Research paper

Efficient and realistic 3-D boundary element simulations of underground construction using isogeometric analysis

Gernot Beer [a,*,1], Christian Duenser [a,1], Vincenzo Mallardo [b,1]

[a] *Institute of Structural Analysis, Graz University of Technology, Lessingstraße 25/II, 8010 Graz, Austria*
[b] *Department of Architecture, University of Ferrara, Via Quartieri 8, 44121 Ferrara, Italy*

ARTICLE INFO

ABSTRACT

The paper outlines some recent developments of the boundary element method (BEM) that makes it more user friendly and suitable for a realistic simulation in geomechanics, especially for underground excavations and tunnelling. The innovations refer to the introduction of isogeometric concepts, elasto-plastic analysis and the simulation of ground support. The introduction of isogeometric concepts for the description of the excavation boundaries results in less user and analysis effort, since complex geometries can be modelled with few parameters and degrees of freedom. No mesh generation is necessary. Heterogeneous and inelastic ground conditions are considered via general inclusions and rock bolts via linear inclusions.

A comparison of results of test examples with other numerical methods and analytical solutions confirms the efficiency and accuracy of the proposed implementation. A practical example with a complex geometry is presented.

1. Introduction

Since the publication of the first paper on the topic (Hughes et al., 2005), isogeometric analysis has gained increased popularity. The majority of applications have been with the Finite Element method (FEM) and much less with the BEM. However, the advantage of the BEM, that requires only the discretisation of the boundary, makes it an ideal companion to Computer Aided Design (CAD). First applications of the isogeometric BEM (IGABEM) were published in elasticity in 2-D (Simpson et al., 2012, 2013) and in 3-D (Scott et al., 2013). Other applications followed (for example see An et al., 2018; Fang et al., 2020). In Marussig et al. (2015) the concept of a geometry independent field approximation, which involved a decoupling of the geometry definition and the approximation of the unknown, was first introduced and was later adopted by others (Atroshchenko et al., 2018). The seamless integration of BEM and CAD was discussed in Marussig (2016). In a recent book published on the isogeometric BEM (Beer et al., 2019) it was shown how geometrical information can be taken directly from CAD data and that efficient and accurate simulations with very few unknowns can be obtained.

One fact that has hampered the widespread use of the BEM is that fundamental solutions, on which the method is based, exist only for elastic material properties and homogeneous domains. Fundamental solutions can be obtained for anisotropic materials, but they are very

complicated (Pan and Chou, 1976). To overcome the fact that the original BEM can only deal with homogeneous and elastic domains, several workarounds were introduced. Among them we mention the introduction of boundary element regions, to consider a piecewise heterogeneous domain (Banerjee and Raveendra, 1986) and the coupling of the BEM with the Finite Element Method (FEM) where the FEM regions model non-linear behaviour (Wendland, 1990). The topic of elastic inclusions was recently dealt with in Sun et al. (2020). The concept of including non-linear effects by adding a volume integral was first introduced by Brebbia and Walker (1980) and Banerjee (1994) and later expanded in Gao and Davies (2011). Various ways of avoiding the generation of a volume mesh were presented (see for example Tanaka et al., 2001), but their application is limited to finite domain problems. All of the mentioned solutions involve the introduction of errors or increase the discretisation effort.

In this paper we show several innovations that make the BEM suitable for realistic simulations in underground construction. This means that the BEM simulation can consider ground support, heterogeneous ground conditions and non-linear material behaviour. We first introduce the theoretical background of the BEM with volume effects. Then the evaluation of the arising boundary and volume integrals is discussed in some detail. This includes the description of the excavation geometry with NURBS patches and the definition of subdomains, where

---

material properties differ from the ones used to compute the fundamental solutions or which behave in an inelastic way. Rock bolts are also modelled as (linear) subdomains and their analytical integration allows many of them to be used in a simulation with a small increase in the numerical effort.

Two test examples are included, which test the accuracy and efficiency of the simulation of ground support and elasto-plastic material behaviour. Finally a practical example with some complexity is presented.

It is emphasised that the simulation approach presented here does not involve any mesh generation. Instead, geometries are defined by NURBS patches using data generated by CAD programs or input data, in a similar data format, that are user generated.

2. Theory

In the following we will use the word **inclusion** to specify part of the analysis domain that is not modelled by boundary elements, i.e. parts of the domain that has different material properties or behaves inelastically. This also applies to the ground support.

As will be explained, we use the concept of initial stresses inside inclusions to consider those volume effects, which are not considered by the boundary discretisation.

In the following we will use matrix algebra and it is therefore necessary to convert the stress and strain tensors $\sigma_{ij}, \epsilon_{ij}$ to pseudo-vectors σ, ϵ using *Voigt* notation:

$$\sigma = \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{23} \\ \sigma_{13} \end{Bmatrix} \quad \text{and} \quad \epsilon = \begin{Bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ \epsilon_{12} + \epsilon_{21} \\ \epsilon_{23} + \epsilon_{32} \\ \epsilon_{13} + \epsilon_{31} \end{Bmatrix} \quad (1)$$

The initial stress vector σ_0, due the fact that a point inside the inclusion has properties that are different to the ones used for computing the fundamental solutions, is given by:

$$\sigma_0 = (\mathbf{D} - \mathbf{D}_{incl})\epsilon \quad (2)$$

where ϵ is the total strain, $\mathbf{D}$ is the constitutive matrix for computing the fundamental solutions and $\mathbf{D}_{incl}$ is the corresponding matrix for the inclusion point.

Remark. While the matrix $\mathbf{D}$ is restricted to an isotropic elasticity matrix, the constitutive matrix $\mathbf{D}_{incl}$ can be quite general, ranging from the sparsely populated isotropic elasticity matrix to a fully populated anisotropic elasticity matrix. Indeed, later on we will use an elasto-plastic constitutive matrix for $\mathbf{D}_{incl}$.

In the following we first establish the governing integral equations and then discuss in detail how the arising volume and surface integrals are evaluated.

2.1. Governing integral equations

Consider a domain Ω with a boundary Γ, containing a subdomain Ω_0 where initial stresses $\sigma_0(\hat{x})$ are present. We apply the theorem by Betti and the **collocation method** to arrive at the governing integral equations. This means that we set the work done on the boundary Γ by tractions T times displacements $\mathbf{u}$ equal to the work done by displacements U times tractions $\mathbf{t}$. We assume T and U to be fundamental solutions of the governing differential equation at $\hat{x}$ due to a source at $\tilde{x}_n$ and $\mathbf{u}, \mathbf{t}$ to be boundary values. If initial stresses are present, additional work is done in the domain Ω_0 by the initial stresses $\sigma_0(\hat{x})$ times the fundamental solution for strains $\mathsf{E}(\tilde{x}_n, \hat{x})$. The integral

equation can be written as (see Brebbia et al., 1984; Aliabadi, 2002; Banerjee and Butterfield, 1981):

$$\int_\Gamma \mathsf{T}(\tilde{x}_n, \hat{x})\mathbf{u}(\hat{x}) \, d\Gamma(\hat{x}) = \int_\Gamma \mathsf{U}(\tilde{x}_n, \hat{x}) \, \mathbf{t}(\hat{x}) \, d\Gamma(\hat{x})$$
$$+ \int_{\Omega_0} \mathsf{E}(\tilde{x}_n, \hat{x})\sigma_0(\hat{x}) d\Omega_0(\hat{x}). \quad (3)$$

where $\tilde{x}_n$ are the coordinates of the **collocation point** n. To be able to solve the integral equations they have to be regularised. The regularised integral equations are written as :

$$\int_\Gamma \mathsf{T}(\tilde{x}_n, \hat{x})(\mathbf{u}(\hat{x}) - \mathbf{u}(\tilde{x}_n)) \, d\Gamma(\hat{x}) - \mathbf{A}_n \mathbf{u}(\tilde{x}_n) = \int_\Gamma \mathsf{U}(\tilde{x}_n, \hat{x}) \, \mathbf{t}(\hat{x}) \, d\Gamma(\hat{x})$$
$$+ \int_{\Omega_0} \mathsf{E}(\tilde{x}_n, \hat{x})\sigma_0(\hat{x}) d\Omega_0(\hat{x}). \quad (4)$$

where $\mathbf{A}_n = \mathbf{0}$ for finite domain problems and $\mathbf{A}_n = \mathbf{I}$ for infinite domain problems. The derivation of Eq. (4) and the fundamental solutions U and T are presented in Beer et al. (2019). The fundamental solution E is given by:

$$E_{ijk} = \frac{-C}{r^2} \left[C_3(r_{,k}\delta_{ij} + r_{,j}\delta_{ik}) - r_{,i}\delta_{jk} + C_4 \, r_{,i}r_{,j}r_{,k} \right] \quad (5)$$

where $r = |\hat{x} - \tilde{x}|$, $r_{,i} = \frac{r_i}{r}$ and δ_{ij} is the *Kronecker Delta*. The constants are: $C = \frac{1}{16\pi G(1-\nu)}$, $C_3 = 1 - 2\nu$ and $C_4 = 3$ where G is the shear modulus and ν the Poisson's ratio.

The tensor E_{ijk} is converted to a matrix E:

$$\mathsf{E} = \begin{bmatrix} E_{111} & E_{122} & E_{133} & E_{112} + E_{121} & E_{123} + E_{132} & E_{113} + E_{131} \\ E_{211} & E_{222} & E_{233} & E_{212} + E_{221} & E_{223} + E_{232} & E_{213} + E_{231} \\ E_{311} & E_{322} & E_{333} & E_{312} + E_{321} & E_{323} + E_{332} & E_{313} + E_{331} \end{bmatrix} \quad (6)$$

3. Discretisation of integral equations

To be able to solve Eqs. (4) we have to discretise them. This involves 2 steps:

- The subdivision of the boundary domain into patches and the volume domain into inclusions
- The approximation of the unknown boundary values and the approximation of initial stresses.

This will be discussed in the subsequent sections separately for the boundary and volume integrals.

3.1. Discretisation of the boundary integrals

For the numerical solution of the boundary integral equations the integrals are expressed as sum of integrals over patches:

$$\int_\Gamma \mathsf{U}(\tilde{x}_n, \hat{x}) \, \mathbf{t}(\hat{x}) \, d\Gamma(\hat{x}) - \int_\Gamma \mathsf{T}(\tilde{x}_n, \hat{x})(\mathbf{u}(\hat{x}) - \mathbf{u}(\tilde{x}_n)) \, d\Gamma(\hat{x}) + \mathbf{A}_n \mathbf{u}(\tilde{x}_n) =$$
$$\sum_{e=1}^E \int_{\Gamma_e} \mathsf{U}(\tilde{x}_n, \hat{x}) \, \mathbf{t}^e(\hat{x}) \, d\Gamma_e(\hat{x}) - \sum_{e=1}^E \int_{\Gamma_e} \mathsf{T}(\tilde{x}_n, \hat{x})\mathbf{u}^e(\hat{x}) d\Gamma_e$$
$$+ \left[\sum_{e=1}^E \left(\int_{\Gamma_e} \mathsf{T}(\tilde{x}_n, \hat{x}) d\Gamma_e \right) + \mathbf{A}_n \right] \mathbf{u}(\tilde{x}_n) \quad (7)$$

where e specifies the patch number and E is the total number of patches. In the following the geometry of patches is specified using NURBS basis functions. The advantage of this is that some geometrical shapes such as cylinder and spheres can be described exactly with few parameters. For further information on NURBS and how (7) is obtained the reader is referred to Beer et al. (2019).

There are 3 types of patches that are useful for geomechanics modelling: finite, infinite and trimmed patches. In addition we introduce a special patch with a cutout that can be used for modelling intersections.
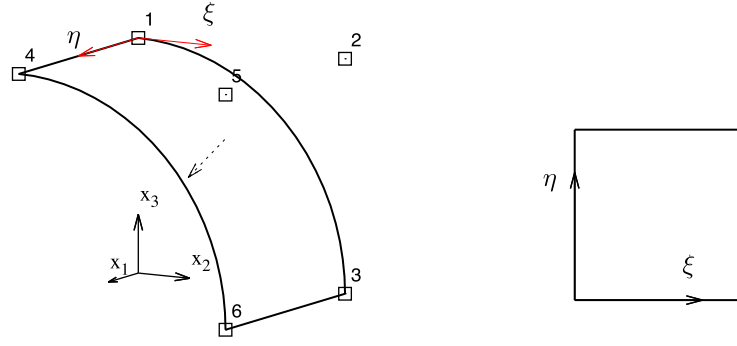
Fig. 1. A finite patch with control points (numbered squares). Left: in the global, right: in the local coordinate system. Also shown is the "outward normal".
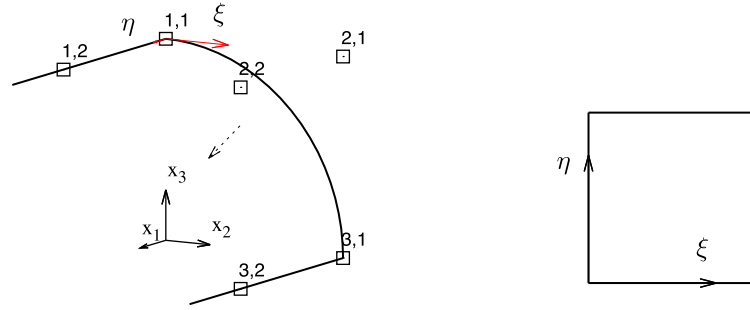

Fig. 2. Example of an infinite patch. Left in the global and right in the local coordinate system.

3.1.1. Geometry definition of finite patches

In Fig. 1 we show an example of a finite patch. The mapping from the local $\xi(\xi, \eta)$ to the global $\boldsymbol{x}$ coordinate system is given by

$$\boldsymbol{x}(\xi, \eta) = \sum_{i=1}^{I} R_i(\xi, \eta) \boldsymbol{x}_i. \tag{8}$$

where $R_i(\xi, \eta)$ are NURBS basis functions and the control points (coordinates $\boldsymbol{x}_i$) are numbered consecutively, first in the ξ- and then in the η-direction.

The vectors tangential to the surface are given by

$$\mathbf{v}_\xi = \frac{\partial \boldsymbol{x}}{\partial \xi} = \begin{pmatrix} \frac{\partial x_1}{\partial \xi} \\ \frac{\partial x_2}{\partial \xi} \\ \frac{\partial x_3}{\partial \xi} \end{pmatrix} \quad \text{and} \quad \mathbf{v}_\eta = \frac{\partial \boldsymbol{x}}{\partial \eta} = \begin{pmatrix} \frac{\partial x_1}{\partial \eta} \\ \frac{\partial x_2}{\partial \eta} \\ \frac{\partial x_3}{\partial \eta} \end{pmatrix} \tag{9}$$

and the unit vector normal is

$$\mathbf{n} = \frac{\mathbf{v}_\xi \times \mathbf{v}_\eta}{J}. \tag{10}$$

The Jacobian is

$$J = |\mathbf{v}_\xi \times \mathbf{v}_\eta|. \tag{11}$$

The direction of the "outward normal" depends on how the control points are numbered.

