unknowns in the boundary value problem to be solved. In the case of deformation theory the degrees of freedom correspond to the displacement components, whereas in the case of groundwater flow the degrees-of-freedom are the groundwater heads. For consolidation problems degrees-of-freedom are both displacement components and (excess) pore pressures. In addition to the interpolation functions it is described which type of numerical integration over elements is used in the program.

6.1 Interpolation functions of point elements

Point elements are elements existing of only one single node. Hence, the displacement field of the element $\underline{u}$ itself is only defined by the displacement field of this single node $\underline{v}$:

$$\underline{u} = \underline{v} \qquad \textit{Eq. [124]}$$

with:

$$\underline{u} = \begin{pmatrix} u_x & u_y \end{pmatrix}^T \quad and \quad \underline{v} = \begin{pmatrix} v_x & v_y \end{pmatrix}^T \qquad (PLAXIS\ 2D)$$

and

$$\underline{u} = \begin{pmatrix} u_x & u_y & u_z \end{pmatrix}^T \quad and \quad \underline{v} = \begin{pmatrix} v_x & v_y & v_z \end{pmatrix}^T \qquad (PLAXIS\ 3D)$$

6.1.1 *Structural elements*

6.1.1.1 *Fixed-end anchors*

In PLAXIS fixed-end anchors are considered to be point elements. The contribution of this element to the stiffness matrix can be derived from the traction the fixed-end anchor imposes on a point in the geometry due to the displacement of this point (see Eq. [13]). As a fixed-end anchor has only an axial stiffness and no bending stiffness, it is more convenient to rotate the global displacement field $\underline{v}$ to the displacement field $\underline{v}^*$ such that the first axis of the rotated coordinate system coincides with the direction of the fixed-end anchor:

$$\underline{v}^* = \mathbf{R}_\theta \underline{v} \qquad \textit{Eq. [125]}$$

where $\mathbf{R}_\theta$ denotes the rotation matrix. As only axial displacements are relevant, the element will only have one degree of freedom in the rotated coordinate system. The traction in the rotated coordinate system t^* can be derived as:

$$t^* = D^S u^* \qquad \textit{Eq. [126]}$$

where

D^S = the constitutive relationship of an anchor as defined in the Material Models Manual.

Converting the traction in the rotated coordinate system to the traction in the global coordinate system $\underline{t}$ by using the rotation matrix again and substituting Eq. [124] gives:

$$\underline{t} = \mathbf{R}_\theta^T D^S \mathbf{R}_\theta \underline{v} \qquad \textit{Eq. [127]}$$

Substituting this equation in Eq. [13] gives the element stiffness matrix of the fixed-end anchor $\mathbf{K}^S$:

$$\mathbf{K}^S = \mathbf{R}_\theta^T D^S \mathbf{R}_\theta \qquad \textit{Eq. [128]}$$

In case of elastoplastic behaviour of the anchor the maximum tension force is bound by $F_{max,tens}$ and the maximum compression force is bound by $F_{max,comp}$.

6.2 Interpolation functions and numerical integration of line elements

Within an element existing of more than one node the displacement field $\underline{u} = \begin{pmatrix} u_x & u_y \end{pmatrix}^T$ (PLAXIS 2D) or $\underline{u} = \begin{pmatrix} u_x & u_y & u_z \end{pmatrix}^T$ (PLAXIS 3D) and is obtained from the discrete nodal values in a vector $\underline{v} = \begin{pmatrix} v_x & v_y & \cdots & v_n \end{pmatrix}^T$ using interpolation functions assembled in matrix **N**:

$$\underline{u} = \mathbf{N}\underline{v}$$

Hence, interpolation functions **N** are used to interpolate values inside an element based on known values in the nodes. Interpolation functions are also denoted as shape functions.

Let us first consider a line element. Line elements are the basis for line loads, beams and node-to-node anchors. The extension of this theory to areas and volumes is given in the subsequent sections. When the local position, ξ, of a point (usually a stress point or an integration point) is known, one can write for a displacement component u:

$$u(\xi) = \sum_{i=1}^{n} N_i(\xi) v_i \qquad \textit{Eq. [129]}$$

where

v_i	=	Nodal values
$N_i(\xi)$	=	Value of the shape function of node *i* at position ξ
u(ξ)	=	Resulting value at position ξ
n	=	Number of nodes per element

6.2.1 *Interpolation functions of line elements*

Interpolation functions or shape functions are derived in a local coordinate system. This has several advantages like programming only one function per element type, a simple application of numerical integration and allowing higher-order elements to have curved edges.

6.2.1.1 *2-node line elements*

In Figure 3 (on page 33), an example of a 2-node line element is given. In contrast to a 3-node line element or a 5-node line element in the PLAXIS 2D program, this element is not compatible with an area element in the PLAXIS 2D or PLAXIS 3D program or a volume element in the PLAXIS 3D program. The 2-node line elements are the basis for node-to-node anchors. The shape functions N_i have the property that the function value is equal to unity at node *i* and zero at the other node. For 2-node line elements the nodes are located at ξ=*-1* and ξ=*1*. The shape functions are given by:

$$N_1 = \frac{1}{2}(1 - \xi)$$

$$N_2 = \frac{1}{2}(1 + \xi) \qquad \textit{Eq. [130]}$$

2-node line elements provide a first-order (linear) interpolation of displacements.

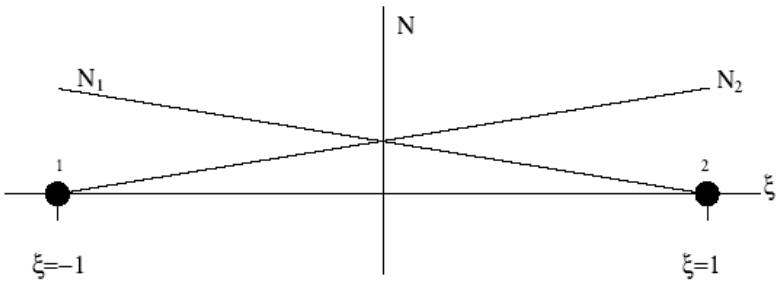


Figure 3: Shape functions for a 2-node line element

6.2.1.2 *3-node line elements*

In Figure 4 (on page 34), an example of a 3-node line element is given, which is compatible with the side of a 6-node triangle in the PLAXIS 2D or PLAXIS 3D program or a 10-node volume element in the PLAXIS 3D program, since these elements also have three nodes on a side. The shape functions N_i have the property that the function value is equal to unity at node *i* and zero at the other nodes. For 3-node line elements, where nodes 1, 2 and 3 are located at ξ = -1, 0 and 1 respectively, the shape functions are given by:

$$N_1 = -\frac{1}{2}(1-\xi)\xi$$
$$N_2 = (1+\xi)(1-\xi) \qquad \textit{Eq. [131]}$$
$$N_3 = \frac{1}{2}(1+\xi)\xi$$

3-node line elements provide a second-order (quadratic) interpolation of displacements. These elements are the basis for line loads and beam elements.

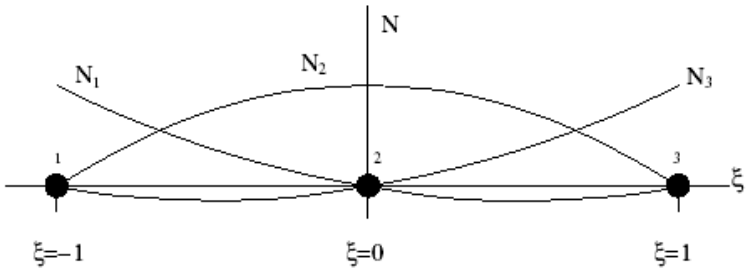


Figure 4: Shape functions for a 3-node line element

6.2.1.3 *5-node line elements*

In Figure 5 (on page 35), an example of a 5-node line element is given, which is compatible with the side of a 15-node triangle in the PLAXIS 2D program, since these elements also have five nodes on a side. The shape functions N_i have the property that the function value is equal to unity at node *i* and zero at the other nodes. For 5-node line elements, where nodes 1, 2, 3, 4 and 5 are located at ξ = -1, -0.5, 0, 0.5 and 1 respectively, the shape functions are given by:

$$N_1 = -(1-\xi)(1-2\xi)\xi(-1-2\xi)/6$$
$$N_2 = 4(1-\xi)(1-2\xi)\xi(-1-\xi)/3$$
$$N_3 = (1-\xi)(1-2\xi)(-1-2\xi)(-1-\xi) \qquad \textit{Eq. [132]}$$
$$N_4 = 4(1-\xi)\xi(-1-2\xi)(-1-\xi)/3$$
$$N_5 = -(1-2\xi)\xi(-1-2\xi)(-1-\xi)/6$$

5-node line elements provide a fourth-order (quartic) interpolation of displacements. These elements are the basis for line loads and beam elements.

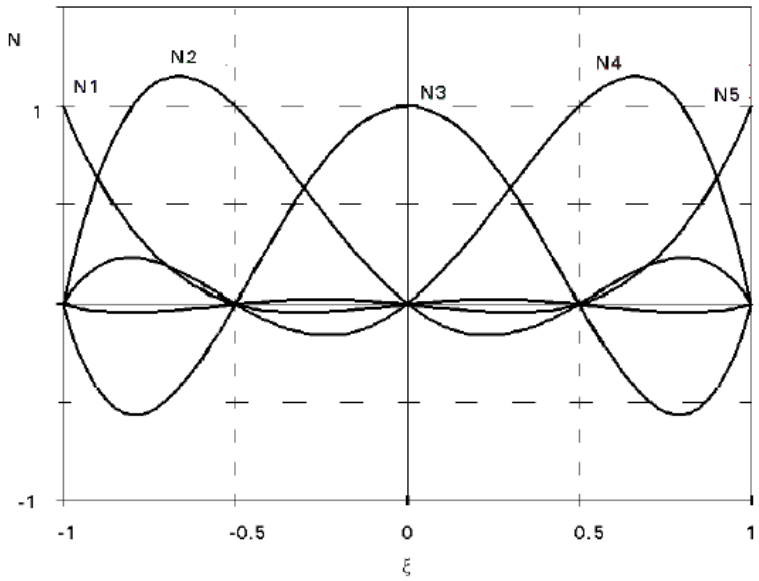


Figure 5: Shape functions for a 5-node line element

6.2.2 *Structural elements*

Structural line elements in the PLAXIS program are based on the line elements as described in the previous sections. However, there are some differences.

6.2.2.1 *Node-to-node anchors*

Node-to-node anchors are springs that are used to model ties between two points. A node-to-node anchor consists of a 2-node element with both nodes shared with the elements the node-to-node anchor is attached to. Therefore, the nodes have three d.o.f.s in the global coordinate system. However, as a node-to-node anchor can only sustain normal forces, only the displacement in the axial direction of the node-to-node anchor is relevant. Therefore it is more convenient to rotate the global coordinate system to a coordinate system in which the first axis coincides with the direction of the anchor. This rotated coordinate system is denoted as the *x*, y*, z** coordinate system and is similar to the (*1, 2, 3*) coordinate system used in the Material Models Manual. In this rotated coordinate system, these elements have only one d.o.f. per node (a displacement in axial direction).

The shape functions for these axial displacements are already given by Eq. [130]. Using index notation, the axial displacement can now be defined as:

$$u_x^* = N_i v_{ix}^* \qquad \textit{Eq. [133]}$$

where

v_{ix}^* = the nodal displacement in axial direction of node *i*.

