Accepted Manuscript

Computer methods in applied mechanics and engineering

Isogeometric Boundary Element analysis with elasto-plastic inclusions. Part 2: 3-D problems

Gernot Beer, Vincenzo Mallardo, Eugenio Ruocco, Benjamin Marussig, Jürgen Zechner, Christian Dünser, Thomas-Peter Fries

PII: S0045-7825(16)30961-6
DOI: http://dx.doi.org/10.1016/j.cma.2016.11.007
Reference: CMA 11217

To appear in: *Comput. Methods Appl. Mech. Engrg.*

Received date: 22 August 2016
Revised date: 5 November 2016
Accepted date: 5 November 2016

Isogeometric Boundary Element Analysis with elasto-plastic inclusions. Part 2: 3-D problems

Gernot Beer[a,b,*], Vincenzo Mallardo[c], Eugenio Ruocco[d], Benjamin Marussig[a], Jürgen Zechner[a,e], Christian Dünser[a], Thomas-Peter Fries[a]

[a]Institute of Structural Analysis, Graz University of Technology, Lessingstraße 25/II, 8010 Graz, Austria
[b]Centre for Geotechnical and Materials Modelling, University of Newcastle, Callaghan, NSW 2308, Australia
[c]Department of Architecture, University of Ferrara, Via della Ghiara 36, 44121 Ferrara, Italy
[d]Department of Civil Engineering, Design, Building and Environment, Second University of Naples, Via Roma 28, 81031 Aversa, Caserta, Italy
[e]TailSiT GmbH, Tailored Simulation Tools, Nikolaiplatz 4, 8020 Graz, Austria

Abstract

A novel approach is presented for the isogeometric Boundary Element analysis of domains that contain predefined inclusions with material properties that are different to the rest of the domain and which also may exhibit inelastic behavior. The paper is complementary to a paper published recently in this journal, which focussed on plane problems. The theoretical framework and implementation are extended to three-dimensional problems here.

In our approach the geometry of inclusions is described using NURBS surfaces. The advantage over currently used methods is that no discretization into cells is required, in order to evaluate the arising volume integrals. The implementation is verified on test cases by comparison with either exact solutions or ones computed by the Finite Element method.

Keywords: BEM, isogeometric analysis, elasto-plasticity, inclusions

1. Introduction

The Boundary Element method (BEM) is ideally suited for a seamless integration with NURBS based Computer Aided Design (CAD) programs [1]. The use of isogeometric methods [2] allows simulations to use geometry information directly form CAD, without the generation a mesh. Implementations of the isogeometric BEM for linear solid mechanics problems and for homogeneous regions have been published recently [3, 4, 5, 6, 7, 8, 9, 10, 11, 12] .

The isogeometric BEM has been extended in a previous paper, published in this journal [13], to allow the simulation of heterogeneous, inelastic domains under plane

[*] Corresponding author. Tel.: +43 316 873 6181, fax: +43 316 873 6185, mail: *gernot.beer@tugraz.at*, web: *www.ifb.tugraz.at*

strain/stress conditions. Here the theoretical framework and implementation is extended to 3-D problems. In the following we attempt to avoid duplication of material from the mentioned 2-D paper and readers are encouraged to read this publication first.

Using the theorem of Betti as explained in [14], the boundary integral equation can be written in incremental form and in matrix notation as:

$$\begin{aligned}\mathbf{c}\dot{\mathbf{u}}(\mathbf{y}) &= \int\limits_S \mathbf{U}(\mathbf{y},\mathbf{x})\dot{\mathbf{t}}(\mathbf{x})\,\mathrm{d}S(\mathbf{x}) + \int\limits_{S_0} \mathbf{U}(\mathbf{y},\bar{\mathbf{x}})\dot{\mathbf{t}}_0(\bar{\mathbf{x}})\,\mathrm{d}S_0(\bar{\mathbf{x}}) \\ &- \int\limits_S \mathbf{T}(\mathbf{y},\mathbf{x})\dot{\mathbf{u}}(\mathbf{x})\,\mathrm{d}S(\mathbf{x}) + \int\limits_{V_0} \mathbf{U}(\mathbf{y},\bar{\mathbf{x}})\dot{\mathbf{b}}_0(\bar{\mathbf{x}})\,\mathrm{d}V_0(\bar{\mathbf{x}})\end{aligned} \tag{1}$$

where $\mathbf{c}$ is a free term, $\mathbf{U}(\mathbf{y},\mathbf{x})$ and $\mathbf{T}(\mathbf{y},\mathbf{x})$ are matrices containing fundamental solutions for the displacements and tractions at a point $\mathbf{x}$ due to a source at a point $\mathbf{y}$ [15], $\dot{\mathbf{u}}(\mathbf{x})$ and $\dot{\mathbf{t}}(\mathbf{x})$ are increments of the displacement and traction vectors on the surface S, defining the problem domain. $\dot{\mathbf{b}}_0(\bar{\mathbf{x}})$ are increments of body force at point $\bar{\mathbf{x}}$ inside the inclusion and $\dot{\mathbf{t}}_0(\bar{\mathbf{x}})$ are increments of tractions related to the body force acting on surface S_0, bounding V_0.

The integral equations can be solved for the unknowns $\mathbf{u}$ or $\mathbf{t}$ by discretization. As in majority of previous work on the isogeometric BEM we use the collocation method, i.e. we write the integral equations for a finite number (N) of source points, $\mathbf{y}_n$

$$\begin{aligned}\mathbf{c}\,\dot{\mathbf{u}}\,(\mathbf{y}_n) &= \int\limits_S \mathbf{U}\,(\mathbf{y}_n,\mathbf{x})\,\dot{\mathbf{t}}\,(\mathbf{x})\,\mathrm{d}S(\mathbf{x}) + \int\limits_{S_0} \mathbf{U}\,(\mathbf{y}_n,\bar{\mathbf{x}})\,\dot{\mathbf{t}}_0\,(\bar{\mathbf{x}})\,\mathrm{d}S_0(\bar{\mathbf{x}}) \\ &- \int\limits_S \mathbf{T}\,(\mathbf{y}_n,\mathbf{x})\,\dot{\mathbf{u}}\,(\mathbf{x})\,\mathrm{d}S(\mathbf{x}) + \int\limits_{V_0} \mathbf{U}\,(\mathbf{y}_n,\bar{\mathbf{x}})\,\dot{\mathbf{b}}_0\,(\bar{\mathbf{x}})\,\mathrm{d}V_0(\bar{\mathbf{x}})\end{aligned} \tag{2}$$

with $n = \{1,\ldots,N\}$.

For the discretization of the surface integrals over S we divide the boundary into patches and use a geometry independent field approximation approach for each patch, i.e. we use different basis functions for the description of the geometry and for the field values.

$$\begin{aligned}\mathbf{x}^e &= \sum_{k=1}^{K} R_k(s,t)\,\mathbf{x}_k^e \\ \mathbf{u}^e &= \sum_{k=1}^{K^d} R_k^d(s,t)\,\mathbf{u}_k^e \\ \mathbf{t}^e &= \sum_{k=1}^{K^t} R_k^t(s,t)\,\mathbf{t}_k^e\end{aligned} \tag{3}$$

In above equations the superscript e refers to the number of the patch, R_k, R_k^d and R_k^t are NURBS basis functions with respect to the local coordinates s,t for the geometry, displacements and tractions respectively. The parameters $\mathbf{x}_k^e$ specify the location of control points. The parameters $\mathbf{u}_k^e$ and $\mathbf{t}_k^e$ relate to the displacements and tractions. K,

K^d, K^t specify the number of parameters for each patch.