3.1.2. Geometry definition of infinite patches

Here we introduce a patch definition that is useful for the simulation in geomechanics where one sometimes has to consider a surface that tends to infinity (Beer, 2015). In this case we define an infinite patch as shown in Fig. 2. The mapping for a patch that extends to infinity in the η-direction is given by

$$\boldsymbol{x} = \sum_{j=1}^{2} \sum_{i=1}^{I} R_{ij}^\infty(\xi, \eta) \boldsymbol{x}_{ij} \tag{12}$$

where

$$R_{ij}^\infty(\xi, \eta) = R_i(\xi) M_j^\infty(\eta) \tag{13}$$

and the special infinite basis functions are

$$M_1^\infty = \frac{1 - 2\eta}{1 - \eta} \quad \text{and} \quad M_2^\infty = \frac{\eta}{1 - \eta}. \tag{14}$$

The vectors in the tangential directions are given by

$$\mathbf{v}_\xi = \frac{\partial \boldsymbol{x}}{\partial \xi} = \sum_{j=1}^{2} \sum_{i=1}^{I} \frac{\partial R_i(\xi)}{\partial \xi} M_j^\infty(\eta) \boldsymbol{x}_{ij} \tag{15}$$

$$\mathbf{v}_\eta = \frac{\partial \boldsymbol{x}}{\partial \eta} = \sum_{j=1}^{2} \sum_{i=1}^{I} R_i(\xi) \frac{\partial M_j^\infty(\eta)}{\partial \eta} \boldsymbol{x}_{ij} \tag{16}$$

where

$$\frac{\partial M_1^\infty}{\partial \eta} = \frac{-1}{(1 - \eta)^2} \quad \text{and} \quad \frac{\partial M_2^\infty}{\partial \eta} = \frac{1}{(1 - \eta)^2}. \tag{17}$$

The unit vector normal is computed as for the finite patch. It is noted that the Jacobian J tends to infinity as η tends to 1.

3.1.3. Trimmed patches

Patches can be trimmed using trimming curves, resulting in more complex geometries. The trimming curves are defined in patch coordinates ξ, η. A trimmed patch is shown in Fig. 3. More information about trimming can be found in Beer (2015).

3.1.4. Special patches

Special patches can be used to model intersections of curved surfaces with flat surfaces with a minimum of effort. A special patch is shown in Fig. 4.

Its geometry is defined by:

$$\boldsymbol{x}(\xi, \eta) = (1 - \eta)\boldsymbol{x}^I + \eta \boldsymbol{x}^{II} \tag{18}$$

where

$$\boldsymbol{x}^I(\xi) = \sum_{j=1}^{J} R_j(\xi) \boldsymbol{x}_j^I \quad \text{and} \quad \boldsymbol{x}^{II}(\xi) = \sum_{j=1}^{J} R_j(\xi) \boldsymbol{x}_j^{II} \tag{19}$$

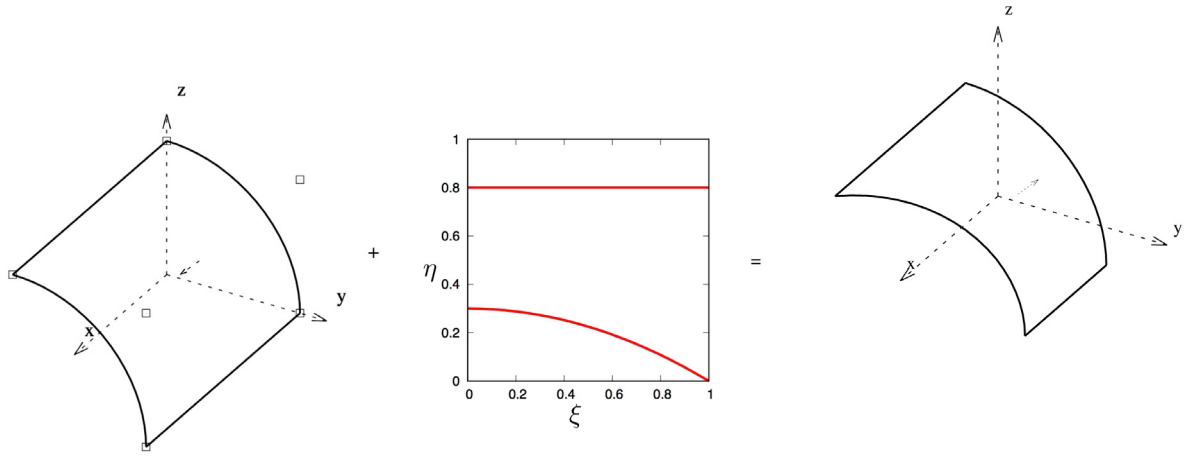The superscript II refers to the inner and I to the outer bounding curve.

Fig. 3. Figure showing a normal patch, trimming curves and the resulting trimmed patch.
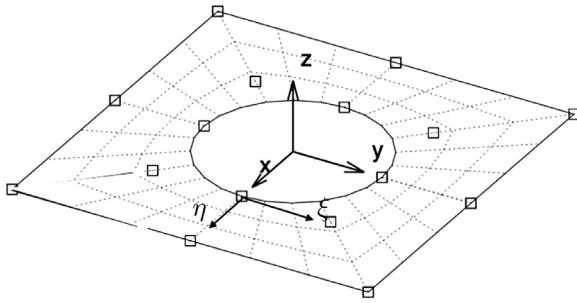

Fig. 4. Example of a special patch with a circular cutout.

The vectors in ξ and η directions are given by:

$$\mathbf{v}_\xi = (1-\eta)\mathbf{v}^I + \eta\mathbf{v}^{II} \qquad \text{and} \qquad \mathbf{v}_\eta = \mathbf{x}^I - \mathbf{x}^{II} \tag{20}$$

where

$$\mathbf{v}^I(\xi) = \sum_{j=1}^{J} \frac{\partial R_j(\xi)}{\partial \xi} \mathbf{x}_j^I \qquad \text{and} \qquad \mathbf{v}^{II}(\xi) = \sum_{j=1}^{J} \frac{\partial R_j(\xi)}{\partial \xi} \mathbf{x}_j^{II} \tag{21}$$

The outward normal and the Jacobian are computed the same way as for finite patches.

3.1.5. Defining geometry with NURBS

CAD programs use NURBS to describe geometrical shapes (cylinder, spheres or general smooth shapes). If two shapes intersect trimming is applied. Since CAD programs are mainly designed for visualisation, the resulting intersection geometry may have small gaps. The interested reader may consult (Beer et al., 2019), where a whole chapter is devoted to the topic of how data from CAD can be used for a BEM simulation. However, instead of asking the CAD program to compute the intersection geometry and then extract the necessary data, which is complicated, one may write a MATLAB function that computes the intersection geometry without gaps. For the practical example we have developed a function that does this.

Here we show on an example how easy it is to define complex geometrical shapes with NURBS. We start with the definition of a circular excavation and of 2 cross-passages. Only a few lines of input data (shown in Fig. 5 on the left and middle) are required. The two geometries are then intersected resulting in the geometry definition on the right of Fig. 5 consisting of a trimmed patch, normal patches, infinite patches and special patches. Note that no mesh generation is necessary. The dotted lines indicate integration regions (see the section on integration below).

3.1.6. Approximation of boundary values

To be able to solve the patch integrals in Eq. (7), the boundary values must be approximated. For the approximation we also use NURBS basis functions. For normal patches the unknown boundary values are approximated by

$$\hat{\mathbf{u}}^e(\xi,\eta) = \sum_{k=1}^{K} \hat{R}_k(\xi,\eta)\, \hat{\mathbf{u}}_k^e$$

$$\hat{\mathbf{t}}^e(\xi,\eta) = \sum_{k=1}^{K} \hat{R}_k(\xi,\eta)\, \hat{\mathbf{t}}_k^e. \tag{22}$$

where $\hat{R}_k(\xi,\eta)$ are NURBS basis functions (the hat indicating that they may be different to the ones used for describing the geometry) and $\hat{\mathbf{u}}_k^e, \hat{\mathbf{t}}_k^e$ are parameter values. It should be noted that in contrast to Lagrange polynomials, parameter values do not in general represent real values.

For infinite patches we have 2 choices for the displacements:

• *Plane strain*: Displacements are constant to infinity:

$$\hat{\mathbf{u}}^e(\xi,\eta) = \hat{\mathbf{u}}^e(\xi,\eta=0) = \sum_{k=1}^{K^\infty} \hat{R}_k(\xi)\, \hat{\mathbf{u}}_k^{e\infty} \tag{23}$$

• *Decay*: Displacements decay to zero as infinity is approached

$$\hat{\mathbf{u}}^e(\xi,\eta) = (1-\eta)\hat{\mathbf{u}}^e(\xi,\eta=0) = (1-\eta)\sum_{k=1}^{K^\infty} \hat{R}_k(\xi)\, \hat{\mathbf{u}}_k^{e\infty} \tag{24}$$

where K^∞ is the number of parameters and $\hat{\mathbf{u}}_k^{e\infty}$ are the parameter values on the finite boundary.

Our refinement philosophy is to take the NURBS functions that define the geometry of the problem and refine them as necessary using knot insertion and order elevation.

Known values are defined by

$$\bar{\mathbf{u}}^e(\xi,\eta) = \sum_{k=1}^{\bar{K}} \bar{R}_k(\xi,\eta)\, \bar{\mathbf{u}}_k^e$$

$$\bar{\mathbf{t}}^e(\xi,\eta) = \sum_{k=1}^{\bar{K}} \bar{R}_k(\xi,\eta)\, \bar{\mathbf{t}}_k^e. \tag{25}$$

where $\bar{R}_k$ are basis functions, which may be different from the ones defining the geometry and the unknown values.

Inserting the approximations into the patch integrals allows the boundary parameters can be taken outside:

$$\int_{\Gamma_e} \mathsf{U}(\tilde{\mathbf{x}}_n,\hat{\mathbf{x}})\, \mathbf{t}^e(\hat{\mathbf{x}})\, d\Gamma_e(\hat{\mathbf{x}}) = \int_{\Gamma_e} \mathsf{U}(\tilde{\mathbf{x}}_n,\hat{\mathbf{x}}) \left(\sum_{k=1}^{K} R_k(\xi,\eta)\, \hat{\mathbf{t}}_k^e \right) d\Gamma_e(\hat{\mathbf{x}}) \tag{26}$$

$$= \sum_{k=1}^{K} \left(\int_{\Gamma_e} \mathsf{U}(\tilde{\mathbf{x}}_n,\hat{\mathbf{x}}) R_k(\xi,\eta)\, d\Gamma_e(\hat{\mathbf{x}}) \right) \hat{\mathbf{t}}_k^e$$
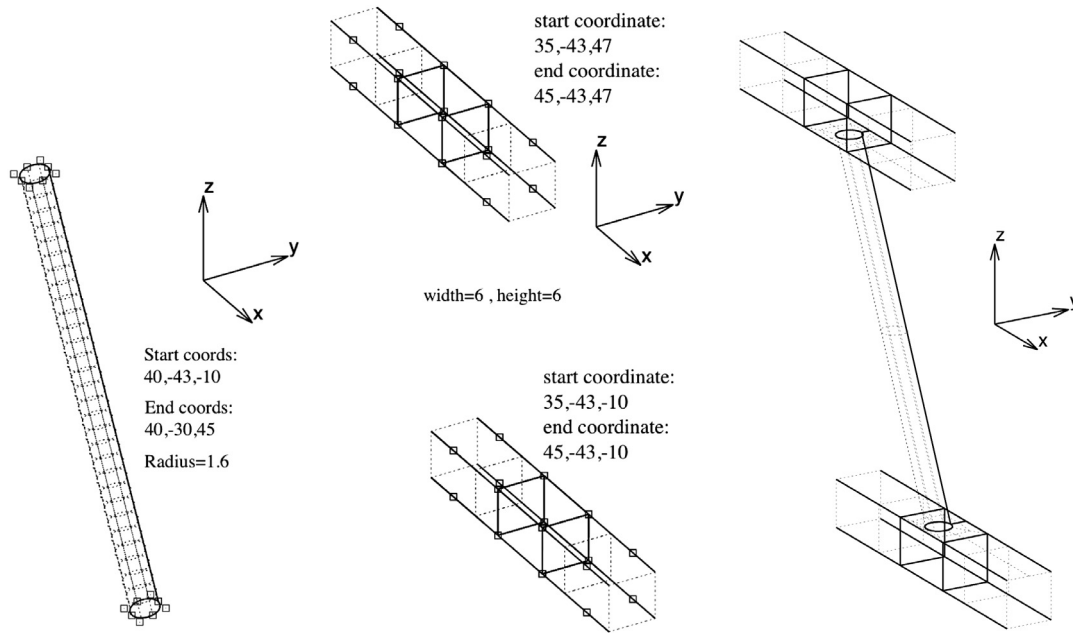
Fig. 5. Example of definition of geometry with NURBS, showing input data required. Left: circular excavation, middle: cross-passages, right: combined geometry.

$$\int_{\Gamma_e} \mathsf{T}(\tilde{\boldsymbol{x}}_n, \hat{\boldsymbol{x}}) \, \boldsymbol{u}^e(\hat{\boldsymbol{x}}) \, d\Gamma_e(\hat{\boldsymbol{x}}) = \int_{\Gamma_e} \mathsf{U}(\tilde{\boldsymbol{x}}_n, \hat{\boldsymbol{x}}) \left(\sum_{k=1}^{K} R_k(\xi, \eta) \, \hat{\boldsymbol{u}}_k^e \right) d\Gamma_e(\hat{\boldsymbol{x}}) \quad (27)$$

$$= \sum_{k=1}^{K} \left(\int_{\Gamma_e} \mathsf{T}(\tilde{\boldsymbol{x}}_n, \hat{\boldsymbol{x}}) R_k(\xi, \eta) \, d\Gamma_e(\hat{\boldsymbol{x}}) \right) \hat{\boldsymbol{u}}_k^e$$

where the hat and overbar has been omitted, because this depends if the values are known or unknown. This requires only the integration of fundamental solutions times the basis functions, which will be discussed later.

3.2. Discretisation of volume integral

The volume integral is solved numerically by dividing the volume into inclusions, defining each one geometrically. The integral is replaced by a sum of integrations over inclusions:

$$\int_{\Omega_0} \mathsf{E}(\tilde{\boldsymbol{x}}_n, \hat{\boldsymbol{x}}) \boldsymbol{\sigma}_0(\hat{\boldsymbol{x}}) d\Omega_0(\hat{\boldsymbol{x}}) = \sum_{ni=1}^{Ni} \int_{\Omega_{ni}} \mathsf{E}(\tilde{\boldsymbol{x}}_n, \hat{\boldsymbol{x}}) \boldsymbol{\sigma}_0(\hat{\boldsymbol{x}}) d\Omega_{ni}(\hat{\boldsymbol{x}}). \quad (28)$$

where Ni is the number of inclusions and Ω_{ni} specifies the inclusion domain. For the numerical treatment an approximation of the initial stress $\boldsymbol{\sigma}_0$ is assumed inside the inclusion and the integrals are solved numerically or analytically as will be explained later.

Two types of inclusions are considered: General inclusion that represent a volume of material that has different elastic properties or behaves inelastically and linear inclusions to represent rock bolts. It should be noted that a C^0 continuity of displacements exists at the interface between the inclusion and the domain.