The nodal displacements in the rotated coordinate system can be rotated to give the nodal displacements in the global coordinate system:

$$\underline{v}_i = \mathbf{R}_\theta^T v_{ix}^* \qquad Eq.\ [134]$$

where the nodal displacement vector in the global coordinate system is denoted by $\underline{v}_i = \begin{pmatrix} v_{ix} & v_{iy} \end{pmatrix}^T$ in the PLAXIS 2D program and $\underline{v}_i = \begin{pmatrix} v_{ix} & v_{iy} & v_{iz} \end{pmatrix}^T$ in the PLAXIS 3D program and $\mathbf{R}_\theta$ denotes the rotation matrix. For further elaboration into the element stiffness matrix see Derivatives of interpolation functions (on page 37) and Calculation of element stiffness matrix (on page 40).

6.2.2.2 Beam elements

The 3-node beam elements are used to describe semi-one-dimensional structural objects with flexural rigidity. Beam elements are slightly different from 3-node line elements in the sense that they have six degrees of freedom per node instead of three in the global coordinate system, i.e. three translational d.o.f (u_x, u_y, u_z) and three rotational d.o.f (φ_x, φ_y, φ_z).

The rotated coordinate system is denoted as the *x*, y*, z** coordinate system and is similar to the (*1, 2, 3*) coordinate system used in the Material Models Manual. The beam elements are numerically integrated over their cross section using 4 (2x2) point Gaussian integration. In addition, the beam elements are numerically integrated over their length using 4-point Gaussian integration according to Table 1 (on page 39). Beam elements have only one local coordinate (ξ). The element provides a quadratic interpolation (3-node element) of the axial displacement (see Eq. [131]). Using index notation, the axial displacement can now be defined as:

$$u_x^* = N_i v_{ix}^* \qquad Eq.\ [135]$$

where v_{ix}^* denotes the nodal displacement in axial direction of node *i*. As Mindlin's theory has been adopted the shape functions for transverse displacements may be the same as for the axial displacements (see Eq. [131] and Eq. [132]). Using index notation, the transverse displacements can now be defined as:

$$\underline{u}_w^* = \mathbf{N}_{iw}^* \underline{v}_{iw}^* \qquad Eq.\ [136]$$

The local transverse displacements in any point is given by:

$$\underline{u}_w^* = \begin{bmatrix} u_y^* & u_z^* \end{bmatrix}^T \qquad Eq.\ [137]$$

The matrix $\mathbf{N}_{iw}^*$ for transverse displacements is defined as:

$$\mathbf{N}_{iw}^* = \begin{bmatrix} N_i(\xi) & 0 \\ 0 & N_i(\xi) \end{bmatrix} \qquad Eq.\ [138]$$

The local nodal transverse displacements and rotations of node *i* are given by v_{iw}^*:

$$\underline{v}_{iw}^* = \begin{bmatrix} \underline{v}_{iy}^* & \underline{v}_{iz}^* \end{bmatrix}^T \qquad Eq.\ [139]$$

where $\underline{v}_{iy}^*$ and $\underline{v}_{iz}^*$ denote the nodal transverse displacements.

As the displacements and rotations are fully uncoupled according to Mindlin's theory, the shape functions for the rotations may be different than the shape functions used for the displacements. However, in PLAXIS the same functions are used. Using index notation, the rotations can now be defined as:

$$\underline{\varphi}_w^* = \mathbf{N}_{iw}^* \underline{\psi}_{iw}^* \qquad Eq.\ [140]$$

where the matrix $\mathbf{N}^{*}_{iw}$ is defined by Eq. [138]. The local rotations in any point is given by:

$$\varphi^{*}_{w} = \begin{bmatrix} \varphi^{*}_{y} & \varphi^{*}_{z} \end{bmatrix}^{T} \qquad Eq.\ [141]$$

The local nodal rotations of node *i* are given by v^{*}_{iz}:

$$\Psi^{*}_{iw} = \begin{bmatrix} \Psi^{*}_{iy} & \Psi^{*}_{iz} \end{bmatrix}^{T} \qquad Eq.\ [142]$$

where ψ^{*}_{iy} and ψ^{*}_{iz} denote the nodal rotations. For further elaboration into the element stiffness matrix see Derivatives of interpolation functions (on page 37) and Calculation of element stiffness matrix (on page 40).

6.2.3 *Derivatives of interpolation functions*

To compute the element stiffness matrix first the derivatives of the interpolation functions should be derived.

6.2.3.1 *Node-to-node anchors*

As node-to-node anchors can only sustain axial forces, only the axial strains are of interest: $\varepsilon^{*}=du^{*}/dx^{*}$. Using the chain rule for differentiation gives:

$$\varepsilon^{*} = \frac{d\,u^{*}}{d\,x^{*}} = \frac{d\,u^{*}}{d\,\xi}\frac{d\,\xi}{d\,x^{*}} \qquad Eq.\ [143]$$

where (using index notation)

$$\frac{d\,u^{*}}{d\,\xi} = \frac{d\,N_i}{d\,\xi}v_i^{*} \qquad Eq.\ [144]$$

and

$$\frac{d\,x^{*}}{d\,\xi} = \frac{d\,N_i}{d\,\xi}x_i^{*} \qquad Eq.\ [145]$$

The parameter x_i^{*} denotes the coordinate of the nodes in the rotated coordinate system. In case of 2-node line elements Eq. [145] can be simplified to:

$$\frac{d\,x^{*}}{d\,\xi} = \frac{L}{2} \qquad Eq.\ [146]$$

where

L = length of the element in the global coordinate system.

Inserting Eq. [144], Eq. [146], Eq. [172] into Eq. [143] will give:

$$\varepsilon^{*} = B_i^{*}v_{ix}^{*} \qquad Eq.\ [147]$$

where the rotated strain interpolation function B_i^{*} is given by:

$$B_i^{*} = \frac{2}{L}\frac{d\,N_i}{d\,\xi} \qquad Eq.\ [148]$$

Rotating the local nodal displacements back to the global coordinate system gives:

$$\mathbf{B}_i = \frac{2}{L}\frac{d\ N_i}{d\ \xi}\mathbf{R}_\theta \qquad \textit{Eq. [149]}$$

Note that this strain interpolation function is still a function of the local coordinate ξ as the shape functions N_i are a function of ξ.

6.2.3.2 *Beam elements*

In case of axial displacements in the rotated coordinate system, the strain interpolation matrix for beams can be derived from Eq. [143] till Eq. [145]. As node *2* of a 3-node beam element is located in the middle of the element by default, Eq. [145] can be simplified to Eq. [146]. So, the strain interpolation matrix in the global coordinate system for the longitudinal displacements of beams is given by:

$$\mathbf{B}_{ix} = \frac{2}{L}\frac{d\ N_i}{d\ \xi}\mathbf{R}_\theta \qquad \textit{Eq. [150]}$$

To derive the shear forces in a Mindlin beam, a shear strain interpolation matrix is needed to define the stiffness matrix. In contrast to the Bernoulli beam theory, the Mindlin beam theory does not neglect the transverse shear deformations and thus the rotation (γ) of a beam cannot be calculated simply as a derivative of the transverse displacement:

$$\begin{bmatrix}\gamma_{12}\\ \gamma_{13}\end{bmatrix} = \begin{bmatrix}\frac{d\ u_y^*}{d\ x^*} - \varphi_z\\ \frac{d\ u_z^*}{d\ x^*} - \varphi_y\end{bmatrix}\mathbf{B}_{iw}^* \upsilon_{iw}^* - \mathbf{N}_{iw}^* \varphi_{iw}^* \qquad \textit{Eq. [151]}$$

where

$$\mathbf{B}_{iw}^* = \begin{bmatrix}\frac{d\ N_i}{d\ x^*} & 0\\ 0 & \frac{d\ N_i}{d\ x^*}\end{bmatrix} \qquad \textit{Eq. [152]}$$

and φ_{iw}^* is defined by Eq. [139]. Using the chain rule for differentiation (Eq. [143]) and the assumption of Eq. [146], Eq. [150] can be simplified to:

$$\mathbf{B}_{iw}^* = \frac{2}{L}\begin{bmatrix}\frac{d\ N_i}{d\ \xi} & 0\\ 0 & \frac{d\ N_i}{d\ \xi}\end{bmatrix} \qquad \textit{Eq. [153]}$$

Using Eq. [134] to rotate local nodal transverse displacements and rotations to global nodal displacements and rotations and inserting this equation in Eq. [153] gives:

$$\mathbf{B}_{iw} = \frac{2}{L}\begin{bmatrix}\frac{d\ N_i}{d\ \xi} & 0\\ 0 & \frac{d\ N_i}{d\ \xi}\end{bmatrix}\mathbf{R}_\theta \qquad \textit{Eq. [154]}$$

Note that this strain interpolation function is still a function of the local coordinate ξ as the shape functions N_i are a function of ξ.

In case of bending moments, a curvature interpolation matrix is needed to define the stiffness matrix. The curvature interpolation function describes the kinematic relationship between curvatures and displacements:

$$\underline{\kappa}^* = \begin{bmatrix} \kappa_2 \\ \kappa_3 \end{bmatrix} = \begin{bmatrix} \dfrac{d\,\varphi_y^*}{d\,x^*} \\ \dfrac{d\,\varphi_z^*}{d\,x^*} \end{bmatrix} = \mathbf{B}_{iw}^* \underline{\Psi}_{iw}^* \qquad \textit{Eq. [155]}$$

6.2.4 *Numerical integration of line elements*

To compute the element stiffness matrix first the derivatives of the interpolation functions should be derived.

In order to obtain the integral over a certain line, the integral is numerically estimated as:

$$\int_{\xi=-1}^{1} F(\xi)\, d\,\xi \approx \sum_{i=1}^{k} F(\xi_i) w_i \qquad \textit{Eq. [156]}$$

where

$F(\xi_i)$ = value of the function F at position ξ_i
w_i = weight factor for point i

A total of k sampling points is used. A method that is commonly used for numerical integration is Gaussian integration, where the positions ξ_i and weights w_i are chosen in a special way to obtain high accuracy. For Gaussian-integration a polynomial function of degree *2k-1* can be integrated exactly by using *k*points. The position and weight factors of the integration are given in Table 1 (on page 39). Note that the sum of the weight factors is equal to 2, which is equal to the length of the line in local coordinates. The types of integration used for the 2-node line elements and the 3-node line elements are shaded.

Table 1: Gaussian integration

Points	ξ_i	w_i	max. polyn. degree
1 point	0.000000...	2	1
2 points	±0.577350...	1	3
3 points	± 0.774596...	0.55555... (5/9)	5
	0.000000...	0.88888... (8/9)	
4 points	± 0.861136...	0.347854...	7
	± 0.339981...	0.652145...	
5 points	± 0.906179...	0.236926...	9
	± 0.538469...	0.478628...	
	0.000000...	0.568888...	

6.2.5 *Calculation of element stiffness matrix*

6.2.5.1 *Node-to-node anchors*

The element stiffness matrix of a node-to-node anchor is calculated by the integral (see also Eq. [25]):

$$\mathbf{K}^e = \int \mathbf{B}^T D^a \mathbf{B} \; d\; V \qquad \textit{Eq. [157]}$$

where

D^a = elastic constitutive relationship of the node-to-node anchor as discussed in the Material Models Manual

As the strain interpolation matrix is still a function of the local coordinate ξ it will make more sense to solve the integral of Eq. [157] in the local coordinate system. Applying the change of variables theorem to change the integral to the local coordinate system gives:

$$\mathbf{K}^e = \int \mathbf{B}^T D^a \mathbf{B} \frac{d\; x^*}{d\; \xi} \; d\; V^* \qquad \textit{Eq. [158]}$$

In case of a 2-node line element, $dx^*/d\xi=L/2$. This integral is estimated by numerical integration as described in Numerical integration of line elements (on page 39). In fact, the element stiffness matrix is composed of submatrices $\mathbf{K}^e_{ij}$ where *i* and *j* are the local nodes. The process of calculating the element stiffness matrix can be formulated as:

$$\mathbf{K}^e_{ij} = \sum_k \mathbf{B}^T_i D^a \mathbf{B}_j \frac{d\; x^*}{d\; \xi} w_k \qquad \textit{Eq. [159]}$$

In case of elastoplastic behaviour of the anchor the maximum tension force is bound by $F_{max,tens}$ and the maximum compression force is bound by $F_{max,comp}$ (PLAXIS 3D).

