A Fast Boundary Element Method with Hierarchical Matrices for Elastostatics and Elasto-Plasticity

Zur Erlangung des akademischen Grades eines
Doktors der technischen Wissenschaften
ausgeführte Dissertation

eingereicht an der
Fakultät für Bauingenieurwissenschaften
der Technischen Universität Graz

von

Jürgen Zechner

Berichter: Prof. Dipl.Ing. Dr.tech. Gernot Beer
Prof. Attilio Frangi

Abstract

Boundary element methods offer advantages for the simulation of problems involving a large or infinite domain. Only the surface needs to be meshed and the radiation condition is fulfilled by the fundamental solutions. Unfortunately, the method leads to fully populated system matrices resulting in a quadratic numerical complexity in terms of storage and numerical effort. To overcome this drawback, fast summation methods have been developed, reducing the complexity to almost linear or even linear behavior. This is achieved mainly by splitting the domain into a near- and a far-field, where contributions of the latter are subject to numerical approximation. This thesis covers the application of the concept of Hierarchical Matrices to elastostatic and elasto-plastic problems with a collocation-based boundary element method. The application of the Adaptive Cross Approximation as well as kernel interpolation techniques are presented. The main objective is to efficiently solve inhomogeneous and non-linear problems. In this context, the focus is on the evaluation of internal results with body forces and the special treatment of discrete volume potentials. The successful implementation into a multi-purpose code and the application to various real-world problems with adequate accuracy of the results is demonstrated.

Kurzfassung

Einer der größten Vorteile der Randelementmethode ist unendliche Gebiete oder den Halbraum exakt beschreiben zu können. Da nur die Oberfläche des Problems modelliert werden muss wird im Vergleich zu anderen numerischen Methoden die Anzahl der benötigten Freiheitsgrade zwar verringert, die resultierenden Systemmatrizen sind jedoch unsymmetrisch und voll besetzt. Die daraus folgende quadratische Komplexität für Rechenspeicher und Rechenzeit macht die Methode für größere Aufgabenstellungen unattraktiv. Schnelle Summationsmethoden teilen das diskretisierte Gebiet üblicherweise in ein Nah- und Fernfeld. Dabei wird der Einfluss des Fernfelds auf die Lösung approximiert und somit der numerische Aufwand reduziert. Der Inhalt der vorliegenden Arbeit ist die Anwendung von Hierarchischen Matrizen mit annähernd linearer Komplexität auf die Randelementmethode für elastische und elasto-plastische Probleme. Zur Approximation der, auf der Kollokationsmethode basierenden Gleichungssysteme werden zwei Ansätze, nämlich die Konstruktion von Niedrigrangmatrizen mittels adaptiver Kreuzapproximation und mittels Kerninterpolation behandelt. Das Hauptaugenmerk richtet sich auf die Berechnung von Feldgrößen im Gebiet und auf die Approximation von diskreten Volumenpotentialen, mit dem Ziel schnell und effektiv inhomogene und nichtlineare Aufgabenstellungen mit der Randelementmethode behandeln zu können. Die beschriebene Methode wurde in ein Computerprogramm implementiert und anhand von praxisnahen Aufgabenstellungen verifiziert.

Contents

1 Introduction

The boundary element method (BEM) is well suited for solving problems involving an unbounded domain or problems with high stress concentration. This is because the radiation condition is exactly fulfilled so no mesh truncation is necessary. In general, the meshing procedure is greatly simplified because, for homogeneous problems, only the boundary of the domain has to be considered. For problems in continuum mechanics, the BEM is expedient if the considered body has inclusions or distributions of initial strains or stresses. Formulations have been developed for inhomogeneous problems with body forces induced by gravity, thermal effects and, like in this work, plasticity. However, today only few commercial software packages are available which utilize the BEM.

In contrary, the finite element method (FEM) is applied to a wide field of engineering applications and can deal with many, even anisotropic, inhomogeneuous and highly-nonlinear physical models. The method takes advantage of its sparse system matrices and is easy to parallelize. Thus, the solving of problems with many degrees of freedom is possible.

There has been a progressive development of computer hardware over the last decades. In this context, one of the problems with the BEM is its computational efficiency. For large scale problems, i.e. problems which are about to be solved with the highest possible number of degrees of freedom, it is a requirement for the method to show only linear or almost linear increase of computational effort. To achieve this, fast solution algorithms, which can reduce complexity to the optimal linear behaviour, have been developed. Their application to different BEM formulations has been in focus of many researchers in recent years.

The aim of the thesis is to address problems in elastostatics and for elasto-plasticity by introducing hierarchical matrices ($\mathcal{H}$-matrices) and their adaption to software solutions in this context.

1.1 Motivation and Objective

In this work, the choice for a pure boundary element method formulation is motivated by problems arising in soil and rock engineering which involve infinite and semi-infinite domains like tunneling as shown in Figure 1.1 or other actions in underground construction [62]. Today in contrast to the FEM there are no commercial BEM programs available that can be used to simulate underground excavations efficiently and accurately. For the last decade the computer software *BEFE++* has been developed[1], with the aim of closing

[1] Information on the software package can be found at *www.ifb.tugraz.at*

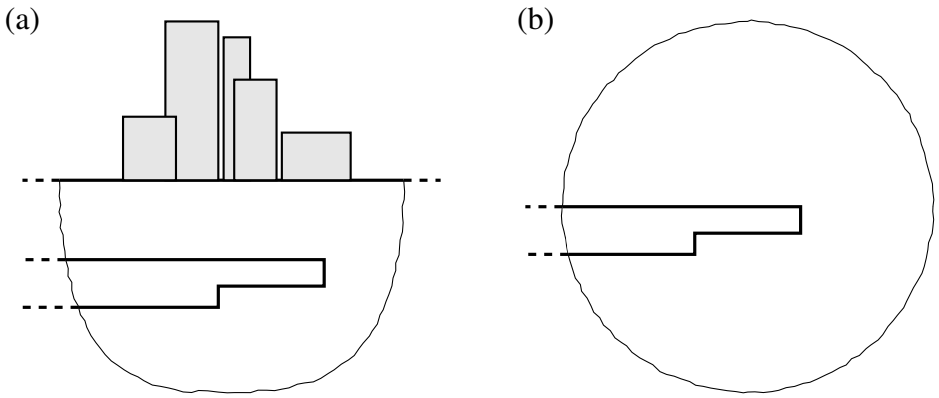

Figure 1.1: Semi-infinite (a) and infinite (b) domain for problems in underground construction

this gap. With the enhancements described in [13] nonlinear material behavior, support systems like rock bolts and steel arches as well as small inclusions in the rock material are simulated by the BEM. Since the method can easily deal with evolving boundaries, its application to sequential excavation problems is was within reach. Deep tunnels, for instance, are modeled as an exterior problem and the surrounding rock-material is only locally subject to elasto-plastic behavior. Instead of modeling the whole domain, the BEM just needs a discrete model of the tunnel boundary and possibly earth's surface as a semi-infinite domain. Although the main focus is on underground construction, the application of the considered approach is not restrictive and can be applied to any small strain elasto-static or elasto-plastic problem.

The main objective of the thesis at hand has been the development of a fast simulation method for problems in elastostatics and elasto-plasticity to overcome the described drawbacks. This is achieved by using the technique of $\mathcal{H}$-matrices applied to an elasto-plastic BEM solution procedure. By doing this, near optimal complexity in terms of storage and computational effort can be reached. In the context of $\mathcal{H}$-matrices, two approaches are followed for the approximation of system matrices: First, the adaptive cross approximation (ACA) algorithm, usually applied to scalar problems, is adapted to work with the emerging vector valued problems. Second, a matrix approximation technique for elasto-static and elasto-plastic boundary element methods by means of kernel interpolation in the context of $\mathcal{H}$-matrices is developed.

Furthermore, fast solution procedures are established, allowing nonlinear system of equations to be solved rapidly and with desired accuracy. This is achieved by using fast matrix-matrix or matrix-vector operations provided by $\mathcal{H}$-matrices algebra. The solution procedure is carried out iteratively and the matrix format is taken to calculate approximate inverse matrices for the preconditioning in reasonable time.

Finally, the algorithms are implemented into a computational framework. Particular attention is paid to the effective use of multiple core computer environments and to a

realization in such a way that it also allows the solution of other physical models.

1.2 State of the Art

Two major fields of research are touched on by this work. In particular, the BEM and its application to elastostatic and elasto-plastic problems, as well as fast summation methods and their adaptation. The following overview does not claim to be complete. For a detailed background on historical developments or approaches beyond the scope of this work, the reader is referred to the overview papers [28] and [110] as well as to the textbooks of Bonnet [17], Gaul et al. [41], Sauter and Schwab [95] and Steinbach [101]. The textbook of Chen and Zhou [27] covers nonlinear applications to the BEM and provides a deeper mathematical inside.

Elasticity and Plasticity The first BEM for elastostatic problems has been developed by Rizzo [88], for elasto-plastic ones by Swedlow and Cruse [103] and there have been several to follow enhancing this technique. One of the most important developments, not only for engineers, has been the adoption of isoparametric elements developed by Lachat and Watson [65]. In [105] a complete and correct formulation for three-dimensional elasto-plastic problems has been published.

Telles and Carrer [106] were among the first to successfully implement a BEM formulation in an implicit manner and presented a solution for transient problems [26] too. Bonnet and Mukherjee [19] proposed an initial strain formulation in which the consistent tangential operator [97], delivered by many material laws used in FEM, can be used to obtain convergence of quadratic order for the iterative solution procedure. Gao and Davies [39] achieved the same by using a plastic multiplier. The latter approach reduces the unknowns for the iterative solution technique but prohibits the application of arbitrary material laws as a black box. A good overview of the method in elastostatic and plastic applications can be found in [18]. The book of Beer et al. [12] describes many principles used in this particular work.

The collocation method, in which the boundary integral equation (BIE) is solved at carefully chosen points, has been used in the papers mentioned above. Generally, system matrices are unsymmetrical in this case. The symmetric Galerkin boundary element method (SGBEM) overcomes that by using weighted residuals as well as a hypersingular BIE. The resulting system of equation is block skew symmetric and can be solved by iterative solvers in a robust way. Maier and Polizzotto [67] applied this to static; Frangi and Maier [36] to transient elasto-plastic problems.

A major drawback of treating elasto-plastic problems by the BEM is the need for a domain discretization in order to deal with the inhomogeneous and nonlinear part of the domain. Although the domain triangulation is only used to carry out integrals and therefore does not introduce additional degrees of freedom, the computational effort for the originating system matrices is remarkable, hence allowing only small regions of plasticity.

Moreover, the boundary only nature of the method is lost. Much effort has been given in recent years to avoid the usage of such domain cells for inhomogeneous or nonlinear problems. An overview of methods for volume integration is given in [58]. One possible approach for the calculation of such Newton or volume potentials is to use particular solutions derived by means of the FEM [99] or by globally defined interpolation functions as by the dual reciprocity method (DRM) initially proposed by Nardini and Brebbia [75]. The latter has been extended to elasto-plasticity in [3]. The radial integration method (RIM) for the elasto-plastic BEM, presented by Gao [37], uses radial basis functions to transfer the domain integral into radial and boundary integrals. However, a comparison of the DRM to the direct approach found in [59] recommends the direct method for accuracy. For infinite and semi-infinite domains the RIM is not directly applicable since it requires a finite domain. Because of the sensitivity to the location of the interpolation points the use of radial basis functions could lead to unstable results and to an ill-conditioned system of equations. That's why the direct, cell based approach has been taken in that work.

Fast Methods For large scale computing, the BEM suffers from fully populated system matrices, which result in a quadratic complexity in terms of storage and computational effort. To overcome this, fast summation methods for non local problems been developed and applied successfully to the BEM. The most important ones are the fast multipole method (FMM) [90] and $\mathcal{H}$-matrices [50]. Different and in contrast to those agglomerating methods are the wavelet method [16] and fast Fourier transformation (FFT) based approaches [85]. For the latter methods, there are hardly any reports of applications to elasticity or even elasto-plasticity. The FMM has been taken to solve elastic problems for instance by Of et al. [78], for plastic problems by Wang and Yao [109] in two dimensions only.

As previously mentioned, volume potentials are needed for the elasto-plastic analysis by the BEM. The first publication on fast evaluation of Newton potentials goes back to a paper by Ewald [35] in 1921. For elasticity and SGBEM Of et al. [79] proposed an approach to the FMM. Fast, cell based approaches by means of FFT can be found for instance in [31] and [60], however, for Poisson's equation only.