3.2.1. Geometrical definition of general inclusion

General inclusions are defined by bounding NURBS surfaces. We establish a local coordinate system $\boldsymbol{s} = (s, t, r)^{\mathrm{T}} = [0, 1]^3$ as shown in Fig. 6 and map from local $\boldsymbol{s}$ coordinates to global $\boldsymbol{x}$ coordinates. The global coordinates of a point $\boldsymbol{x}$ with the local coordinates $\boldsymbol{s}$ are given by

$$\boldsymbol{x}(s, t, r) = (1 - r) \, \boldsymbol{x}^I(s, t) + r \, \boldsymbol{x}^{II}(s, t) \quad (29)$$

where

$$\boldsymbol{x}^I(s, t) = \sum_{k=1}^{K^I} R_k^I(s, t) \, \boldsymbol{x}_k^I \quad \text{and} \quad \boldsymbol{x}^{II}(s, t) = \sum_{k=1}^{K^{II}} R_k^{II}(s, t) \, \boldsymbol{x}_k^{II}. \quad (30)$$

The superscript I relates to the bottom (red) surface and II to the top (green) bounding surface and $\boldsymbol{x}_k^I$, $\boldsymbol{x}_k^{II}$ are control point coordinates. K^I and K^{II} represent the number of control points, $R_k^I(s, t)$ and $R_k^{II}(s, t)$ are NURBS basis functions. Note that there is a one to one mapping between the local surface coordinates ξ, η and the local coordinates s, t.

The derivatives are given by

$$\frac{\partial \boldsymbol{x}(s, t, r)}{\partial s} = (1 - r) \frac{\partial \boldsymbol{x}^I(s, t)}{\partial s} + r \frac{\partial \boldsymbol{x}^{II}(s, t)}{\partial s}$$

$$\frac{\partial \boldsymbol{x}(s, t, r)}{\partial t} = (1 - r) \frac{\partial \boldsymbol{x}^I(s, t)}{\partial t} + r \frac{\partial \boldsymbol{x}^{II}(s, t)}{\partial t} \quad (31)$$

$$\frac{\partial \boldsymbol{x}(s, t, r)}{\partial r} = -\boldsymbol{x}^I(s, t) + \boldsymbol{x}^{II}(s, t)$$

where for example:

$$\frac{\partial \boldsymbol{x}^I(s, t)}{\partial s} = \sum_{k=1}^{K^I} \frac{\partial R_k^I(s, t)}{\partial s} \, \boldsymbol{x}_k^I \quad \text{and} \quad \frac{\partial \boldsymbol{x}^{II}(s, t)}{\partial s} = \sum_{k=1}^{K^{II}} \frac{\partial R_k^{II}(s, t)}{\partial s} \, \boldsymbol{x}_k^{II}. \quad (32)$$

The Jacobi matrix of this mapping is

$$\mathbf{J} = \begin{pmatrix} \frac{\partial \boldsymbol{x}}{\partial s} \\ \frac{\partial \boldsymbol{x}}{\partial t} \\ \frac{\partial \boldsymbol{x}}{\partial r} \end{pmatrix} \quad (33)$$

and the Jacobian is $J = |\mathbf{J}|$.

3.2.2. Geometry definition of linear inclusion

This type of inclusions is used to model cables and rock bolts. Here we assume that the geometry is defined by a linear NURBS curve and that the bar has a circular cross-section with radius R over which the stress and strain are assumed constant. *The assumption is that the area of the cross-section of the inclusion is significantly smaller than that of the medium it is embedded in, allowing simplifications to be introduced for the integration.* We establish a local coordinate system $s = [0, 1]$ as shown on the right in Fig. 7. The global coordinates of a point $\boldsymbol{x}$ with the local coordinate s are given by

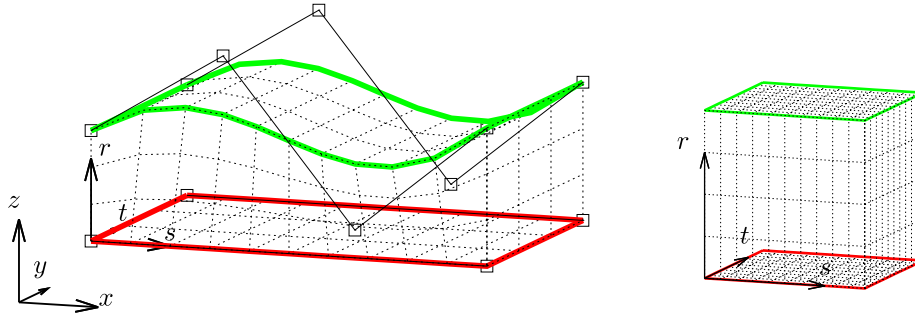$$\boldsymbol{x}(s) = \sum_{k=1}^{K} R_k(s) \, \boldsymbol{x}_k \quad (34)$$

Fig. 6. Mapping of 3-D inclusion showing the bottom and top NURBS surfaces and the associated control points defining the inclusion: Left in global $\boldsymbol{x}$, right in local $\boldsymbol{s}$ space. Also shown are subregions for the volume integration.
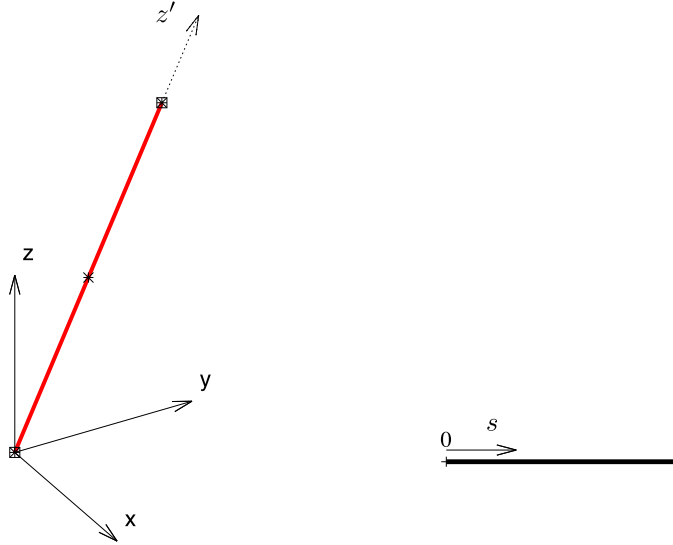

Fig. 7. Definition of linear inclusion by a NURBS curve with control points as hollow squares in global (left) and local (right) coordinates.

where K is the number of control points, $R_k(s)$ are NURBS basis functions and $\boldsymbol{x}_k$ are control point coordinates. We also define a local coordinate system whereby the z' axis is along the bar, specified by unit vector $\mathbf{v}_{z'}$. The vector along the bar is given by

$$\mathbf{V}_{z'} = \frac{\partial \boldsymbol{x}(s)}{\partial s} = \sum_{k=1}^{K} \frac{\partial R_k(s)}{\partial s} \, \boldsymbol{x}_k \tag{35}$$

The Jacobian is

$$J = \sqrt{\mathrm{V}_{z'_x}^2 + \mathrm{V}_{z'_y}^2 + \mathrm{V}_{z'_z}^2} \tag{36}$$

The unit vector in z' direction is given by

$$\mathbf{v}_{z'} = \mathbf{V}_{z'}/J \tag{37}$$

3.2.3. Approximation of initial stress

For the numerical integration, introduced below, we need the values of the initial stress at Gauss points. It is inefficient to compute $\boldsymbol{\sigma}_0$ at every Gauss point, whose location changes according to the location of $\tilde{\boldsymbol{x}}_n$. Instead we compute the initial stress at a fixed grid of points inside the inclusion. The value of initial stress at a point with the local coordinates $\boldsymbol{s} = (s, t, r)^{\mathrm{T}}$ for general inclusions and $\boldsymbol{s} = s$ for linear inclusions can be obtained by interpolation between grid points:

$$\boldsymbol{\sigma}_0(\boldsymbol{s}) = \sum_{l=1}^{L} M_l^{\sigma}(\boldsymbol{s}) \boldsymbol{\sigma}_{0l} \tag{38}$$

where $\boldsymbol{\sigma}_{0l}$ is the initial stress vector at grid point l with the local coordinate $\boldsymbol{s}_l$. L is the total number of inclusion points and $M_l^{\sigma}(\boldsymbol{s})$ are linear or constant basis functions, which will be shown later.

4. Numerical integration of boundary integrals

The boundary integrals to be solved are:

$$\mathbf{U}_{nk}^e = \int_{\Gamma_e} \mathsf{U}(\tilde{\boldsymbol{x}}_n, \hat{\boldsymbol{x}}) \hat{R}_k(\xi, \eta) \, d\Gamma_e(\hat{\boldsymbol{x}})$$

$$\mathbf{T}_{nk}^e = \int_{\Gamma_e} \mathsf{T}(\tilde{\boldsymbol{x}}_n, \hat{\boldsymbol{x}}) \hat{R}_k(\xi, \eta) \, d\Gamma_e(\hat{\boldsymbol{x}}) \tag{39}$$

$$\mathbf{T}_{n}^e = \int_{\Gamma_e} \mathsf{T}(\tilde{\boldsymbol{x}}_n, \hat{\boldsymbol{x}}) \, d\Gamma_e(\hat{\boldsymbol{x}})$$

They are evaluated numerically, using Gauss Quadrature. The integration scheme now depends on the location of the collocation point. If it is outside the patch we use regular integration otherwise we have to use singular integration.

Initially we divide the patch into **integration regions** depending on the following:

- The location of the collocation points. They should be on integration region boundaries.
- The aspect ratios of each integration region. It should be moderate and this is particularly important for singular integration.

4.1. Regular integration

For regular integration we have to consider that the value of integrand tends to infinity as the collocation point is approached. To maintain an adequate precision of integration is crucial to the quality of the results. Therefore the number of Gauss points has to be increased

near the collocation point. There is no analytical formula to determine the number of Gauss points required for a certain precision, but estimates have been worked out in Beer et al. (2019). The number depends on the size of the integration region and the proximity of the collocation point. The best strategy is to limit the number of Gauss points available and to subdivide the integration region into subregions. A Quadtree method that increases the number of Gauss points near the collocation point is most efficient.

Gauss Quadrature requires limits which range from -1 to $+1$. Therefore we introduce new local coordinates inside each subregion $\bar{\xi} = (\bar{\xi}, \bar{\eta})^{\mathrm{T}} = [-1, 1]^2$. The transformation to the patch coordinate system $\xi = (\xi, \eta)^{\mathrm{T}} = [0, 1]^2$ is given by:

$$\xi = \frac{\Delta \xi_s}{2}(1 + \bar{\xi}) + \xi_{s1} \tag{40}$$

$$\eta = \frac{\Delta \eta_s}{2}(1 + \bar{\eta}) + \eta_{s1} \tag{41}$$

where $\Delta \xi_s \times \Delta \eta_s$ is the size of the subregion and ξ_{s1}, η_{s1} are the starting coordinates.

The integration can be written as:

$$\mathbf{U}^e_{nk} = \sum_{s=1}^{S} \sum_{i=1}^{g_\xi(s)} \sum_{j=1}^{g_\eta(s)} \mathsf{U}\left(\tilde{\mathbf{x}}_n, \hat{\mathbf{x}}(\bar{\xi}_i, \bar{\eta}_j)\right) R_k(\xi(\bar{\xi}_i), \eta(\bar{\eta}_j)) \frac{\Delta \xi_s}{2} \frac{\Delta \eta_s}{2} J(\bar{\xi}_i, \bar{\eta}_j) W_i W_j$$

$$\mathbf{T}^e_{nk} = \sum_{s=1}^{S} \sum_{i=1}^{g_\xi(s)} \sum_{j=1}^{g_\eta(s)} \mathsf{T}\left(\tilde{\mathbf{x}}_n, \hat{\mathbf{x}}(\bar{\xi}_i, \bar{\eta}_j)\right) R_k(\xi(\bar{\xi}_i), \eta(\bar{\eta}_j)) \frac{\Delta \xi_s}{2} \frac{\Delta \eta_s}{2} J(\bar{\xi}_i, \bar{\eta}_j) W_i W_j \tag{42}$$

$$\mathbf{T}^e_{n} = \sum_{s=1}^{S} \sum_{i=1}^{g_\xi(s)} \sum_{j=1}^{g_\eta(s)} \mathsf{T}\left(\tilde{\mathbf{x}}_n, \hat{\mathbf{x}}(\bar{\xi}_i, \bar{\eta}_j)\right) \frac{\Delta \xi_s}{2} \frac{\Delta \eta_s}{2} J(\bar{\xi}_i, \bar{\eta}_j) W_i W_j$$

where $g_\xi(s), g_\eta(s)$ is the number of Gauss points in ξ, η directions, W_i, W_j are Gauss weights, S is the number of subregions and J is the Jacobian of the transformation from global coordinates $\mathbf{x}$ to local ξ coordinates.

4.2. Singular integration

If the collocation point is part of the subregion then the integral involving U is weakly singular. The integral is solved by subdividing the integration region into triangular subregions with the collocation point at the apex (Fig. 8). This means that the Jacobian tends to zero as the collocation point is approached. The singular integration can now be written as

$$\triangle \mathbf{U}^e_{nk} = \sum_{n_t=1}^{N_t} \sum_{i=1}^{g_\xi} \sum_{j=1}^{g_\eta} \mathsf{U}\left(\tilde{\mathbf{x}}_n, \hat{\mathbf{x}}(\bar{\xi}_i, \bar{\eta}_j)\right) R_k(\xi(\bar{\xi}_i), \eta(\bar{\eta}_j)) 0.25 \ J_{\xi, n_t}(\bar{\xi}_i, \bar{\eta}_j) J(\bar{\xi}_i, \bar{\eta}_j) W_i W_j \tag{43}$$

where N_t is the number of triangles. There are now two Jacobians involved, one for the transformation from the patch coordinates to triangular coordinates ($J_{\xi, n_t}(\bar{\xi}_i, \bar{\eta}_j)$) which tends to zero and one for the transformation from patch to global coordinates ($J(\bar{\xi}_i, \bar{\eta}_j)$).

5. Integration of integrals over Ω_0

For the inclusion integrals we use numerical integration for general inclusions and analytical integration for linear inclusions. For the numerical integration we subdivide the inclusion into integration regions with the same criteria as used for the boundary integration and apply Gauss quadrature. When point n is part of the integration region we have to invoke singular integration, if it is not a regular one.

5.1. General inclusions

The integral to be solved for each inclusion is :

$$\mathbf{B}^{ni}_{0n} = \int_{\Omega_{ni}} \mathsf{E}(\tilde{\mathbf{x}}_n, \hat{\mathbf{x}}) \sigma_0(\hat{\mathbf{x}}) d\Omega_{ni}(\hat{\mathbf{x}}) \tag{44}$$
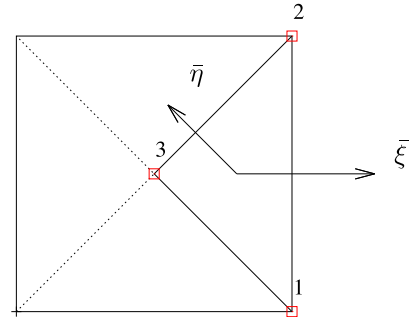

Fig. 8. Definition of a triangular subregion. The collocation point is located at point 3.