6.2.5.2 *Beam elements (PLAXIS 3D)*

In case of axial forces, the element stiffness matrix is given by Eq. [157] till Eq. [159]. In case of shear forces the stiffness matrix of a beam is calculated by the integral:

$$\mathbf{K}^e = \int \mathbf{B}^T_\varphi \mathbf{D}^s \mathbf{B}_\varphi \; d\; V \qquad \textit{Eq. [160]}$$

where $\mathbf{D}^s$ denotes the constitutive relationship of a beam in shearing (see Material Models Manual):

$$\mathbf{D}^s = \begin{bmatrix} kGA & 0 \\ 0 & kGA \end{bmatrix} \qquad \textit{Eq. [161]}$$

In case of bending moments the stiffness matrix of a beam is calculated by the integral:

$$\mathbf{K}^e = \int \mathbf{B}_\varphi^T \mathbf{D}^b \mathbf{B}_\varphi \, d\, V \qquad Eq.\ [162]$$

where $\mathbf{D}^b$ denotes the constitutive relationship of a beam in bending (see Material Models Manual):

$$\mathbf{D}^b = \begin{bmatrix} EI_2 & EI_{23} \\ EI_{23} & EI_3 \end{bmatrix} \qquad Eq.\ [163]$$

To solve the integral of Eq. [158] in the local coordinate system, the change of variables theorem should be applied:

$$\mathbf{K}^e = \int \mathbf{B}_\varphi^T \mathbf{D}^b \mathbf{B}_\varphi \frac{d\, x^*}{d\, \xi} \, d\, V^* \qquad Eq.\ [164]$$

In PLAXIS, for 3-node beam elements $dx^*/d\xi=L/2$. This integral is estimated by numerical integration as described in Numerical integration of line elements (on page 39). In fact, the element stiffness matrix is composed of submatrices $\mathbf{K}_{ij}^e$ where *i* and *j* are the local nodes. The process of calculating the element stiffness matrix can be formulated as:

$$\mathbf{K}_{ij}^e = \sum_k \mathbf{B}_{i\varphi}^T D^b \mathbf{B}_{j\varphi} \frac{d\, x^*}{d\, \xi} w_k \qquad Eq.\ [165]$$

6.3 Interpolation functions and numerical integration of area elements

Areas and surfaces in PLAXIS 2D are formed by 6-node or 15-node triangular elements. For the areas and surfaces in PLAXIS 3D only the 6-node triangular elements are available. The interpolation functions and the type of integration of these elements are described in the following subsections.

6.3.1 *Interpolation functions of area elements*

6.3.1.1 *6-node triangular elements*

The 6-node triangles are one of the options for the basis for the soil elements in PLAXIS 2D and the basis for plate elements and distributed loads in PLAXIS 3D.

For triangular elements there are two local coordinates (ξ and η). In addition we use an auxiliary coordinate $\zeta = 1-\xi-\eta$. 6-node triangular elements provide a second-order interpolation of displacements. The shape functions N_i have the property that the function value is equal to unity at node *i* and zero at the other nodes. The shape functions can be written as (see the local node numbering as shown in Figure 6 (on page 42)):

N_1	$= \zeta(2\zeta-1)$
N_2	$= \xi(2\xi-1)$
N_3	$= \eta(2\eta-1)$

N_4	$= 4\zeta\xi$)
N_5	$= 4\xi\eta$
N_6	$= 4\eta\zeta$

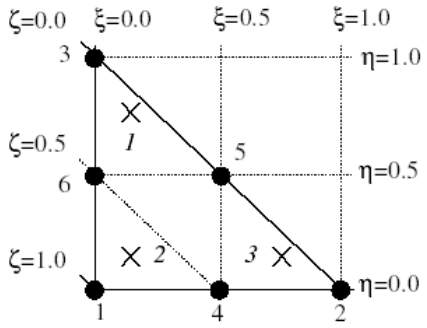


Figure 6: Local numbering and positioning of nodes (•) and integration points (x) of a 6-node triangular elementx) of a 6-node triangular element

6.3.1.2 *15-node triangular elements*

The 15-node triangles are one of the options for the basis for soil elements in PLAXIS 2D. For triangular elements there are two local coordinates (ξ and η). In addition we use an auxiliary coordinate $\zeta = 1-\xi-\eta$. For 15-node triangles the shape functions can be written as (see the local node numbering as shown in Figure 7 (on page 43)): .

N_1	$= \zeta(4\zeta-1)(4\zeta-2)(4\zeta-3)/6$
N_2	$= \xi(4\xi-1)(4\xi-2)(4\xi-3)/6$
N_3	$= \eta(4\eta-1)(4\eta-2)(4\eta-3)/6$
N_4	$= 4\zeta\xi(4\zeta-1)(4\xi-1)$
N_5	$= 4\xi\eta(4\xi-1)(4\eta-1)$
N_6	$= 4\eta\zeta(4\eta-1)(4\zeta-1)$
N_7	$= 8\xi\zeta(4\zeta-1)(4\zeta-2)/3$
N_8	$= 8\zeta\xi(4\xi-1)(4\xi-2)/3$
N_9	$= 8\eta\xi(4\xi-1)(4\xi-2)/3$
N_{10}	$= 8\xi\eta(4\eta-1)(4\eta-2)/3$
N_{11}	$= 8\zeta\eta(4\eta-1)(4\eta-2)/3$
N_{12}	$= 8\eta\zeta(4\zeta-1)(4\zeta-2)/3$
N_{13}	$= 32\eta\xi\zeta(4\zeta-1)$
N_{14}	$= 32\eta\xi\zeta(4\xi-1)$
N_{15}	$= 32\eta\xi\zeta(4\eta-1)$

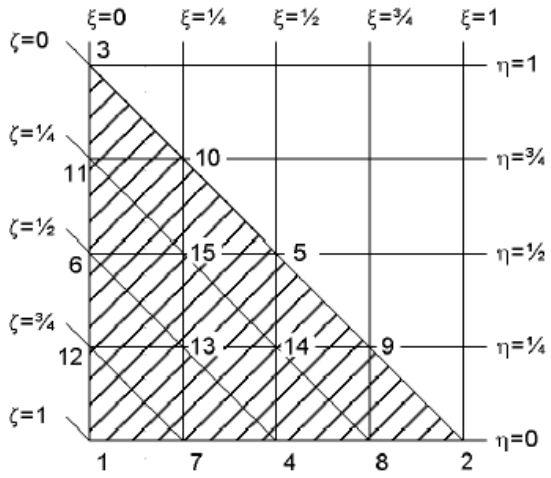


Figure 7: Local numbering and positioning of nodes of a 15-node triangular element

6.3.2 *Structural elements*

Structural area elements in the PLAXIS program, i.e. plates and interfaces are based on the are elements as described in the previous sections. However there are some differences.

6.3.2.1 *Plate elements*

Plate elements are different from the 6-node triangles which have three degrees of freedom per node. As the plate elements cannot sustain torsional moments, the plate elements have only 5 d.o.f per node in the rotated coordinate system, i.e.:

- one axial displacement (u_x^*);
- two transverse displacements (u_y^* and u_z^*);
- two rotations (φ_y^* and φ_z^*).

These elements are directly integrated over their cross section and numerically integrated using 3 point Gaussian integration. The position of the integration points is indicated in Figure 8 (on page 44) and corresponds with Table 3 (on page 45).

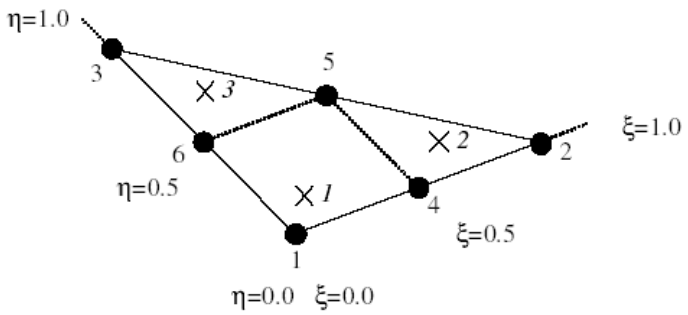


Figure 8: Local numbering and positioning of nodes (•) and integration points (x) of a 6-node plate triangle.•) and integration points (x) of a 6-node plate triangle.

6.3.2.2 *Interface elements*

Differently from the plate elements, interface elements have pairs of nodes instead of single nodes. The interface elements are numerically integrated using 6 point Gauss integration. The distance between the two nodes of a node pair is zero. Each node has three translational degrees of freedom (u_x, u_y, u_z). As a result, interface elements allow for differential displacements between the node pairs (slipping and gapping). The position and weight factors of the integration points are given in Table 2 (on page 44).

Table 2: 6-Point Gaussian integration for 12-node triangular elements

Point	ξ_i	η_i	w_i
1	0.091576...	0.816848	0.109952
2	0.091576	0.091576	0.109952
3	0.816848	0.091576	0.109952
4	0.108103	0.445948	0.223382
5	0.445948	0.108103	0.223382
6	0.445948	0.445948	0.223382

For more information see Dunavant (1985) (on page 68).

6.3.3 *Numerical integration of area elements*

As for line elements, one can formulate the numerical integration over areas as:

$$\iint F(\xi, \eta)\, d\,\xi\, d\,\eta \approx \sum_{i=1}^{k} F(\xi_i, \eta_i) w_i \qquad \textit{Eq. [166]}$$

The PLAXIS program uses Gaussian integration within the area elements.

6.3.3.1 *6-node triangular elements*

For 6-node triangular elements the integration is based on 3 sample points (Figure 6 (on page 42)). The position and weight factors of the integration points are given in Table 3 (on page 45). Note that the sum of the weight factors is equal to 1.

Table 3: 3-point Gaussian integration for 6-node triangular elements

Point	ξ_i	η_i	w_i
1	1/6	2/3	1/3
2	1/6	1/6	1/3
3	2/3	1/6	1/3

6.3.3.2 *15-node triangular elements*

For 15-node elements 12 sample points are used. The position and weight factors of the integration points are given in Table 4 (on page 45). Note that, in contrast to the line elements, the sum of the weight factors is equal to 1.

Table 4: 12-point Gaussian integration for 15-node triangular elements

Point	ξ_i	η_i	ζ_i	w_i
1, 2 & 3	0.063089...	0.063089...	0.873821...	0.050845...
4...6	0.249286...	0.249286...	0.501426...	0.116786...
7...12	0.310352...	0.053145...	0.636502...	0.082851...

6.4 Interpolation functions and numerical integration of volume elements

The soil volume in the PLAXIS program is modelled by means of 10-node tetrahedral elements. The interpolation functions, their derivatives and the numerical integration of this type of element are described in the following subsections.