Hierarchical Matrices The technique of $\mathcal{H}$-matrices for multidimensional problems originates from the panel cluster method developed by Hackbusch and Nowak [55] which has also been applied to elastic problems [57]. Initially, truncated Taylor expansions [51] were used to approximate system matrices until the interpolation of the kernel function was proposed in [53] and rigorously analyzed in [47]. Unfortunately, and up to a certain level, these schemes depend on the fundamental solution of the underlying problem. That might be the reason why there are no reports of the application to elastic or plastic problems in the literature. Bebendorf [6] introduced the ACA technique to find low rank representations in the context of $\mathcal{H}$-matrices, first for the Nyström method and then for collocation BEM [10]. The algorithm uses only a few entries of the original matrix. Thus, it is a black box method

and obviously easy to implement in existing codes. To combine the advantages of kernel interpolation and ACA, Börm and Grasedyck [23] proposed the hybrid cross approximation (HCA) algorithm. For elastostatics ACA has been applied together with the SGBEM in [9], for elastodynamics in [71] and [72]. In [4] nonlinear elliptic problems are treated efficiently by $\mathcal{H}$-matrices. Benedetti et al. [14] applied the approximation technique to solve crack problems. $\mathcal{H}$-matrices have been used for the evaluation of stresses at internal points in terms of a post-processing procedure in [66]. Both papers do not consider body forces or a nonlinear algorithm and it is not perfectly clear how the scalar based ACA is applied to vector valued problems. The computational complexity achieved by $\mathcal{H}$-matrices is almost linear up to a logarithmic factor. To get the optimal linear complexity, hierarchical matrices with linear complexity ($\mathcal{H}^2$-matrices) have been presented in [53] and [54] as an algebraic counterpart to FMM and applied to boundary integral operators in [25]. In [5] an ACA based approach to construct matrix approximations with linear complexity was introduced.

1.3 Introductory Example

This section introduces the concepts touched on in this work explained in a simple one-dimensional example. For that purpose, the boundary value problem of an elastic bar with axial loading is taken and solved by applying the concepts of boundary element methods. A short introduction to one dimension elasto-plasticity is given and finally, the concept of matrix approximation in that context is described.

Boundary Value Problem Consider a column $\Omega = \{x|0 < x < \ell\}$, fixed at $x = 0$ and subject to axial loading F at $x = \ell$ and $n(x)$ acting along the column as shown in

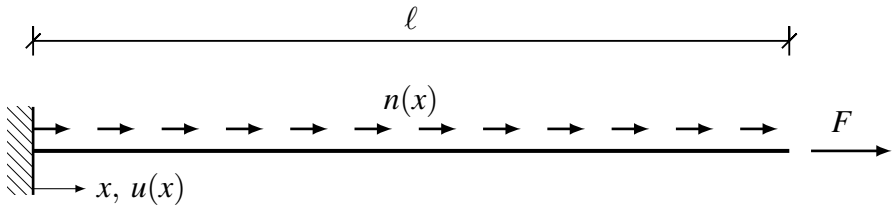

Figure 1.2: Elastic column under axial loading

Figure 1.2. The displacement $u(x)$ is the primal variable. The problem is described by the inhomogeneous, ordinary second order differential equation and the given boundary conditions

$$\begin{aligned} -E(x)A(x)u''(x) &= n(x) \quad \forall x \in \Omega = \{0,\ldots,\ell\} \\ u(0) &= 0 \\ N(\ell) &= F \end{aligned} \tag{1.1}$$

with Young's modulus $E(x)$ and the cross-section $A(x)$. Keeping $EA = E(x)A(x)$ constant and by taking a linear load $n(x) = n_0\,(1 - x/l)$ the analytic solution for the displacement $u(x)$ and the strain $\varepsilon(x) = u'(x)$ reads

$$\begin{aligned} u(x) &= \frac{F}{EA}x + \frac{n_0}{2EA}\frac{x}{\ell}\left(\frac{x^2}{3} - x\ell + \ell^2\right) \\ \varepsilon(x) &= \frac{F}{EA} + \frac{n_0}{2EA}\frac{1}{\ell}\left(x^2 - 2x\ell + \ell^2\right) \end{aligned} \tag{1.2}$$

The axial force $N(x) = EAu'(x)$ and the axial stress $\sigma(x) = Eu'(x)$ are then directly calculated.

Now, the problem is solved by using the concepts of the BEM following the explanations of Antes [2], Hartmann [56] and Gaul et al. [41]. As a fact, the boundary of this problem reduces to only two points. The *fundamental solutions* for the displacement and the axial force of the homogeneous part of (1.1) for a unit load at a point y are given by

$$u^*(x,y) = \frac{1}{2EA}|x-y| \quad \text{and} \quad N^*(x,y) = EA\frac{\partial}{\partial x}u^*(x,y) = \frac{1}{2}\operatorname{sgn}(x-y)\,.$$

Generally, (1.1) can be solved by multiplying a test function v and integration by parts over the problem domain. In the BEM the test function v is the corresponding fundamental solution. This leads to the representation of the solution

$$u(y) = \left[N(x)u^*(x,y) - u(x)\frac{N^*(x,y)}{EA}\right]_{x=0}^{\ell} + \int_0^{\ell} n(x)u^*(x,y)\,\mathrm{d}x \tag{1.3}$$

at any point $y \in \Omega$. By inserting the limits for x, shifting y to the boundary points and rearranging with respect to the boundary conditions (1.3) gives

$$\begin{pmatrix} 0 & -1 \\ l & 1 \end{pmatrix}\begin{pmatrix} \frac{N(0)}{EA} \\ u(\ell) \end{pmatrix} = -\begin{pmatrix} \frac{F\ell}{EA} \\ 0 \end{pmatrix} - \frac{1}{EA}\int_0^{\ell}\begin{pmatrix} x \\ (\ell - x) \end{pmatrix} n(x)\,\mathrm{d}x. \tag{1.4}$$

The domain integral on the right hand side represents the influence of the body forces $n(x)$. By solving equation (1.4) all boundary values are known and can be inserted in (1.3) to get the displacements $u(y)$ at any point in the domain Ω. The displacement field yields the exact solution given in (1.2). The strain $\varepsilon(y)$ is calculated by taking the derivation of (1.3) with respect to the spatial coordinate

$$\varepsilon(y) = \left[N(x)\varepsilon^*(x,y)\right]_{x=0}^{\ell} + \int_0^{\ell} n(x)\varepsilon^*(x,y)\,\mathrm{d}x. \tag{1.5}$$

Calculating stresses is straight forward by applying the material law $\sigma(x) = E\varepsilon(x)$ to (1.5). The derived strain fundamental solution can be written as

$$\varepsilon^*(x,y) = \frac{1}{2}\,\mathrm{sgn}\,(x-y)\,.$$

The integrals in (1.3) and (1.5) need to be separated at point y so that the integrant is a smooth.

Elasto-Plastic Material In the book of Simo and Hughes [96] the basic concepts of computational plasticity are explained on a simple Coulomb friction device illustrated in Figure 1.3. The device expands linear elastically due to a stress σ. E denotes the Young's

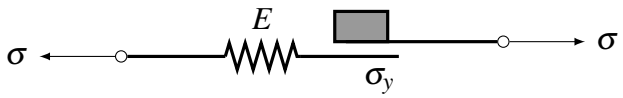

Figure 1.3: Coulomb friction-device representing 1-D plastic behavior

Modulus as elastic constant. The change of the length is described by the total strain ε. The total strain is split into an elastic part, associated with the spring, and a plastic part which is related to the friction element.

$$\varepsilon = \varepsilon_e + \varepsilon_p$$

For the spring-element it follows directly that

$$\sigma = E\varepsilon_e = E\,(\varepsilon - \varepsilon_p)\,.$$

The process in the friction element is described within a time increment Δt. If the stress exceeds the yield stress $\sigma > \sigma_y$ the friction device starts slipping. In that case, the *plastic strain-rate* is defined as

$$\dot{\varepsilon}_p = \frac{\partial \varepsilon_p}{\partial t}$$

where the over-dot $\dot{(\,)} = \frac{\partial}{\partial t}$ denotes the time derivative. To describe the behavior of the complete Coulomb friction device one needs to introduce a *yield condition* such as

$$F(\sigma) := |\sigma| - \sigma_y \leq 0 \tag{1.6}$$

for perfect plasticity. All stresses σ fulfilling condition (1.6) are admissible stresses. In multidimensional plasticity, the surface of these stresses is called the *yield surface*. If $F(\sigma) < 0$ the element exhibits purely elastic behavior with $\dot{\varepsilon}_p = 0$ and $\dot{\sigma} = E\dot{\varepsilon}$. If $F(\sigma) = 0$ slip occurs with a constant rate $\gamma \geq 0$ in the direction of σ. γ is called consistency parameter or *plastic multiplier*. Since $\gamma = 0$ for $F(\sigma) < 0$, the condition $\gamma F(\sigma) = 0$ holds. These

conditions are known as the Kuhn-Tucker conditions.
To describe the complete non-linear behavior numerically, the problem is linearized and solved incrementally in every n-th time step $\Delta t_n = t_{n+1} - t_n$. From this linearization one gets an incremental change of the strain state $\dot{\varepsilon}$. This induces a trial state of stress $\sigma_{trial} = \sigma_n + E\dot{\varepsilon}_{n+1}$ as an elastic predictor which needs to be checked by condition (1.6). If fulfilled, the incremental step $\dot{\varepsilon}_{n+1}$ is purely elastic. If not, the trial state needs to be corrected onto the yield surface to become admissible. This is usually done by the radial return algorithm (RRA) and is described by the function $\bar{\sigma}$

$$\sigma_{n+1} = \bar{\sigma}(\sigma_n, \varepsilon_n, \dot{\varepsilon}_{n+1}). \tag{1.7}$$

The function $\bar{\sigma}$ now returns the admissible stress state σ_{n+1}. Hence, a plastic residuum $\dot{\sigma}_p = \sigma_{trial} - \sigma_{n+1}$ is left. The complete state after the RRA can be described by

$$\begin{aligned}
\sigma_{n+1} &= \sigma_{trial} - \operatorname{sgn}(\sigma_{trial})\dot{\gamma}E \\
\varepsilon_{n+1} &= \varepsilon_n + \dot{\varepsilon}_{n+1} \\
\varepsilon_{p,n+1} &= \varepsilon_{p,n} + \operatorname{sgn}(\sigma_{trial})\dot{\gamma}
\end{aligned}$$

with the plastic stress- and strain-rates

$$\begin{aligned}
\dot{\sigma}_p &= \sigma_{trial} - \sigma_{n+1} \\
\dot{\varepsilon}_p &= \varepsilon_{p,n+1} - \varepsilon_{p,n}.
\end{aligned}$$

The described behavior of the Coulomb friction element is shown in Figure 1.4 for positive tensile stress. This is the simplest form of plasticity. In practical experiments with metals a hardening effect is observed. It is shown in Figure 1.5 that the admissible stress space expands with increasing plastic strain ε_p or strain rate $\dot{\varepsilon}_p$. This effect is described by the plastic modulus K and the hardening variable α. The yield condition then changes to

$$f(\sigma, \alpha) := |\sigma| - \left(\sigma_{y,0} + K\alpha\right) \leq 0 \tag{1.8}$$

for $\alpha \geq 0$. A simple definition for the hardening variable α would be

$$\alpha = \int_0^n |\dot{\varepsilon}_p| \, \mathrm{d}t.$$

Taking the aforementioned modifications into account, the RRA returns

$$\alpha_{n+1} = \alpha_n + \dot{\gamma} \qquad \text{with} \qquad \dot{\gamma} = \frac{F(\sigma_{trial}, \alpha_n)}{E + K}.$$

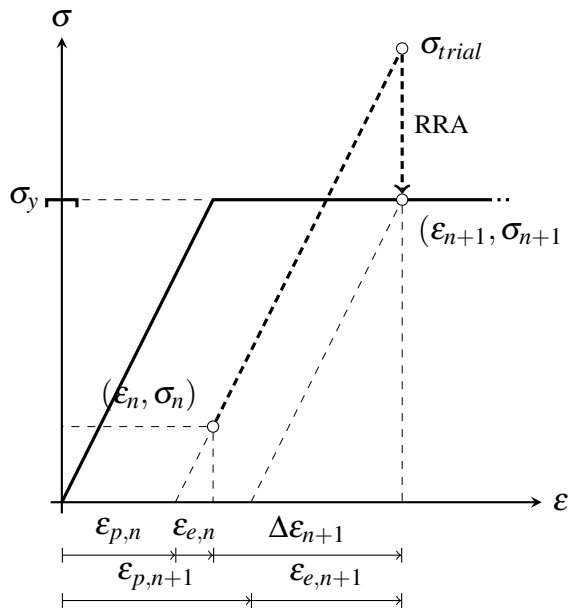

Figure 1.4: Elastic- perfectly-plastic material behavior RRA

Finally, a tangential modulus

$$H = \begin{cases} E, & \gamma = 0 \quad \text{(elastic)} \\ \frac{EK}{E+K}, & \gamma > 0 \quad \text{(plastic)} \end{cases}$$

can be defined, with $K = 0$ for a perfectly plastic material law.

Elasto-Plastic Column Since plasticity is a local phenomenon, such kind of problems are usually non-linear and not solvable directly in a closed form. Instead, the problem is linearized and solved iteratively applying Newton's method [81]. As mentioned in the previous paragraph, $\dot{\sigma}_p > 0$ introduces unbalanced internal forces in the linearized trial step. These forces need to be re-applied to the initial problem in terms of body forces, initial strains or initial stresses since (1.4) holds for elastic material behavior only.