Introducing the interpolation of initial stresses (38) we have:

$$\mathbf{B}^{ni}_{0n} = \int_{\Omega_{ni}} \mathsf{E}(\tilde{\mathbf{x}}_n, \hat{\mathbf{x}}) \sum_{j=1}^{J} M_j^\sigma(\hat{\mathbf{x}}) \sigma_{0j} = \sum_{j=1}^{J} \mathbf{B}^{ni}_{0nj} \sigma_{0j} \tag{45}$$

where

$$\mathbf{B}^{ni}_{0nj} = \int_{\Omega_{ni}} \mathsf{E}(\tilde{\mathbf{x}}_n, \hat{\mathbf{x}}) M_j^\sigma(\hat{\mathbf{x}}) d\Omega_{ni}(\hat{\mathbf{x}}) \tag{46}$$

5.1.1. Regular integration

To maintain adequate precision of integration we subdivide the integration region into subregions depending on the size of the integration region and the proximity of point n. For sub-region n_s the transformation from the inclusion (s) coordinates to the coordinates used for Gauss integration $\bar{\xi} = (\bar{\xi}, \bar{\eta}, \bar{\zeta})^{\mathrm{T}} = [-1, 1]^3$ is given by

$$s = \frac{\Delta s_n}{2}(1 + \bar{\xi}) + s_{n_s}$$
$$t = \frac{\Delta t_n}{2}(1 + \bar{\eta}) + t_{n_s} \tag{47}$$
$$r = \frac{\Delta r_n}{2}(1 + \bar{\zeta}) + r_{n_s}$$

where $\Delta s_n \times \Delta t_n \times \Delta r_n$ denotes the size of the sub-region and s_n, t_n, r_n are the edge coordinates. The Jacobian of this transformation is $J_\xi^n = \frac{1}{8} \Delta s_n \Delta t_n \Delta r_n$.

We can write:

$$\mathbf{B}^{ni}_{0nj} = \sum_{n_s=1}^{N_s} \int_{-1}^{1} \int_{-1}^{1} \int_{-1}^{1} \mathsf{E}\left(\tilde{\mathbf{x}}_n, \bar{\mathbf{x}}(\bar{\xi}, \bar{\eta}, \bar{\zeta})\right) M_j^\sigma\left(\bar{\mathbf{x}}(\bar{\xi}, \bar{\eta}, \bar{\zeta})\right) J(s) J_\xi^{n_s} \ d\bar{\xi} d\bar{\eta} d\bar{\zeta} \tag{48}$$

where $J(\mathbf{s})$ is the Jacobian of the mapping between s and $\mathbf{x}$ coordinate systems.

Applying Gauss integration we have:

$$\mathbf{B}^{ni}_{0nj} \approx \sum_{n_s=1}^{N_s} \sum_{g_s=1}^{G_s} \sum_{g_t=1}^{G_t} \sum_{g_r=1}^{G_r} \mathsf{E}\left(\tilde{\mathbf{x}}_n, \bar{\mathbf{x}}(\bar{\xi}_{g_s}, \bar{\eta}_{g_t}, \bar{\zeta}_{g_r})\right) M_j^\sigma\left(\bar{\mathbf{x}}(\bar{\xi}_{g_s}, \bar{\eta}_{g_t}, \bar{\zeta}_{g_r})\right) J(s) J_\xi^{n_s} W_{g_s} W_{g_t} W_{g_r} \tag{49}$$

where N_s is the number of subregions and G_s, G_t and G_r are the number of Gauss points (which depends on the proximity of n) and $\bar{\xi}_{g_s}, \bar{\eta}_{g_t}, \bar{\zeta}_{g_r}$ the Gauss point coordinates in s, t and r directions, respectively. $W_{g_s} W_{g_t} W_{g_r}$ are Gauss weights.

5.1.2. Singular integration

If the integration region includes the point $\mathbf{x}_n$, then the integrand tends to infinity as the point is approached. To deal with the integration involving the weakly singular Kernel we compute the Gauss points in a local coordinate system, where the Jacobian tends to zero as the singularity point is approached. Singular integration of general inclusions is discussed in detail in Beer et al. (2019).
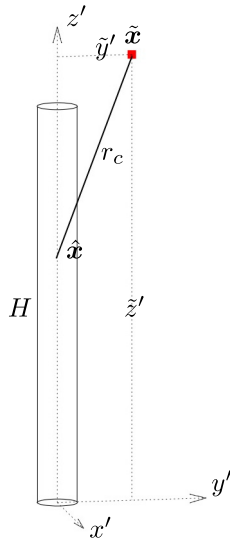
Fig. 9. Analytical computation of regular integral for a subregion of length H of a linear inclusion.
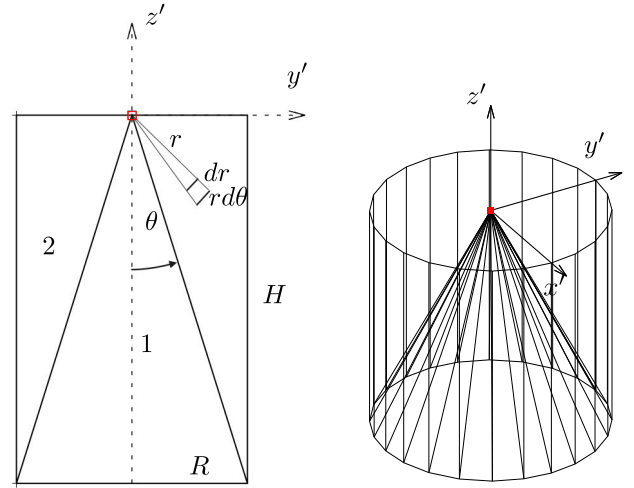

Fig. 10. Explanation of singular integration by subdivision into conical subregions. Left: section through bar, right: Axonometric view. Singular point is marked by red square.

5.2. Linear inclusion, reinforcement bar

For the linear inclusions we can apply analytical integration. We model the bar as a cylindrical region with radius R and length H and assume the initial stress to be in the local z' direction and to vary linearly along the bar. Since the initial stresses are computed from the strains, this means that the displacements along the bar can have a quadratic variation. We consider two types of integration: one where point $\tilde{x}$ is outside the inclusion (regular integration) and one where it is not (singular integration).

5.2.1. Analytical computation of regular integral

Since we assume that the cross-sectional area is significantly smaller than the surrounding medium we can assume that E is constant over the cross-section. In addition we note that the result will multiply with the initial stresses in local directions ($\{\sigma'_0\}$). The integral to be solved is therefore:

$$\mathbf{B}'_{0nl} = \int_\Omega \frac{1}{r_c^2} \tilde{\mathbf{E}}' M_l^\sigma d\Omega_i(\hat{x}) \tag{50}$$

where the prime indicates that the result is computed in the local x', y', z' coordinate system (Fig. 9). r_c is the distance between the source point and a point on the axis of the inclusion.

The integral to be solved is:

$$\triangle\mathsf{E}'_{ij} = -C \int_V \frac{1}{r_c^2} \left[C_3(r'_{,k}\delta_{ij} + r'_{,j}\delta_{ik}) - r'_{,i}\delta_{jk} + C_4\, r'_{,i}r'_{,j}r'_{,k} \right] M_l^\sigma dV \tag{51}$$

We choose the local axes such that $\tilde{x}' = 0$ as follows. The vector pointing in the x' direction is given by:

$$\mathbf{V}_{x'} = (\tilde{x} - \hat{x}) \times \mathbf{v}_{z'} \tag{52}$$

and the one in y' direction is:

$$\mathbf{V}_{y'} = \mathbf{v}_{z'} \times \mathbf{v}_{x'} \tag{53}$$

where the capital letter indicates that the vector is not normalised.

If point $\tilde{x}$ is along the axis of the bar this computation does not work and then we assume

$$\mathbf{v}_{x'} = \mathbf{v}_y \times \mathbf{v}_{z'} \tag{54}$$

where $\mathbf{v}_y$ is a vector in global y-direction.

For the computation of the fundamental solution we have:

$$r_1 = 0 \quad r_2 = -\bar{y}' \quad r_3 = z' - \tilde{z}' \quad r = r_c = \sqrt{\bar{y}'^2 + (z' - \tilde{z}')^2} \tag{55}$$

and

$$r'_{,1} = 0 \quad r_{,2} = -\frac{\bar{y}'}{r_c} \quad r'_{,3} = \frac{z' - \tilde{z}'}{r_c} \qquad dV = \pi R^2 d\bar{z}' \tag{56}$$

The integral to be solved is:

$$\triangle\mathsf{E}'_{ij} = \pi R^2 C \int_{z'=0}^H \frac{1}{r_c^2} \left[C_3(r_{,k}\delta_{ij} + r_{,j}\delta_{ik}) - r_{,i}\delta_{jk} + C_4\, r_{,i}r_{,j}r_{,k} \right] M_l^\sigma(z')dz' \tag{57}$$

where the linear interpolation functions are given by:

$$M_1^\sigma(z') = \frac{z'}{H} \quad M_2^\sigma = 1 - \frac{z'}{H} \tag{58}$$

The analytical solution in Voigt notation is provided in Appendix A.

Since the result of the multiplication with σ'_0 has to be in global coordinates a transformation to the global system is necessary:

$$\mathbf{B}_{0nj} = \mathbf{T}\mathbf{B}'_{0nj} \tag{59}$$

where $\mathbf{T}$ is the transformation matrix given by:

$$\mathbf{T} = \begin{pmatrix} v_{x'_x} & v_{y'_x} & v_{z'_x} \\ v_{x'_y} & v_{y'_y} & v_{z'_y} \\ v_{x'_z} & v_{y'_z} & v_{z'_z} \end{pmatrix} \tag{60}$$

5.2.2. Analytical computation of singular integral

Referring to Fig. 10 we subdivide the bolt into 2 subregions and obtain the following integrals in polar coordinates:

$$\triangle\mathsf{E}'_1 = \int_{\phi=0}^{2\pi} \int_{\theta=\pi-\tilde{\theta}}^\pi \int_{r=0}^{\frac{H}{\cos(\pi-\theta)}} \frac{1}{r^2} \tilde{\mathbf{E}}' \sin\theta dr\ r^2 M_l^\sigma d\theta d\phi \tag{61}$$

$$\triangle\mathsf{E}'_2 = \int_{\phi=0}^{2\pi} \int_{\theta=\pi/2}^{\pi-\tilde{\theta}} \int_{r=0}^{\frac{R}{\sin(\pi-\theta)}} \frac{1}{r^2} \tilde{\mathbf{E}}' \sin\theta dr\ r^2 M_l^\sigma d\theta d\phi$$

with $\tilde{\theta} = \arctan(R/H)$. It can be seen that the r^2 term cancels out which means that the integrand is no longer singular.

The shape functions can be expressed in cylindrical coordinates in two different formats, depending on the position of the collocation point, i.e.:

$$M_1^\sigma(r,\theta) = -r\frac{\cos\theta}{H} \tag{62}$$

$$M_2^\sigma(r,\theta) = 1 + r\frac{\cos\theta}{H} \tag{63}$$

if the collocation point is on the top of the cylinder,

$$M_1^\sigma(r, \theta) = r \frac{\cos \theta}{H} \tag{64}$$

$$M_2^\sigma(r, \theta) = 1 - r \frac{\cos \theta}{H} \tag{65}$$

if the collocation point is on the bottom of the cylinder. In such a way we have that:

$$\Delta E'^{\bar{x}_n = top}(j, k) = -\Delta E'^{\bar{x}_n = bottom}(j, k) \tag{66}$$

The terms of $\Delta E' = \Delta E'_1 + \Delta E'_2$, in Voigt notation, different from zero are provided in Appendix B.

As before a transformation to the global system is necessary:

$$\mathbf{B}_{0nj} = \mathbf{T} \mathbf{B}'_{0nj} \tag{67}$$

6. System of equations

The discretised integral equations can be written in matrix form as:

$$[\mathbf{L}]\{\mathbf{x}\} = \{\mathbf{r}\} + [\mathbf{B}_0]\{\sigma_0\} \tag{68}$$

where $\mathbf{L}$ is an assembled left hand side, $\mathbf{x}$ is the vector of unknowns and $\mathbf{r}$ is the assembled right hand side involving known boundary values (for details of derivation see Beer et al., 2019). $[\mathbf{B}_0]$ is a matrix where the rows refer to the collocation points n and which multiplies with vector $\{\sigma_0\}$ that gathers all initial stress components at grid points inside the inclusions.

7. Computation of values at grid points inside the inclusion

To compute the initial stresses at grid points we need to compute the strains at these points. Even though it is possible to compute the strains directly using integral equations this is rather cumbersome as it involves the integration of strongly singular functions and involves complicated methods to isolate the singularity (see for example Gao and Davies, 2011). To avoid this we compute the displacements first and then compute the strains using a method that is used in the Finite Element Method.

7.1. Computation of displacements

The displacement vector $\mathbf{u}$ at a grid point x inside the inclusion is given by:

$$\mathbf{u}(\mathbf{x}) = \int_\Gamma [\mathsf{U}(\mathbf{x}, \hat{\mathbf{x}})\, \mathbf{t}(\hat{\mathbf{x}}) - \mathsf{T}(\mathbf{x}, \hat{\mathbf{x}})\, \mathbf{u}(\hat{\mathbf{x}})]\, d\Gamma(\hat{\mathbf{x}})$$
$$+ \int_{\Omega_0} \mathsf{E}(\mathbf{x}, \hat{\mathbf{x}})\sigma_0(\hat{\mathbf{x}}) d\Omega_0(\hat{\mathbf{x}}) \tag{69}$$

After the solution the values $\mathbf{u}(\hat{\mathbf{x}})$ and $\mathbf{t}(\hat{\mathbf{x}})$ are known, so the integrals can be evaluated numerically.

We gather displacement vectors at all grid points in a vector $\{\mathbf{u}\}$ and obtain:

$$\{\mathbf{u}\} = [\hat{\mathbf{A}}]\mathbf{x} + \{\bar{\mathbf{c}}\} + [\bar{\mathbf{B}}_0]\{\sigma_0\} \tag{70}$$

where $[\hat{\mathbf{A}}]$ is an assembled matrix that multiplies with the unknown $\mathbf{x}$ and $\{\bar{\mathbf{c}}\}$ collects the displacement contribution due to given BC's. $[\bar{\mathbf{B}}_0]$ is similar to $[\mathbf{B}_0]$ except that the grid point coordinates x_i replace the source point coordinates $\tilde{x}_n$.

Because of the singularity of T the displacements cannot be computed on the problem boundary. So if the inclusion point lies on a boundary patch we recover the displacement from the computed boundary values. For points on a patch boundary (x_k) we replace Eq. (69) by:

$$\mathbf{u}(\mathbf{x}_k) = \sum_i^I R_i^u(\xi_k, \eta_k)\mathbf{u}_i^e \tag{71}$$

where $R_i^u(\xi, \eta)$ are the NURBS basis functions used for approximating the displacements in patch e, that contains the point $\mathbf{x}_k$ and ξ_k, η_k are the local coordinates of the point. The matrix $[\hat{\mathbf{A}}]$ and the vector $\{\bar{\mathbf{c}}\}$ have to be modified for these grid points, whereas $[\bar{\mathbf{B}}_0]$ will contain zero rows in this case.

7.2. Computation of strains, general inclusions

To compute the strains we interpolate the displacements between grid points and obtain for the displacement at a point with the local coordinate s:

$$\mathbf{u}(s) = \sum_{n=1}^N M_n(s)\mathbf{u}_n \tag{72}$$

where $\mathbf{u}_n$ is the displacement vector at grid point n and N is the number of grid points. The interpolation functions $M_n(s, r, t)$ are obtained by considering grid coordinates s, t, r.

Rewriting (72) for a point with the local coordinates s, t, r in terms of local interpolation functions we have:

$$\mathbf{u}(s, t, r) = \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K L_i(s)L_j(t)L_k(r)\mathbf{u}_{n(i,j,k)} \tag{73}$$

where $L_i(s)$, $L_j(t)$, $L_k(r)$ are piecewise constant, linear or quadratic interpolation functions of the local coordinates s, t, r respectively and I, J, K specify the span of the function in the local directions s, t, r, i.e. 1 for constant, 2 for linear and 3 for quadratic interpolation. $n(i, j, k)$ is the grid node number corresponding to i, j, k. The interpolation functions have zero values outside the span.