6.4.1 *10-node tetrahedral element*

The 10-node tetrahedral elements are created in the 3D mesh procedure. This type of element provides a second-order interpolation of displacements. For tetrahedral elements there are three local coordinates (ξ, η and ζ). The shape functions *Ni* have the property that the function value is equal to unity at node *i* and zero at the other nodes. The shape functions of these 10-node volume elements can be written as (see the local node numbering as shown in Figure 9 (on page 46)):

N_1	$= (1-\xi-\eta-\zeta)(1-2\xi-2\eta-2\zeta)$
N_2	$= \zeta(2\zeta-1)$
N_3	$= \xi(2\xi-1)$
N_4	$= \eta(2\eta-1)$
N_5	$= 4\zeta(1-\xi-\eta-\zeta)$
N_6	$= 4\xi\zeta$
N_7	$= 4\xi(1-\xi-\eta-\zeta)$
N_8	$= 4\eta(1-\xi-\eta-\zeta)$
N_9	$= 4\eta\zeta$
N_{10}	$= 4\xi\eta$

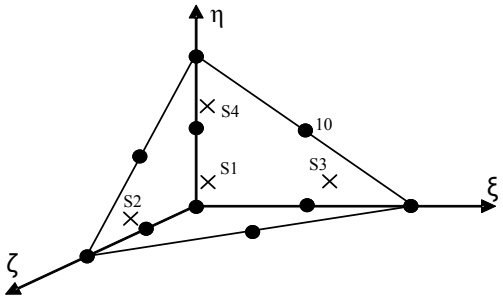


Figure 9: Local numbering and positioning of nodes (•) and integration points (x) of a 10-node tetrahedral elementx) of a 10-node tetrahedral element

The soil elements have three degrees of freedom per node: u_x, u_y and u_z. The shape function matrix $\mathbf{N}_i$ can now be defined as:

$$\mathbf{N}_i = \begin{bmatrix} N_i & 0 & 0 \\ 0 & N_i & 0 \\ 0 & 0 & N_i \end{bmatrix} \qquad \textit{Eq. [167]}$$

and the nodal displacement vector $\underline{v}_i$ is defined as:

$$\underline{v}_i = \begin{bmatrix} v_{ix} & v_{iy} & v_{iz} \end{bmatrix}^T \qquad \textit{Eq. [168]}$$

6.4.2 *Derivatives of interpolation functions*

In order to calculate Cartesian strain components from displacements, such as formulated in Eq. [10], derivatives need to be taken with respect to the global system of axes (*x,y,z*).

$$\underline{\varepsilon} = \mathbf{B}_i \upsilon_i \qquad \textit{Eq. [169]}$$

where

$$\mathbf{B} = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_i}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_i}{\partial z} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} & 0 \\ 0 & \frac{\partial N_i}{\partial z} & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} & 0 & \frac{\partial N_i}{\partial x} \end{bmatrix} \qquad \textit{Eq. [170]}$$

Within the elements, derivatives are calculated with respect to the local coordinate system (ξ, η, ζ).

The relationship between local and global derivatives involves the Jacobian *J*:

$$\begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} = J \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} \qquad \textit{Eq. [171]}$$

Or inversely:

$$\begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} = J^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{bmatrix} \qquad \textit{Eq. [172]}$$

The local derivatives $\partial N_i / \partial \xi$, *etc.*, can easily be derived from the element shape functions, since the shape functions are formulated in local coordinates. The components of the Jacobian are obtained from the differences in nodal coordinates. The inverse Jacobian J^{-1} is obtained by numerically inverting *J*. The Cartesian strain components can now be calculated by summation of all nodal contributions:

$$\begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{bmatrix} = \sum_i \mathbf{B}_i \begin{bmatrix} v_{ix} \\ v_{iy} \\ v_{iz} \end{bmatrix} \qquad Eq.\ [173]$$

where v_i are the displacement components in node *i*.

6.4.3 *Numerical integration of volume elements*

As for line and areas, one can formulate the numerical integration over volumes as:

$$\iiint F(\xi,\ \eta,\ \zeta)\ d\ \xi\ d\ \eta\ d\ \zeta \approx \sum_{i=1}^{k} F(\xi_i,\ \eta_i,\ \zeta_i) w_i \qquad Eq.\ [174]$$

The PLAXIS program uses Gaussian integration within the tetrahedral elements. The integration is based on 4 sample points. The position and weight factors of the integration points are given in Table 5 (on page 48). See Figure 9 (on page 46) for the local numbering of integration points. Note that the sum of the weight factors is equal to 1/6.

Table 5: 4-point Gaussian integration for 10-node triangular elements

Point	ξ_i	η_i	ζ_i	w_i
1	0.138197...	0.138197...	0.138197...	1/24
2	0.138197...	0.138197...	0.585410...	1/24
3	0.585410...	0.138197...	0.138197...	1/24
4	0.138197...	0.585410...	0.138197...	1/24

6.4.4 *Calculation of element stiffness matrix*

The element stiffness matrix, $\mathbf{K}^e$, is calculated by the integral (see also Eq. [25]):

$$\mathbf{K}^e = \int \mathbf{B}^T \mathbf{D}^e \mathbf{B}\ d\ V \qquad Eq.\ [175]$$

As it is more convenient to calculate the element stiffness matrix in the local coordinate system, the change of variables theorem should be applied to change the integral to the local coordinate system:

$$\mathbf{K}^e = \int \mathbf{B}^T \mathbf{D}^e \mathbf{B} j\ d\ V^* \qquad Eq.\ [176]$$

where *j* denotes the determinant of the Jacobian.

The integral is estimated by numerical integration as described in Numerical integration of volume elements (on page 48). In fact, the element stiffness matrix is composed of submatrices $\mathbf{K}_{ij}^{e}$ where *i* and *j* are the local nodes. The process of calculating the element stiffness matrix can be formulated as:

$$\mathbf{K}_{ij}^{e} = \sum_{k} \mathbf{B}_{i}^{T} D^{e} \mathbf{B}_{j} j w_{k} \qquad Eq.\ [177]$$

In case of plastic deformations of the soil only the elastic part of the soil stiffness will be used in the stiffness matrix whereas the plasticity is solved for iteratively.

6.5 Special elements *(PLAXIS 3D)*

As special elements in PLAXIS embedded beams will be considered. Embedded beams are based on the embedded beam approach by Sadek & Shahrour (2004) (on page 69). Embedded beams consist of beam elements to model a pile, anchor or rock bolt and embedded interface elements to model the interaction between the soil and the beam at the beam skin as well as at the foot of a pile.

6.6 *Embedded beams (PLAXIS 3D)*

The embedded beam has been developed to describe the interaction of a pile, anchor, or rock bolt with its surrounding soil or rock. The interaction at the skin and at the foot is described by means of embedded interface elements. The pile, anchor, or rock bolt is considered as a beam which can cross a volume element at any place with any arbitrary orientation (Figure 10 (on page 49)). Due to the existence of the beam element three extra nodes are introduced inside the volume element.

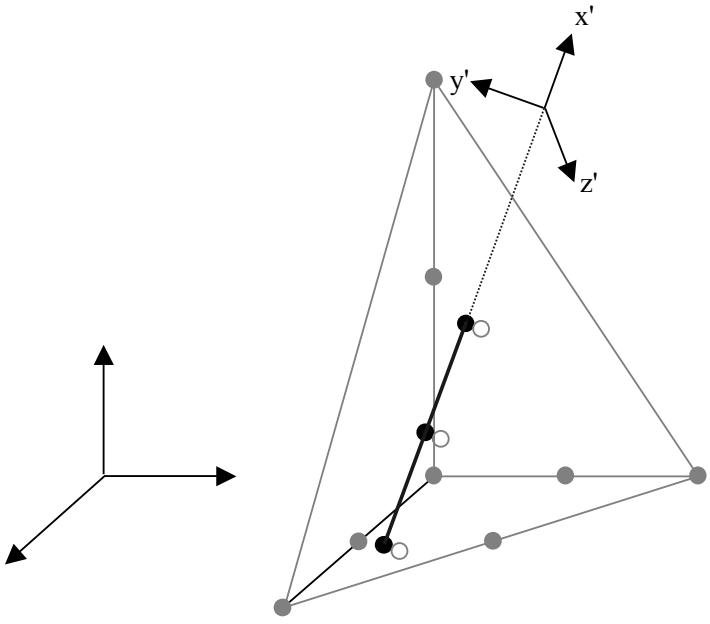


Figure 10: Illustration of the embedded beam element denoted by the solid line. The blank grey circles denote the virtual nodes of the soil element.

6.6.1 *Finite element discretisation*

The finite element discretisation of the pile or rock bolt is similar to beam elements, as discussed in Beam elements (PLAXIS 3D) (on page 40). The finite element discretisation of the interaction with the soil will be discussed in this chapter. Using the standard notation the displacement of the soil $\underline{u}_s$ and the displacement of the beam $\underline{u}_b$ can be discretised as:

$$\underline{u}_s = \mathbf{N}_s \underline{v}_s \qquad \underline{u}_b = \mathbf{N}_b \underline{v}_b \qquad Eq.\ [178]$$

where $\mathbf{N}_s$ and $\mathbf{N}_b$ are the matrices containing the interpolation functions of the soil elements and the beam elements respectively (see 10-node tetrahedral element (on page 45) and Beam elements (PLAXIS 3D) (on page 40)) and $\underline{v}_s$ and $\underline{v}_b$ are the nodal displacement vectors of the soil elements and the beam elements respectively.

6.6.2 *Interaction at the skin*

First, the interaction between the soil and the beam at the beam skin surface will be described by embedded interface elements. These interface elements are based on 3-node line elements with pairs of nodes instead of single nodes. One node of each pair belongs to the beam element, whereas the other (virtual) node is a point in the 10-node tetrahedral element (Figure 10 (on page 49)). The interaction can be represented by a skin traction $\underline{t}^{skin}$. The development of the skin traction can be regarded as an incremental process:

$$\underline{t}^{skin} = \underline{t}_0^{skin} + \Delta\underline{t}^{skin} \qquad Eq.\ [179]$$

In this equation $\underline{t}_0^{skin}$ denotes the initial skin traction and $\Delta\underline{t}^{skin}$ denotes the skin traction increment. The constitutive relation between the skin traction increment and the relative displacement increment is formulated as:

$$\Delta\underline{t}^{skin} = \mathbf{T}^{skin} \Delta\underline{u}_{rel} \qquad Eq.\ [180]$$

In this relation $\mathbf{T}^{skin}$ denotes the material stiffness of the embedded interface element in the global coordinate system. The increment in the relative displacement vector $\Delta\underline{u}_{rel}$ is defined as the difference in the increment of the soil displacement and the increment of the beam displacement:

$$\Delta\underline{u}_{rel} = \Delta\underline{u}_b - \Delta\underline{u}_s = \mathbf{N}_b \Delta\underline{v}_b - \mathbf{N}_s \Delta\underline{v}_s = \mathbf{N}_{rel} \Delta\underline{v}_{rel} \qquad Eq.\ [181]$$

where

$$\mathbf{N}_{rel} = \begin{bmatrix} \mathbf{N}_b & -\mathbf{N}_s \end{bmatrix} \qquad Eq.\ [182]$$

and

$$\Delta\underline{v}_{rel} = \begin{bmatrix} \Delta\underline{v}_b \\ \Delta\underline{v}_s \end{bmatrix} \qquad Eq.\ [183]$$

Looking to the virtual work equation (Eq. [6]) the traction increment at the beam skin can be discretised as:

$$\int \delta \underline{u}_{rel}^{T} \Delta \underline{t}^{skin} dS = \delta \underline{v}_{rel}^{T} \int \mathbf{N}_{rel}^{T} \mathbf{T}^{skin} \mathbf{N}_{rel} dS \Delta \underline{v}_{rel} = \mathbf{K}^{skin} \Delta \underline{v}_{rel}$$

In this formulation the element stiffness matrix $\mathbf{K}^{skin}$ represents the interaction between the beam and the soil at the skin and consists of four parts:

$$\mathbf{K}^{skin} = \begin{bmatrix} \mathbf{K}_{bb}^{skin} & \mathbf{K}_{bs}^{skin} \\ \mathbf{K}_{sb}^{skin} & \mathbf{K}_{ss}^{skin} \end{bmatrix} \qquad Eq.\ [184]$$