For the bar in Figure 1.2, consider an elasto-plastic material law with isotropic hardening shown in Figure 1.5. The problem domain Ω is now discretized by finite cell elements as shown in Figure 1.6. One cell describes a Coulomb friction device. Additionally, the body forces $n(x)$ are interpolated

$$n(x) \approx \sum_{j=1}^{N} n(x_j)\varphi_j(x)$$

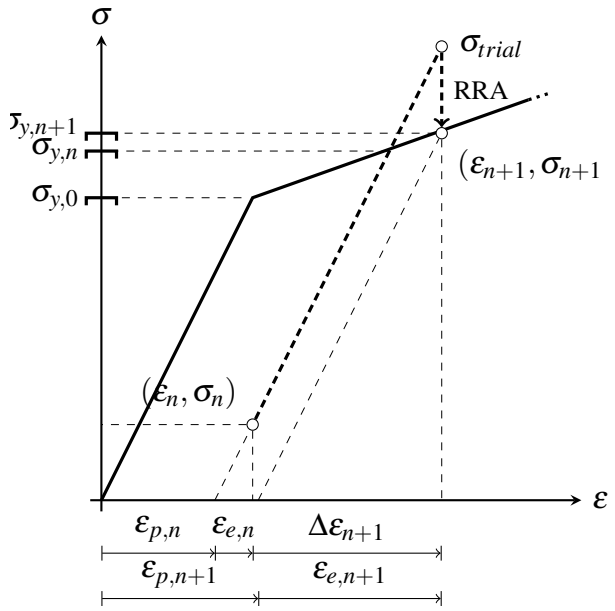

Figure 1.5: Elastic-plastic material behavior with strain hardening effect and RRA

along the bar. The interpolation-functions $\varphi_j(x)$ are chosen to be piecewise constant

$$\varphi_j(x) = \begin{cases} 1 & \frac{\ell(j-1)}{N} \leq x < \frac{\ell j}{N} \\ 0 & \text{else} \end{cases}$$

along the cells. After solving (1.4) the strain and stress for every finite cell is calculated

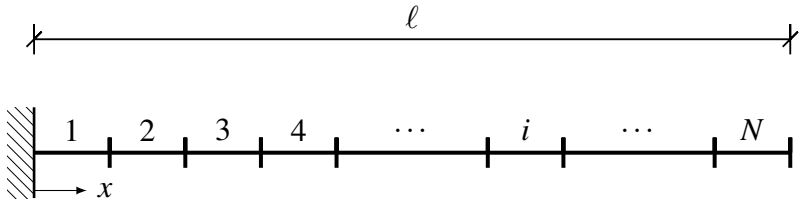

Figure 1.6: Cell-discretization of the elastic column

by (1.5). Now the stress in the element is checked against the yield criterion (1.8). If the condition is fulfilled for all cells, the load results in a purely elastic reaction. Cells where the condition doesn't hold are subject to plastic strain. The initial plastic strains $\dot{\varepsilon}_p$ or the corresponding initial stresses $\dot{\sigma}_p$ need to be applied as artificial body forces in the

following step. This is done by adding a domain-term

$$\cdots + \int_0^\ell \sigma_p(x)\varepsilon^*(x,y)\,\mathrm{d}x$$

to the right hand side of (1.3) and

$$\cdots + \int_0^\ell \sigma_p(x)\varepsilon^{*\prime}(x,y)\,\mathrm{d}x,$$

to (1.5). The procedure shown in Algorithm 1 is repeated until the residual stresses $\dot{\sigma}_p$ are zero.

Algorithm 1 Algorithm for the numeric solution of the elasto-plastic column

```
1:  procedure SOLVE ELASTO-PLASTIC COLUMN
2:      compute elastic column
3:      for all internal cells do
4:          determine initial elastic stress σ_trial,0
5:          RRA(σ_trial,0, Δε_0) → valid stress σ_0
6:          calculate residual stress σ_p
7:      end for
8:      while norm of residual ‖σ̇_p‖ > ε_σ do
9:          update incremental unknown boundary values u̇ and Ṅ
10:         for all internal cells do
11:             calculate trial stress σ_trial,n
12:             RRA(σ_trial,n, Δε_n) → σ_n+1
13:             calculate σ̇_p
14:         end for
15:     end while
16:     determine final boundary values and stress state
17: end procedure
```

Matrix Approximation The internal point evaluation of stresses $\dot{\sigma}_n$ resulting for every cell-midpoint of the discretized column can be written in matrix-vector notation

$$\dot{\boldsymbol{\sigma}} = \mathbf{b} + \mathbf{E}\dot{\boldsymbol{\sigma}}_p, \quad \mathbf{E} \in \mathbb{R}^{N\times N} \tag{1.9}$$

where **b** represents the linear, purely elastic part of the solution. The second term involves the residual plastic stresses. The coefficients of **E** are

$$\mathbf{E}[i,j] = \int_{\Omega_j} \varepsilon^{*\prime}(x_i, y)\varphi_j(y)\,\mathrm{d}y. \tag{1.10}$$

The indices i and j indicate the point of evaluation and cell to be integrated. Since $\varepsilon^{*\prime}$ is a globally defined function, the matrix **E** is fully populated. Therefore the matrix-vector product needs $\mathcal{O}(N^2)$ operations. The required storage scales with $\mathcal{O}(N^2)$ as well. That means if the number of cells is doubled, the computational effort increases by the factor $2^2 = 4$. That could become unacceptable for a high amount of cells. To overcome this, fast summation methods have been introduced, which provides linear $\mathcal{O}(N)$ or near linear $\mathcal{O}(N\log^{\alpha} N)$ complexity instead. It is important to note that for $i = j$ the function to be integrated is not smooth and the integral needs to be treated separately. To achieve the reduced complexity, the main objective is to find a data sparse representation of the system matrix. At this point low rank matrices ($\mathcal{R}_k$-matrices) are introduced [52]. A $\mathcal{R}_k$-matrix can be represented by the factorization

$$\mathbf{R}_k = \mathbf{A}\cdot\mathbf{B}^T \quad \mathbf{A}\in\mathbb{R}^{r\times k}, \mathbf{B}\in\mathbb{R}^{c\times k}.$$

Here, k denotes the rank of $\mathbf{R}_k$, r and c the rows and columns respectively. The rank of a matrix is determined by the number of linear independent columns or rows. If $k \ll \min\{r,c\}$ the storage $\mathsf{St} = k(r+c)$ can be significantly smaller than for full matrices where $\mathsf{St}_F = rc$. The computational complexity for a matrix-vector product $\mathsf{Op} = \mathcal{O}(2k(r+c))$ instead of $\mathsf{Op}_F = \mathcal{O}(rc)$ for the full representation.

In this example one would like to approximate the full system matrix **E** by a $\mathcal{R}_k$-matrix in such a way that a certain accuracy $\|\mathbf{E}-\mathbf{R}_k\| < \epsilon_{\mathcal{H}}$ measured in a matrix norm could be reached. Investigating the matrix-entries (1.10), this could eventually be achieved by separation of the variables x and y. A separation is only possible if $x \neq y$. This motivates a *clustering* of the whole matrix with respect to its geometry. In that special example this also corresponds to a clustering with respect to the indices. First, the discretized column is divided into two parts holding the index sets $I_1 = \{1,\ldots,N/2-1\}$ and $I_2 = \{N/2,\ldots,N\}$. In relation to that, the system matrix **E** is divided into 4 block-matrices as depicted in Figure 1.7. Now, the blocks $\mathbf{E}_{ij}$ represent matrix entries for the index sets $I_i \times I_j$. Clearly, the off-diagonal matrix blocks $\mathbf{E}_{12}$ and $\mathbf{E}_{21}$ are candidates for a data-sparse representation by separation of variables, while the diagonal matrix-blocks are not. To generate more *admissible* matrix-blocks, the same concept is applied to the sub-matrices $\mathbf{E}_{11}$ and $\mathbf{E}_{22}$. Again, resulting in a block-wise representation. This approach could be repeated recursively up to a certain level. Matrices of this form are called $\mathcal{H}$-matrices [52].

In the following section we consider the matrix block $\mathbf{E}_{12}$ as an approximation in the $\mathcal{R}_k$-matrix format. The observation points x_i lie in the interval $[0,\ell/2]$ and the correlated cells in $[\ell/2,\ell]$. The idea is to introduce a degenerated kernel which allows the treatment

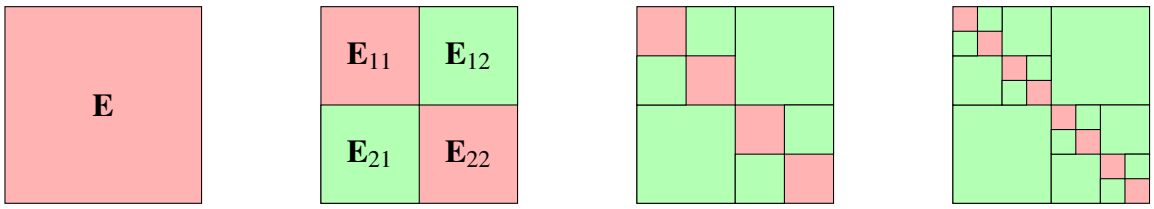

Figure 1.7: Different levels of clustering for the system matrix **E**

of x and y separately. This can be achieved by means of truncated Taylor-expansion or, in that case, by polynomial interpolation

$$\varepsilon'^{*}(x,y) \approx \sum_{\nu=1}^{k} \sum_{\mu=1}^{k} L_\nu(x)\varepsilon'^{*}(\bar{x}_\nu,\bar{x}_\mu)L_\mu(y). \tag{1.11}$$

using Lagrange polynomials

$$L_\nu(x) = \prod_{\mu=0,\mu\neq\nu}^{k} \frac{x-\bar{x}_\mu}{\bar{x}_\nu-\bar{x}_\mu} \tag{1.12}$$

where all $\bar{x}$ denote the support points to be chosen. For the matrix entries of $\mathbf{E}_{12}$ this leads to the matrix-form

$$\mathbf{E}_{12} = \mathbf{V}\cdot\mathbf{S}\cdot\mathbf{W}^T \qquad \mathbf{V}\in\mathbb{R}^{r\times k}, \mathbf{S}\in\mathbb{R}^{k\times k}, \mathbf{W}\in\mathbb{R}^{c\times k}$$

with the entries

$$\begin{aligned}
\mathbf{V}[i,\nu] &= L_\nu(x_i) && i\in\{1,\ldots,N/2\}\\
\mathbf{S}[\nu,\mu] &= \varepsilon^{*\prime}(\bar{x}_\nu,\bar{y}_\mu) && \nu,\mu\in\{1,\ldots,k\}\\
\mathbf{W}[j,\mu] &= \int_{\Omega_j} L_\mu(y_j)\varphi_j(y)\,\mathrm{d}y && j\in\{N/2+1,\ldots,N\}
\end{aligned}$$

Evaluations of the original kernel $\varepsilon^{*\prime}$ are only carried out for $\mathbf{S}$ at the supports points $\bar{x}$ and $\bar{y}$ of the interpolation polynomials L_ν and L_μ. The $\mathcal{R}_k$-matrix representation is easily calculated by $\mathbf{A}=\mathbf{V}$ and $\mathbf{B}=\mathbf{S}\mathbf{W}^T$. The error of that approximation is specified by the order of interpolation, more specifically the rank k of $\mathbf{E}_{12}$.

Remark: Since in case of this example, the integral kernel $\phi(x,y)=\varepsilon^{*\prime}(x,y)$ is constant for the admissible matrix blocks, a constant interpolation using one support point per index set is exact. Usually, the kernels are of no polynomial type and involve singularities like $\ln|x-y|$ or $|x-y|^{-d}$ and no exact representation with interpolants is possible.

The described procedure is repeated for every admissible sub-matrix in Figure 1.7. The

entries of inadmissible blocks, in that example the matrix diagonal, are evaluated by direct integration of (1.10). With the $\mathcal{H}$-matrix at hand, the numerical solution of the described procedure is carried out with almost linear complexity with respect to the size of the problem.

1.4 Outline

The outline for the remainder of the thesis at hand is as follows. *Chapter 2* introduces the model problem and gives an introduction to the solution by means of boundary integral equations.
For the approximative numerical solution of such integral equations a BEM formulation for linear elastostatics and the extension to nonlinear elasto-plastic analysis is described in *chapter 3*. Generally, three-dimensional problems are treated but the theory is presented in two for sake of simplicity.
In *chapter 4*, the basic concepts of matrix approximation by means of $\mathcal{R}_k$-matrices are introduced. The format of $\mathcal{H}$-matrices, their generation and some operations is presented.
Since this work aims for practical applications of $\mathcal{H}$-matrices to known BEM techniques, *chapter 5* is dedicated to the fast solution of elastostatic and alasto-plastic systems of equations as well as details and special techniques used for the implementation into a program code.
Chapter 6 is devoted to numerical results. Some benchmark examples are shown to validate the implementation. Furthermore, several real word examples give an idea of the performance of the implemented alorithm. Parameter studies related to the $\mathcal{H}$-matrix structure and the quality of far-field approximations are given.
Finally, *chapter 7* recapitulates the work at hand. Drawbacks and weaknesses are pointed out and an outlook for possible further developments is given.