Replacing the 3 sums by one we have

$$\mathbf{u}(s, t, r) = \sum_{n=1}^N M_{n(i,j,k)}(s, t, r)\mathbf{u}_{n(i,j,k)} \tag{74}$$

where N is the total number of grid points and

$$M_{n(i,j,k)}(s, t, r) = L_i(s)L_j(t)L_k(r) \tag{75}$$

The derivatives of the displacements are given by

$$\frac{\partial \mathbf{u}(s, t, r)}{\partial s} = \sum_{n=1}^N \frac{\partial M_n}{\partial s}\mathbf{u}_n \tag{76}$$

$$\frac{\partial \mathbf{u}(s, t, r)}{\partial t} = \sum_{n=1}^N \frac{\partial M_n}{\partial t}\mathbf{u}_n \tag{77}$$

$$\frac{\partial \mathbf{u}(s, t, r)}{\partial r} = \sum_{n=1}^N \frac{\partial M_n}{\partial r}\mathbf{u}_n \tag{78}$$

where

$$\frac{\partial M_n}{\partial s} = \frac{\partial L_i(s)}{\partial s}L_j(t)L_k(r) \tag{79}$$

$$\frac{\partial M_n}{\partial t} = L_i(s)\frac{\partial L_j(t)}{\partial t}L_k(r) \tag{80}$$

$$\frac{\partial M_n}{\partial r} = L_i(s)L_j(t)\frac{\partial L_k(r)}{\partial r} \tag{81}$$

Unfortunately we cannot use NURBS for the interpolation functions as they are based on parameter values instead of real values. Therefore they cannot be used to interpolate the real displacement values at internal points. We use Lagrange polynomials instead.

The strains are given by:

$$\epsilon_x = \frac{\partial u_x}{\partial x} = \sum_{n=1}^N \frac{\partial M_n}{\partial x}u_{xn} \tag{82}$$

$$\epsilon_y = \frac{\partial u_y}{\partial y} = \sum_{n=1}^N \frac{\partial M_n}{\partial y}u_{yn} \tag{83}$$

$$\epsilon_z = \frac{\partial u_z}{\partial z} = \sum_{n=1}^N \frac{\partial M_n}{\partial z}u_{zn} \tag{84}$$

$$\gamma_{xy} = \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} = \sum_{n=1}^{N} \frac{\partial M_n}{\partial x} u_{yn} + \sum_{n=1}^{N} \frac{\partial M_n}{\partial y} u_{xn} \tag{85}$$

$$\gamma_{zy} = \frac{\partial u_z}{\partial y} + \frac{\partial u_y}{\partial z} = \sum_{n=1}^{N} \frac{\partial M_n}{\partial z} u_{yn} + \sum_{n=1}^{N} \frac{\partial M_n}{\partial y} u_{zn} \tag{86}$$

$$\gamma_{xz} = \frac{\partial u_x}{\partial z} + \frac{\partial u_z}{\partial x} = \sum_{n=1}^{N} \frac{\partial M_n}{\partial x} u_{zn} + \sum_{n=1}^{N} \frac{\partial M_n}{\partial z} u_{xn} \tag{87}$$

The strains at grid point k can be written in matrix notation:

$$\epsilon(\mathbf{x}_k) = \hat{\mathbf{B}}(\mathbf{x}_k)\{\mathbf{u}\} \tag{88}$$

where

$$\hat{\mathbf{B}}(\mathbf{x}_k) = \begin{pmatrix} \mathbf{B}_1 & \mathbf{B}_2 & \cdots \end{pmatrix} \tag{89}$$

and

$$\mathbf{B}_i = \begin{pmatrix} \frac{\partial M_i}{\partial x} & 0 & 0 \\ 0 & \frac{\partial M_i}{\partial y} & 0 \\ 0 & 0 & \frac{\partial M_i}{\partial z} \\ \frac{\partial M_i}{\partial y} & \frac{\partial M_i}{\partial x} & 0 \\ 0 & \frac{\partial M_i}{\partial z} & \frac{\partial M_i}{\partial y} \\ \frac{\partial M_i}{\partial z} & 0 & \frac{\partial M_i}{\partial x} \end{pmatrix} \tag{90}$$

The global derivatives of M_n are given by:

$$\begin{pmatrix} \frac{\partial M_n}{\partial x} \\ \frac{\partial M_n}{\partial y} \\ \frac{\partial M_n}{\partial z} \end{pmatrix} = \mathbf{J}^{-1} \begin{pmatrix} \frac{\partial M_n}{\partial s} \\ \frac{\partial M_n}{\partial t} \\ \frac{\partial M_n}{\partial r} \end{pmatrix} \tag{91}$$

where $\mathbf{J}$ is the Jacobian matrix Eq. (33). For a linear inclusion we compute the strain in local directions as is shown later.

Gathering all strain vectors at grid points in $\{\epsilon\}$ we can write:

$$\{\epsilon\} = [\hat{\mathbf{B}}]\{\mathbf{u}\} \tag{92}$$

After substitution of $\{\mathbf{u}\}$ we obtain:

$$\{\epsilon\} = [\hat{\mathbf{B}}]\left([\hat{\mathbf{A}}]\mathbf{x} + \{\bar{\mathbf{c}}\} + [\bar{\mathbf{B}}_0]\{\boldsymbol{\sigma}_0\}\right) \tag{93}$$

The initial stresses are computed by

$$\{\boldsymbol{\sigma}_0\} = \begin{bmatrix} \mathbf{D} - \mathbf{D}_{incl} \end{bmatrix} \{\epsilon\} = \begin{bmatrix} \mathbf{D} - \mathbf{D}_{incl} \end{bmatrix} [\hat{\mathbf{B}}]\left([\hat{\mathbf{A}}]\mathbf{x} + \{\bar{\mathbf{c}}\} + [\bar{\mathbf{B}}_0]\{\boldsymbol{\sigma}_0\}\right) \tag{94}$$

where $\begin{bmatrix} \mathbf{D} - \mathbf{D}_{incl} \end{bmatrix}$ is a matrix containing $\mathbf{D} - \mathbf{D}_{incl}$ as sub-matrices on the diagonal.

7.2.1. Computation of strain for linear inclusions

For linear inclusions it is convenient to work with the strain in local coordinates. If we assume the bolt to be fully grouted, i.e. no slip is allowed between the bolt and the domain it is embedded in and that the Poisson's ratio of the bolt has no effect, the only strain that has to be considered is the one along the bar[2]:

$$\epsilon_{z'} = \frac{\partial u_{z'}}{\partial z'} = \sum_{n=1}^{N} \frac{\partial M_n}{\partial z'} u_{z'n} = \sum_{n=1}^{N} \frac{\partial M_n}{\partial s} \frac{1}{J} (\mathbf{v}_{z'} \cdot \mathbf{u}_n) \tag{95}$$

where J is the Jacobian and $\mathbf{v}_{z'}$ is a unit vector in z' direction.

Eq. (90) now becomes

$$\mathbf{B}_n = \frac{1}{J} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{\partial M_n}{\partial s} v_{z'_x} & \frac{\partial M_n}{\partial s} v_{z'_y} & \frac{\partial M_n}{\partial s} v_{z'_z} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{96}$$

---

[2] It should be noted that this restriction can be lifted, i.e. slip can be considered.

The local initial stress vector is given by:

$$\{\boldsymbol{\sigma}_0'\} = (\mathbf{D}' - \mathbf{D}_{incl}')\{\epsilon'\} \tag{97}$$

where

$$(\mathbf{D}' - \mathbf{D}_{incl}') = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & E - E_{incl} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{98}$$

where E and E_{incl} is the Young' modulus of the domain and the inclusion respectively and

$$\{\boldsymbol{\sigma}_0'\} = \begin{Bmatrix} 0 \\ 0 \\ \sigma_{0z'} \\ 0 \\ 0 \\ 0 \end{Bmatrix} \tag{99}$$

8. Solution procedure

A solution that already includes the effect of inclusions that have different elastic properties is possible by combining equation (68) with (93). Eq. (93) can be written in the following form:

$$\{\epsilon\} = [\hat{\mathbf{C}}]\{\mathbf{x}\} + \{\bar{\mathbf{c}}\} + [\bar{\mathbf{C}}_0]([\mathbf{D}] - [\mathbf{D}_{incl}])\{\epsilon\} \tag{100}$$

where:

$$[\hat{\mathbf{C}}] = [\hat{\mathbf{B}}][\hat{\mathbf{A}}], \qquad [\hat{\mathbf{C}}_0] = [\hat{\mathbf{B}}][\bar{\mathbf{B}}_0], \qquad \{\bar{\mathbf{c}}\} = [\hat{\mathbf{B}}]\{\bar{\mathbf{c}}\}, \tag{101}$$

Eq. (100) along with Eq. (68) form the following linear system of equations:

$$\begin{pmatrix} [\mathbf{L}] & -[\mathbf{B}_0]([\mathbf{D}] - [\mathbf{D}_{incl}]) \\ -[\hat{\mathbf{C}}] & [\mathbf{I}] - [\hat{\mathbf{C}}_0]([\mathbf{D}] - [\mathbf{D}_{incl}]) \end{pmatrix} \begin{pmatrix} \{\mathbf{x}\} \\ \{\epsilon\} \end{pmatrix} = \begin{pmatrix} \{\mathbf{r}\} \\ \{\bar{\mathbf{c}}\} \end{pmatrix} \tag{102}$$

that can be solved in terms of boundary unknowns and internal strains.

It is also possible to obtain a system of equations that only multiplies with the boundary unknown:

$$[\mathbf{L}]'\{\mathbf{x}\} = \{\mathbf{r}\}' \tag{103}$$

where $[\mathbf{L}]'$ and $\{\mathbf{r}\}'$ are modified left and right hand sides.

We rewrite the strain vector as:

$$\{\epsilon\} = ([\mathbf{I}] - [\hat{\mathbf{C}}_0]([\mathbf{D}] - [\mathbf{D}_{incl}])^{-1})([\hat{\mathbf{C}}]\{\mathbf{x}\} + \{\bar{\bar{\mathbf{c}}}\}) = [\mathbf{A}]\{\mathbf{x}\} + \{\mathbf{b}\} \tag{104}$$

where

$$[\mathbf{A}] = ([\mathbf{I}] - [\hat{\mathbf{C}}_0]([\mathbf{D}] - [\mathbf{D}_{incl}]))^{-1}[\hat{\mathbf{C}}] \qquad \{\mathbf{b}\} = ([\mathbf{I}] - [\hat{\mathbf{C}}_0]([\mathbf{D}] - [\mathbf{D}_{incl}]))^{-1}\{\bar{\bar{\mathbf{c}}}\} \tag{105}$$

Eq. (104) can be inserted in Eq. (68) in order to obtain:

$$[\mathbf{L}]\{\mathbf{x}\} = \{\mathbf{r}\} + [\mathbf{B}_0]([\mathbf{D}] - [\mathbf{D}_{incl}])([\mathbf{A}]\{\mathbf{x}\} + \{\mathbf{b}\}) \tag{106}$$

and, hence, the following system of equations can be obtained

$$([\mathbf{L}] - [\mathbf{B}_0])([\mathbf{D}] - [\mathbf{D}_{incl}])[\mathbf{A}]\{\mathbf{x}\} = \{\mathbf{r}\} + [\mathbf{B}_0]([\mathbf{D}] - [\mathbf{D}_{incl}])\{\mathbf{b}\} \tag{107}$$

The matrices in Eq. (103) are defined by:

$$[\mathbf{L}]' = ([\mathbf{L}] - [\mathbf{B}_0])([\mathbf{D}] - [\mathbf{D}_{incl}])[\mathbf{A}] \tag{108}$$

$$\{\mathbf{r}\}' = \{\mathbf{r}\} + [\mathbf{B}_0]([\mathbf{D}] - [\mathbf{D}_{incl}])\{\mathbf{b}\} \tag{109}$$

8.1. Elasto-plasticity, Newton–Raphson method

If the stress at an inclusion point exceeds the elastic limit incremental/iterative elasto-plastic procedures, well known in the FEM, are

applied. A detailed discussion of these methods is beyond the scope of this paper. A very good description can be found in Smith et al. (2013).

First we consider that $[\mathbf{D}_{incl}] = [\mathbf{D}_{e,incl}]$ for the case of elastic behaviour and $[\mathbf{D}_{incl}] = [\mathbf{D}_{ep,incl}]$ for the case of elasto-plastic behaviour, where $[\mathbf{D}_{e,incl}]$ is the elastic constitutive matrix and $[\mathbf{D}_{ep,incl}]$ is the elasto-plastic constitutive matrix. We proceed in an incremental/iterative way and check after each increment if the yield function $F(\sigma)$ is smaller or greater than zero.

The increment of stress in the plastic regime is:

$$\triangle\sigma = \mathbf{D}_{ep,incl}\triangle\epsilon^* \tag{110}$$

where $\triangle\epsilon^*$ is the in-elastic strain increment (i.e. the one that occurs after the stress has reached a state where $F(\sigma)=0$) . The in-elastic ($\triangle\epsilon^*$) and elastic ($\triangle\epsilon^e$) strain increments can be computed by:

$$\triangle\epsilon^* = f\triangle\epsilon \quad , \quad \triangle\epsilon^e = (1-f)\triangle\epsilon \tag{111}$$

where $\triangle\epsilon$ is the total plastic strain increment and

$$f = \frac{F_{new}}{F_{new} - F_{old}} \quad \text{if } f > 0 \quad f = 0 \text{ otherwise} \tag{112}$$

F_{new} is the value of F at the end of the increment, F_{old} is the value at the beginning.

The initial stress due to plasticity is given by:

$$\triangle\sigma_0^p = (\mathbf{D} - \mathbf{D}_{ep,incl})\triangle\epsilon^* = (\mathbf{D} - \mathbf{D}_{ep,incl})f\triangle\epsilon \tag{113}$$

If the increment in total strain $\triangle\epsilon$ has occurred while traversing the yield surface then:

$$\sigma_0^e = (\mathbf{D} - \mathbf{D}_{e,incl})(1-f)\triangle\epsilon \tag{114}$$

The total initial stress increment, including plastic effects, is given by

$$\triangle\sigma_0 = \triangle\sigma_0^e + \triangle\sigma_0^p = ((1-f)(\mathbf{D} - \mathbf{D}_{e,incl}) + f(\mathbf{D} - \mathbf{D}_{ep,incl}))\triangle\epsilon = \mathbf{D}'\triangle\epsilon \tag{115}$$

where

$$\mathbf{D}' = ((1-f)(\mathbf{D} - \mathbf{D}_{e,incl}) + f(\mathbf{D} - \mathbf{D}_{ep,incl})) \tag{116}$$

If the strain increment is totally elastic (f=0) we have:

$$\triangle\sigma_0 = \mathbf{D}'\triangle\epsilon = (\mathbf{D} - \mathbf{D}_{e,incl})\triangle\epsilon \tag{117}$$

If the strain increment is totally plastic (f=1) we have:

$$\triangle\sigma_0 = \mathbf{D}'\triangle\epsilon = (\mathbf{D} - \mathbf{D}_{ep,incl})\triangle\epsilon \tag{118}$$

To start the simulation we set f in Eq. (116) equal to zero i.e. $\mathbf{D}' = \mathbf{D}_{e,incl}$ and obtain the first result:

$$[\mathbf{L}]'\{\mathbf{x}_0\} = \{\mathbf{r}\}' \tag{119}$$

With this result we compute the value of the yield function at internal points and compute a load factor λ that reduces the load to the one where first yield occurred. We then reduce the results to first yield:

$$\{\mathbf{x}_0\} => \lambda\{\mathbf{x}_0\} \tag{120}$$

We also adjust the stresses at internal points to the new load level. For the first load step ($n=1$) the matrix $\mathbf{D}'$ is updated with $f = 1$ for the point where the first yield has occurred.