For an external Neumann problem for example the system of equations

$$[\mathbf{T}]\{\mathbf{u}\} = \{\mathbf{F}\} + \{\mathbf{F}\}_0 \tag{4}$$

is obtained where $[\mathbf{T}]$ is an assembled matrix with coefficients related to Kernel $\mathbf{T}$ and $\{\mathbf{u}\}$ is a vector that collects all displacement components on points $\mathbf{y}_n$. On the right hand side of (4) the vector $\{\mathbf{F}\}$ is related to the given tractions and $\{\mathbf{F}\}_0 = \{\mathbf{F}\}_0^{S_0} + \{\mathbf{F}\}_0^{V_0}$ is related to the body force effects, i.e. to the integrals over S_0 and V_0 in (2).

Details of the implementation of the isogeometric BEM for 3-D elastic homogeneous domains can be found in [15, 16]. Here we concentrate on the definition of the geometry of the inclusions and on the evaluation of the volume integrals.

2. Basic approach and previous work

The basic approach is to solve the problem in an iterative way. First the elastic problem is solved considering an elastic homogeneous domain. Then the solution is modified to account for the presence of inclusions and inelastic behavior.

The procedure can be summarized as follows:

1. Solve the elastic, homogeneous problem and determine the increment of stress $\dot{\boldsymbol{\sigma}}$ inside the inclusion V_0.
2. Determine an increment in initial stress $\dot{\boldsymbol{\sigma}}_0$ due to the fact that the elastic material properties of the inclusion are different from the ones used for the fundamental solutions and/or due to the fact that the elastic limit has been exceeded.
3. Convert $\dot{\boldsymbol{\sigma}}_0$ to body force and traction increments $\dot{\mathbf{b}}_0$, $\dot{\mathbf{t}}_0$.
4. Compute new right hand side by evaluating the arising volume and surface integrals.
5. Solve for the new right hand side and compute a new increment of stress $\dot{\boldsymbol{\sigma}}$ inside the inclusion.
6. Repeat 2 to 5 until $\dot{\boldsymbol{\sigma}}_0$ is sufficiently small.

2.1. Elastic inclusions

Elastic inclusions can be modeled with the multi-region method (see for example [14]) and this involves an additional discretization and increases the number of unknowns. Here we include their treatment in the iterative process required for plasticity.

To compute the initial stress increment for the case where the inclusions have elastic properties which are different to the ones used for the fundamental solutions we use the relation between increments of stress $\dot{\boldsymbol{\sigma}}$ and strain $\dot{\boldsymbol{\epsilon}}$ in Voigt notation

$$\dot{\boldsymbol{\sigma}} = \mathbf{C}\,\dot{\boldsymbol{\epsilon}} \tag{5}$$

$$\dot{\boldsymbol{\epsilon}} = \mathbf{C}^{-1}\,\dot{\boldsymbol{\sigma}} \tag{6}$$

where $\mathbf{C}$ is the constitutive matrix for the domain, used for the computation of the fundamental solutions. The difference in stress between the inclusion and the domain and therefore the initial stress increment can be computed by

$$\dot{\boldsymbol{\sigma}}_0 \quad = \quad (\mathbf{C}_i - \mathbf{C})\,\dot{\boldsymbol{\epsilon}} \tag{7}$$

where $\mathbf{C}_i$ is the constitutive matrix for the inclusion.

2.2. Inelastic behavior

If the inclusion experiences inelastic behavior then additional initial stresses are generated. Here we use the concept of visco-plasticity, but it is obvious that the method presented here can also be applied to elasto-plasticity. In visco-plasticity we specify a visco-plastic strain rate

$$\frac{\partial \boldsymbol{\epsilon}^{vp}}{\partial t} = \frac{1}{\eta}\Phi(F)\frac{\partial Q}{\partial \boldsymbol{\sigma}} \tag{8}$$

where η is a viscosity parameter, F is the yield function, Q the plastic potential [17]. The function Φ is defined as:

$$\Phi(F) = 0 \quad if \quad F <= 0 \tag{9}$$

$$\Phi(F) = F \quad if \quad F > 0. \tag{10}$$

The visco-plastic strain increment during a time increment Δt can be computed by an explicit scheme by solving

$$\dot{\boldsymbol{\epsilon}}^{vp} = \frac{\partial \boldsymbol{\epsilon}^{vp}}{\partial t}\Delta t. \tag{11}$$

The time step Δt can not be chosen freely and if chosen too large, oscillatory behavior will occur in the solution. Suitable time step values can be found in [18]. The initial stress increment is given by

$$\dot{\boldsymbol{\sigma}}_0 = \mathbf{C}\,\dot{\boldsymbol{\epsilon}}^{vp}. \tag{12}$$

2.3. Previous work

Previous work on this topic has been discussed in detail in [13] and this will not be repeated here. The common approach for the evaluation of the necessary domain integrals in these publications is to use cells. Figure 1 shows a possible discretisation of an inclusion into linear cells. Cells are identical to isoparametric elements, i.e. the definition of their geometry involves a mapping to a local coordinate system ξ, η, ζ. For the evaluation of the contribution of each cell to the volume integral, initial stresses are usually interpolated between nodal points, using Lagrange ploynomials.

The disadvantage of using cells and the motivation for the work presented here, is that this approach involves additional discretisation effort. In addition, because continuous shape functions are used to interpolate the initial stresses inside cells, a problem arises when the elasto-plastic boundary crosses a cell. In this case, errors would be introduced in the integration of the volume terms. The approach presented here, involving a single mapping, allows the region of integration to be subdivided into subregions that accurately represent the plastic zones, thereby avoiding the integration error that may occur with cells. Another motivation is that in applications in geomechanics,

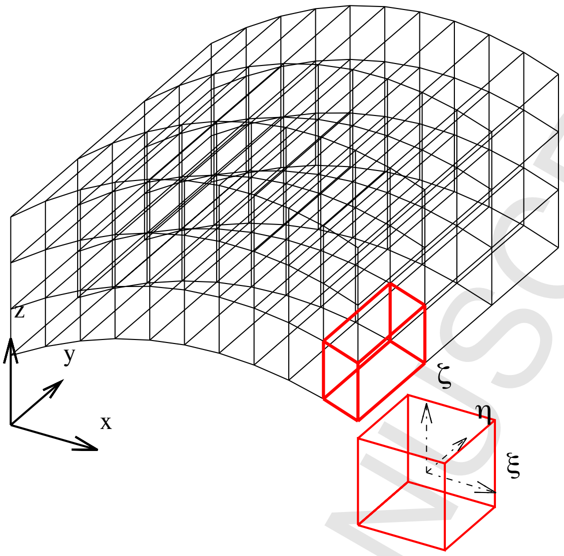

Figure 1: Example of the discretisation of an inclusion with linear cells, showing the mapping of the highlighted cell to a local coordinate system.

geological inclusions are usually defined in CAD data by their bounding surfaces. This opens up the possibility of using this information directly in the simulation.

In the following it is first outlined how the geometry of inclusions is defined using NURBS surfaces and how the arising volume and surface integrals are numerically evaluated. Then the computation of $\{\mathbf{F}\}_0$ in (4) is shown. Finally, the accuracy and efficiency of the proposed approach is demonstrated on two test examples.