2 Boundary Integral Equations

This chapter is dedicated to linear elasticity and the representation by means of BIEs. In that context, all necessary operators are introduced which allow a compact notation of the mathematical statements. The explanations are extended to nonlinearities induced by elasto-plastic behavior of the material. Many textbooks on BIEs and boundary element methods for different fields of application have been written. Therefore, the explanations in this work are kept quite short and mainly introduce the used notation. I.e. the books of Beer et al. [12], Gaul et al. [41] and Telles [104] give a good introduction to the method for engineers and present some special applications. A mathematical point of view provide Steinbach [101], Sauter and Schwab [95] as well as Bonnet [17]. From the latter books, the operator notation is taken and introduced in the following.

2.1 Elastic Continuum

The deformations of the material to be considered are assumed to be small enough such that all resulting equations are linear. In this case, a Cartesian coordinate system spanned by the orthonormal unit vectors $(\mathbf{e}_1, \ldots, \mathbf{e}_d)$ with d being the spatial dimension is sufficient and only the reference position $\boldsymbol{x} \in \mathbb{R}^d$ of a material-point is needed in order to describe all necessary relations. The domain is solid, isotropic and linearly elastic. Therefore, the *kinematic* relation between the Cauchy strain tensor $\boldsymbol{\varepsilon}$ and the displacement u is

$$\mathcal{B}u(\boldsymbol{x}) = \boldsymbol{\varepsilon}(\boldsymbol{x}) = \frac{1}{2}\left((\nabla u(\boldsymbol{x}))^{\mathsf{T}} + \nabla u(\boldsymbol{x})\right)$$

with $\mathcal{B}$ defining the strain operator. The *constitutive* relationship between stress $\boldsymbol{\sigma}$ and strain follows Hooke's law defining the stress operator $\mathcal{S}$

$$\mathcal{S}u(\boldsymbol{x}) = \boldsymbol{\sigma}(\boldsymbol{x}) = \mathsf{C}(\lambda, \mu) : \boldsymbol{\varepsilon}(\boldsymbol{x}) \tag{2.1}$$

with C denoting the fourth order stiffness tensor. From a physical point of view the Lamé constants λ and μ can be replaced by the more meaningful Young's modulus E and Poisson's ratio ν

$$\lambda = \frac{E \cdot \nu}{(1-2\nu)(1+\nu)} \quad \text{and} \quad \mu = \frac{E}{2(1+\nu)}$$

with $\mu > 0$ and $\lambda > -2\mu/3$ respectively. μ is also known as shear modulus G. Note that the tensors $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^{\mathsf{T}}$, $\boldsymbol{\sigma} = \boldsymbol{\sigma}^{\mathsf{T}}$ and C are symmetric. Moreover, C is considered to be constant

and independent of $\boldsymbol{x}$. With the equations of *equilibrium*

$$\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{x}) + f(\boldsymbol{x}) = 0 \tag{2.2}$$

the partial differential equation (PDE) for linear elastostatics in terms of displacements

$$\mathcal{L}u(\boldsymbol{x}) = -(\lambda + 2\mu)\nabla \cdot \nabla u(\boldsymbol{x}) - \mu \nabla \times (\nabla \times u(\boldsymbol{x})) = f(\boldsymbol{x}) \tag{2.3}$$

with body forces f is formulated. $\mathcal{L}$ denotes the Lamé-Navier operator. For two-dimensional problems, two different basic assumptions can be made. For *plane strain* the displacements in the x_3-direction are not considered in the above equations or operators. The stress component $\sigma_{33} \neq 0$ exists, albeit all other components containing the index 3 are zero. In case of *plane stress* problems, the stress-state is considered to be two-dimensional. Hence $\sigma_{i3} = \sigma_{3i} = 0$.

Boundary Value Problem Let us consider a Lipschitz domain $\Omega \in \mathbb{R}^d$ and its boundary $\Gamma = \partial\Omega$, subject to prescribed boundary conditions. The part of Γ over which displacement $u(\boldsymbol{y})$ is given is the *Dirichlet* boundary Γ_D. On the *Neumann* boundary Γ_N the surface tractions $t(\boldsymbol{y})$ are given and are related to the stresses by the unit *outward* normal vector $\mathbf{n}$. The boundary values u and t are called *Cauchy data*. In the following discussion the *boundary trace*

$$\mathrm{Tr}\, u(\boldsymbol{x}) = \lim_{\mathbf{x}\to\mathbf{y}} u(\boldsymbol{x}) = u(\boldsymbol{y}) \quad \boldsymbol{x} \in \Omega, \boldsymbol{y} \in \Gamma$$

and the *conormal derivative* or traction operator

$$\mathcal{T}u(\boldsymbol{x}) = \lim_{\mathbf{x}\to\mathbf{y}} \boldsymbol{\sigma}(\boldsymbol{x})\mathbf{n}(\boldsymbol{y}) = t(\boldsymbol{y}) \quad \boldsymbol{x} \in \Omega, \boldsymbol{y} \in \Gamma$$

are introduced [19, 101]. $\mathcal{T}$ involves surface derivatives and the relationship between stresses and strains. The operator is represented by

$$\mathcal{T}u(\boldsymbol{x}) = \lambda \nabla \cdot u(\boldsymbol{y})\mathbf{n}(\boldsymbol{y}) + 2\mu \nabla u(\boldsymbol{y}) \cdot \mathbf{n}(\boldsymbol{y}) + \mu \mathbf{n}(\boldsymbol{y}) \times (\nabla \times u(\boldsymbol{y})) \tag{2.4}$$

and maps displacements to surface tractions. The boundary splits into a Neumann and a Dirichlet part such that $\Gamma = \Gamma_N \cup \Gamma_D$ and $\Gamma_N \cap \Gamma_D = 0$. Thus, the following problem is well posed. Let $\Omega_p \subset \Omega$ be a part of the domain which is subjected to body forces f. A typical domain is shown in Figure 2.1. This leads to the following inhomogeneous boundary value problem: Find a displacement field $u(\boldsymbol{x})$ so that

$$\begin{aligned} \mathcal{L}u(\boldsymbol{x}) &= f(\boldsymbol{z}) \quad && \forall \boldsymbol{x} \in \Omega, \forall \boldsymbol{z} \in \Omega_p \\ t(\boldsymbol{y}) &= g_N(\boldsymbol{y}) \quad && \forall \boldsymbol{y} \in \Gamma_N \\ u(\boldsymbol{y}) &= g_D(\boldsymbol{y}) \quad && \forall \boldsymbol{y} \in \Gamma_D. \end{aligned} \tag{2.5}$$

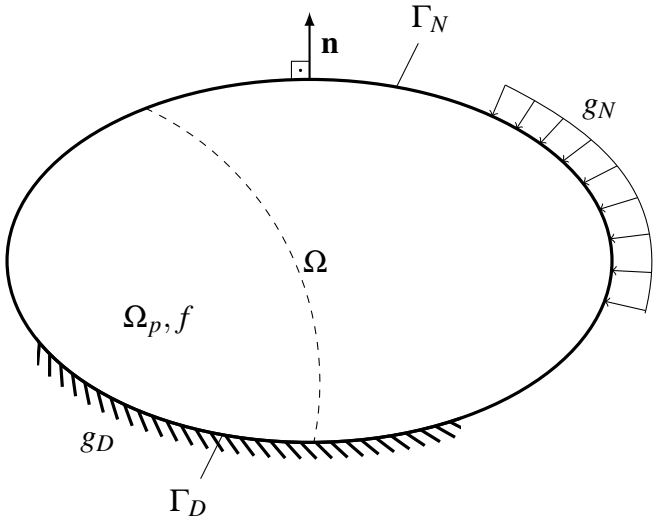

Figure 2.1: Fixed elastic body subject to external and body force loading

Here, g_N is the prescribed Neumann data in terms of surface tractions and g_D represents the prescribed Dirichlet data in terms of displacements. For application to problems in underground construction, infinite or semi-infinite domains are considered as depicted in Figure 2.2. During the construction of deep tunnels or caverns the surface settlements are negligible, and therefore an infinite region is analyzed. For all other types of tunneling the settlements are of interest and the influence on buildings can be evaluated by taking a semi-infinite region. Since there is usually no prescribed displacement on such surfaces, the boundary condition

$$\begin{aligned} u(\boldsymbol{y}) &= u_0 \qquad \forall \boldsymbol{y} \in \Gamma_D = \partial B_R \\ t(\boldsymbol{y}) &= g_N(\boldsymbol{y}) \quad \forall \boldsymbol{y} \in \Gamma_N \end{aligned}$$

needs to be solved. The Dirichlet boundary lies on the surface ∂B_R of a sphere B_R with radius $R \to \infty$ tending to infinity. In practical examples u_0 usually equals zero. It will be seen later that for the BEM such problems are treated on Γ_N only, thus as an exterior Neumann problem.

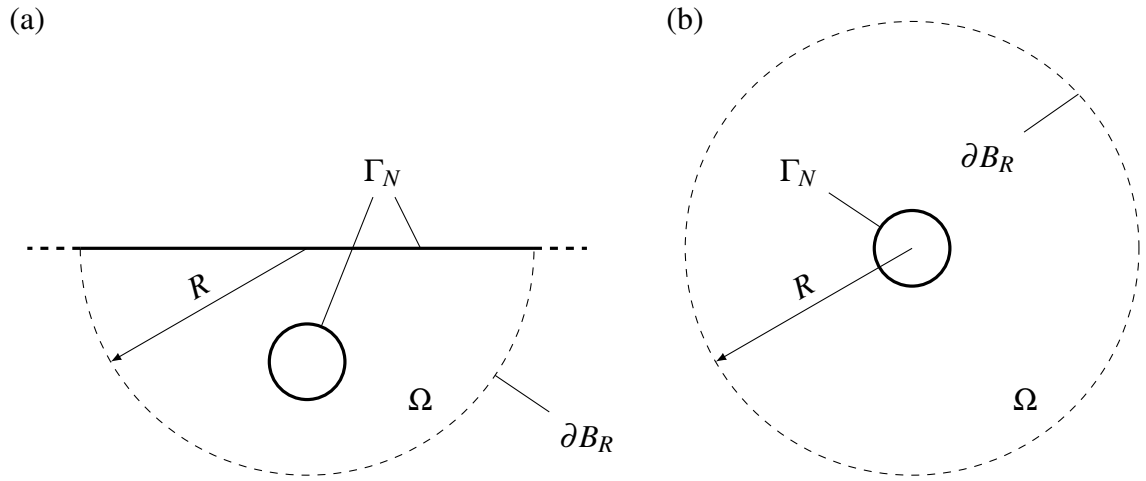

Figure 2.2: Semi-infinite (a) and infinite (b) exterior domains

2.2 Linear Elastic Problems

Boundary Integrals For linear elasticity, the boundary value problem can be solved by means of boundary integrals. The starting point is the *weak form*

$$\int_{\Omega} (\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{x})) \cdot v(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} + \int_{\Omega_p} f(\boldsymbol{z}) \cdot v(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z} = 0 \tag{2.6}$$

of equation (2.3). Using the divergence theorem

$$\int_{\Omega} \nabla \cdot \boldsymbol{\sigma}(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = \int_{\Gamma} \mathrm{Tr}\, \boldsymbol{\sigma}(\boldsymbol{x}) \cdot \mathbf{n}(\boldsymbol{y}) \, \mathrm{d}s_{\boldsymbol{y}} \quad \forall \boldsymbol{x} \in \Omega$$

with $\boldsymbol{y} \in \Gamma$ and the identity for the divergence of

$$\nabla \cdot (\boldsymbol{\sigma}^{\mathsf{T}}(\boldsymbol{x}) \cdot v(\boldsymbol{x})) = \boldsymbol{\sigma}(\boldsymbol{x}) : \nabla v(\boldsymbol{x}) + \nabla \boldsymbol{\sigma}(\boldsymbol{x}) \cdot v(\boldsymbol{x}), \tag{2.7}$$