The matrix $\mathbf{K}_{bb}^{skin}$ represents the contribution of the beam nodes to the interaction, the matrix $\mathbf{K}_{ss}^{skin}$ represents the contribution of the soil nodes to the interaction and the matrices $\mathbf{K}_{bs}^{skin}$ and $\mathbf{K}_{sb}^{skin}$ are the mixed terms:

$$\begin{aligned} \mathbf{K}_{bb}^{skin} &= \int \mathbf{N}_b^T \mathbf{T}^{skin} \mathbf{N}_b \, d\, S \\ \mathbf{K}_{bs}^{skin} &= \int \mathbf{N}_b^T \mathbf{T}^{skin} \mathbf{N}_s \, d\, S \\ \mathbf{K}_{sb}^{skin} &= \int \mathbf{N}_s^T \mathbf{T}^{skin} \mathbf{N}_b \, d\, S \\ \mathbf{K}_{ss}^{skin} &= \int \mathbf{N}_s^T \mathbf{T}^{skin} \mathbf{N}_s \, d\, S \end{aligned} \qquad Eq.\ [185]$$

These integrals are numerically estimated using:

$$\int_{\xi=-1}^{1} F(\xi) \, d\, \xi \approx \sum_{i=1}^{k} F(\xi_i) w_i \qquad Eq.\ [186]$$

However, instead of Gauss integration Newton-Cotes integration is used. In this method the points ξ_i are chosen at the position of the nodes, see Table 6 (on page 51). The type of integration used for the embedded interface elements is shaded. In case of plastic deformations of the embedded interface elements only the elastic part of the interface stiffness will be used in the stiffness matrix whereas the plasticity is solved for iteratively.

Table 6: Newton-Cotes integration

Nodes	ξ_i	w_i
2 nodes	± 1	1
3 nodes	± 1, 0	1/3, 4/3
4 nodes	± 1,± 1/3	1/4, 3/4
5 nodes	± 1,± 1/2, 0	7/45, 32/45, 12/45

6.6.3 *Interaction at the foot*

The interaction of the embedded beam at the foot is described by an embedded interface element. This interaction can be represented by a foot force vector $\underline{t}^{foot}$. Like the development of the skin traction the development of the foot force is an incremental process:

$$\underline{t}^{foot} = \underline{t}_0^{foot} + \Delta \underline{t}^{foot} \qquad Eq.\ [187]$$

In this equation $\underline{t}_0^{foot}$ denotes the initial force and $\Delta\underline{t}^{foot}$ denotes the force increment at the foot. The constitutive relation between the skin traction increment and the relative displacement increment is formulated as:

$$\Delta\underline{t}^{foot} = \mathbf{D}^{foot}\Delta\underline{u}_{rel} \qquad Eq.\ [188]$$

In this relation $\mathbf{D}^{foot}$ denotes the material stiffness matrix of the spring element at the foot of the embedded beam in the global coordinate system. As for the skin interaction the force increment at the foot of the beam can be discretised by means of the virtual work (Eq. [6]), as:

$$\delta u_{rel}^T \Delta t^{foot} = \delta v_{rel}^T \mathbf{N}_{rel}^T \mathbf{D}^{foot} \mathbf{N}_{rel} \Delta\underline{v}_{rel} = \mathbf{K}^{foot}\Delta\underline{v}_{rel} \qquad Eq.\ [189]$$

The stiffness matrix at the foot is represented by $\mathbf{K}^{foot}$ and consists of four parts:

$$\mathbf{K}^{foot} = \begin{bmatrix} \mathbf{K}_{bb}^{foot} & \mathbf{K}_{bs}^{foot} \\ \mathbf{K}_{sb}^{foot} & \mathbf{K}_{ss}^{foot} \end{bmatrix} \qquad Eq.\ [190]$$

In this equation $\mathbf{K}_{bb}^{foot}$ represents the contribution of the beam nodes, $\mathbf{K}_{ss}^{foot}$ represents the contribution from the soil nodes and $\mathbf{K}_{bs}^{foot}$ and $\mathbf{K}_{sb}^{foot}$ are the mixed terms:

$$\begin{aligned} \mathbf{K}_{bb}^{foot} &= \mathbf{N}_b^T \mathbf{D}^{foot} \mathbf{N}_b \\ \mathbf{K}_{bs}^{foot} &= -\mathbf{N}_b^T \mathbf{D}^{foot} \mathbf{N}_s \\ \mathbf{K}_{sb}^{foot} &= -\mathbf{N}_s^T \mathbf{D}^{foot} \mathbf{N}_b \\ \mathbf{K}_{ss}^{foot} &= \mathbf{N}_s^T \mathbf{D}^{foot} \mathbf{N}_s \end{aligned} \qquad Eq.\ [191]$$

In case of plastic deformations of the embedded interface element only the elastic part of the interface stiffness will be used in the stiffness matrix whereas the plasticity is solved for iteratively.

7 Convergence of non-linear calculations in PLAXIS

Soil behaviour is usually not-linear which implies that either plasticity occurs or that the associated material stiffness matrix is not linear (i.e stiffness can depend on: stiffness, strain, suction, temperature etc). Therefore, the system of equations must be solved incrementally using an iterative process (see Calculation Process (on page 73) and Understanding the iterative convergence process (on page 75)). This process can be considered sufficiently accurate if the global, local residual and integrated local error are less than a defined tolerated error. Depending on the type of problem modelled, PLAXIS also considers additional criteria to enforce that both globally and locally out-of-balance forces are close to zero.

PLAXIS solves different model global errors (out-of-balance) depending on the physical nature of the problem, as shown below:

Physical problem	Field Variable	Conjugate flux
Soil stress analysis	Displacement (u)	Force (F)
Structural stress analysis	Rotation (Φ)	Moment (M)
Groundwater flow analysis	Pore water pressure (p)	GW Flow (q_{GW})

In general, the global error should always be less than a tolerated error:

$$GlobalError \; < \; Tolerated \; Error$$

Note:

- The tolerated error can be provided by the user, PLAXIS usually uses a default value of 0.01 or 1%. In PLAXIS, the tolerated error for each criterion checked upon is calculated based on the global error.
- Coupled problems require to fulfill the convergence criteria for each field independently except for consolidation analysis, for which only soil stress and eventually structural stresses if structural elements have been activated in a consolidation phase.

Due to the high non-linearity nature of the physical problems modelled, the obtention of accurate non-linear solutions require not only the consideration of global criteria but also additional local convergence criteria that will be presented in the following sections.

7.1 Convergence criteria for deformation analysis

The criteria evaluated to satisfy the convergence of finite element solutions in deformation analysis are divided in three main categories **Global error checking**, **Local error checking**, and **Integrated local error**.. The convergence criteria for deformation analysis are applicable for the following PLAXIS analyses:

- Plastic analysis
- Consolidation analysis
- Dynamic analysis
- Safety analysis
- Fully coupled analysis (deformation component)

Also, as an overview, the global and local error checking criteria for deformation are subdivided as follows:

1. **Global error criteria**
   - Check on force residuals
   - Check on moment residuals
2. **Local error criteria**
   - Inaccurate plastic points for soil elements
   - Inaccurate plastic points for interface elements
   - Inaccurate plastic points for beams
   - Inaccurate plastic points for embedded beams
   - Inaccurate plastic points for plates
   - Inaccurate plastic points for anchors
   - Inaccurate plastic points for geogrids
3. **Integrated (or total) local error criteria**
   - Accuracy in the plastic zone
   - Accuracy for embedded beams

7.1.1 Global error criteria

7.1.1.1 Check on force residuals

A global error indicator for nodal forces is systematically calculated with the consideration of the total number of degrees of freedom:

$$GlobalErrorForce = \frac{\| r \|_2}{CSP \| f_{ext}^{inact} \|_2 + \| f_{\text{int}}^{act} \|_2} = \frac{\| f_{\text{int}} - f_{ext} \|_2}{CSP \| f_{ext}^{inact} \|_2 + \| f_{\text{int}}^{act} \|_2} \qquad \textit{Eq. [192]}$$

where

$\| . \|_2$ = Quadratic norm

And CSP (Current Stiffness Parameter) is defined as:

$$\text{CSP} = \frac{change\ of\ elastic\ strain\ energy\ increment}{change\ of\ total\ strain\ energy\ increment} = \frac{\int_{V_{act}} \Delta\varepsilon D^{e} dV}{\int_{V_{act}} \Delta\varepsilon \Delta\sigma dV} \qquad Eq.\ [193]$$

where

$$ToleratedForceError = ToleratedError$$

Hint: In PLAXIS the *ToleratedError* is set by default as 0.01.

7.1.1.2 Check on moment residuals

A global error indicator for moment in structural elements is calculated with respect to Moment for structural elements having rotational degrees of freedom (dof).

$$GlobalErrorMoment = \frac{\| m_{\text{int}} - m_{ext} \|_{\infty}}{m_{ref}} \qquad Eq.\ [194]$$

where

$\| . \|_{\infty}$ = Infinite norm.

m_{ref} = Moment contribution of each external force component.

The reference moment m_{ref} is computed as follows:

$$m_{ref} = \sum_{el} \int \left| B_{beam}^{T} \sigma \right|_{moment} dV \qquad Eq.\ [195]$$

The convergence criteria for moment reads:

$$GlobalErrorMoment < ToleratedMomentError$$

By default:

$$ToleratedMomentError = ToleratedError$$

Note: This criterium only applies to structural elements with rotational degrees of freedom.

7.1.2 Local error criteria

PLAXIS also enforces several local convergence checks, where local error indicators are determined mainly based on the information derived from plastic points .

A point is defined as plastically inaccurate if:

$$PlasticLocalError > ToleratedPlasticLocalError$$

where the *PlasticLocalError* is calculated differently depending on the type of elements. This concept is considered for:

- soil elements
- interface elements

Convergence is satisfied if:

$$Number\ of\ inaccurate\ soil\ plastic\ points < ToleratedInaccurateSoilPlasticPointPercentage*number\ of\ soil\ plastic\ points + 3$$

and

$$Number\ of\ inaccurate\ interface\ plastic\ points < ToleratedInaccurateInterfacePlasticPointPercentage*number\ of\ soil\ plastic\ points + 3$$

Hint:

By default:

$ToleratedInaccurateSoilPlasticPointPercentage$ and $ToleratedInaccurateInterfacePlasticPointPercentage$ are set to 0.1

Note:

- The criteria checks should be satisfied for each independent counting.
- Only inaccurate plastic points for soil and interface are being reported in the calculation progress window.

In PLAXIS 3D additional independent inaccurate plastic points counting are also considered for the following structural elements:

- Beams
- Plates
- Anchors
- Geotextiles

7.1.2.1 Inaccurate plastic points for soil elements

For soil elements, PLAXIS calculates the following quantities (or values) for each stress point:

$$SoilPlasticLocalError_{\ j} = \frac{\| \ \sigma_{eq,j} - \sigma_{c,j} \|}{\max\left(\tau_{\max,j},\ c,\ 1\ kPa\right)} \qquad Eq.\ [196]$$

where (see Figure 11 (on page 57)):

$$\sigma_{c,j} = \sigma_0 + D^e(\Delta\varepsilon_j - \Delta\varepsilon_{p,j}) = \sigma_0 + D^e(\Delta\varepsilon_{e,j})$$
$$\sigma_{eq,j} = \sigma_{c,j-1} + D^e\delta\varepsilon_j \qquad \textit{Eq. [197]}$$

The equilibrium stress is calculated based on the linearized soil behaviour at FE model level whereas the constitutive stress is the one computed by consideration of the real soil constitutive behaviour at stress point.