3. Geometry definition for inclusions

The first task is the description of the geometry of the subdomain V_0. For this we propose to use a mapping method introduced recently for trimmed surfaces in [8] and [15]. This means that the domain of an inclusion is defined by two NURBS surfaces and a linear interpolation between them.

We establish a local coordinate system $\mathbf{s} = (s,t,r)^\mathsf{T} = [0,1]^3$ as shown in Figure 2 and perform all computations such as integration and differentiation in this system and then map it to the global x,y,z-system. The global coordinates of a point $\mathbf{x}$ with the local coordinates $\mathbf{s}$ are given by

$$\mathbf{x}(s,t,r) = (1-r)\,\mathbf{x}^{I}(s,t) + r\,\mathbf{x}^{II}(s,t) \tag{13}$$

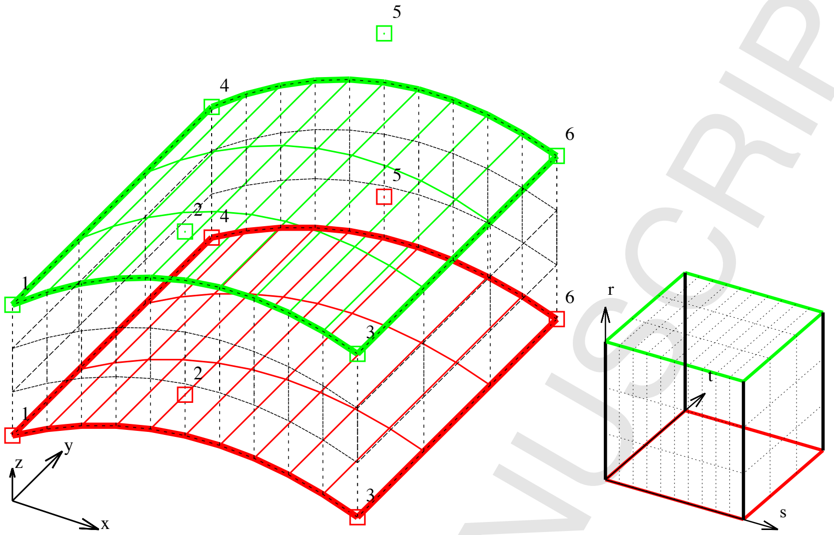

Figure 2: Example of the geometry definition of an inclusion with the proposed approach. Left: the inclusion in the global coordinate system. The bounding surfaces are colour coded and the associated control points depicted by coloured and numbered hollow squares. Right: the local map. Note that in contrast to the cell based approach no discretization and only one map is involved.

where

$$\mathbf{x}^{I}(s,t)=\sum_{k=1}^{K^{I}}R_{k}^{I}(s,t)\,\mathbf{x}_{k}^{I}\quad\text{and}\quad\mathbf{x}^{II}(s,t)=\sum_{k=1}^{K^{II}}R_{k}^{II}(s,t)\,\mathbf{x}_{k}^{II}. \tag{14}$$

The superscript I relates to the bottom (red) surface and II to the top (green) surface and $\mathbf{x}_{k}^{I}$, $\mathbf{x}_{k}^{II}$ 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.

Remark 1: It should be noted that in this mapping approach the bottom and top surfaces may have a different number of control points.

The derivatives are given by

$$\begin{aligned}\frac{\partial\mathbf{x}(s,t,r)}{\partial s}&=(1-r)\frac{\partial\mathbf{x}^{I}(s,t)}{\partial s}+r\frac{\partial\mathbf{x}^{II}(s,t)}{\partial s}\\ \frac{\partial\mathbf{x}(s,t,r)}{\partial t}&=(1-r)\frac{\partial\mathbf{x}^{I}(s,t)}{\partial t}+r\frac{\partial\mathbf{x}^{II}(s,t)}{\partial t}\\ \frac{\partial\mathbf{x}(s,t,r)}{\partial r}&=-\mathbf{x}^{I}(s,t)+\mathbf{x}^{II}(s,t)\end{aligned} \tag{15}$$

where for example:

$$\frac{\partial \mathbf{x}^{I}(s,t)}{\partial s} = \sum_{k=1}^{K^{I}} \frac{\partial R_k^{I}(s,t)}{\partial s}\mathbf{x}_k^{I} \quad \text{and} \quad \frac{\partial \mathbf{x}^{II}(s,t)}{\partial s} = \sum_{k=1}^{K^{II}} \frac{\partial R_k^{II}(s,t)}{\partial s}\mathbf{x}_k^{II}. \tag{16}$$

The Jacobian matrix of this mapping is

$$\mathbf{J} = \begin{pmatrix} \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} & \frac{\partial z}{\partial s} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} & \frac{\partial z}{\partial t} \\ \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} & \frac{\partial z}{\partial r} \end{pmatrix} \tag{17}$$

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

Remark 2: The implementation explained in the following is not restricted to the simple geometry description, outlined above. Any method that allows the mapping of the geometry to a unit cube can be applied. In the following we have used this description mainly in order to simplify the explanation of the method.

4. Computation of $\{\mathbf{F}\}_0$

Here we discuss the computation of the right hand side during iteration. This involves the evaluation of the integrals in Equation (2) over S_0 and V_0 using Gauss quadrature. For an accurate integration it is necessary to subdivide the integration region into subregions. The subdivision strategy is determined by several factors:

1. The limited span of basis functions as governed by the knot vectors.
2. The extent of the plastic zone.
3. An imposed limit on the number of Gauss points available for integration.

4.1. Computation of the surface integral over S_0

For the numerical integration Gauss Quadrature is used. Following the implementation of the integration procedures for the elastic BEM outlined in [15] we distinguish between singular integration when the collocation point is inside the integration region, nearly singular integration when it is close to the region and regular integration when it is further away.

4.1.1. Regular and nearly singular integration

If the collocation point is not inside the integration region then the number of integration points depends on the proximity of the point $\mathbf{y}$ to the region and the size of the region. In the implementation a Quadtree method is used, meaning that the size of the integration region is subdivided until the criterion for the maximum number of Gauss points defined by the user is fulfilled (for further details see [15]). We first integrate over the two surfaces defining the inclusion (i.e. over $s,t = [0,1]^2$ for $r = 0,1$) and then over the edges. A possible subdivision for a given location of the collocation

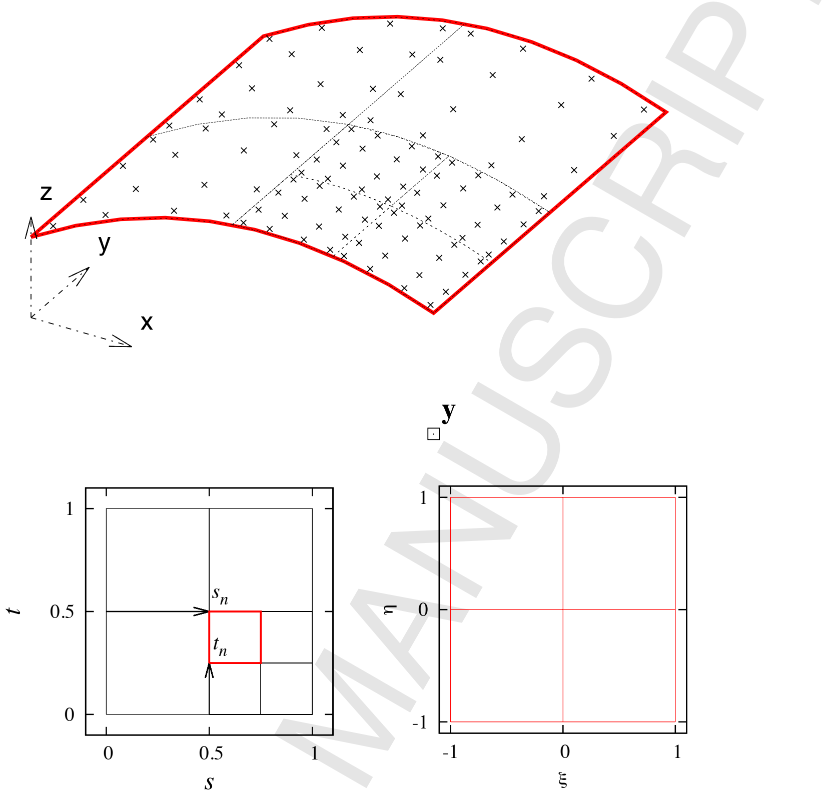

Figure 3: Explanation of the Quadtree method: Surface in the global x,y,z, the local s,t and the ξ,η system. The collocation point is depicted by a hollow square and **y**, Gauss points are depicted by crosses.