Green's first identity for the linear elastic problem

$$\int_{\Omega} \nabla \boldsymbol{\sigma}(\boldsymbol{x}) v(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = \int_{\Gamma} \mathrm{Tr}\, \boldsymbol{\sigma}(\boldsymbol{x}) \mathbf{n}(\boldsymbol{y}) v(\boldsymbol{y}) \, \mathrm{d}s_{\boldsymbol{y}} - \int_{\Omega} \boldsymbol{\sigma}(\boldsymbol{x}) : \nabla v(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \tag{2.8}$$

is obtained. Since the Lamé operator $\mathcal{L}$ relates the displacement field u with $\nabla\boldsymbol{\sigma}$ and $\mathcal{T}$ maps u to surface tractions $t \in \Gamma$, identity (2.8) could be written in a more compact way

$$\int_{\Omega} \mathcal{L}u(\boldsymbol{x}) v(\boldsymbol{x}) = \int_{\Gamma} \mathcal{T}u(\boldsymbol{x}) v(\boldsymbol{y}) \, \mathrm{d}s_{\boldsymbol{y}} - a(u, v)$$

with

$$a(u,v) = \int_\Omega \sigma_u(\boldsymbol{x}) : \varepsilon_v(\boldsymbol{x}) \,\mathrm{d}\Omega$$

being a bilinear form. Since σ_u is symmetric, the gradient of the test function ∇v in (2.8) equals the symmetric ε_v. The introduced subscripts denote the field the quantities are related to. Usually, above equation is the starting point for the FEM. For the BEM formulation the next steps are crucial: the above described procedure is repeated exchanging u and v. Using the *reciprocal work principle* [41] as a generalization of *Betti's reciprocity theorem* [12], the bilinear form is realized to be symmetric $a(u,v) = a(v,u)$. With $\mathcal{L}u = f$ equation (2.9) is derived.

$$\int_\Omega u(\boldsymbol{x})\mathcal{L}v(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} = \int_\Gamma u(\boldsymbol{y})\mathcal{T}v(\boldsymbol{x}) \,\mathrm{d}s_{\boldsymbol{y}} - \int_\Gamma \mathcal{T}u(\boldsymbol{x})v(\boldsymbol{y}) \,\mathrm{d}s_{\boldsymbol{y}} + \int_{\Omega_p} f(\boldsymbol{z})v(\boldsymbol{z}) \,\mathrm{d}\boldsymbol{z} \tag{2.9}$$

Considering for the moment that there are no body forces $f = 0$, the next step is to get rid of the last remaining domain-integral. This is done by finding a function u^* which satisfies

$$\mathcal{L}u^*(\boldsymbol{x},\boldsymbol{y}) = \delta(\boldsymbol{y} - \boldsymbol{x})$$

with the *Dirac-delta* δ and its properties

$$\delta(x) = \begin{cases} +\infty & x = 0 \\ 0 & \text{else} \end{cases} \quad \text{and} \quad u(\tilde{x}) \;= \int_{-\infty}^{+\infty} \delta(x - \tilde{x})u(x) \,\mathrm{d}x.$$

u^* is called a *fundamental solution* of the underlying problem. In case of elasto-static problems this is *Kelvin's* fundamental solution $u^*(\boldsymbol{x},\boldsymbol{y}) \equiv \mathsf{U}(\boldsymbol{x},\boldsymbol{y})$ for displacements [12, 41], which is symmetric. It represents the displacements at a point $\boldsymbol{x}$ induced by a unit point force in point $\boldsymbol{y}$ of an unbounded domain. Actually, the fundamental solution does not directly depend on the variables $\boldsymbol{x}$ and $\boldsymbol{y}$ but on their distance $|\boldsymbol{x} - \boldsymbol{y}|$ and is therefore translationally invariant. Finally, the test function v in (2.9) is chosen to be $\mathsf{U}(\boldsymbol{x},\boldsymbol{y})$. Evaluating the traction fundamental solution by

$$\mathsf{T}(\boldsymbol{x},\boldsymbol{y}) = \mathcal{T}_{\boldsymbol{y}}\mathsf{U}(\boldsymbol{x},\boldsymbol{y}) \tag{2.10}$$

results in the representation formula or *Somigliana's identity* for the displacement of a point $\boldsymbol{x} \in \Omega$

$$u(\boldsymbol{x}) = \int_\Gamma \mathsf{U}(\boldsymbol{x},\boldsymbol{y})t(\boldsymbol{y}) \,\mathrm{d}s_{\boldsymbol{y}} - \int_\Gamma \mathsf{T}(\boldsymbol{x},\boldsymbol{y})u(\boldsymbol{y}) \,\mathrm{d}s_{\boldsymbol{y}} + \int_{\Omega_p} \mathsf{U}(\boldsymbol{x},\boldsymbol{z})f(\boldsymbol{z}) \,\mathrm{d}\boldsymbol{z} \quad \forall \boldsymbol{y} \in \Gamma,\, \boldsymbol{z} \in \Omega_p \tag{2.11}$$

expressed by boundary integrals and a domain integral for the contribution of body forces. It is common for all fundamental solutions used in this work to have a singularity of order

$$\mathcal{O}\left(\frac{1}{r^k}\right) \quad \text{or} \quad \mathcal{O}\left(\ln\frac{1}{r}\right) \quad \text{with} \quad r = |\boldsymbol{x}-\boldsymbol{y}| \tag{2.12}$$

when the distance r between $\boldsymbol{x}$ and $\boldsymbol{y}$ goes to zero. For boundary integrals in three dimensions a kernel with $k=1$ is called *weakly-*, *strongly-* with $k=2$ and *hyper-*singular with $k=3$. In two dimensions the weakly singular term is $\ln r$, the strongly singular yields $k=1$ and the hyper singular kernel $k=2$.

Unbounded Domains One of the major advantages of the BEM is the analysis of infinite and semi-infinite (exterior) domains. This is because of the peculiar characteristics of the fundamental solutions in context of the integral equation. They fulfill the *Sommerfeld radiation condition* [32]. As shown in Figure 2.2, an artificial circular or spherical boundary ∂B_R with radius R is considered to enclose Γ and the region with body forces Ω_p. The representation formula can be written as

$$\begin{aligned} u(\boldsymbol{x}) = \int_\Gamma \mathsf{U}(\boldsymbol{x},\boldsymbol{y}) t(\boldsymbol{y})\,\mathrm{d}s_{\boldsymbol{y}} - \int_\Gamma \mathsf{T}(\boldsymbol{x},\boldsymbol{y}) u(\boldsymbol{y})\,\mathrm{d}s_{\boldsymbol{y}} + \int_{\Omega_p} \mathsf{U}(\boldsymbol{x},\boldsymbol{z}) f(\boldsymbol{z})\,\mathrm{d}\boldsymbol{z} \\ + \int_{\partial B_R} \mathsf{U}(\boldsymbol{x},\boldsymbol{y}) t(\boldsymbol{y})\,\mathrm{d}s_{\boldsymbol{y}} - \int_{\partial B_R} \mathsf{T}(\boldsymbol{x},\boldsymbol{y}) u(\boldsymbol{y})\,\mathrm{d}s_{\boldsymbol{y}} \end{aligned} \tag{2.13}$$

Because of the above mentioned singular behavior, it can be seen that the last two integrals

$$\lim_{R\to\infty} \left(\int_{\partial B_R} \mathsf{U}(\boldsymbol{x},\boldsymbol{y}) t(\boldsymbol{y})\,\mathrm{d}s_{\boldsymbol{y}} - \int_{\partial B_R} \mathsf{T}(\boldsymbol{x},\boldsymbol{y}) u(\boldsymbol{y})\,\mathrm{d}s_{\boldsymbol{y}} \right) = 0 \tag{2.14}$$

vanish and the representation formula (2.11) is valid for exterior problems too.

Boundary Integral Operators The representation formula (2.11) and therefore the displacements $u \in \Omega$ can be evaluated if the full Cauchy data u and t on the boundary $\boldsymbol{y} \in \Gamma$ is given. Initially this is not the case and a solution of the underlying problem is not possible in this form. To achieve this, the point $\boldsymbol{x}$ gets shifted to the boundary by the limiting process $\boldsymbol{x} \ni \Omega \to \boldsymbol{x} \in \Gamma$. This is understood as application of the boundary trace operator Tr to the whole formula [101] and yields the BIE of first order

$$u(\boldsymbol{x}) = \mathrm{Tr}\int_\Gamma \mathsf{U}(\boldsymbol{x},\boldsymbol{y}) t(\boldsymbol{y})\,\mathrm{d}s_{\boldsymbol{y}} - \mathrm{Tr}\int_\Gamma \mathsf{T}(\boldsymbol{x},\boldsymbol{y}) u(\boldsymbol{y})\,\mathrm{d}s_{\boldsymbol{y}} + \mathrm{Tr}\int_{\Omega_p} \mathsf{U}(\boldsymbol{x},\boldsymbol{z}) f(\boldsymbol{z})\,\mathrm{d}\boldsymbol{z} \quad \boldsymbol{x} \in \Gamma.$$

Remark: To achieve a symmetric formulation, the trace operator Tr *and* the traction operator $\mathcal{T}$ are applied to (2.11). With the boundary integral equation of second order and with the concept of weighted residuals [41], a skew symmetric formulation can be formulated. For further insight and the application of the SGBEM the reader is referred to [102] and [95, 101].

Because of the singular fundamental solutions, the mentioned limiting process has to be done carefully. Again, the author would like to refer to the textbooks [95, 101] for detailed analysis of that process and to [12] and the references therein for the practical treatment of the singular integrals. For convenience, boundary integral operators are introduced [95]. The *single layer* operator

$$(\mathcal{V}t)(\boldsymbol{x}) = \int_{\Gamma} \mathsf{U}(\boldsymbol{x},\boldsymbol{y}) t(\boldsymbol{y}) \,\mathrm{d}s_{\boldsymbol{y}} \quad \forall \boldsymbol{x},\boldsymbol{y} \in \Gamma$$

is weakly singular and causes no implications with the trace operation. More challenging is the derivation of the *double layer* operator

$$(\tilde{\mathcal{K}}u)(\boldsymbol{x}) = \int_{\Gamma} \mathsf{T}(\boldsymbol{x},\boldsymbol{y}) u(\boldsymbol{y}) \,\mathrm{d}s_{\boldsymbol{y}} \quad \forall \boldsymbol{x},\boldsymbol{y} \in \Gamma \setminus B_{\varepsilon}(\boldsymbol{x}).$$

The integral only exists as a *Cauchy principal value*, where the radius ε of a sphere B_{ε} around $\boldsymbol{x}$ is treated in a limiting process $\varepsilon \to 0$. The remainder of that process is an integral free term which is

$$\mathcal{C}u(\boldsymbol{x}) = cu(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \Gamma \tag{2.15}$$

with $c = 1/2$ for smooth surfaces. For non-smooth surfaces the value of c depends on the angle of the kink at $\boldsymbol{x}$ and on Poisson's ratio ν in elasticity [69]. For sake of simplicity we substitute

$$(\mathcal{K}u)(\boldsymbol{x}) = \mathcal{C}\mathcal{I}u(\boldsymbol{x}) + \tilde{\mathcal{K}}u(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \Gamma$$

with $\mathcal{I}$ being the identity. Finally, with the *Newton* operator

$$(\tilde{\mathcal{N}}f)(\boldsymbol{x}) = \int_{\Omega} \mathsf{U}(\boldsymbol{x},\boldsymbol{z}) f(\boldsymbol{z}) \,\mathrm{d}\boldsymbol{z} \quad \forall \boldsymbol{x} \in \Gamma, \forall \boldsymbol{z} \in \Omega$$

taking the body forces f into account, the boundary integral equation yields

$$(\mathcal{K}u)(\boldsymbol{x}) = (\mathcal{V}t)(\boldsymbol{x}) + (\tilde{\mathcal{N}}f)(\boldsymbol{x}). \tag{2.16}$$

A comprehensive and detailed analysis of the properties of these operators can be found in [95] or [101] among other, rather mathematical, textbooks.

Strains and Stresses For many engineering problems, interior point evaluations are of interest. Displacements are directly calculated with the representation formula (2.11). Moreover, for the elasto-plastic analysis described in section 2.3 stresses are significant. It is now assumed that all evaluations are carried out at points $\tilde{\boldsymbol{x}} \in \Omega$ inside the domain.