We apply the rest of the loading in n steps. The increment in load is:

$$\triangle\{\mathbf{r}\}' = \frac{1-\lambda}{n}\{\mathbf{r}\}' \tag{121}$$

At each increment we solve:

$$[\mathbf{L}]'\{\triangle\mathbf{x}_i\} = \{\triangle\mathbf{r}\}' \tag{122}$$

For the second and subsequent load steps we update the matrix $\mathbf{D}'$ with f according to the current state of stress. Standard return algorithms can be applied to ensure that the stresses stay on the yield surface.
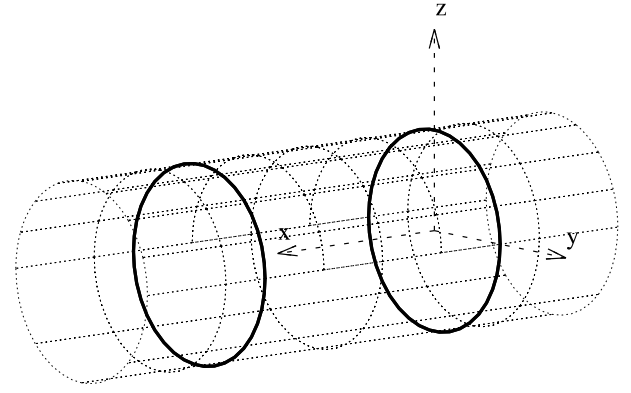

Fig. 11. Geometry of test example.

Remark. Note that incremental/iterative procedures need only be applied for elasto-plastic behaviour. When inclusions are defined that have different elastic behaviour the solution is obtained without iteration.

9. Test examples

We test the implementation on an example of a circular, infinitely long, tunnel in an infinite domain subjected to a virgin stress (see Fig. 11). Since the aim of the examples is to test the accuracy of the implementation and not to perform a realistic simulation, we use non-dimensional parameters. For the domain we assume E=1 and $\nu = 0$ and for the radius of the tunnel, R=1. The results of the IGABEM simulation are compared with a FEM analysis using the software PLAXIS and an analytical calculation.

9.1. Elastic analysis with no ground support

In order to establish a comparable regime between the IGABEM and PLAXIS simulations we first conduct an elastic simulation without ground support and with a virgin stress $\sigma_x = 0, \sigma_y = 0, \sigma_z = -1$.

9.1.1. Discretisation with PLAXIS

The infinite domain is approximated by providing an artificial boundary at some distance from the tunnel. Since this distance affects the results we examine the error introduced by varying the distances from the tunnel centre to the artificial boundary to 2.5, 5 and 10 times the tunnel diameter (2.5D, 5D, 10D). The boundary conditions at the outer boundary of the mesh (shown in Fig. 12) are that displacements normal to the faces of the cuboid are set to zero. The fact that the tunnel is infinitely long can be modelled by extending the mesh along the tunnel axis to 10 m (=5D). The mesh consists of quadratic 10-noded tetrahedral elements. For load case 0 the virgin stresses are assigned to all elements of the mesh and the excavation is then simulated by deactivating the elements inside the tunnel. This will give the same result as a much simpler plane strain simulation but the same mesh will be used for the second test example where a 3-D analysis is necessary.

9.1.2. Discretisation with IGABEM

Only the boundary of the tunnel is discretised as the infinite domain is explicitly considered by the fundamental solutions. The excavation boundary is defined by 16 control points and basis functions of order 2 (quadratic) along the tunnel walls and of order 1 (linear) along the tunnel axis are used. It should be noted that this geometrical description exactly represents a circular tunnel. To simulate the infinite extent of the tunnel we use plane strain infinite patches. The excavation of the tunnel is simulated by assigning a virgin stress state. Excavation forces on the boundary are then automatically computed. The discretisation is
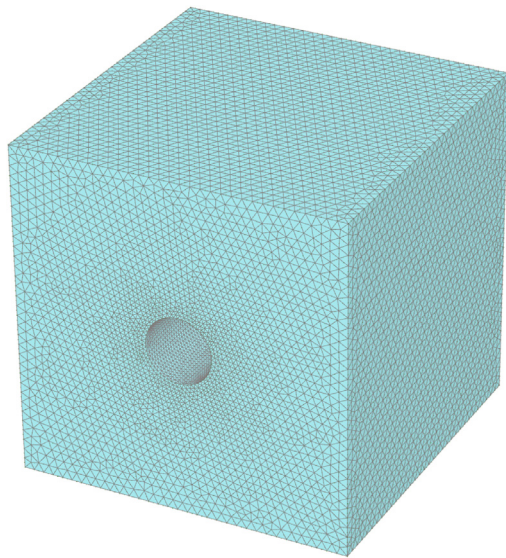
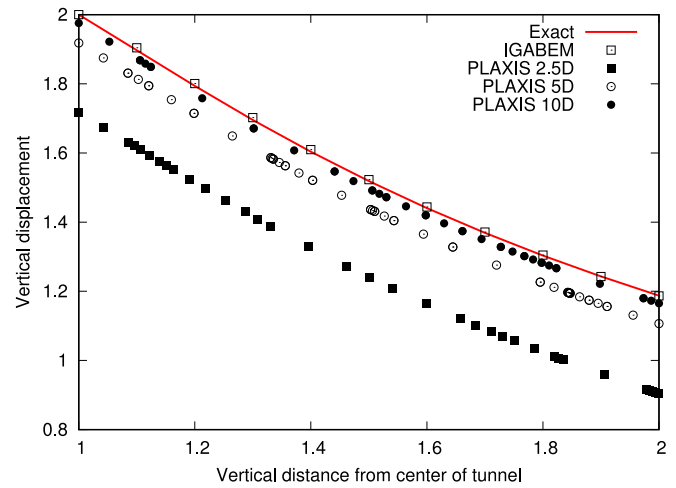**Fig. 12.** Finite element mesh with distance 2.5D from the tunnel centre to the outer boundary.

Fig. 14. Variation of the vertical displacement above the tunnel: Comparison of FEM results with the BEM result and the exact solution.
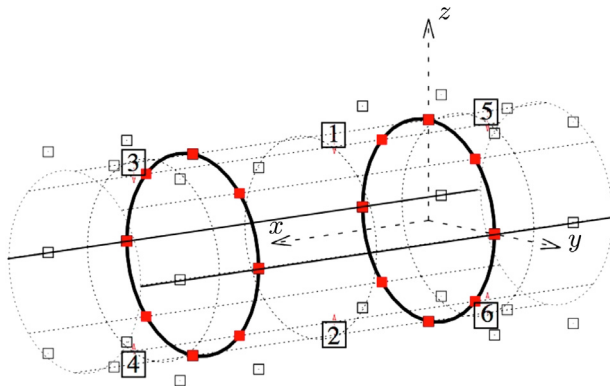

Fig. 13. Discretisation of tunnel into 6 patches. Patch numbers are shown. Patches 3 to 6 are infinite. Control points are shown as hollow squares, collocation points as filled squares.

Fig. 15. Geometry of the elastic simulation with bolts.

shown in Fig. 13. For the approximation of the displacements the same basis functions as for the description of the geometry are used, resulting in the collocation points shown. The discretisation has 48 degrees of freedom.

9.1.3. Comparison of results

To investigate the effect of the artificial boundary on the results of the FEM analysis we compare the values of vertical displacements along a vertical line above the tunnel with the exact result (Kirsch solution Kirsch, 1898) and the IGABEM result. It can be seen in Fig. 14 that the FEM results converge to the exact results, while the IGABEM result already is in excellent agreement. The computing time for the PLAXIS simulation was 3 min 53 s and for the IGABEM simulation 8 s.

9.2. Elastic simulation with rock bolts

This example is designed to test the implementation of rock bolts. Three rock bolts are installed at the top of the tunnel (Fig. 15). The diameter of the bolts is 0.05 and the elastic modulus is twice the value of the domain (i.e. $E_{bolt} = 2$). It is noted that the consideration of the rock bolts renders the analysis three dimensional. The aim is to study
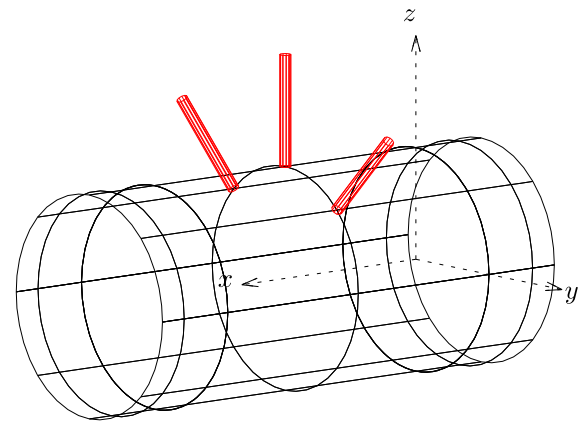
the local influence of the rock bolts, assuming that further away plane strain conditions prevail.

9.2.1. Discretisation with PLAXIS

In PLAXIS rock bolts are simulated as embedded beams. The geometry of the bolt can be arbitrarily located in the finite element mesh, thus nodes of the bolt do not need to coincide with nodes of the finite element mesh. As the beam cross section is small, compared with its length, the bending stiffness is almost zero, therefore it can be assumed that the beam acts as a bolt. In Fig. 16 the three bolts are shown embedded in the surrounding FEM mesh.

9.2.2. Discretisation with IGABEM

The bolts are defined as linear inclusions. In order to get results comparable to PLAXIS the continuity of displacements is changed to C^0 at the point where the rock bolts meet the boundary patch. This is done be inserting knots into the basis functions that approximate the unknown. This increases the number of collocation points and the degrees of freedom. The IGABEM discretisation is shown in Fig. 17 and has 108 degrees of freedom.

9.2.3. Comparison of results

We compare the displacements along the rock bolts in Fig. 18. Good agreement can be observed. The computation time of the PLAXIS simulation was 6 min and 12 s and for the IGABEM simulation 24 s.
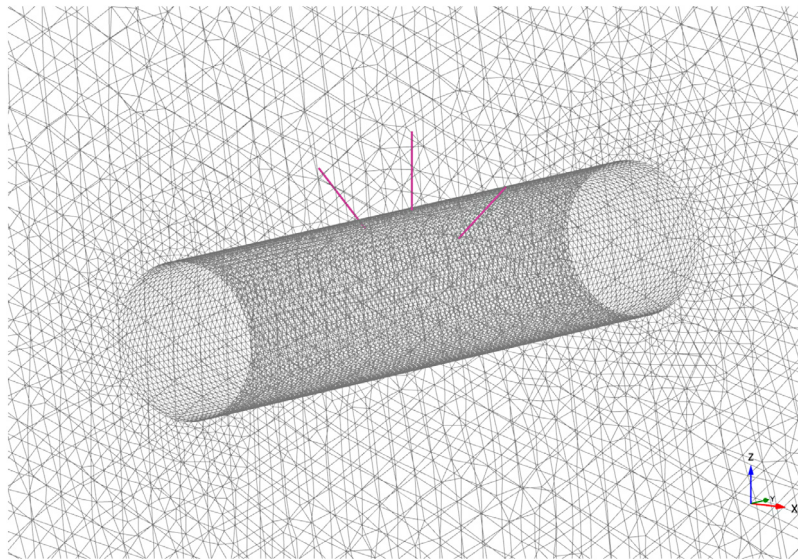
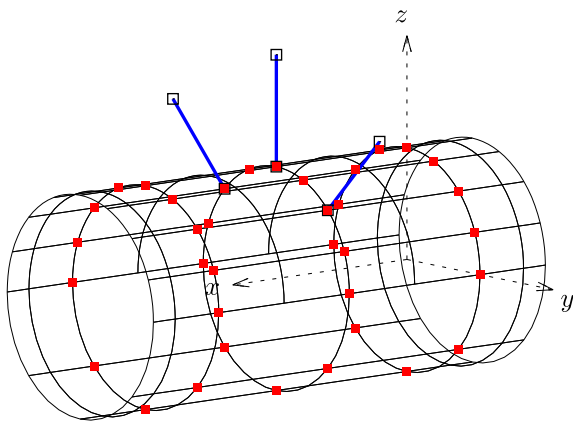**Fig. 16.** Bolts simulated as embedded beams with PLAXIS.

Fig. 17. Discretisation of the problem with IGABEM. Shown are the bolts in blue with the associated control points as hollow squares. The collocation points obtained after knot insertion are shown as red filled squares.
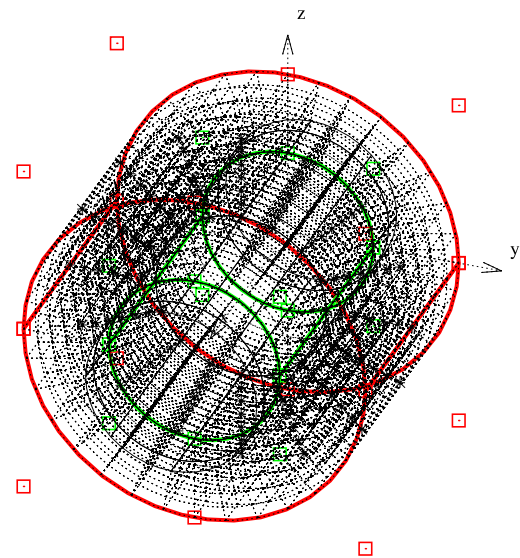

Fig. 19. Volume discretisation for the plasticity example. Shown are the two (green and red) surfaces defining the inclusion and the control points as hollow squares. Also shown are the internal points as stars and the subdivision into integration regions for the volume integration. These regions are automatically determined depending on the location of internal points and on the aspect ratio.
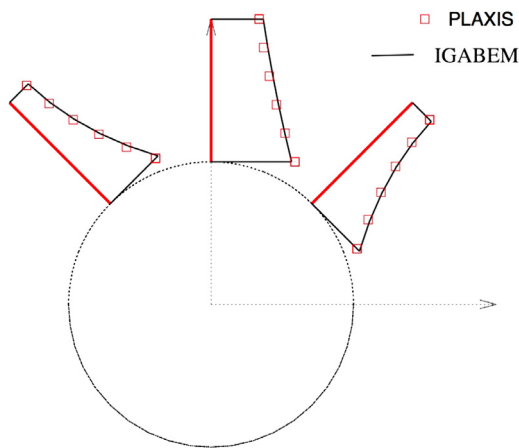

Fig. 18. Comparison of displacements along the rock bolts.