point **y**, using the Quadtree method is shown in Figure 3. For the integration along the bounding surfaces I and II the global locations of Gauss points are computed by

$$\mathbf{x}^I(\mathrm{s},\mathrm{t}) = \sum_{k=1}^{K^I} R_k^I(s,t)\,\mathbf{x}_k^I \quad \text{and} \quad \mathbf{x}^{II}(s,t) = \sum_{k=1}^{K^{II}} R_k^{II}(s,t)\,\mathbf{x}_k^{II} \tag{18}$$

The Jacobian of this transformation is J^i. Gauss integration requires the use of a local coordinate system $\boldsymbol{\xi} = [-1,1]^2$. The transformation is given by

$$s = \frac{\Delta s_n}{2}(1+\xi) + s_n \quad \text{and} \quad t = \frac{\Delta t_n}{2}(1+\eta) + t_n \tag{19}$$

where $\Delta s_n, \Delta t_n$ denotes the size of the integration region n_s and s_n, t_n are the local coordinates of the edge of the subregion. The Jacobian of this transformation is J_s^n.

For the integration along the edges we have for example for edge e_1 at $t = 0$:

$$\bar{\mathbf{x}}(s,r) = (1-r)\,\mathbf{x}_1^I(s) + r\,\mathbf{x}_1^{II}(s) \tag{20}$$

where $\mathbf{x}_1^I(s)$ and $\mathbf{x}_1^{II}(s)$ are the coordinates of a point on the top and bottom curves of the bounding surfaces with the local coordinate $t = 0$. The Jacobian of this transformation is J^{e_1}. The transformation to the $\boldsymbol{\xi}$ coordinate system is given by

$$s = \frac{\Delta s_n}{2}(1+\xi) + s_n \quad \text{and} \quad r = \frac{\Delta r_n}{2}(1+\eta) + r_n \tag{21}$$

where $\Delta s_n, \Delta r_n$ denotes the size of the integration region n_s and s_n, r_n are the local coordinates of the edge of the subregion. The Jacobian of this transformation is $J^n_{e_1}$

We can now write the sub vector of $\{\mathbf{F}\}_0^{S_0}$ related to the collocation point n as

$$\mathbf{F}_{0n}^{S_0} = \sum_{i=1}^{2}\sum_{n_s=1}^{N_s}\int_{-1}^{1}\int_{-1}^{1}\mathbf{U}(\mathbf{y}_n,\bar{\mathbf{x}})\dot{\mathbf{t}}_0(\bar{\mathbf{x}})J^i J_s^{n_s}\;\mathrm{d}\xi\,\mathrm{d}\eta + \sum_{j=1}^{4}\sum_{n_s=1}^{N_s}\int_{-1}^{1}\int_{-1}^{1}\mathbf{U}(\mathbf{y}_n,\bar{\mathbf{x}})\dot{\mathbf{t}}_0(\bar{\mathbf{x}})Je_j J_{e_j}^{n_s}\;\mathrm{d}\xi\,\mathrm{d}\eta \tag{22}$$

where N_s is the number of subregions.

4.1.2. Singular integration

If the collocation point is located on S_0 the integrand approaches infinity as the point is approached. Here we apply the method that has been used for dealing with weakly singular integrals over surface S. It involves the transformation to a local coordinate system where the Jacobian tends to zero as the collocation point is approached. This method is well documented (see for example [15]). An example of a singular integration is given in Figure 4.

4.2. Computation of the volume integral over V_0

For subregion n_s of integration the transformation from $\mathbf{s}$ coordinates to $\boldsymbol{\xi} = (\xi,\eta,\zeta)^{\mathsf{T}} = [-1,1]^3$ is given by

$$\begin{aligned} s &= \tfrac{\Delta s_n}{2}(1+\xi) + s_{n_s} \\ t &= \tfrac{\Delta t_n}{2}(1+\eta) + t_{n_s} \\ r &= \tfrac{\Delta r_n}{2}(1+\zeta) + r_{n_s} \end{aligned} \tag{23}$$

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

The sub vector of $\{\mathbf{F}\}_0^{V_0}$ related to collocation point n can be written as:

$$\mathbf{F}_{0n}^{V_0} = \sum_{n_s=1}^{N_s}\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1}\mathbf{U}(\mathbf{y}_n,\bar{\mathbf{x}}(\xi,\eta,\zeta))\,\dot{\mathbf{b}}_0(\bar{\mathbf{x}}(\xi,\eta,\zeta))\,J(\mathbf{s})J_\xi^{n_s}\;\mathrm{d}\xi\,\mathrm{d}\eta\,\mathrm{d}\zeta \tag{24}$$

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

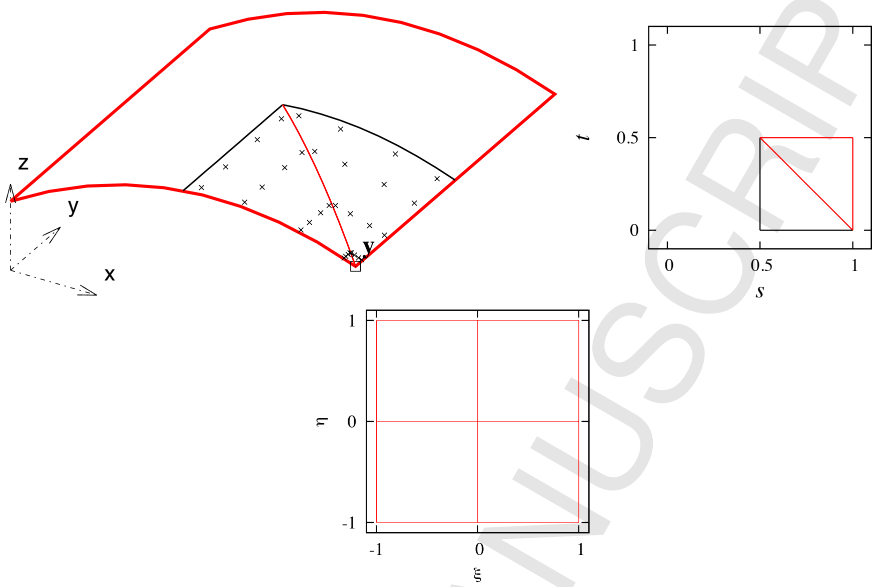

Figure 4: Subdivision into integration regions when the collocation point **y** is part of S_0 in global x,y,z, local s,t and ξ,η coordinates. The black lines indicate the limits of the integration region, the red thin lines the subdivision into a triangular subregion. Gauss points are depicted by crosses.