The expression for the strain evaluation stems from the utilization of the strain operator $\mathcal{B}$ to the representation formula (2.11). The resulting operators are marked by the prime symbol $()'$. Since the original integral-kernels are smooth for $\tilde{\boldsymbol{x}} \neq \boldsymbol{y}$, the differentiation and the integration is interchanged and the following integral operators

$$(\mathcal{V}'_\varepsilon t)(\tilde{\boldsymbol{x}}) = \int_\Gamma \mathcal{B}\mathsf{U}(\tilde{\boldsymbol{x}}, \boldsymbol{y}) t(\boldsymbol{y}) \, \mathrm{d}s_{\boldsymbol{y}} \quad \forall \boldsymbol{y} \in \Gamma$$

and

$$(\mathcal{K}'_\varepsilon t)(\tilde{\boldsymbol{x}}) = \int_\Gamma \mathcal{B}\mathsf{T}(\tilde{\boldsymbol{x}}, \boldsymbol{y}) u(\boldsymbol{y}) \, \mathrm{d}s_{\boldsymbol{y}} \quad \forall \boldsymbol{y} \in \Gamma$$

are defined. Here, the derivatives implied by $\mathcal{B}$ are carried out with respect to $\tilde{\boldsymbol{x}}$. If $|\tilde{\boldsymbol{x}} - \boldsymbol{y}| \to 0$, the *volume operator* [27]

$$(\tilde{\mathcal{N}}'_\varepsilon f)(\boldsymbol{x}) = \oint_\Omega \mathcal{B}\mathsf{U}(\boldsymbol{x}, \boldsymbol{z}) f(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z} + \tilde{\mathcal{C}}\mathcal{I} f(\boldsymbol{z}) \quad \forall \boldsymbol{z} \in \Omega$$

is strongly singular and exists only as a Cauchy principal value. The necessary limiting process is symbolically denoted by the integral symbol $\oint$. Again, $\tilde{\mathcal{C}}$ is an integral free term depending on Poisson's ration ν only. Using this notation, Somigliana's identity for strain is written as

$$\boldsymbol{\varepsilon}(\tilde{\boldsymbol{x}}) = (\mathcal{V}'_\varepsilon t)(\tilde{\boldsymbol{x}}) - (\mathcal{K}'_\varepsilon u)(\tilde{\boldsymbol{x}}) + (\tilde{\mathcal{N}}'_\varepsilon f)(\tilde{\boldsymbol{x}}) \quad \forall \tilde{\boldsymbol{x}} \in \Omega. \tag{2.17}$$

Multiplying (2.17) with C or, in other words, applying the stress operator $\mathcal{S}$ to (2.11) yields the formula for the evaluation of internal stress

$$\boldsymbol{\sigma}(\tilde{\boldsymbol{x}}) = (\mathcal{V}'_\sigma t)(\tilde{\boldsymbol{x}}) - (\mathcal{K}'_\sigma u)(\tilde{\boldsymbol{x}}) + (\tilde{\mathcal{N}}'_\sigma f)(\tilde{\boldsymbol{x}}) \quad \forall \tilde{\boldsymbol{x}} \in \Omega. \tag{2.18}$$

The subscripts σ and ε stand for stresses or strains in question.

The evaluation of stresses of points $\Omega \ni \tilde{\boldsymbol{x}} \to \boldsymbol{x} \in \Gamma$ need special treatment since, Tr is applied. Due to the aforementioned application of $\mathcal{B}$, the singularity-order k of the integral kernels has been increased. Thus, the kernel of $\mathcal{K}'$ is hyper-singular. Moreover, the strain and stress field can be discontinuous, i.e at corners and edges. The mentioned integral operator in the BIE of the second kind exists as a Hadamard finite part [95], whose evaluation is possible with different approaches. Furthermore, formulations which involve only strongly singular integral kernels by shifting derivatives to the displacement field

exist [17]. A short, but not complete comparison of different approaches working out hypersingular integrals for the stress evaluation on the boundary can be found in [111]. To overcome the rather cumbersome treatment with this problem, a simple *stress recovery* procedure based on derivatives of surface displacements u and the direct use of surface tractions t can be used carefully [12].

As seen in this section, it is possible with boundary integral equations to deal with homogeneous boundary value problems with $f = 0$ in an elegant way. If the considered domain is subject to body forces $f \neq 0$, an inhomogeneous boundary value problem needs to be worked out. In special cases, i.e. if f is a harmonic or constant function, it is possible to convert the domain integrals into boundary integrals [17]. If the domain Ω_p subject to body forces is limited and f possibly non-smooth as well, one needs to find particular solutions to do so. This can be achieved with finite elements [99] or with globally defined interpolation functions such as radial basis functions. In relation the DRM has been a topic of many publications (see [83] and references therein). The RIM follows a similar approach. Unfortunately, both methods are not directly applicable to infinite or semi-infinite domain problems.

2.3 Elasto-Plastic Problems

The basic concept of small strain plasticity has been introduced in section 1.3. The remainder of this chapter is devoted to going through the noteworthy points for two and three-dimensional problems while keeping the notation consistent. As noted in section 1.2, there are many different approaches with the BEM treating elasto-plastic problems. Most of them rely on internal point evaluations as described in section 2.2 and the corresponding incorporation of the "plastic term" in the integral equations by body force terms. In this work, a direct method with integration over the plastic region is employed.

Due to the nonlinearity, a solution for arbitrary domains is not possible in closed form. Therefore, the problem is solved incrementally in terms of linearized pseudo-time steps $n \in \{0,\dots,N\}$ with $\dot{t} = t_{n+1} - t_n$ as described for the one-dimensional problem in section 1.3. Here, the over-dot $\dot{(\cdot)}$ denotes the derivative with respect to pseudo-time, therefore the difference of a quantity between the steps n and $n+1$ is called *rate*. The rate of plastic strain is usually discontinuous at the boundary of the plastic, inhomogeneous domain. Here, the domain subject to plasticity is considered to be $\Omega_p \subset \Omega$. The elastic relationship between incremental strains and stresses (2.1) only holds for the linear elastic case. Thus, the strain rate is composed of an elastic and plastic part $\dot{\varepsilon} = \dot{\varepsilon}_e + \dot{\varepsilon}_p$. As a result, the stress increment is expressed in terms of elastic and plastic contributions

$$\dot{\sigma}(\boldsymbol{x}) = \mathsf{C}(\lambda,\mu) : (\dot{\varepsilon}(\boldsymbol{x}) - \dot{\varepsilon}_p(\boldsymbol{x})) = \dot{\sigma}_e(\boldsymbol{x}) - \dot{\sigma}_p(\boldsymbol{x}). \tag{2.19}$$

The relations

$$\begin{aligned} \dot{\sigma}_e(\boldsymbol{x}) &= \dot{\sigma}(\boldsymbol{x}) + \dot{\sigma}_p(\boldsymbol{x}) = \mathsf{C}(\lambda,\mu) : \dot{\varepsilon}(\boldsymbol{x}) \\ \dot{\sigma}_p(\boldsymbol{x}) &= \mathsf{C}(\lambda,\mu) : \dot{\varepsilon}_p(\boldsymbol{x}) \end{aligned} \tag{2.20}$$

are considered using Betti's reciprocity theorem (2.9), which gives an additional integral over Ω_p in the representation formula (2.11). This yields to the incremental form

$$\dot{u}(\boldsymbol{x}) = \int_\Gamma \mathsf{U}(\boldsymbol{x},\boldsymbol{y})\dot{t}(\boldsymbol{y})\,\mathrm{d}s_{\boldsymbol{y}} - \int_\Gamma \mathsf{T}(\boldsymbol{x},\boldsymbol{y})\dot{u}(\boldsymbol{y})\,\mathrm{d}s_{\boldsymbol{y}} + \int_{\Omega_p} \mathsf{E}(\boldsymbol{x},\boldsymbol{z})\dot{\sigma}_p(\boldsymbol{z})\,\mathrm{d}\boldsymbol{z} \quad \boldsymbol{x}\in\Omega,\, \boldsymbol{z}\in\Omega_p \tag{2.21}$$

with the initial plastic stresses $\dot{\sigma}_p$ and the fundamental solution for strain $\mathsf{E}(\boldsymbol{x},\boldsymbol{y}) = \mathcal{B}\mathsf{U}(\boldsymbol{x},\boldsymbol{y})$. To improve legibility, the body forces f are neglected from now on. However, $f \neq 0$ can be easily considered by the formulation derived in the following. Also, initial stains $\dot{\varepsilon}_p$ (2.21) could be used instead by taking

$$\dot{u}(\boldsymbol{x}) = \int_\Gamma \mathsf{U}(\boldsymbol{x},\boldsymbol{y})\dot{t}(\boldsymbol{y})\,\mathrm{d}\Gamma - \int_\Gamma \mathsf{T}(\boldsymbol{x},\boldsymbol{y})\dot{u}(\boldsymbol{y})\,\mathrm{d}\Gamma + \int_{\Omega_p} \mathsf{S}(\boldsymbol{x},\boldsymbol{z})\dot{\varepsilon}_p(\boldsymbol{z})\,\mathrm{d}\Omega \quad \boldsymbol{x}\in\Omega,\, \boldsymbol{z}\in\Omega_p$$

with $\mathsf{S}(\boldsymbol{x},\boldsymbol{y}) = \mathcal{S}\mathsf{U}(\boldsymbol{x},\boldsymbol{y})$. Now the Newton operators

$$(\mathcal{N}_E\sigma)(\boldsymbol{x}) = \int_\Omega \mathsf{E}(\boldsymbol{x},\boldsymbol{z})\sigma(\boldsymbol{z})\,\mathrm{d}\boldsymbol{z} \quad \forall \boldsymbol{x}\in\Gamma$$
$$(\bar{\mathcal{N}}_S\varepsilon)(\boldsymbol{x}) = \int_\Omega \mathsf{S}(\boldsymbol{x},\boldsymbol{z})\varepsilon(\boldsymbol{z})\,\mathrm{d}\boldsymbol{z} \quad \forall \boldsymbol{x}\in\Gamma$$

and the corresponding volume operators for the strain and stress evaluation

$$(\mathcal{N}'_\varepsilon\sigma)(\boldsymbol{x}) = \mathcal{B}(\mathcal{N}\sigma)(\boldsymbol{x}) \quad \forall \boldsymbol{x}\in\Omega_p$$
$$(\mathcal{N}'_\sigma\sigma)(\boldsymbol{x}) = \mathcal{S}(\mathcal{N}\sigma)(\boldsymbol{x}) \quad \forall \boldsymbol{x}\in\Omega_p$$

are defined. By setting $\mathcal{N} = \mathcal{N}_E$ for better readability it follows the incremental BIE with initial stress

$$(\mathcal{K}\dot{u})(\boldsymbol{x}) = (\mathcal{V}\dot{t})(\boldsymbol{x}) + (\mathcal{N}\dot{\sigma}_p)(\boldsymbol{x}) \quad \forall \boldsymbol{x}\in\Gamma \tag{2.22}$$

and the formula for the incremental internal strains

$$\dot{\varepsilon}(\tilde{\boldsymbol{x}}) = (\mathcal{V}'_\varepsilon\dot{t})(\tilde{\boldsymbol{x}}) - (\mathcal{K}'_\varepsilon\dot{u})(\tilde{\boldsymbol{x}}) + (\mathcal{N}'_\varepsilon\dot{\sigma}_p)(\tilde{\boldsymbol{x}}) \quad \forall \tilde{\boldsymbol{x}}\in\Omega \tag{2.23}$$

and stresses

$$\dot{\sigma}(\tilde{\boldsymbol{x}}) = (\mathcal{V}'_\sigma\dot{t})(\tilde{\boldsymbol{x}}) - (\mathcal{K}'_\sigma\dot{u})(\tilde{\boldsymbol{x}}) + (\mathcal{N}'_\sigma\dot{\sigma}_p)(\tilde{\boldsymbol{x}}) \quad \forall \tilde{\boldsymbol{x}}\in\Omega. \tag{2.24}$$

The equations for initial strains can be derived in a similar way.

Linearization In the following paragraph, the solution strategy for the non-linear system of equations is described. Due to the history dependencies the given Cauchy data

$$g = \sum_{s=0}^{S} g_s = \sum_{s=0}^{S} \omega_s g$$

can be subdivided into load steps s such that they are weighted by a factor ω_s. As a result the problem is linearized and the field quantities are accumulated after every load step. Exemplarily, we consider an exterior Neumann problem $\Gamma = \Gamma_N$ (2.13) with a domain subject to plastic behavior Ω_p. Figure 2.3 shows a sketch of such a problem in tunneling.