Soil plastic stress points *j* are inaccurate if:

$$SoilPlasticLocalError_j > ToleratedSoilPlasticLocalError$$

By default:

$$ToleratedSoilPlasticLocalError = ToleratedError$$

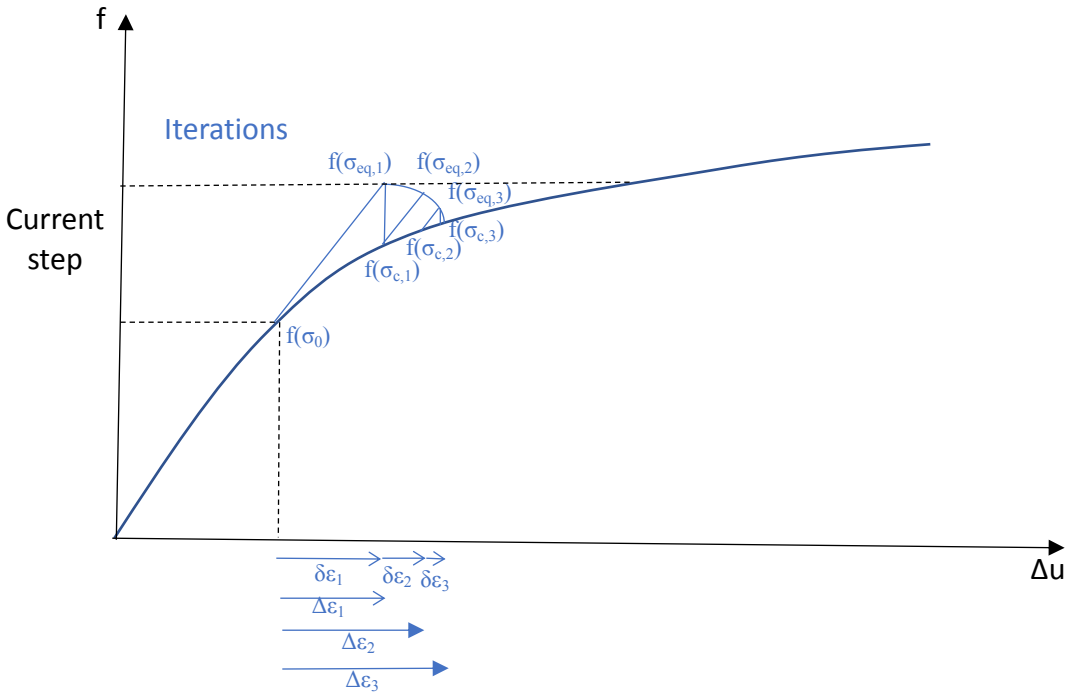


Figure 11: Equilibrium and constitutive stress concept

In PLAXIS 3D convergence check on number of inaccurate soil plastic points are skipped if:

$$GlobalErrorForcer_j < 0.01 \times ToleratedError$$

7.1.2.2 Inaccurate plastic points for interfaces

For interface elements, PLAXIS also calculates the following quantity for each stress point:

$$InterfacePlasticLocalError_j = \frac{\| \tau_{eq,j} - \tau_{c,j} \|}{\max(\tau_{\max,j},\ c,\ 1kPa)} \qquad \textit{Eq. [198]}$$

Interface plastic points *j* are inaccurate if:

$$InterfacePlasticLocalError_{j} > ToleratedInterfacePlasticLocalError$$

By default:

$$ToleratedInterfacePlasticLocalError = ToleratedError$$

Note:

- Convergence checks on number of inaccurate interface plastic points are skipped if:

$$GlobalErrorForce < 0.1 * ToleratedError$$

7.1.2.3 Inaccurate plastic points for beams

For beam elements, PLAXIS calculates the following error for each stress point:

$$BeamPlasticLocalError_{j} = \frac{\| \sigma_{eq,j}^{normal} - \sigma_{c,j}^{normal} \|}{\| \sigma_{eq,j}^{normal} \|} \qquad Eq.\ [199]$$

Beam plastic points *j* are inaccurate if:

$$BeamPlasticLocalError_{j} > ToleratedBeamPlasticLocalError$$

By default:

$$ToleratedBeamPlasticLocalError = ToleratedError$$

7.1.2.4 Inaccurate plastic points for plates

For plate elements, PLAXIS 3D calculates the following for each stress point:

$$PlatePlasticLocalError_{j} = \frac{\| F_{eq,j}^{in-plane} - F_{c,j}^{in-plane} \|}{\| F_{eq,j}^{in-plane} \|} \qquad Eq.\ [200]$$

Plate plastic points *j* are inaccurate if:

$$PlatePlasticLocalError_{j} > ToleratedPlatePlasticLocalError$$

BY default:

$$ToleratedPlatePlasticLocalError = ToleratedError$$

7.1.2.5 Inaccurate plastic points for anchors

For anchor elements, PLAXIS 3D calculates the following for each stress point:

$$AnchorPlasticLocalError_j = \frac{\| F_{eq,j} - F_{c,j} \|}{\| F_{eq,j} \|} \qquad Eq.\ [201]$$

Anchor plastic points *j* are inaccurate if:

$$AnchorPlasticLocalError_j > ToleratedAnchorPlasticLocalError$$

BY default:

$$ToleratedAnchorPlasticLocalError = ToleratedError$$

7.1.2.6 Inaccurate plastic points for geogrids

For geogrid elements, PLAXIS 3D calculates the local error quantities for each stress point. Local error calculation depends here whether the material is defined as isotropic.

$$GeogridPlasticLocalError_j = \frac{\| F_{eq,j}^{in-plane} - F_{c,j}^{in-plane} \|}{\| F_{eq,j}^{in-plane} \|} \qquad Eq.\ [202]$$

or anisotropic:

$$GeogridPlasticLocalError_{1,j} = \frac{\| \sigma_{eq,j}^{1} - \sigma_{c,j}^{1} \|}{\| \sigma_{eq,j}^{1} \|} \qquad Eq.\ [203]$$

$$GeogridPlasticLocalError_{2,j} = \frac{\| \sigma_{eq,j}^{2} - \sigma_{c,j}^{2} \|}{\| \sigma_{eq,j}^{2} \|} \qquad Eq.\ [204]$$

Geogrid plastic points *j* are inaccurate if:

$$GeogridPlasticLocalError_j > ToleratedGeogridPlasticLocalError$$

By default:

$$ToleratedGeogridPlasticLocalError = ToleratedError$$

In PLAXIS 3D convergence check on number of inaccurate geogrid plastic points are ignored if:

$$GlocabErrorForce < 0.01 * ToleratedError$$

7.1.2.7 Accuracy for embedded beams

For embedded beams additional convergence criteria are considered to check the accuracy of couplings springs (or special interfaces).

In PLAXIS, an embedded beam error is computed for each individual embedded beam element *p* as:

$$\text{EmbeddedBeamInterfaceError}_p = \frac{\left| F^p_{foot,eq} + \int_{Length} T^p_{skin,eq} dl - F^p_{foot,c} - \int_{Length} T^p_{skin,c} dl \right|}{\max\left(F_{\max},\ 1.0\right)} \qquad \textit{Eq. [205]}$$

where:

$$F_{\max} = \max_{p=1,N_{eb}} \left(\left| F^p_{foot,eq} + \int_{Length} T^p_{skin,eq} dl \right| ,\ \left| F^p_{foot,c} + \int_{Length} T^p_{skin,c} dl \right| \right) \qquad \textit{Eq. [206]}$$

The concept of constitutive and equivalent forces for $F_{foot,c}$, $F_{foot,eq}$and T_{skin} is also considered in the same fashion as for stress in soil (Figure 11 (on page 57)). The equilibrium forces are calculated based on the linearized pile tip/skin behaviour whereas the constitutive forces are the ones computed by consideration of real constitutive behaviour of coupling springs along embedded beam length and at embedded beam tip.

This way inaccurate 3D embedded beams could be identified when:

$$EmbeddedBeamInterfaceError_p > ToleratedEmbeddedBeamInterfaceError$$

By default:

$$ToleratedEmbeddedBeamInterfaceError = \begin{cases} ToleratedError\ if\ GlobalError & > & ToleratedError \\ 5 \times ToleratedError\ if\ GlobalError & < & 0.01 * ToleratedError \\ 2 \times ToleratedError & & Otherwise \end{cases}$$

The convergence criteria are relatively strict in PLAXIS 3D as no inaccurate embedded beam are allowed which can be expressed as:

$$\max_{p=1,N_{eb}} \left(\text{EmbeddedBeamInterfaceError}_p \right) < \text{ToleratedEmbeddedBeamInterfaceError}$$

7.1.3 Integrated local error - Accuracy in the plastic zone

7.1.3.1 Accuracy in the plastic zone

The plastic global error also called the integrated local error is defined as:

$$PlasticGlobalError = \frac{\Sigma_{j=1,N} \int_{V_{p,j}} (PlasticLocalError_j) dV}{\Sigma_{j=1,N} \int_{V_{p,j}} dV} \qquad Eq.\ [207]$$

where $PlasticLocalError_j$ has been defined in Local error criteria (on page 55).

Converge is satisfied if:

$$PlasticGlobalError < ToleratedPlasticGlobalError$$

$$ToleratedPlasticGlobalError = TolerateError$$

The plastic error is calculated in PLAXIS 3D only for:

1. Soil elements for plastic points only with
   $$SoilPlasticGlobalError < ToleratedSoilPlasticGlobalError$$
2. Interfaces
   $$InterfacePlasticGlobalError < ToleratedInterfacePlasticGlobalError$$
3. Anchors
   $$AnchorPlasticGlobalError < ToleratedAnchorPlasticGlobalError$$
4. Geogrids
   $$GeotextileGeogridPlasticGlobalError < ToleratedGeogridPlasticGlobalError$$

Unless:

$$GlobalErrorForce < 0.01 * Tolerated\,Error$$

7.2 Convergence criteria for flow analysis

The convergence criteria for flow analysis are applicable for the following PLAXIS analyses:

- Steady state flow.
- Transient flow.
- Fully coupled analysis.

As an overview of the PLAXIS convergence criteria for flow analysis are presented as follows:

1. Global error criteria
   - Flow error check
   - Change of heat storage for thermal flow calculation check

- Unsaturated behaviour convergence checks for groundwater flow analysis

2. Local error indicators for groundwater flow calculation

- Inaccurate nodes for boundary conditions
  - Inaccurate seepage nodes
  - Inaccurate well (extraction) nodes
  - Inaccurate drain nodes
  - Inaccurate ponding nodes
- Maximum allowable number of iterations for checking status change.

3. Special case: Steady-state calculation criteria

- Local groundwater flow error
- Average pore pressure change
- Maximum pore pressure change
- Inaccurate nodes for specific boundary conditions

7.2.1 Global error criteria

7.2.1.1 Flow error check

A global error indicator for nodal flux is systematically calculated with the consideration of the total number of degrees of freedom:

$$GlobalFlowError = \frac{\Sigma \| \; q_{init} - q_{ext} \|_{\; closed\ nodes}}{\max\left(\Sigma \| \; q_{ext}^{out} \|_{\; open \;\; nodes}, \Sigma \| \; q_{ext}^{in} \|_{\; open \;\; nodes}\right)} \qquad Eq.\ [208]$$

Note: Fluxes are computed differently in groundwater flow than in heat flow (Darcy flux *GWFlowError* vs heat flux *ThFlowErr*).

Convergence is satisfied if:

$$GlobalFlowError < TolerGlobalFlowError$$

Hint: *TotalGlobalFlowError* is by default set to 0.01. This condition is enforced for each (pseudo) time step.