Applying Gauss integration we have:

$$\mathbf{F}_{0n}^{V_0} \approx \sum_{n_s=1}^{N_s} \sum_{m=1}^{M} \sum_{l=1}^{L} \sum_{k=1}^{K} \mathbf{U}\left(\mathbf{y}_n, \bar{\mathbf{x}}(\xi_m, \eta_l, \zeta_k)\right) \dot{\mathbf{b}}_0\left(\bar{\mathbf{x}}(\xi_m, \eta_l, \zeta_k)\right) J(\mathbf{s})\, J_{\xi}^{n_s}\, W_m W_l W_k \tag{25}$$

where N_s is the number of integration regions and M, L and K are the number of integration points in ξ, η and ζ directions respectively. To determine the number of Gauss points necessary for an accurate integration we consider that, whereas there is usually a moderate variation of body force, the Kernel $\mathbf{U}$ is $\mathcal{O}(r^{-1})$ so the number of integration points has to be increased if $\mathbf{y}_n$ is close to V_0.

If the integration region includes the collocation point $\mathbf{y}_n$, then the integrand tends to infinity as the point is approached. To deal with the integration involving the weakly singular Kernel we perform the integration in a local coordinate system, where the Jacobian tends to zero as the singularity point is approached. For this we divide the integration region into tetrahedral sub-regions. The transformation from the local $\boldsymbol{\xi}$ coordinate system, in which the Gauss coordinates are defined, to global coordinates involves the following transformation steps:

1. from $\boldsymbol{\xi}$ to a local system $(\sigma, \tau, \rho)^{\mathsf{T}} = [0,1]^3$
2. from (σ, τ, ρ) to $\mathbf{s}$
3. from $\mathbf{s}$ to $\mathbf{x}$

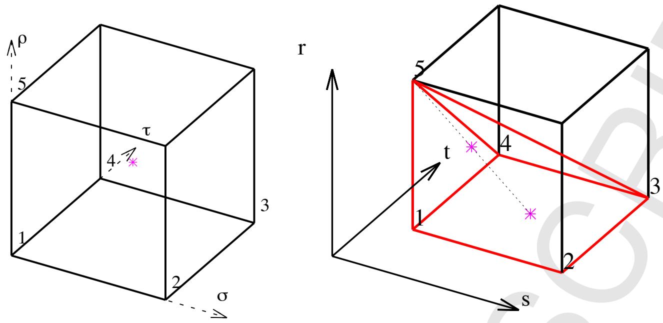

Figure 5: Singular volume integration, showing a tetrahedral subregion and the mapping from the local σ, τ, ρ system to the $\mathbf{s}$ coordinate system. A point with the local coordinates $\sigma = \tau = \rho = 0.5$ (i.e. $\xi = \eta = \zeta = 0$) is shown.

Steps 1 and 3 have already been discussed, so we concentrate on explaining the second step. Referring to Figure 5 we assume that the singular point is an edge point of the integration region.

For this case the transformation is as follows: First we determine the local coordinates $\mathbf{s}_1$ to $\mathbf{s}_5$ of the edge points of the tetrahedron, with 5 being the singularity point. Next we define a linear NURBS surface with points 1 to 4 and map the coordinates of the point (σ, τ) onto this surface:

$$\mathbf{s}_0(\sigma, \tau) = \sum_{i=1}^{4} R_i(\sigma, \tau)\, \mathbf{s}_i \tag{26}$$

where $R_i(\sigma, \tau)$ are linear basis functions. The final map is obtained by interpolation in the ρ direction:

$$\mathbf{s}(\sigma, \tau, \rho) = (1 - \rho)\, \mathbf{s}_0(\sigma, \tau) + \rho\, \mathbf{s}_5 \tag{27}$$

The Jacobian matrix of this transformation is given by:

$$\mathbf{J} = \begin{pmatrix} (1-\rho)\frac{\partial \mathbf{s}_0}{\partial \sigma} \\ (1-\rho)\frac{\partial \mathbf{s}_0}{\partial \tau} \\ \mathbf{s}_5 - \mathbf{s}_0 \end{pmatrix} \tag{28}$$

The Jacobian of this transformation tends to zero as the singular point is approached.

4.3. Implementation

$\{\mathbf{F}\}_0$ has to be recomputed at each iteration and this involves considerable computational work. It is therefore convenient to pre-compute matrices that multiply with $\boldsymbol{\sigma}_0$ and $\mathbf{b}_0$. At each iteration step the computation only involves a matrix multiplication:

$$\{\mathbf{F}\}_0 = \mathbf{B}\{\sigma_0\} + \mathbf{B}^V\{\mathbf{b}_0\} \tag{29}$$

where $\mathbf{B}$ and $\mathbf{B}^V$ are pre-computed matrices and $\{\sigma_0\}$ and $\{\mathbf{b}_0\}$ are vectors containing a list of values of initial stress and body force at points inside the inclusion.

4.4. Computation of $\dot{\mathbf{t}}_0$ *and* $\dot{\mathbf{b}}_0$

For the evaluation of the integrals (22) and (25), the values of $\dot{\mathbf{t}}_0$ and $\dot{\mathbf{b}}_0$ must be computed at the Gauss points and their determination involves stress or strain evaluations inside the inclusions. Since the location of quadrature points vary depending on the location of the collocation point $\mathbf{y}_n$, this would involve a very large number of NURBS evaluations for each iteration. In addition some computations would involve singular integration.

It is therefore convenient to compute the required values at regular grid points inside the inclusion and then interpolate or extrapolate the values to the locations of the Gauss points. An advantage of this scheme is that required differentiations can be carried out numerically using finite differences. A regular grid of points is therefore established in the the local ($\mathbf{s}$) coordinate system of the inclusion.

Remark 3: It should be noted that theoretically it is possible to compute the strain and subsequently the initial stress and body force (using a derived fundamental solution) exactly at the Gauss points. However, this would involve a considerable computational overhead and the authors feel that for practical problems this is not really necessary. So, in order to reduce the computational effort the mentioned approximations (using a regular grid with interpolation and numerically computing the derivatives) have been introduced. The test examples show that convergence to accurate results is achieved as the number of internal points is increased.

After computing the initial stress increment $\dot{\boldsymbol{\sigma}}_0$ using the procedures outlined in sections 2.1 and 2.2, the initial traction increments $\dot{\mathbf{t}}_0$ are computed from the initial stresses by

$$\dot{\mathbf{t}}_0 = \begin{pmatrix} \dot{\sigma}_{0x} & \dot{\tau}_{0xy} & \dot{\tau}_{0xy} \\ \dot{\tau}_{0xy} & \dot{\sigma}_{0y} & \dot{\tau}_{0yz} \\ \dot{\tau}_{0zx} & \dot{\tau}_{0zy} & \dot{\sigma}_{0z} \end{pmatrix} \mathbf{n} \tag{30}$$

where $\mathbf{n}$ is the unit outward normal vector to the surface S_0. The body force increment $\dot{\mathbf{b}}_0$ can be computed by

$$\dot{\mathbf{b}}_0 = -\begin{pmatrix} \frac{\partial \dot{\sigma}_{0x}}{\partial x} + \frac{\partial \dot{\tau}_{0xy}}{\partial y} + \frac{\partial \dot{\tau}_{0xz}}{\partial z} \\ \\ \frac{\partial \dot{\tau}_{0yx}}{\partial x} + \frac{\partial \dot{\sigma}_{0y}}{\partial y} + \frac{\partial \dot{\tau}_{0yz}}{\partial z} \\ \\ \frac{\partial \dot{\tau}_{0zx}}{\partial x} + \frac{\partial \dot{\tau}_{0zy}}{\partial y} + \frac{\partial \dot{\sigma}_{0z}}{\partial z} \end{pmatrix} \tag{31}$$