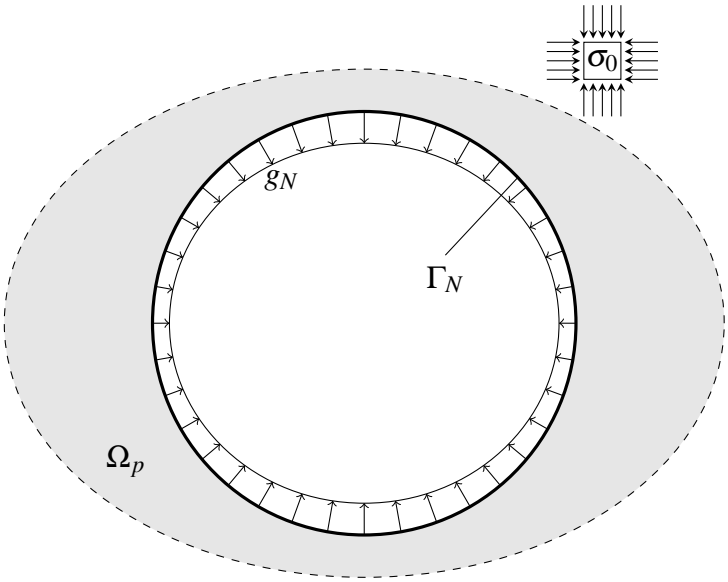

Figure 2.3: Exterior Neumann problem with a potentially plastic region Ω_p for a tunnel excavation

For excavation, the given Neumann data g_N are derived from the known virgin stress state σ_0 and the unit outward normal to the boundary. With the unknown displacements on the left hand side and the given tractions on the right, the boundary integral equation is

$$(\mathcal{K}\dot{u})(\boldsymbol{x}) = (\mathcal{V}\dot{g}_N)(\boldsymbol{x}) + (\mathcal{N}\dot{\sigma}_p)(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \Gamma_N. \tag{2.25}$$

The subscript denoting the load steps are omitted with $g_N := g_{N,s}$ and $u := u_s$. Since $\mathcal{K}$ is invertible for exterior problems [95], the uniquely solvable boundary integral equation (2.25) is reformulated in a way such that the unknowns u are inserted in the incremental form of (2.18) resulting in

$$\dot{\sigma}(\tilde{\boldsymbol{x}}) = (\mathcal{V}'_\sigma \dot{g}_N)(\boldsymbol{x}) - \mathcal{K}'_\sigma \left((\mathcal{K}^{-1}\mathcal{V}\dot{g}_N)(\boldsymbol{x}) + (\mathcal{K}^{-1}\mathcal{N}\dot{\sigma}_p)(\tilde{\boldsymbol{x}}) \right) + (\mathcal{N}'_\sigma \dot{\sigma}_p)(\tilde{\boldsymbol{x}}). \tag{2.26}$$

Separating boundary and domain terms and with the substitutions

$$\dot{b}_\sigma = \left(\mathcal{V}'_\sigma - \mathcal{K}'_\sigma \mathcal{K}^{-1}\mathcal{V}\right)\dot{g}_N \quad \text{and}$$
$$\mathcal{W}_\sigma = \mathcal{K}^{-1}\mathcal{N} + \mathcal{N}'_\sigma.$$

equation (2.26) yields to

$$\dot{\sigma}(\tilde{\boldsymbol{x}}) = \dot{b}_\sigma(\boldsymbol{x}) + (\mathcal{W}_\sigma \dot{\sigma}_p)(\tilde{\boldsymbol{x}}) \tag{2.27}$$

for the stress rate in $\tilde{\boldsymbol{x}} \in \Omega_p$. At this point the spatial variables $\boldsymbol{x}$, $\tilde{\boldsymbol{x}}$ and $\boldsymbol{z}$ are omitted. For equilibrium (2.27) needs to be fulfilled. The nonlinearity is induced by the RRA which maps from an possibly inadmissible trial state to an admissible state on the yield surface between steps n and $n+1$. This mapping is expressed in terms of the function

$$\bar{\sigma}(\sigma_n, \varepsilon_n, \dot{\varepsilon}_n, h_n) \mapsto (\sigma_{n+1}, h_{n+1}). \tag{2.28}$$

Here, h denotes the set of parameters needed for the RRA, i.e. hardening and history parameters as well as the Lamé constants μ and λ. The values $\sigma_n, \varepsilon_n, \dot{\varepsilon}_n, h_n$ at step n are known. With the relations (2.20) and $\dot{\sigma}_e$ denoting the elastic trial state, the residual plastic stress is $\dot{\sigma}_p = \dot{\sigma}_e - \dot{\sigma}$ where $\dot{\sigma}$ is determined by (2.28). Using these relations, a residuum

$$\mathsf{R}(\dot{\sigma}_e) := \mathcal{W}_\sigma\left(\dot{\sigma} - \dot{\sigma}_e\right) - \dot{b}_\sigma + \dot{\sigma} \tag{2.29}$$

is specified [17, 106]. Now, equilibrium is found iteratively by updating the elastic trial stress until $\mathsf{R}(\dot{\sigma}_e) = 0$. This is done using Newton's method

$$\mathsf{R}(\dot{\sigma}_e) + \mathcal{D}\mathsf{R}(\dot{\sigma}_e)\Delta\dot{\sigma}_e = 0 \tag{2.30}$$

where the differential operator $\mathcal{D}$ contains the partial derivative of the components of $\dot{\sigma}$ and results to the *Jacobi-matrix* $\mathsf{J}_i(\dot{\sigma}_e) = \mathcal{D}\mathsf{R}(\dot{\sigma}_e)$ with

$$\mathsf{J}_i(\dot{\sigma}_e)[r,c] = \frac{\partial \mathsf{R}(\dot{\sigma}_e)_r}{\partial(\dot{\sigma}_e)_c}$$

and the Newton iteration step i. In this work, the solution is sought using a (simplified) *modified Newton's method* [81] with a constant, non-updated J_0. Since the original method for problems of this type exhibits quadratic convergence, the modified version is less powerful. Nevertheless, the modified method converges to the right solution in that case.

To address the convergence rate, enhanced solution procedures have been published. I.e. equation (2.27) is the foundation for implicit solution procedures as explained in [106]. In [3] and [39] a better convergence is reached solving for the plastic multiplier (see section 1.3). However, this implies a incorporation of the yield criterion into the algorithm. This is not acceptable for the intended application of this work. Finally, the approach of Bonnet and Mukherjee [19] which incorporates the consistant tangential operator (CTO) provided by many yield criteria, can be applied at this place with the following implica-

tions. As a result, optimal quadratic complexity can be achieved. Therefore, (2.29) is rewritten to provide residual strains

$$\mathsf{R}(\dot{\varepsilon}) := \mathcal{W}_{\varepsilon}\left(\bar{\sigma}(\sigma_n, \varepsilon_n, \dot{\varepsilon}_n, h_n) - \sigma_n - \mathsf{C}(\mu, \lambda) : \dot{\varepsilon}\right) - \dot{b}_{\varepsilon} + \dot{\varepsilon} \tag{2.31}$$

using the relations (2.20) and the substitutions

$$\dot{\sigma} = \sigma_{n+1} - \sigma_n \quad \text{and} \quad \sigma_{n+1} = \bar{\sigma}(\sigma_n, \varepsilon_n, \dot{\varepsilon}_n, h_n). \tag{2.32}$$

By applying $\mathcal{D}$ to equation (2.31), $\mathsf{J}(\dot{\varepsilon})$ becomes

$$\mathsf{J}(\dot{\varepsilon}) = \mathcal{W}_{\varepsilon}\left(\mathsf{C}_{n+1} - \mathsf{C}(\mu, \lambda)\right) + \mathcal{I} \tag{2.33}$$

with the local CTO

$$\mathsf{C}_{n+1} = \frac{\partial \bar{\sigma}(\sigma_n, \varepsilon_n, \dot{\varepsilon}_n, h_n)}{\partial \dot{\varepsilon}}. \tag{2.34}$$

Merging the continuously updated equations (2.33) and (2.30) together, the additive correction $\Delta\dot{\varepsilon}$ is calculated with

$$-\mathsf{J}(\dot{\varepsilon})\Delta\dot{\varepsilon} = \mathsf{R}(\dot{\varepsilon}) \tag{2.35}$$

until equilibrium $\mathsf{R}(\dot{\varepsilon}) = 0$ is reached. It is remarkable that for this approach a rate-formulation is not necessary any more since the iterative process is carried out within (2.35). As seen in (2.31) and (2.34), the RRA can be used as a black box providing valid strain-stress states and the CTO. Hence, this formulation works with many different material models without any modification.

3 Boundary Element Method

The only impreciseness introduced so far is based on the underlying model description of the PDE, which may be wrong, i.e. for materials subject to large deformations, and the linearization of the nonlinear system of equations for problems in plasticity. It is clear that the integral equations presented in chapter 2 are not analytically solvable for arbitrary domains. Hence, the geometry as well as the field values are subject to approximations. Therefore, the boundary of the domain is subdivided into boundary elements. They give rise to the name of the numerical method described in this chapter: the boundary element method. At this point, the discretization of the underlying equations is introduced as well as the strategy for the solution of mixed boundary value problem in elasticity and elasto-plasticity.

3.1 Discretization

The parametrization of the boundary Γ as well as the zone subject to body forces Ω_p is based on a mesh. This is a geometrical partition of the treated boundary or domain. Even though methods exist with mesh refinement based on estimated approximation errors, the mesh is kept constant during the analysis in the present work.

Spatial Discretization The approximations Γ_h and $\Omega_{p,h}$ of boundary and region with body forces are given by triangulation creating disjoint sets of *boundary elements* τ and finite *cell elements* ϖ

$$\Gamma \approx \Gamma_h = \bigcup_{e=1}^{E} \tau_e \quad \text{and} \quad \Omega \approx \Omega_h = \bigcup_{c=1}^{C} \varpi_c.$$

Each element is occupied by two or more, globally defined nodes $\boldsymbol{p}_i \in \mathbb{R}^d$. The set of I corresponding global node numbers $P_e = \{p_1, \ldots, p_i, \ldots p_I\}$ is called element incidences where the subscript i denotes the corresponding local node number of element e. Since the boundary elements are two-dimensional surface patches in three dimensions or line patches in two, the function

$$\mathcal{X}_\tau(\xi_1, \ldots, \xi_{d-1}) : \mathbb{R}^{d-1} \mapsto \mathbb{R}^d \tag{3.1}$$

is a coordinate transformation mapping local coordinates ξ_i of the reference element $\hat{\tau} = (-1,1)^{d-1}$ to the global coordinates of $\tau = (a,b)^{d-1}$. In the following, $\boldsymbol{\xi}$ stands for all

intrinsic local element coordinates. The simplest element shape in three dimensions is a triangle and is considered to be completely flat. Hence, the outward normal $\mathbf{n}$ is constant all over the element and $\mathcal{X}_\tau$ is linear. A four-point quadrilateral element is described by a bilinear $\mathcal{X}_\tau$ and is flat along the edges but can have curved diagonals. For quadratic description of the geometry in three dimensions, six-node triangular or eight-node quadrilateral elements are used. In two dimensions the linear and quadratic description is provided by a two- or three-node line elements respectively.

For the domain discretization with finite cell elements, the mapping function

$$\mathcal{X}_\varpi(\xi_1,\ldots,\xi_d) : \mathbb{R}^d \mapsto \mathbb{R}^d$$

is used. Generally, the geometry and therefore any point $\boldsymbol{x}$ on elements τ or ϖ are described by

$$\boldsymbol{x} = \mathcal{X}(\boldsymbol{\xi}) = \sum_{i=1}^{I} \varphi_i(\boldsymbol{\xi})\boldsymbol{p}_i \tag{3.2}$$

with the shape functions φ_i depending on the local coordinates $\boldsymbol{\xi}$. In the context of the BEM, cells are only used for integration over a volume. Hence, points of cells ϖ touching the boundary do not necessarily need to coincide with points of elements τ for the boundary approximation. The concept of boundary approximation can be found in [41] and other related textbooks. Details on the straight forward domain discretization in context with the BEM are pointed out in [12].

Field Variables Still to be completed is the approximation of the Cauchy data or other field values on the reference elements $\hat{\tau}$ and $\hat{\varpi}$ in terms of the local coordinates $\boldsymbol{\xi}$. With (3.1) in hand, every function on in the parameter space of $\hat{\tau}$ and $\hat{\varpi}$ can be transformed to the Cartesian coordinate system. For the approximation of displacements and tractions the idea of Rüberg and Schanz [93] is adopted. Subsequently, the displacements u are approximated by locally and piecewise defined C^0-continuous trial functions

$$\mathbf{u}_e(\boldsymbol{\xi}) = \sum_{i=1}^{M} \varphi_i(\boldsymbol{\xi})\mathbf{u}_i(\boldsymbol{\xi_i}) \quad \varphi \in S_h \tag{3.3}$$

for M support points $\boldsymbol{\xi_i}$ and $\varphi_i(\boldsymbol{\xi_i}) = 1$. The space of continuous and discontinuous trial functions on the discretized boundary is denoted by

$$S_h(\Gamma_h) = \operatorname{span}\{\varphi_e\}_{e=1}^{E} \quad \text{and} \quad S_h^-(\Gamma_h) = \operatorname{span}\{\psi_e\}_{e=1}^{E}.$$

For the surface tractions t trial functions ψ need to have a continuity of class C^{-1} since t can have jumps across element-corners and edges or at points where known values are

discontinuous. Independent of the displacement discretization

$$\mathbf{t}_e(\boldsymbol{\xi}) = \sum_{i=1}^{N} \psi_i(\boldsymbol{\xi}) \mathbf{t}_i(\boldsymbol{\xi_i}) \quad \psi \in S_h^-. \tag{3.4}$$

To evaluate displacements or surface tractions in Cartesian coordinates, the only thing left to do is use (3.2), which results in $\mathbf{u}_e(\boldsymbol{x}) = \mathbf{u}_e(\mathcal{X}(\boldsymbol{\xi}))$ and $\mathbf{t}_e(\boldsymbol{x}) = \mathbf{t}_e(\mathcal{X}(\boldsymbol{\xi}))$. The same concept is applied to the other field variables f, ε and σ, i.e.

$$\boldsymbol{\sigma}_e(\boldsymbol{\xi}) = \sum_{i=1}^{L} \theta_i(\boldsymbol{\xi}) \boldsymbol{\sigma}_i(\boldsymbol{\xi_i}), \tag{3.5}$$

again with trial functions in S_h and S_h^- chosen with respect to the type of data. If the trial functions for the geometry and the field values are chosen to be the same, the elements are called isoparametric [65].