7.2.1.2 Unsaturated behaviour convergence check

In case of unsaturated behavior in groundwater flow analysis, additional checks are being performed as the hydraulic conductivity and storativity. Matrices are computed at the beginning of each time step assuming particular values for the degree-of-saturation and relative permeability. A time step is judged valid if:

$$AverageSaturationChange < ToleratedSaturationChange$$

and

$$AverageKrChange < ToleratedKrChange$$

where:

$$AverageKrChange < ToleratedKrChange$$

$$AverageSaturationChange = \frac{\| Sat_{new} - Sat_{old} \|}{\| Sat_{old} \|} \qquad Eq.\ [209]$$

and

$$AverageKrChange = \frac{\| K_{rel,new} - K_{rel,old} \|}{\| K_{rel,old} \|} \qquad Eq.\ [210]$$

In PLAXIS, it is considered:

$$ToleratedKrChange = ToleratedSaturationChange = ToleratedGlobalFlowError$$

Some additional checks are also being considered to prevent excessive changes of the relative permeability and/or degree of saturation in the unsaturated zone:

$$AverageSaturationChange < ExcessiveSaturationChange$$

and

$$AverageKrChange < ExcessiveKrChange$$

with

$$ExcessiveKrChange = ExcessiveSaturationChange = 10 * TolerGlobalFlowError$$

If excessive changes of degree-of-saturation or relative permeability are being detected, then a new step attempt will be taken with a smaller time step size (downscaling factor is 1.5)

If the resulting changes of degree-of-saturation and/or relative permeabilities after the step size increase leads to:

$$AverageSaturationChange > ExcessiveSaturationChange$$

and

$$AverageKrChange > ExcessiveKrChange$$

Then the upscaling is ignored, and the step size is scaled back to its previous value.

7.2.2 Local error criteria

As for deformation analysis, flow analysis in PLAXIS also enforces additional local convergence check. Those are related to monitoring inaccurate nodes for specific boundary conditions for groundwater flow calculation.

7.2.2.1 Inaccurate nodes for boundary conditions

a. Inaccurate seepage nodes:

Seepage nodes can be open or closed. If water wants to flow out, the pore pressure should be zero. If the water tries to flow in, the node should be closed. In PLAXIS, a seepage node which change status between two consecutive time steps is set as inaccurate.

The following convergence criteria is then enforced:

$$Numbers\ of\ inaccurate\ seepage\ nodes\ < \max(int(ToleratedInaccurateSeepageNodesPercentage * numbers\ of\ seepage\ nodes),\ 1)$$

Hint:

$$ToleratedInaccurateSeepageNodesPercentage = 0.001$$

b. Inaccurate well (extraction) nodes:

If the water head in the well nodes goes below minimum specified head and if the active pore pressure in the same node is positive, then the pore pressure in the well node is set to P_{max}. A well node which change status between two consecutive time steps is set as inaccurate. The corresponding convergence criteria on inaccurate well nodes reads:

$$Number\ of\ inaccurate\ well\ nodes\ < 1$$

c. Inaccurate drain nodes:

If inflow is being detected in a drain node then it is switched to closed node. A drain node which change status between two consecutive time steps is set as inaccurate.

The corresponding convergence criteria on inaccurate drain nodes reads:

$$Number\ of\ inaccurate\ drain\ nodes < 1$$

d. Inaccurate ponding nodes:

If the head to get prescribed inflow is exceeded the nodes BC is changed to prescribed head. A well node which change status between two consecutive time steps is set as inaccurate.

The corresponding convergence criteria on inaccurate drain nodes reads:

$$Numbers\ of\ inaccurate\ ponding\ nodes < int(ToleratedInaccuratePondingNodesPercentage*numbers\ of\ seepage\ nodes),\ 1)$$

where:

Hint:

$$ToleratedInaccuratePondingNodesPercentage = 0.01$$

7.2.2.2 Maximum allowable number of iterations

The check on node status for each of the boundary condition types is done over a Maximum allowable number of iterations. Beyond this, the boundary conditions status check will be ignored. **The maximum allowable number of iterations for checking status change are**:

- For seepage nodes: 10
- For drain nodes: 10
- For well nodes: 10
- For ponding nodes: 2

7.2.3 Special case: Steady-state calculation criteria

A steady state calculation is also solved using an implicit time stepping scheme (pseudo time step). For each pseudo-time step, the global error indicator for nodal flux must satisfied the global flow error convergence criteria as given in Flow error check (on page 62).

$$GlobalFlowError < TolerGlobalFlowError$$

Hint:

- *TolerGlobalFlowError* is internally set to 0.01 for steady state flow analysis and cannot be explicitly set by user.

For steady state groundwater flow calculation, the same convergence criteria as considered in transient analysis with respect to unsaturated behavior are also being enforced (see Unsaturated behaviour convergence check (on page 62)). Moreover, additional specific flow checks are being considered and which are only considered for steady state groundwater flow analysis.

7.2.3.1 Local groundwater flow check

An additional condition on flow error at local level should be satisfied during steady-state flow analysis:

$$LocalFlowError < TolerGlobalFlowError$$

where:

$$LocalFlowError = \frac{\left(\max_{closed\ nodes} \mid q_{ex} - q_{in} \mid\right)^2}{max\left(\Sigma \| \ q_{ext}^{out} \|_{\ open\ nodes}, \Sigma \| \ q_{ext}^{in} \|_{\ open\ nodes}\right)^2} \qquad Eq.\ [211]$$

where:

$$TolerLocalFlowError < 10 \times TolerGlobalFlowError$$

7.2.3.2 Average pore pressure change

The following steady state regime condition should be satisfied as well:

$$AveragePorePressureChange < TolerAveragePorePressureChange$$

where:

$$AveragePorePressureChange = \frac{\Sigma_{i=1,Nodes}\left({p_{new,i}}^2 - {p_{old,i}}^2\right)}{\Sigma_{i=1,Nodes}\left({p_{old,i}}^2\right)} \qquad Eq.\ [212]$$

Hint:

$$TolerAveragePorePressureChange = 0.005$$

Note: New and old subscripts refer to two consecutive pseudo-time steps.

7.2.3.3 Maximum pore pressure change

The number of inaccurate pore pressure nodes are counted during a steady state groundwater flow analysis. A flow node *i* is inaccurate if:

$$SSErrND(i) > TolerSSGWFNodeError$$

$$SSErrND(i) = \frac{{p_{new,i}}^2 - {p_{old,i}}^2}{{p_{old,i}}^2} \qquad Eq.\ [213]$$

By default:

$$TolerSSGWFNodeError = 100 \times TolerAveragePorePressureChange$$

Note: A check on the maximum pore pressure change (the most inaccurate pore pressure node) will be enforced.

$$MaximumPorePressureError < TolerMaximumPorePressureError$$

$$MaximumPorePressureError = \max_{i=1,Nnodes} \left(\frac{{p_{new,i}}^2 - {p_{old,i}}^2}{{p_{old,i}}^2} \right) \qquad Eq.\ [214]$$

By default:

$$TolerMaximumPorePressureError = 10 \times TolerAveragePorePressureChange$$

However, if the number of steps is larger than 10 and either:

a. The number of inaccurate pore pressure nodes *nSSErrNod* satisfies:

$$nSSErrNod < ToleratedInaccurateNodesPercentageSSFlow \times \text{total number of flow nodes (active and inactive)}$$

Hint: *ToleratedInaccurateNodesPercentageSSFlow* = 0.01 (default).

b. or the global flow error *GlobalFlowError* satisfies:

$$GlobalFlowError < 0.01 \times TolerGlobalFlowError$$

Then the criteria on maximum pore pressure change is disregarded.

7.2.3.4 Inaccurate nodes for boundary conditions

Inaccurate nodes are also considered during steady state groundwater flow calculation using the same definition as given in Inaccurate nodes for boundary conditions (on page 64). Only this specificity of steady state calculation with respect to transient flow is in the definition of the maximum allowable number of iterations for checking status change.

For steady-state calculation, the maximum allowable numbers of iterations for checking status change are reduced and equal to:

- For seepage nodes: 2
- For drain nodes: 2
- For well nodes: 2
- For ponding nodes: 2

8 Reference

1. (1999). TRIM, Total Risk Integrated Methodology. TRIM FATE Technical Support Document Volume I: Description of Module, EPA/43/D-99/002A. U.S. EPA: TRIM.
2. Alefeld, G., Herzberger, J. (1983). Introduction to Interval Computations. Academic Press, New York.
3. Basabe, J.D.D., Sen, M.K. (2007). Grid dispersion and stability criteria of some common finite-element methods for acoustic and elastic wave equations. Geophysics, 72(06), T81-T95.
4. Bathe, K.J., Koshgoftaar, M.R. (1979). Finite element free surface seepage analysis without mesh iteration. Int. J. Num. An. Meth Geo, 3, 13-22.
5. Biot, M.A. (1956). General solutions of the equations of elasticity and consolidation for porous material. Journal of Applied Mechanics, 23(2).
6. Das, B.M. (1995). Fundamentals of soil dynamics. Elsevier.
7. Dong, W., Shah, H.C. (1987). Vertex method for computing functions of fuzzy variables. Fuzzy Sets & Systems, 24, 65-78.
8. Dunavant, D.A. (1985). High degree efficient symmetrical gaussian quadrature rules for the triangle. International journal for numerical methods in engineering, 21, 1129-1148.
9. Goos, G., Hartmanis, J. (eds.) (1985). Interval Mathematics 1985. Springer Verlag, Berlin.
10. Haigh, S.K., Ghosh, B., Madabhushi, S.P.G. (2005). Importance of time step discretisation for nonlinear dynamic finite element analysis. Canadian Geotechnical Journal, 42, 957-963.
11. Hughes, T.J.R. (1987). The finite element method, linear static and dynamics analisis. Prencice Hall Int.
12. Jaulin, L., Kieffer, M., Didrit, P., Walter, E. (2001). Applied Interval Analysis. Springer, London.
13. Joyner, W.B., Chen, A.T.F. (1975). Calculation of non linear ground response in earthquake. Bulletin of Seismological Society of America, 65, 1315-1336.
14. Kearfott, R.B., Kreinovich, V. (eds.) (1996). Applications of Interval Computations. Kluwer Academic Publishers, Dordrecht.
15. Kelly, K.R., Ward, R.W., Treitel, S., Alford, R.M. (1976). Synthetic seismograms: a finite difference approach. Geophysics, 41(01), 2-27.
16. Kramer, S.L. (1996). Geotechnical earthquake engineering. Prentice Hall, New Jersey.
17. Kulpa, Z. (1997). Diagrammatic representation of interval space in proving theorems about interval relations. Reliable Computing, 3(3), 209-217.
18. Lysmer, J., Kuhlmeyer, R.L. (1969). Finite dynamic model for infinite media. ASCE J. of the ENg. Mech. Div., 859-87.
19. Moore, R.E. (1966). Interval Analysis. Engelwood cliffs.
20. Moore, R.E. (1979). Methods and Applications of Interval Analysis. SIAM Studies in Applied Mathematics, Philadelphia.
21. Neumaier, A. (1990). Interval Methods for Systems of Equations. Cambridge University Press, Cambridge.
22. Pradhan, P., Baidya, D.K., Ghosh, D.P. (2004). Dynamic response of foundations resting on layered soil by cone model. Soil Dynamics and Earthquake Engineering, 24(06), 425-434.
23. Ratschek, H., Rokne, J. (1988). New Computer Methods for Global Optimization. Ellis Horwood Limited, Chichester.

Reference

24. Riks, E. (1979). An incremental approach to the solution of snapping and buckling problems. Int. J. Solids & Struct., 15, 529-551.