It is convenient to compute the derivatives with respect to local coordinates $\mathbf{s}$ first and then transform them to global coordinates. For example the global derivatives of σ_x in terms of local derivatives are given by the transformation

$$\boldsymbol{\sigma}_{x,\mathbf{x}} = \mathbf{J}^{-1}\boldsymbol{\sigma}_{x,\mathbf{s}} \tag{32}$$

where

$$\boldsymbol{\sigma}_{x,\mathbf{x}} = \begin{pmatrix} \frac{\partial\sigma_x}{\partial x} \\ \\ \frac{\partial\sigma_x}{\partial y} \\ \\ \frac{\partial\sigma_x}{\partial z} \end{pmatrix} \quad \text{and} \quad \boldsymbol{\sigma}_{x,\mathbf{s}} = \begin{pmatrix} \frac{\partial\sigma_x}{\partial s} \\ \\ \frac{\partial\sigma_x}{\partial t} \\ \\ \frac{\partial\sigma_x}{\partial r} \end{pmatrix} \tag{33}$$

and $\mathbf{J}$ is the Jacobian matrix (17).

The derivatives are numerically computed using finite differences. For grid points inside the inclusion that have other points left and right (or top and bottom) of them we use a central finite difference, whereas for points that only have one point on a side we use forward or backward finite differences.

Remark 4: In the following examples we apply a simple linear interpolation between grid points and a linear extrapolation from grid points to the boundary S_0. Obviously more sophisticated schemes may be applied. However care has to be taken for the case when the variation of the initial stress is discontinuous. Also it has to be considered that in our approach the internal points are used to numerically compute the derivatives of the initial stresses. The simple scheme applied here leads to an increase in the accuracy for determination of the derivatives and for the evaluation of the associated integrals as the number of grid points is increased. The convergence of the solution as a function of the number of internal points is investigated in one of the numerical examples below.

5. Computation of results inside the inclusion

The solution algorithm requires the evaluation of strains and stresses at internal points. The displacements at a point $\mathbf{y}_i$ inside the inclusion is given by:

$$\begin{aligned}\mathbf{u}(\mathbf{y}_i) = &\int_S \mathbf{U}(\mathbf{y}_i,\mathbf{x})\,\mathbf{t}(\mathbf{x})\,\mathrm{d}S(\mathbf{x}) - \int_S \mathbf{T}(\mathbf{y}_i,\mathbf{x})\,\mathbf{u}(\mathbf{x})\,\mathrm{d}S(\mathbf{x}) \\ &+ \int_{S_0} \mathbf{U}(\mathbf{y}_i,\bar{\mathbf{x}})\,\dot{\mathbf{t}}_0(\bar{\mathbf{x}})\,\mathrm{d}S_0(\bar{\mathbf{x}}) + \int_{V_0} \mathbf{U}(\mathbf{y}_i,\bar{\mathbf{x}})\,\dot{\mathbf{b}}_0(\bar{\mathbf{x}})\,\mathrm{d}V_0(\bar{\mathbf{x}})\end{aligned} \tag{34}$$

The strain can be computed by

$$\begin{aligned}\boldsymbol{\epsilon}(\mathbf{y}_i) = &\int_S \mathbf{S}(\mathbf{y}_i,\mathbf{x})\,\mathbf{t}(\mathbf{x})\,\mathrm{d}S(\mathbf{x}) - \int_S \mathbf{R}(\mathbf{y}_i,\mathbf{x})\,\mathbf{u}(\mathbf{x})\,\mathrm{d}S(\mathbf{x}) \\ &+ \int_{S_0} \mathbf{S}(\mathbf{y}_i,\bar{\mathbf{x}})\,\dot{\mathbf{t}}_0(\bar{\mathbf{x}})\,\mathrm{d}S_0(\bar{\mathbf{x}}) + \int_{V_0} \mathbf{S}(\mathbf{y}_i,\bar{\mathbf{x}})\,\dot{\mathbf{b}}_0(\bar{\mathbf{x}})\,\mathrm{d}V_0(\bar{\mathbf{x}})\end{aligned} \tag{35}$$

For the first iteration we set $\dot{\mathbf{b}}_0 = \dot{\mathbf{t}}_0 = 0$. The evaluation of the arising surface and domain integrals follows the methodology described above.

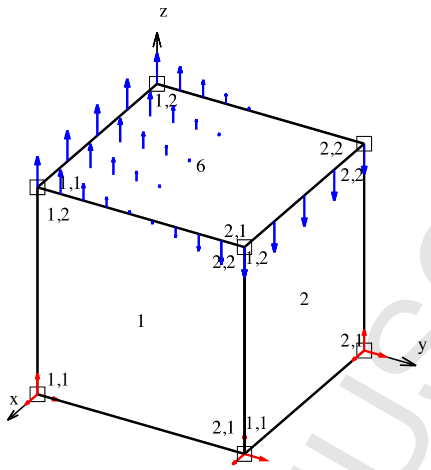

Figure 6: Test example 1: Geometry definition of cube with 6 NURBS patches showing control points in local numbering as hollow squares, loading in blue and boundary conditions as red arrows in restrained directions.

5.1. *Implementation*

Since the strain has to be evaluated at each iteration step it is convenient to pre-compute matrices that multiply with $\boldsymbol{\sigma}_0$ and $\mathbf{b}_0$. If this is done, the computation only involves a matrix multiplication:

$$\epsilon = \{\mathbf{C}\} + [\mathbf{A}]\{\mathbf{u}\} + [\mathbf{D}]\{\boldsymbol{\sigma}_0\} + [\mathbf{D}]^V\{\mathbf{b}_0\} \tag{36}$$

In the above $\{\mathbf{C}\}$ is a vector related to known values, $[\mathbf{A}]$ is a matrix related to unknown values $\{\mathbf{u}\}$ and $[\mathbf{D}]$, $[\mathbf{D}]^V$ are matrices related to the initial stresses and body forces.

6. Test examples

In the following sections the theory is tested on simple examples, where the solution is known or can be obtained using a different numerical model.

6.1. *Test example 1: Cube in bending with elastic inclusion*

The first example tests the algorithm for the case of a single elastic inclusion. It consists of a unit cube $[0,1]^3$, composed of two different materials. Its geometry is defined by six NURBS surfaces (patches) with basis functions of order 1. The cube is loaded with a moment and fixed at the bottom as shown in Figure 6.

The inclusion is defined by two linear NURBS surfaces and assigned a Young's modulus E of half the one used for computing the fundamental solution and no change in the Poisson's ratio ν.

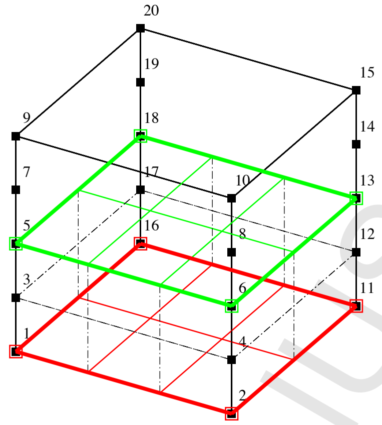

Figure 7: Test example 1: Definition of inclusion with 2 linear NURBS surfaces shown in green and red with the associated control points marked with hollow squares. Collocation points are numbered and shown as black filled squares.