9.3. Elasto-plastic simulation without rock bolts

For this test we can compare with an analytical solution in plane strain due to Duncan–Fama (see for example Hoek and Brown, 1980). For a hydrostatic virgin stress of p_0 (compression positive) and a Mohr–Coulomb yield condition with a friction angle ϕ and cohesion c, the solution for the elasto-plastic radial displacement of the tunnel wall is given by:

$$u_p = \frac{R(1+\nu)}{E} \left(2(1-\nu)(p_0 - p_{cr})\frac{r_p}{R}^2 - (1-2\nu)p_0 \right) \tag{123}$$

where the extent of the plastic zone is given by:

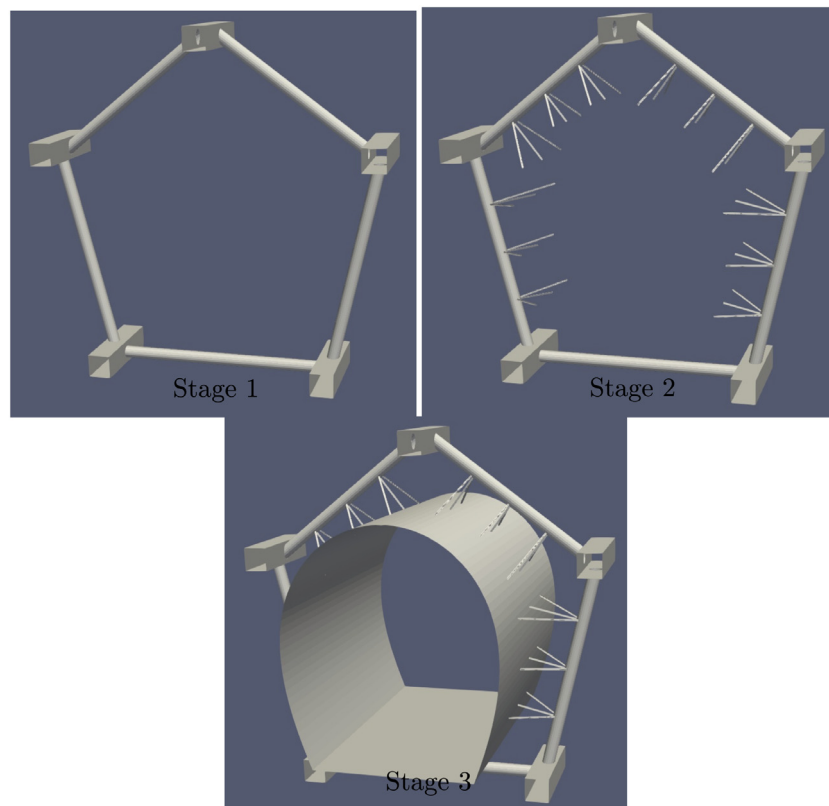$$r_p = R \left(\frac{2p_0(k-1) + \sigma_{cm}}{(1+k)\sigma_{cm}} \right)^{\frac{1}{k-1}} \tag{124}$$

Fig. 20. Example of large cavern: Perspective view of excavation stages.

and

$$\sigma_{cm} = \frac{2c\ cos\phi}{1 - sin\phi} \tag{125}$$

$$k = \frac{1 + sin\phi}{1 - sin\phi} \tag{126}$$

Table 1
Rock mass properties.

E (MPa)	v	c (MPa)	ϕ (degrees)
23000	0.25	1.63	36

9.3.1. Discretisation with IGABEM

For this simulation the excavation is surrounded with a general inclusion extending one radius from the excavation surface as shown in Fig. 19. The properties assigned to the inclusion are $E = 1$, $v = 0$, $\phi = 10°$, $c = 0.5$. A hydrostatic compressive virgin stress $p_0 = 1$ was applied.

9.3.2. Comparison of results

For the above input data the analytical solution for the extent of the plastic zone was computed as 1.3, which means it is within the inclusion. The theoretical solution for the radial displacement was 1.262 which compares well with the IGABEM solution of 1.269. Convergence to 1% of residual was achieved after 6 iterations.

10. Practical example

The practical example relates to the excavation of a large underground cavern with a height of 60 m, a width of 50 m and an extension of 300 m. The cavern is at a depth of 500 m resulting in a vertical virgin compressive stress of 15 MPa with $k_0 = 0.8$. The rock mass properties are listed in Table 1. A Mohr–Coulomb yield condition with a dilation angle of $\psi = 0$ is assumed.

The aim of the simulation is to investigate if a pre-installation of ground support would lead to a safer and more economic excavation. The required excavation stages are shown on a 60 m section of the cavern in Fig. 20. In the first stage a tunnel system consisting of circular tunnels (by raise boring) and rectangular horizontal tunnels (by blasting) is excavated. Next cables are installed from the circular

tunnels of such length that they reach to the surface of the cavern to be excavated. Finally the cavern is excavated. The cross-sectional diameter of the cables is 25 mm with the modulus of elasticity of E=210 GPa.

10.1. The simulation model

For a preliminary analysis we analyse a 60 m section of the cavern. At the edges of the section we use infinite plane strain boundary elements. Nine cables per circular excavation are considered. We start with the definition of the excavation surfaces using the method outlined in Section 3.1.5. For the description of the smooth shape of the cavern walls 4 control points and a basis function of order 3 has been used for each half. For the variation of the unknowns we use the same basis functions as for the description of the geometry except that for the bottom surface the basis function was elevated by one order from linear to quadratic across the cavern.

The resulting simulation model is shown in Fig. 21 for the final excavation stage and has 612 degrees of freedom. **It should be stressed that no mesh generation is involved here. The dotted lines in** Fig. 21 **define integration regions, which are automatically determined depending of the location of collocation points and their aspect ratio.**

Next the inclusions are specified. This relates to the cables and to the part of the domain where it is assumed that non-linear behaviour is taking place. In this study we concentrate on the rock mass behaviour between the tunnels and the excavation surface, so this part is selected.
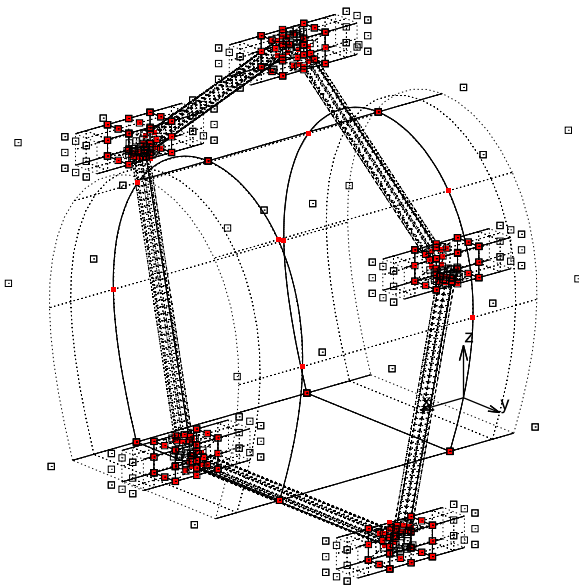
Fig. 21. IGABEM model of excavation surfaces, showing control points as hollow squares and collocation points as filled red squares.
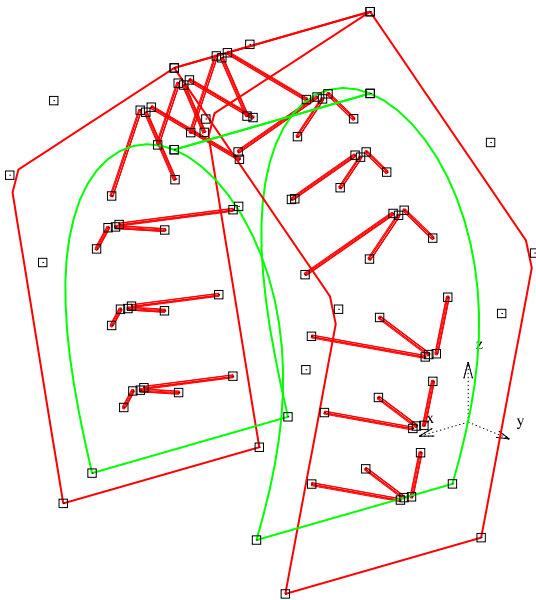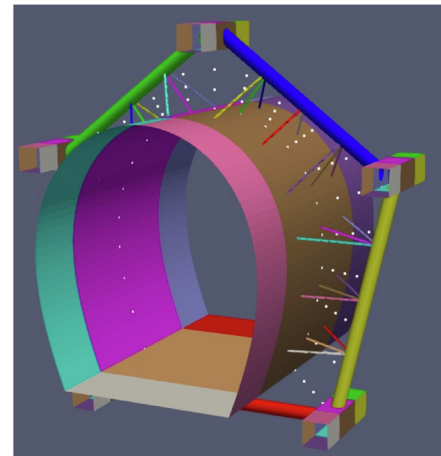

Fig. 23. Perspective view of geometry definition for the final excavation stage. Boundary Patches are colour coded. The grid points inside the general inclusion for modelling non-linear behaviour are shown as white dots.

Fig. 22. Definition of the inclusions showing bolt inclusions and bounding surfaces (red and green lines) defining the part of the domain where non-linear behaviour is modelled.
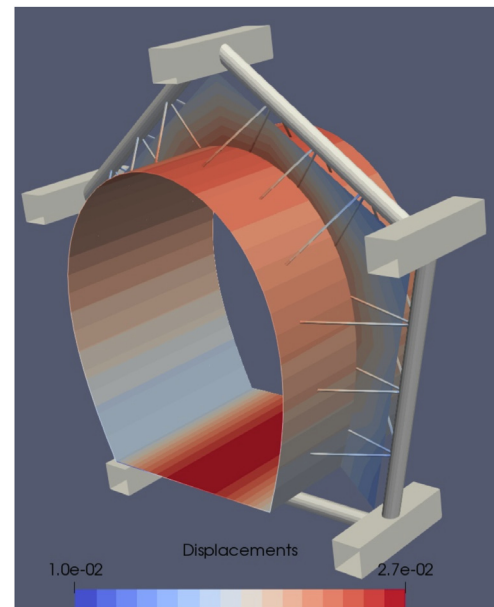

Fig. 24. Result of the analysis: Contours of absolute displacement plotted on the cavern surface, the cables and a plane inside the general inclusion.

The definition of the inclusions is shown in Fig. 22 and the final model in Fig. 23.

10.2. Preliminary results

The novel approach to simulation will be used for a detailed study to determine if pre-installation of ground support can make the excavation of large underground caverns more feasible, safe and economic. Here only a preliminary result is presented in Fig. 24. For this case the convergence to 1% residual was achieved in 10 iterations.

It is clear that more cables are required to provide adequate pre-support but it should be noted that the number of cables can be increased substantially without significantly increasing the simulation effort. This is because cables are generated automatically and because

the numerical effort only increases slightly due to the fact that analytical integration is used. It should be noted that the number of degrees of freedom is not increased by increasing the number of cables.

11. Summary and conclusions

A new approach to simulation for underground excavations using isogeometric methods and NURBS has been presented, which makes it extremely user friendly and efficient. No mesh generation is required and geometric data can be specified in a CAD data format. Since CAD software also uses NURBS and trimmed NURBS surfaces the connection to CAD is natural. Parts of the domain that have different material properties or behave in an inelastic way can be considered but a definition of inclusions is necessary. This means that for most practical applications only near field effects are considered in the simulation, i.e inclusions have limited extent and are located near the region of interest. If this restriction can be accepted, then the software can do

everything a FEM simulation can do, but with a drastic reduction of user effort and execution time.

One of the main contributions of the paper is the presentation of analytically integrated rock bolt inclusions. Numerical integration is the most compute intensive aspect of any BEM simulation. In case of large patches the number of Gauss points can be high. This is because of the nature of the integrals, which exhibit singular behaviour, i.e. where the value of the integrand increases rapidly as the source point is approached. For the numerical volume integration, where the integration is in 3 directions the computational effort is particularly high. However, it is noted that the matrices that require volume integration only depend on geometry and can be precomputed for a particular model. Also, the computations are able to exploit to a very high degree the use of multiple processor hardware.

At the time of writing the simulation model has been written in MATLAB, had a first application to a real problem and will be used for a detailed study to simulate the effect of pre-support for large caverns.

Although concrete arches and shotcrete can be modelled with the general inclusion approach presented here, it becomes cumbersome, when the thickness is small. Therefore, special shotcrete inclusions with a semi-analytical integration are being developed next. It should be pointed out that it is also possible to develop inclusion geometries, that are not restricted to the cuboid shape presented here (in the sense that only two surfaces can be of general shape).

Finally, it is hoped that this contribution will make the BEM more attractive for simulations of underground excavations.

Declaration of competing interest

Acknowledgements

Appendix A. Linear inclusion. Volume regular integral

The analytical solution in Voigt notation (only terms different from zero are listed) is for $l = 1$:

$$\Delta E'(1,4)(\tilde{y}' \neq 0) = 2C\pi R^2 \frac{C_3}{H\tilde{y}'} \left[r_{c0} + \frac{\tilde{z}'\Delta z' - \tilde{y}'^2}{r_{c1}} \right]$$

$$\Delta E'(1,6)(\tilde{y}' \neq 0) = 2C\pi R^2 \frac{C_3}{H} \left[\frac{H}{r_{c1}} + \ln\left(\frac{r_{c0} - \tilde{z}'}{r_{c1} + \Delta z'} \right) \right]$$

$$\Delta E'(1,6)(\tilde{y}' = 0) = 2C\pi R^2 \frac{C_3}{H} \left\{ \frac{\tilde{z}'}{|\Delta z'|} - \frac{\tilde{z}'}{|\tilde{z}'|} - \ln\left[(\Delta z')^{\Delta z'} \left(-\tilde{z}' \right)^{\tilde{z}'} \right] \right\}$$

$$\Delta E'(2,1)(\tilde{y}' \neq 0) = C\pi \frac{R^2}{H\tilde{y}'} \left(\frac{\tilde{y}'^2 - \tilde{z}'\Delta z'}{r_{c1}} - r_{c0} \right)$$

$$\Delta E'(2,2)(\tilde{y}' \neq 0) = C\pi \frac{R^2}{H\tilde{y}'} \left\{ \frac{\tilde{z}'^2 + 2C_3 r_{c0}^2}{r_{c0}} + \right.$$
$$\left. + \frac{\Delta z'\left[\tilde{z}'\Delta z'^2 + (H + \tilde{z}')\tilde{y}'^2 \right] - 2C_3 r_{c1}^2(\tilde{y}'^2 - \tilde{z}'\Delta z')}{r_{c1}^3} \right\}$$

$$\Delta E'(2,3)(\tilde{y}' \neq 0) = C\pi R^2 \frac{\tilde{y}'}{H} \left(\frac{1}{r_{c0}} - \frac{r_{c0}^2 + 2H^2 - 3H\tilde{z}'}{r_{c1}^3} \right)$$

$$\Delta E'(2,5)(\tilde{y}' \neq 0) = 2C\pi \frac{R^2}{H} \left\{ \frac{H\left[(C_3 - 1)H(\Delta z' - \tilde{z}') + C_3 r_{c0}^2 - \tilde{z}'^2 \right]}{r_{c1}^3} + \right.$$
$$\left. \tilde{z}'(\frac{1}{r_{c1}} - \frac{1}{r_{c0}}) + C_3 \ln \frac{r_{c0} - \tilde{z}'}{r_{c1} + \Delta z'} \right\}$$

$$\Delta E'(2,5)(\tilde{y}' = 0) = \Delta E'_{M_1}(1,6)(\tilde{y}' = 0) \tag{A.1}$$

$$\Delta E'(3,1)(\tilde{y}' \neq 0) = C\pi \frac{R^2}{H} \left(\ln \frac{r_{c1} + \Delta z'}{r_{c0} - \tilde{z}'} - \frac{H}{r_{c1}} \right)$$

$$\Delta E'(3,1)(\tilde{y}' = 0) = -\frac{\Delta E'_{M_1}(1,6)(\tilde{y}' = 0)}{2C_3}$$