25. Sadek, M., Shahrour, I. (2004). A three dimensional embedded beam element for reinforced geomaterials. International Journal for Numerical and Analytical Methods in Geomechanics, 28, 931-946.

26. Sluys, L.J. (1992). Wave propagation, Localisation and Dispersion in softening solids. dissertation, Delft University of Technology.

27. Song, E.X. (1990). Elasto-plastic consolidation under steady and cyclic loads. Ph.d thesis, Delft University of Technology, The Netherlands.

28. Tuy, H. (1998). Convex Analysis and Global Optimization. Kluwer Academic Publishers, Dordrecht.

29. Van Langen, H., Vermeer, P.A. (1990). Automatic step size correction for non-associated plasticity problems. Int. J. Num. Meth. Eng., 29, 579-598.

30. Vermeer, P.A. (1979). A modified initial strain method for plasticity problems. In Proc. 3rd Int. Conf. Num. Meth. Geomech. Balkema, Rotterdam, 377-387.

31. Vermeer, P.A., van Langen, H. (1989). Soil collapse computations with finite elements. Ingenieur-Archiv 59, 221-236.

32. Vermeer, P.A., Verruijt, A. (1981). An accuracy condition for consolidation by finite elements. Int. J. for Num. Anal. Met. in Geom., 5, 1-14.

33. Zienkiewicz, O.C. (1967). The finite element method in structural and continuum mechanics. McGrawHill, London, UK.

34. Zienkiewicz, O.C., Taylor, R.L. (1991). The finite element method; Solid and Fluid mechanics, Dynamics and Non-Linearity, volume 2. Mc Graw-Hill, U.K., 4 edition.

Appendices

A Symbols

Symbol	Name
$\underline{b}$	Vector containing the body force
$\mathbf{B}$	Strain interpolation matrix
$\mathbf{D}^e$	Elastic material stiffness matrix representing Hooke's law
f	Yield function
g	Plastic potential function
$\mathbf{k}$	Permeability matrix
$\mathbf{K}$	Stiffness matrix
$\mathbf{L}$	Differential operator
$\mathbf{M}$	Material stiffness matrix
$\mathbf{N}$	Matrix with shape functions
p	(Excess) pore pressure
t	Time
$\underline{t}$	Boundary tractions
$\underline{u}$	Vector with displacement components
$\underline{v}$	Vector with nodal displacements
V	Volume
w	Weight factor
γ	Volumetric weight
$\underline{\varepsilon}$	Vector with strain components

Symbols

Symbol	Name
λ	Plastic multiplier
ξ, η, ζ	Local coordinates
$\underline{\sigma}$	Vector with stress components
ω	Integration constant (explicit ω=0; implicit: ω=1.

B Calculation Process

B.1 Finite element calculation process based on the elastic stiffness matrix

Calculation Process	Formulation
Read input data	
Form stiffness matrix	$\mathbf{K} = \int \mathbf{B}^T \mathbf{D}^e \mathbf{B} \, d\ V$
New step	$i \rightarrow i + 1$
Form new load vector	$\underline{f}_{ex}^{i} = \underline{f}_{ex}^{i-1} + \Delta \underline{f}_{ex}$
Form reaction vector	$\underline{f}_{in} = \int \mathbf{B}^T \underline{\sigma}^{i-1} \, d\ V$
Calculate unbalance	$\Delta \underline{f} = \underline{f}_{ex}^{i} - \underline{f}_{in}$
Reset displacement increment	$\Delta \underline{u} = 0$
New iteration	$j \rightarrow j + 1$
Solve displacements	$\delta \underline{u} = \mathbf{K}^{-1} \Delta \underline{f}$
Update displacement increments	$\Delta \underline{u}^{j} = \Delta \underline{u}^{j-1} + \delta \underline{u}$
Calculate strain increments	$\Delta \underline{\varepsilon} = \mathbf{B} \Delta \underline{u}; \delta \underline{\varepsilon} = \mathbf{B} \delta \underline{u}$
Calculate stresses:	$\underline{\sigma}^{tr} = \underline{\sigma}_c^{i-1} + \mathbf{D}^e \Delta \underline{\varepsilon}\ \underline{\sigma}^{eq} = \underline{\sigma}_c^{i,j-1} + \mathbf{D}^e \delta \underline{\varepsilon}$ $\underline{\sigma}_c^{i,j} = \underline{\sigma}^{tr} - \frac{\langle f(\underline{\sigma}^{tr}) \rangle}{d + h} \mathbf{D}^e \frac{\partial g}{\partial \underline{\sigma}}$
Form reaction vector	$\underline{f}_{in} = \int \mathbf{B}^T \underline{\sigma}^{i,j} \, d\ V$
Calculate unbalance	$\Delta \underline{f} = \underline{f}_{ex}^{i} - \underline{f}_{in}$

Calculation Process

Calculation Process	Formulation
Calculate error	$e = \frac{\Delta \underline{f}}{\underline{f}_{ex}^{i}}$
Accuracy check	if $e > e_{tolerated} \rightarrow$ new iteration
Update displacements	$\underline{u}^{i} = \Delta \underline{u}^{i-1} + \Delta \underline{u}$
Write output data (results)	-
If not finished → new step	-
Finish	-

C

Understanding the iterative convergence process

In nonlinear problems, convergence in the iteration procedure (see Calculation Process (on page 73)) needs to be considered. PLAXIS provides information about the different convergence checks for each iteration of the calculation, which allows users to monitor the progress of the analysis run.

Note: The convergence criteria information is produced by the PLAXIS **Convergence tool** and stored in the called **Convergence log file**. For more information on the Convergence log file please Visit the **Reference Manual>Advanced tools>Convergence log** for PLAXIS calculations.

C.1 Quasi-Newton Raphson method

In nonlinear problems the governing balance equations must be solved iteratively by means of the Newton-Raphson method. The iterative process as schematized in Figure 12 (on page 76) considers the update of the incremental displacement as:

$$\delta u_i = K_i^{-1}\left(f_{ext} - f_{\text{int},i}\right)$$

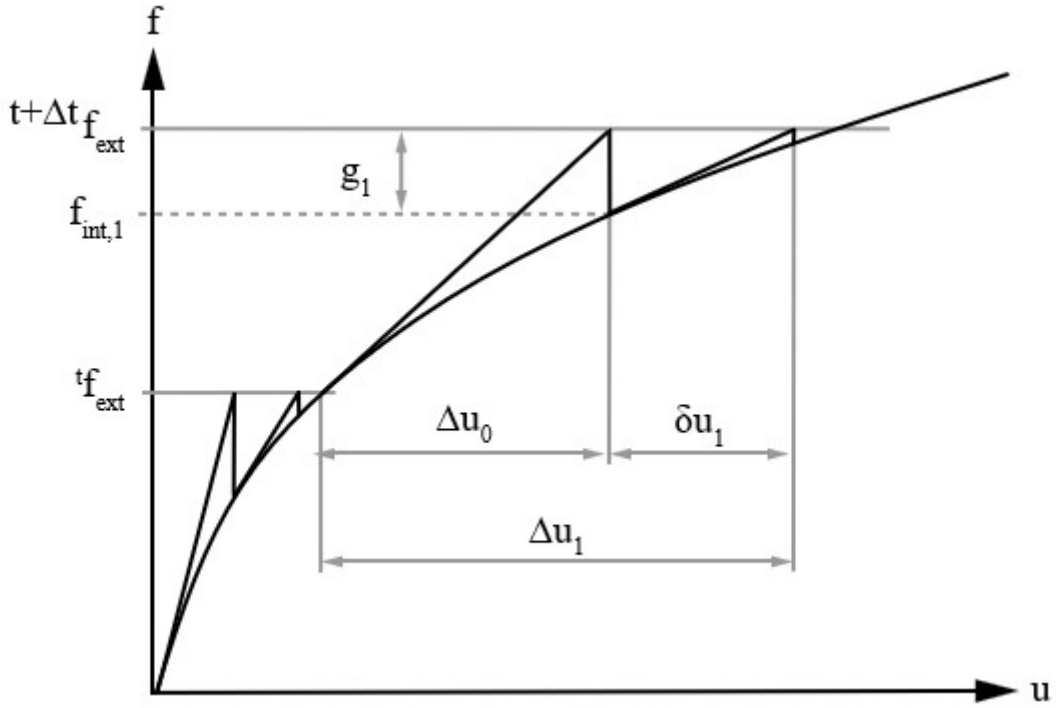


Figure 12: Iteration process

The process is repeated iteratively until the out-of-balance force (f_{ext}–$f_{int,i}$) becomes small compared the applied force itself f_{ext}. A disadvantage of this method is that the stiffness matrix K has to be set up at every iteration and the time-consuming decomposition of the matrix (to compute its inverse) has to be performed every iteration as well.

It is for this computation cost of finding a new stiffness matrix that PLAXIS instead uses a Quasi-newton method inverse at every iteration. The Quasi-Newton method essentially uses the information of previous solution vectors and out-of-balance force vectors during the increment to achieve a better approximation (see Figure 13 (on page 77)) . Unlike Regular Newton-Raphson, the Quasi-Newton method does not set up a completely new stiffness matrix every iteration and is computationally significantly more efficient.

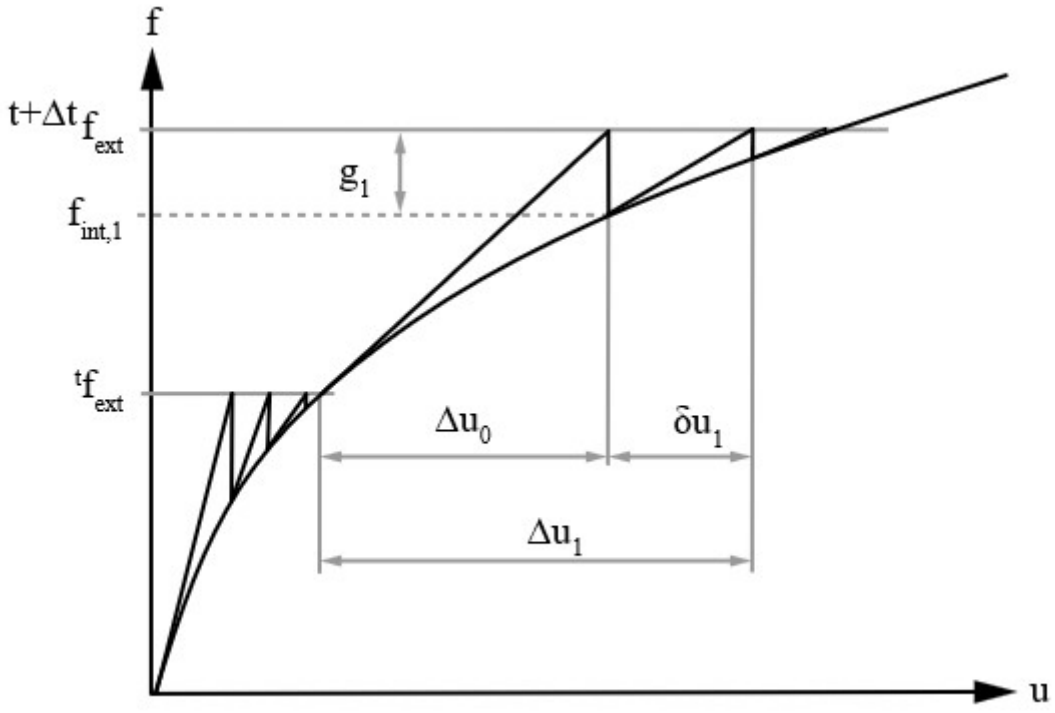


Figure 13: Quasi-Newton iteration