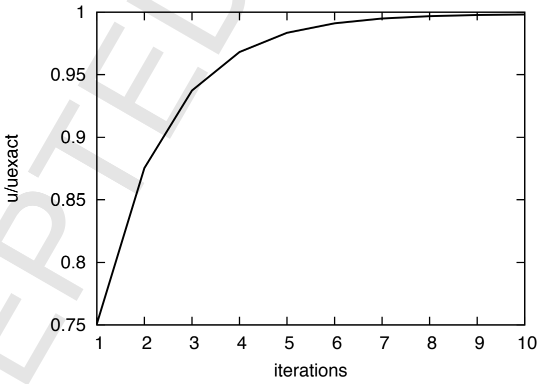

Figure 8: Test example 1: Plot of ratio of maximum computed displacement to the exact one as function of the number of iterations.

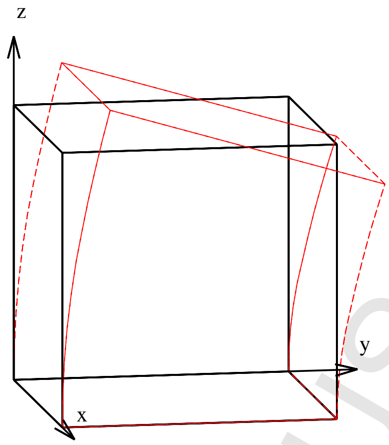

Figure 9: Test example 1: Deformed shape.

For the analysis, the concept of a geometry independent field approximation was used and the basis functions for approximating the displacements were defined using the following knot vectors in the local s,t directions:

- Patch 1: $\Xi_s = 0,0,1,1; \Xi_t = 0,0,0,0.5,0.5,1,1,1$
- Patch 2: $\Xi_s = 0,0,1,1; \Xi_t = 0,0,0,0.5,0.5,1,1,1$
- Patch 3: $\Xi_s = 0,0,1,1; \Xi_t = 0,0,0,0.5,0.5,1,1,1$
- Patch 4: $\Xi_s = 0,0,1,1; \Xi_t = 0,0,0,0.5,0.5,1,1,1$
- Patch 5: $\Xi_s = 0,0,1,1; \Xi_t = 0,0,1,1$
- Patch 6: $\Xi_s = 0,0,1,1; \Xi_t = 0,0,1,1$

with all weights equal to one. This approximation results in a quadratic variation of the displacements in the vertical direction with a C^0 continuity at the interface between materials. The resulting location of the collocation points are shown in Figure 7 and this will give the exact solution for the applied loading. The results are shown in Figure 8 and Figure 9. It can be seen that convergence to the exact solution is achieved after about 10 iterations.

6.2. *Test example 2: Cantilever beam with an inelastic inclusion*

The second example tests the capability of the method to simulate inelastic material behavior. It is a cantilever beam of dimension $1 \times 1 \times 5$ meters with an equally distributed load of 0.049 kN/m^2 at the end. The elastic modulus was assumed to be 10 kPa and the Poisson's ratio was assumed to be zero. A *von Mises* material law was used with a yield stress of 1 kPa . The geometry is defined by 8 control points

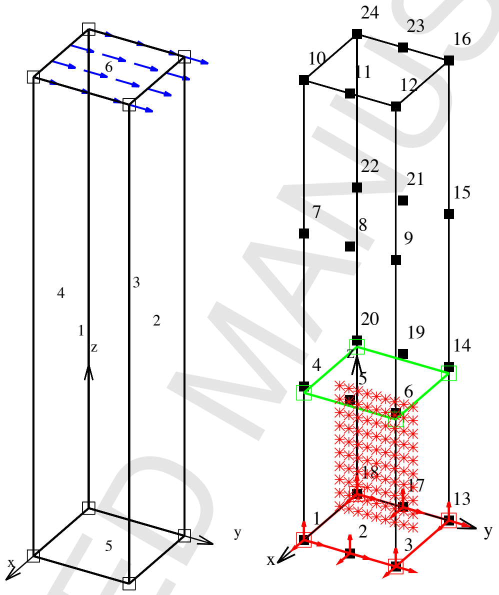

Figure 10: Test example 2: Left: Definition of the geometry with 6 linear NURBS patches (associated control points are shown as hollow squares). Also shown is the loading. Right: Location of collocation points (restrained points are marked with red arrows). Also shown is the definition of the inelastic inclusion with the internal points.

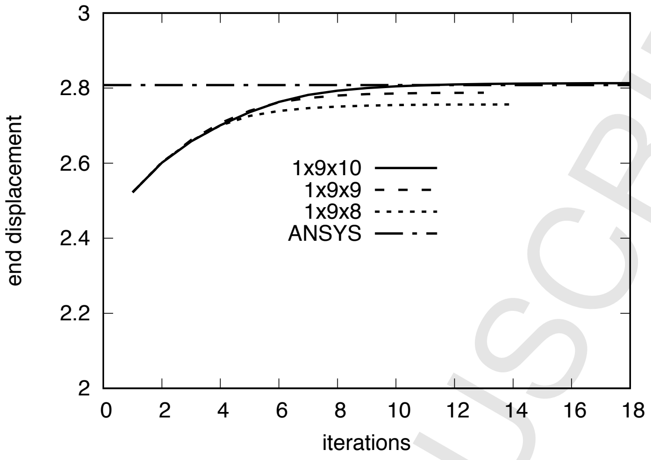

Figure 11: Test example 2: Convergence of the maximum displacement as a function of the number of iterations for three configurations of internal points. The results are compared with an ANSYS Finite Element simulation.

and 6 patches (Figure 10 left) . Figure 10 right shows the collocation points obtained after order elevating two times (from linear to cubic) in the z-direction and one time in the y-direction. The simulation has $24 \times 3 = 72$ unknowns. The elastic maximum displacement for this refinement was 2.5223 *mm* which compares well with the exact solution of 2.5235 and an elastic Finite Element simulation. Also shown is the definition of the inelastic inclusion, depicting the location of internal points for one of the test runs. For this case only one point in x-direction was necessary as the variation of stress is constant in this direction. For the convergence study various combinations of number of points in each directions were tried. The convergence of the displacement with three different internal point configurations is shown in Figure 11 and compared with an non-linear Finite Element simulation with 18 000 unknowns using the commercial code ANSYS. It can be seen that the run with $1 \times 9 \times 10$ internal points (i.e. one in x-direction 9 in y-direction and 10 in z-direction) gives the best agreement with the ANSYS results.