$$\Delta E'(3,2)(\tilde{y}' \neq 0)$$
$$= C\pi \frac{R^2}{H} \left[-\frac{\tilde{z}'}{r_{c0}^2} - \frac{\Delta z'(2H\Delta z' - H\tilde{z}' + r_{c0}^2)}{r_{c1}^3} + \ln \frac{\Delta z' + r_{c1}}{r_{c0} - \tilde{z}'} \right]$$

$$\Delta E'(3,2)(\tilde{y}' = 0) = \Delta E'_{M_1}(3,1)(\tilde{y}' = 0)$$

$$\Delta E'(3,3)(\tilde{y}' \neq 0)$$
$$= C\pi \frac{R^2}{H} \left\{ \frac{H}{r_{c1}^3} \left[(3 + 2C_3)H^2 + 2(1 + C_3)\tilde{y}'^2 - 2(3 + 2C_3)H\tilde{z}' + \right. \right.$$
$$\left. + (3 + 2C_3)\tilde{z}'^2 \right] + \tilde{z}'(\frac{1}{r_{c0}} - \frac{1}{r_{c1}}) - 2(1 + C_3) \ln \frac{r_{c1} + \Delta z'}{r_{c0} - \tilde{z}'} \right\}$$

$$\Delta E'(3,3)(\tilde{y}' = 0) = \frac{1 + C_3}{C_3} \Delta E'_{M_1}(1,6)(\tilde{y}' = 0)$$

$$\Delta E'(3,5)(\tilde{y}' \neq 0)$$
$$= 2C\pi \frac{R^2}{H\tilde{y}'} \left[\frac{(2 + C_3)\tilde{y}'^2 + (1 + C_3)\tilde{z}'^2}{r_{c0}} + \frac{(1 + C_3)\tilde{z}'\Delta z'^3}{r_{c1}^3} - \right.$$
$$\left. - \frac{(2 + C_3)\tilde{y}'^4 + \tilde{y}'^2\Delta z'((3 + C_3)\Delta z' - C_3\tilde{z}')}{r_{c1}^3} \right]$$

and for $l = 2$:

$$\Delta E'(1,4)(\tilde{y}' \neq 0) = 2C\pi R^2 \frac{C_3}{H\tilde{y}'} \left[r_{c1} + \frac{\tilde{z}'\Delta z' - \tilde{y}'^2}{r_{c0}} \right]$$

$$\Delta E'(1,6)(\tilde{y}' \neq 0) = -2C\pi R^2 \frac{C_3}{H} \left[\frac{H}{r_{c0}} + \ln\left(\frac{r_{c0} - \tilde{z}'}{r_{c1} + \Delta z'} \right) \right]$$

$$\Delta E'(1,6)(\tilde{y}' = 0) = 2C\pi R^2 \frac{C_3}{H} \left\{ \frac{\Delta z'}{|\Delta z'|} - \frac{\Delta z'}{|\tilde{z}'|} + \ln\left[(\Delta z')^{\Delta z'} \left(-\tilde{z}' \right)^{\tilde{z}'} \right] \right\}$$

$$\Delta E'(2,1)(\tilde{y}' \neq 0) = C\pi \frac{R^2}{H\tilde{y}'} \left(\frac{y'^2 - \tilde{z}'\Delta z'}{r_{c0}} - r_{c1} \right)$$

$$\Delta E'(2,2)(\tilde{y}' \neq 0)$$
$$= C\pi \frac{R^2}{H\tilde{y}'} \left[\frac{(1 + 2C_3)H^2 + \tilde{z}'^2 - 2H\tilde{z}'(1 + 2C_3) + 2C_3 r_{c0}^2}{r_{c1}} + \right.$$
$$\left. + \frac{\tilde{z}'(r_{c0}^2\Delta z' + H\tilde{y}'^2) - 2C_3 r_{c0}^2(\tilde{y}'^2 - \tilde{z}'\Delta z')}{r_{c0}^3} \right]$$

$$\Delta E'(2,3)(\tilde{y}' \neq 0) = C\pi R^2 \frac{\tilde{y}'}{H} \left(\frac{1}{r_{c1}} - \frac{r_{c0}^2 + H\tilde{z}'}{r_{c0}^3} \right)$$

$$\Delta E'(2,5)(\tilde{y}' \neq 0)$$
$$= 2C\pi \frac{R^2}{H} \left(\frac{\tilde{z}' - C_3 H}{r_{c0}} - \frac{H\tilde{y}'^2}{r_{c0}^3} + \frac{\Delta z'}{r_{c1}} + C_3 \ln \frac{r_{c1} + \Delta z'}{r_{c0} - \tilde{z}'} \right)$$

$$\Delta E'(2,5)(\tilde{y}' = 0) = \Delta E'_{M_2}(1,6)(\tilde{y}' = 0) \tag{A.2}$$

$$\Delta E'(3,1)(\tilde{y}' \neq 0) = C\pi \frac{R^2}{H} \left(-\ln \frac{r_{c1} + \Delta z'}{r_{c0} - \tilde{z}'} + \frac{H}{r_{c0}} \right)$$

$$\Delta E'(3,1)(\tilde{y}' = 0) = -\frac{\Delta E'_{M_2}(1,6)(\tilde{y}' = 0)}{2C_3}$$

$$\Delta E'(3,2)(\tilde{y}' \neq 0) = C\pi \frac{R^2}{H} \left[\frac{\Delta z'}{r_{c1}} + \frac{\tilde{z}'(H\tilde{z}' + r_{c0}^2)}{r_{c0}^3} - \ln \frac{\Delta z' + r_{c1}}{r_{c0} - \tilde{z}'} \right]$$

$$\Delta E'(3,2)(\tilde{y}' = 0) = \Delta E'_{M_2}(3,1)(\tilde{y}' = 0)$$

$$\Delta E'(3,3)(\tilde{y}' \neq 0) = C\pi \frac{R^2}{H} \left[\frac{H\tilde{y}'^2}{r_{c0}^3} - \frac{2(1 + C_3)H + \tilde{z}'}{r_{c0}} - \frac{\Delta z'}{r_{c1}} + \right.$$

$$+ 2(1 + C_3) \ln \frac{r_{c1} + \Delta z'}{r_{c0} - \bar{z}'} \Bigg]$$

$$\Delta E'(3,3)(\tilde{y}' = 0) = \frac{1 + C_3}{C_3} \Delta E'_{M_2}(1,6)(\tilde{y}' = 0)$$

$$\Delta E'(3,5)(\tilde{y}' \neq 0) = 2C\pi \frac{R^2}{H \tilde{y}'} \left[\frac{(1 + C_3)r_{c0}^2 + \tilde{y}'^2 + (1 + C_3)(H^2 - 2H\bar{z}')}{r_{c1}} \right.$$

$$+ \frac{(1 + C_3)\tilde{z}'^3 \Delta z'}{r_{c0}^3} -$$

$$\left. - \frac{(2 + C_3)\tilde{y}'^4 - \tilde{y}'^2 \tilde{z}'(C_3 \Delta z' - (3 + C_3)\tilde{z}')}{r_{c0}^3} \right]$$

where:

$$\Delta z' = H - \bar{z}' \quad r_{c1} = \sqrt{\tilde{y}'^2 + \Delta z'^2} \quad r_{c0} = \sqrt{\tilde{y}'^2 + z'^2}$$
$$\bar{\Delta z}' = \Delta z'/|\Delta z'| \quad \bar{z}' = \tilde{z}'/|\tilde{z}'|\Delta z' \tag{A.3}$$

Appendix B. Linear inclusion. Volume singular integral

The terms of $\Delta E' = \Delta E'_1 + \Delta E'_2$, in Voigt notation, different from zero are for $l = 1$:

$$\Delta E'(1,6) = \Delta E'(2,5) = \frac{C\pi}{4H} \left\{ H^2 \left[8 + 8C_3 - (9 + 8C_3) \cos \tilde{\theta} + \cos 3\tilde{\theta} + \right. \right.$$

$$\left. \left. R^2 \left((3 - 8C_3) \cos \tilde{\theta} + \cos 3\tilde{\theta} + 8C_3 \ln \frac{\cos \tilde{\theta}/2}{\sin \tilde{\theta}/2} \right) \right] \right\}$$

$$\Delta E'(3,1) = \Delta E'(3,2) = \frac{C\pi}{8H} \left[R^2 \left(11 \cos \tilde{\theta} + \cos 3\tilde{\theta} + 8 \ln \frac{\sin \tilde{\theta}/2}{\cos \tilde{\theta}/2} \right) - \right.$$

$$\tag{B.1}$$

$$\left. - 4H^2 \cos \tilde{\theta} \sin^2 \tilde{\theta} \right]$$

$$\Delta E'(3,3) = -C\pi \frac{R^2}{4H} \left[(11 + 8C_3) \cos \tilde{\theta} + \cos 3\tilde{\theta} + 8(1 + C_3) \ln \frac{\sin \tilde{\theta}/2}{\cos \tilde{\theta}/2} \right] +$$

$$C\pi H \left(1 + 4C_3 + 2\cos \tilde{\theta} + \cos 2\tilde{\theta} \right) \sin^2 \tilde{\theta}/2$$

and for $l = 2$

$$\Delta E'(1,6) = \Delta E'(2,5) = \frac{C\pi}{4H} \left\{ H^2 \left[8 + 8C_3 - (9 + 8C_3) \cos \tilde{\theta} + \cos 3\tilde{\theta} \right] + \right.$$

$$+ R \left[(8C_3 - 3)R \cos \tilde{\theta} - R \cos 3\tilde{\theta} + 8(H + 2C_3 H + \right.$$

$$C_3 R \ln \frac{\sin \tilde{\theta}/2}{\cos \tilde{\theta}/2}) + 4H(-1 - 4C_3 + \cos \tilde{2\theta}) \sin \tilde{\theta} \bigg] \bigg\}$$

$$\Delta E'(3,1) = \Delta E'(3,2) = -\frac{C\pi H}{2} \cos \tilde{\theta} \sin^2 \tilde{\theta} - \frac{C\pi R}{8H} \left[R \left(11 \cos \tilde{\theta} + \cos 3\tilde{\theta} + \right. \right.$$

$$8 \ln \frac{\sin \tilde{\theta}/2}{\cos \tilde{\theta}/2} \right) - 2H \left(5 \sin \tilde{\theta} + \sin 3\tilde{\theta} - 4 \right) \bigg] \tag{B.2}$$

$$\Delta E'(3,3) = C\pi H \left(1 + 4C_3 + 2\cos \tilde{\theta} + \cos 2\tilde{\theta} \right) \sin^2 \tilde{\theta}/2 +$$

$$C\pi \frac{R}{4H} \left[(11 + 8C_3)R \cos \tilde{\theta} + R \cos 3\tilde{\theta} + \right.$$

$$8 \left(H + 2C_3 H + (1 + C_3)R \ln \frac{\sin \tilde{\theta}/2}{\cos \tilde{\theta}/2} \right) -$$

$$4H \left(3 + 4C_3 + \cos 2\tilde{\theta} \right) \sin \tilde{\theta} \bigg]$$

References

Aliabadi, M., 2002. The Boundary Element Method, Volume 2: Applications in Solids and Structures. Wiley.

An, Z., Yu, T., Bui, T., Wang, C., Trinh, N., 2018. Implementation of isogeometric boundary element method for 2-D steady heat transfer analysis. Adv. Eng. Softw. 116, 36–49.

Atroshchenko, E., Tomar, S., Xu, G., Bordas, S.P., 2018. Weakening the tight coupling between geometry and simulation in isogeometric analysis: From sub- and super-geometric analysis to Geometry-Independent Field approximaTion (GIFT). Internat. J. Numer. Methods Engrg. 114 (10), 1131–1159. http://dx.doi.org/10.1002/nme.5778.

Banerjee, P., 1994. The Boundar Element Method in Engineering. McGraw-Hill.

Banerjee, P.K., Butterfield, R., 1981. Boundary Element Methods in Engineering Science. McGraw-Hill.

Banerjee, P., Raveendra, S., 1986. Advanced boundary element of two- and three-dimensional problems of elastoplasticity. Internat. J. Numer. Methods Engrg. 23 (6), 985–1002.

Beer, G., 2015. Mapped infinite patches for the NURBS based boundary element analysis in geomechanics. Comput. Geotech. 66, 66–74.

Beer, G., Marussig, B., Duenser, C., 2019. In: Wriggesrs, P., Eberhard, P. (Eds.), The Isogeometric Boundary Element Method. In: Lecture Notes in Applied and Computational Mechanics, vol. 90, Springer Nature.

Brebbia, C.A., Telles, J.C., Wrobel, L., 1984. Boundary Element Techniques. Springer.

Brebbia, C.A., Walker, S., 1980. Boundary Element Techniques in Engineering. Newnes-Butterworths.

Fang, W., An, Z., Yu, T., Bui, T., 2020. Isogeometric boundary element analysis for two-dimensional thermoelasticity with variable temperature. Eng. Anal. Bound. Elem. 110, 80–94.

Gao, X., Davies, T., 2011. Boundary Element Programming in Mechanics, second ed. Cambridge University Press, Cambridge, UK.

Hoek, E., Brown, T., 1980. Underground Excavations in Rock. CRC Press.

Hughes, T.J.R., Cottrell, J.A., Bazilevs, Y., 2005. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput. Methods Appl. Mech. Engrg. 194 (39–41), 4135–4195.

Kirsch, 1898. Die Theorie der Elastizität und die Bedürfnisse der Festigkeitslehre. Zeitschrift des Vereines deutscher Ingen. 42.

Marussig, B., 2016. Seamless Integration of Design and Analysis through Boundary Integral Equations. In: Monographic Series TU Graz: Structural Analysis, Verlag der Technischen Universität Graz, http://dx.doi.org/10.3217/978-3-85125-440-2.

Marussig, B., Zechner, J., Beer, G., Fries, T.-P., 2015. Fast isogeometric boundary element method based on independent field approximation. Comput. Methods Appl. Mech. Engrg. 284, 458–488.

Pan, Y.-C., Chou, T.-W., 1976. Point force solution for an infinite transversely isotropic solid. J. Appl. Mech. 43, 608–612. http://dx.doi.org/10.1115/1.3423941.

Scott, M.A., Simpson, R.N., Evans, J.A., Lipton, S., Bordas, S.P.A., Hughes, T.J.R., Sederberg, T.W., 2013. Isogeometric boundary element analysis using unstructured T-splines. Comput. Methods Appl. Mech. Engrg. 254, 197–221.

Simpson, R.N., Bordas, S.P., Lian, H., Trevelyan, J., 2013. An isogeometric boundary element method for elastostatic analysis: 2D implementation aspects. Comput. Struct. 118, 2–12.

Simpson, R.N., Bordas, S.P., Trevelyan, J., Rabczuk, T., 2012. A two-dimensional isogeometric boundary element method for elastostatic analysis. Comput. Methods Appl. Mech. Engrg. 209, 87–100.

Smith, I., Griffiths, D.V., Margetts, L., 2013. Programming the Finite Element Method. Wiley.

Sun, F., Gong, Y., Dong, C., 2020. A novel fast direct solver for 3D elastic inclusion problems with the isogeometric boundary element method. J. Comput. Appl. Math. 377, 112904.

Tanaka, M., Matsumoto, T., Takakuwa, S., 2001. Dual reciprocity BEM for time-stepping approach to the transient heat conduction problem in nonlinear materials. Comput. Methods Appl. Mech. Engrg. 195(37-40), 4953–4961.

Wendland, W.L., 1990. On the coupling of finite elements and boundary elements. In: Kuhn, G., Mang, H. (Eds.), Discretization Methods in Structural Mechanics. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 405–414.