7. Optimization

The calculation of the strains at internal points, required for computing the initial stress and body force, is very compute intensive as it involves many NURBS evaluations at Gauss points. In order to reduce the number of internal point evaluations it is proposed that the strains are only evaluated at a small number of points of the inclusion (interpolation points) and the values at internal points determined by interpolation, which is carried out in local coordinate system of the inclusion. The strains at

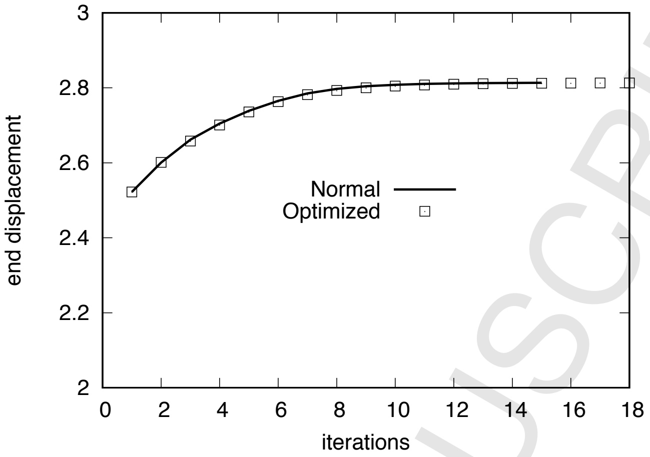

Figure 12: Comparison of normal and optimised method for computing values inside the inclusion.

an inclusion point with the local coordinate s,t,r are given by:

$$\epsilon(s,t,r) = \sum_{i=1}^{I} N_i(s,t,r)\epsilon_i \tag{37}$$

where $N_i(s,t,r)$ are suitable interpolation functions and ϵ_i are the values of strain computed at the interpolation points. To test this we compare the most accurate solution obtained with the previous method with the proposed optimized method. A linear interpolation was used. It can be seen that the approximation due to the interpolation does not alter the result in this case. However, the computation time for the optimised version is about 10% of the normal one and is not significantly increased with the number of internal points.

8. Conclusions and Outlook

The aim of this and a previous paper ([13]), dealing with plane problems, was to remove a main impediment, that has prevented the widespread use of the BEM for heterogeneous and non-linear problems. In the past the application of the BEM to these types of problems has required additional discretisation effort, including a volume discretisation for the evaluation of the arising volume integrals. The emergence of the isogeometric BEM, with the promise of avoiding mesh generation completely, has lead to the idea of applying this technology also to the treatment of volume effects. The work presented here opens the way for simulations without mesh generation, by taking geometrical data directly from CAD. As has been demonstrated in the examples no mesh is generated. Instead the geometry of the problem, as well as that of the inclusions is defined with very few parameters, using NURBS patches. These data can be taken directly from a CAD program. Another feature of the method described here, is that accurate solutions with very few degrees of freedom can be obtained. For the example of the non-linear cantilever beam only 72 degrees of freedom were sufficient to get the same result as with 18 000 degrees of freedom with the FEM.

The main conclusions are

- Accurate results can be obtained with very few degrees of freedom
- Geometry data can be taken directly from CAD programs
- The geometry independent field approximation allows flexible refinement options

The method is currently restricted to cases where the zones of in-elastic behaviour can be specified (as inclusions). However, it is possible to define zones of plasticity automatically and this is a topic of further research.

The area of application that would most benefit from this new approach would be geomechanics since this involves an infinite domain. Domain methods suffer from the fact that meshes need to be truncated. A user friendly BEM method, that can deal with heterogeneous ground conditions and material non-linear behaviour, would be ideal for such applications. In addition, using this approach the information about geological inclusions could be taken directly from the CAD program.

We believe that the most promising way to achieve the "holy grail" of a simulation without mesh generation is via the BEM. This paper presents a further step towards this goal. Open issues are a better and more general mapping method for inclusions, an automatic detection of plastic zones and - most importantly - an optimization of the computational procedures, in order to reduce the significant time spent for the numerical integration.

At the moment the method is restricted to solid mechanics but there may be applications in other fields such as fluid dynamics, where zones of non-linear behavior are treated as volume effects. It is hoped that this paper gives impetus to much needed research in this exiting new area of simulation.

9. Acknowledgements

The work was partially supported by the Austrian science fund FWF, under Grant Number P24974-N30: Fast isogeometric boundary element method. The first author was a visiting professor at the University of Ferrara and the Second University of Naples. The financial support of these institutions is gratefully acknowledged.

References

[1] B. Marussig, Seamless integration of design and analysis through boundary integral equations, Ph.D. thesis, Graz University of Technology (2015).

[2] J. A. Cottrell, T. J. R. Hughes, Y. Bazilevs, Isogeometric Analysis: Toward Integration of CAD and FEA, John Wiley & Sons, Chichester, England, 2009.

[3] R. N. Simpson, S. P. A. Bordas, J. Trevelyan, T. Rabczuk, A two-dimensional isogeometric boundary element method for elastostatic analysis, Computer Methods in Applied Mechanics and Engineering 209–212 (0) (2012) 87–100.

[4] M. A. Scott, R. N. Simpson, J. A. Evans, S. Lipton, S. P. A. Bordas, T. J. R. Hughes, T. W. Sederberg, Isogeometric boundary element analysis using unstructured T-splines, Computer Methods in Applied Mechanics and Engineering 254 (0) (2013) 197 – 221.

[5] B. Marussig, G. Beer, C. Duenser, Isogeometric boundary element method for the simulation in tunneling, Applied Mechanics and Materials 553 (2014) 495–500.

[6] G. Beer, B. Marussig, J. Zechner, C. Duenser, T.-P. Fries, Boundary element analysis with trimmed NURBS and a generalized IGA approach, in: E. Oñate, J. Oliver, A. Huerta (Eds.), 11th World Congress on Computational Mechanics (WCCM XI), 2014, pp. 2445–2456.

[7] B. Marussig, J. Zechner, G. Beer, T.-P. Fries, Fast isogeometric boundary element method based on independent field approximation, Computer Methods in Applied Mechanics and Engineering 284 (2015) 458–488, isogeometric Analysis Special Issue.

[8] G. Beer, B. Marussig, J. Zechner, A simple approach to the numerical simulation with trimmed CAD surfaces, Computer Methods in Applied Mechanics and Engineering 285 (2015) 776–790.

[9] G. Beer, Mapped infinite patches for the NURBS based boundary element analysis in geomechanics, Computers and Geotechnics 66 (2015) 66–74.

[10] V. Mallardo, E. Ruocco, A NURBS boundary-only approach in elasticity, European Journal of Computational Mechanics 25 (2016) 71–90.

[11] V. Mallardo, E. Ruocco, An improved isogeometric Boundary Element Method approach in two dimensional elastostatics, CMES - Computer Modeling in Engineering and Sciences 102 (2014) 373–391.

[12] B. Marussig, J. Zechner, G. Beer, T. P. Fries, Stable isogeometric analysis of trimmed geometries, Computer Methods in Applied Mechanics and Engineeringdoi:10.1016/j.cma.2016.07.040.

[13] G. Beer, B. Marussig, J. Zechner, C. Duenser, T.-P. Fries, Isogeometric boundary element analysis with elasto-plastic inclusions. part 1: plane problems, Computer Methods in Applied Mechanics and Engineering 308 (2016) 552–570.

[14] G. Beer, I. Smith, C. Duenser, The Boundary Element Method with Programming, Springer-Verlag, Wien, 2008.

[15] G. Beer, Advanced numerical simulation methods - From CAD Data directly to simulation results, CRC Press/Balkema, 2015.

[16] G. Beer, S. P. A. Bordas (Eds.), Isogeometric methods for numerical simulation, CISM lecture notes, Springer, 2014.

[17] J. C. Simo, T. J. R. Hughes, Computational Inelasticity, Springer, 1998.

[18] I. C. Cormeau, Numerical stability in quasi-static elasto-viscoplasticity, International Journal for Numerical Methods in Engineering 9 (1) (1975) 109–127.

Highlights

- Accurate results can be obtained with very few degrees of freedom
- Geometry data can be taken directly from CAD programs
- The geometry independent field approximation allows flexible refinement options