Remark: In the implementation used for this work, the user can choose to have an isoparametric ansatz. However, for mixed boundary conditions or Dirichlet problems this will distract the application of the fast solution technique described in chapter 4. Additionally and with no restriction, the interpolation of all field values can be discontinuous. This is possible even for displacements.

Infinite Elements As mentioned in section 2.1 the representation of an infinite or semi-infinite domain is crucial for problems in tunneling or geotechnics. Infinite domains are covered by the radiation condition (2.14) and are inherent in the formulation. For the modeling of earth's surface or a tunnel of infinite extent as well as the description of a fault zone, infinite elements with nodes pointing to infinity and suitable decay functions can be used. For tunnels, the displacements and tractions are constant towards infinity. For half-spaces the displacements decay with $\mathcal{O}(R^{-1})$ and the tractions with $\mathcal{O}(R^{-2})$. Originating from the work of Bettess [15] for the FEM, Beer and Watson [11] developed infinite elements for the BEM solving semi-infinite problems in tunneling and geomechanics. Elements based on that as well as on [74] have been used in this work. Similar and yet improved techniques are found in [29] and [38]. Alternatively, Telles and Brebbia [107] formulated a fundamental solution for traction free half-space problems without the need of elements on the surface. However, since the simulation of structures and buildings should be possible, elements on the surface are needed anyway for a coupled formulation.

3.2 System of Equations

The triangulation allows the approximation of any boundary or volume integral containing an arbitrary kernel $\Phi(\boldsymbol{x}, \boldsymbol{y})$ and field variable b by a finite sum of integrals over boundary

elements

$$\int_{\Gamma} \Phi(\boldsymbol{x},\bar{\boldsymbol{y}}) b(\bar{\boldsymbol{y}}) \,\mathrm{d}s_{\boldsymbol{y}} \approx \sum_{e=1}^{E} \int_{\tau_e} \Phi(\boldsymbol{x},\boldsymbol{y}) b(\boldsymbol{y}) \,\mathrm{d}s_{\boldsymbol{y}} \quad \forall \bar{\boldsymbol{y}} \in \Gamma, \forall \boldsymbol{y} \in \Gamma_h \tag{3.6}$$

or cell elements

$$\int_{\Omega} \Phi(\boldsymbol{x},\bar{\boldsymbol{z}}) b(\bar{\boldsymbol{z}}) \,\mathrm{d}\boldsymbol{z} \approx \sum_{c=1}^{C} \int_{\varpi_c} \Phi(\boldsymbol{x},\boldsymbol{z}) b(\boldsymbol{z}) \,\mathrm{d}\boldsymbol{z} \quad \forall \bar{\boldsymbol{z}} \in \Omega_p, \forall \boldsymbol{z} \in \Omega_{p,h} \tag{3.7}$$

Since only the first BIE is used, a collocation scheme is applied for the discretization of (2.16). This means that the residual of the approximation is minimized pointwise [41]. In order to solve a system of equations, as many *collocation points* $\boldsymbol{x} \in \Gamma_h$ as unknowns are needed. This is determined by the mesh size and supports of trial functions φ and ψ for the unknown Cauchy data. Hence, the sums in (3.6) and (3.7) can be expressed in terms of a matrix-vector product. For the evaluation of internal results $\tilde{\boldsymbol{x}} \in \Omega_{p,h}$ relate to strain or *stress points* depending on the value to be calculated. On account of the elasto-plastic problem, only stresses need to be determined at internal points from now on. The procedure for the evaluation of internal displacements and strains is easily derived in the following statements.

Block System of Matrices As described in section 2.1 the discrete boundary is split into a Dirichlet and a Neumann part $\Gamma_h = \Gamma_{N,h} \cup \Gamma_{D,h}$. On $\Gamma_{N,h}$ the collocation points are chosen to be at the support points of the displacement trial functions φ. In case of interpolation with trial functions in S_h these points coincide at least with edge nodes of the geometry description. On $\Gamma_{D,h}$ the collocation points are located inside the elements on the corresponding support points of trial functions in S_h^-. This prevents redundancies and a rank deficient system of equations. The body forces, e.g. initial strains and stresses inside the domain are usually chosen to be discontinuous. Thus, the corresponding points are inside the cells and located at the support points of θ. In Figure 3.1, an example for linear displacement, constant tractions and constant stresses is given. Squared points mark the collocation and stress points; circular nodes represent geometry nodes. With respect to the location of the collocation points the known Cauchy data and body forces are moved to the right hand side of the equation. Putting the unknown Cauchy data to the left hand side, the discretization of (2.16) leads to the block system of equations

$$\begin{pmatrix} \mathbf{V}_{DD} & -\mathbf{K}_{DN} \\ \mathbf{V}_{ND} & -\mathbf{K}_{NN} \end{pmatrix} \begin{pmatrix} \mathbf{t}_D \\ \mathbf{u}_N \end{pmatrix} = \begin{pmatrix} \mathbf{K}_{DD} & -\mathbf{V}_{DN} \\ \mathbf{K}_{ND} & -\mathbf{V}_{NN} \end{pmatrix} \begin{pmatrix} \mathbf{g}_D \\ \mathbf{g}_N \end{pmatrix} + \begin{pmatrix} \mathbf{N}_{DP} \\ \mathbf{N}_{NP} \end{pmatrix} \boldsymbol{\sigma}_p \tag{3.8}$$

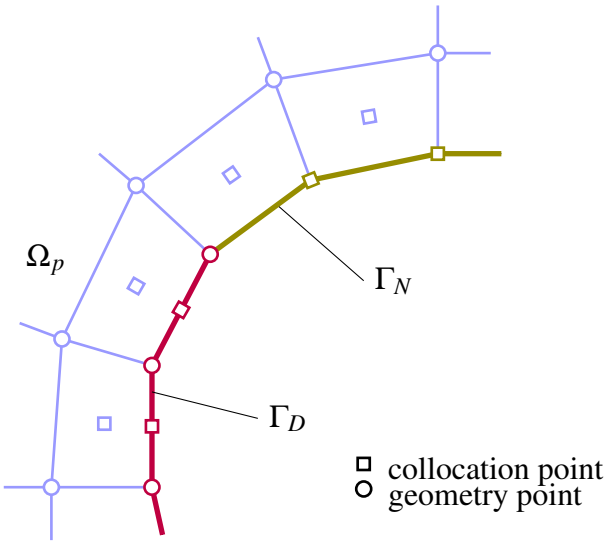

Figure 3.1: Discretized part of a boundary and collocation points with linear ansatz for u and constant ansatz for t and σ

where the entries of sub-matrices are

$$\begin{aligned}
\mathbf{V}_{RC}[r,c] &= \mathcal{V}\psi_c(\boldsymbol{x}_r) \qquad \forall r \in \{1,\ldots,\#(\boldsymbol{x}_r \in \Gamma_{R,h})\} \\
\mathbf{K}_{RC}[r,c] &= \mathcal{K}\varphi_c(\boldsymbol{x}_r) \qquad \forall c \in \{1,\ldots,\#(\psi_c,\, \varphi_c \in \Gamma_{C,h})\} \\
\mathbf{N}_{RC}[r,m] &= \mathcal{N}\varphi_c(\boldsymbol{x}_r) \qquad \forall m \in \{1,\ldots,\#(\theta_c \in \Omega_{p,h})\}
\end{aligned} \tag{3.9}$$

at position (R,C) and denote the discretized single layer, double layer and Newton potential [91]. The first subscript of a sub-matrix denotes the location of the collocation point, and the second that of the element to be integrated over. This means $\mathbf{V}_{ND}$ is the single layer potential for collocation points $\boldsymbol{x}_r \in \Gamma_{N,h}$ and trial functions $\psi_c \in \Gamma_{D,h}$. Therefore, collocation points in the first equation in (3.8) are located on $\Gamma_{D,h}$, on $\Gamma_{N,h}$ in the second. The entries of vectors for Cauchy data and the stresses are

$$\begin{aligned}
\mathbf{t}_D[c] &= t(\boldsymbol{y}_c) \qquad \forall \boldsymbol{y} \in \Gamma_{D,h} \\
\mathbf{g}_N[c] &= t(\boldsymbol{y}_c) \qquad \forall \boldsymbol{y} \in \Gamma_{N,h} \\
\mathbf{u}_N[c] &= u(\boldsymbol{y}_c) \qquad \forall \boldsymbol{y} \in \Gamma_{N,h} \\
\mathbf{g}_D[c] &= u(\boldsymbol{y}_c) \qquad \forall \boldsymbol{y} \in \Gamma_{D,h} \\
\boldsymbol{\sigma}_p[m] &= \sigma_p(\boldsymbol{z}_m) \qquad \forall \boldsymbol{z} \in \Omega_{p,h}
\end{aligned} \tag{3.10}$$

with $c = 1,\ldots,C$, where C is the number of support points of trial functions on the corresponding boundary. Because of the tensor type of the fundamental solutions, matrix entries in (3.9) are understood to be matrix valued by itself. Entries in (3.10) are vector valued i.e. $\mathbf{u}_A[r] = [u(\boldsymbol{y}_c)_i]_{i=1}^d$ or $\boldsymbol{\sigma}_p[m] = [\sigma_p(\boldsymbol{z}_m)_i]_{i=1}^s$ with the number of pseudo stresses s.

No distinction with respect to its location is made for points $\tilde{\boldsymbol{x}} \in \Omega_{p,h}$. Thus, the discrete equation for internal stresses (2.18) reads

$$\boldsymbol{\sigma} = \mathbf{V}'\mathbf{t} - \mathbf{K}'\mathbf{u} + \mathbf{N}'\boldsymbol{\sigma}_p \tag{3.11}$$

also with discrete operators $\mathbf{V}'$, $\mathbf{K}$' and $\mathbf{N}'$ corresponding to (2.18). The entries of sub-matrices and vectors are defined similar to (3.9) and (3.10).

The condition number of the sub-matrices in the first row of (3.8) depends on the inter-element location of $\boldsymbol{x}$. It is suggested in [91] that neighboring points should be located as far away as possible from each other and the optimal points for line elements and triangles are given therein. In [68] and [82], surveys on discontinuous and continuous elements have been done. In the present work the support points for discontinuous shape functions are chosen to be element midpoints and at $\xi = \pm 0.8$ for line elements. For quadrilaterals, the locations result from the tensor product of that coordinates.

Evaluation of Matrix Entries Even though analytic expressions for integrals over boundary elements τ [89] and cells ϖ exist, taking Gaussian quadrature [17] for regular integrals and special analytic procedures for singular integrals has become state of the art. For integration resulting in entries of (3.9), the integral over τ is transformed to an integral over $\hat{\tau}$. For example entries of the single layer potential are calculated by means of

$$\mathbf{V}[r,c] = \int_{\tau_e} \mathsf{U}(\boldsymbol{x}_r, \boldsymbol{y})\psi_c(\boldsymbol{y})\,\mathrm{d}\,\mathrm{s}_{\boldsymbol{y}} = \int_{\hat{\tau}} \mathsf{U}(\boldsymbol{x}_r, \mathcal{X}_\tau(\boldsymbol{\xi}))\psi_c(\boldsymbol{\xi})\sqrt{g_\tau(\boldsymbol{\xi})}\,\mathrm{d}\,\boldsymbol{\xi} \tag{3.12}$$

involving the coordinate transformation (3.2) and Gram's determinant [95]

$$g_\tau(\boldsymbol{\xi}) = \det\left(\mathbf{J}_\tau^\intercal(\boldsymbol{\xi})\mathbf{J}_\tau(\boldsymbol{\xi})\right).$$

The Jacobi matrix is determined by

$$\mathbf{J}(\boldsymbol{\xi}) = \mathcal{D}\mathcal{X}_\tau(\boldsymbol{\xi}) \quad \text{where} \quad \mathbf{J}(\boldsymbol{\xi})[i,j] = \frac{\partial x_i}{\partial \xi_j} \quad \forall i,j \in \{1,\ldots,d-1\}.$$

Now the integral is ready for the Gaussian quadrature

$$\mathbf{V}[r,c] \approx \sum_{g=1}^{G^*} \mathsf{U}(\boldsymbol{x}_r, \mathcal{X}_\tau(\boldsymbol{\xi}_g))\psi_c(\boldsymbol{\xi}_g)\sqrt{g_\tau(\boldsymbol{\xi}_g)}w_g$$

with the coordinates of integration points $\boldsymbol{\xi}_g$ and the corresponding weights w_g. The total number of Gauss points $G^* = G^{d-1}$, with respect to predefined accuracy, is determined by an empiric integration rule [34], which yields the number of integration points G in every local direction. The same procedure is done for all surface integrals. If G is too high, the element to be integrated over is hierarchically subdivided into smaller elements and again

treated by the criterion. If trial functions involved in $\mathbf{K}$ and $\mathbf{K}'$ are in S_h, matrix valued entries are collected from all neighboring elements of the support point and assembled together. For domain integrals involved in $\mathbf{N}$ and $\mathbf{N}'$, the procedure is almost the same except that their intrinsic dimension is equal to d. This has impact on the Gaussian quadrature $G^* = G^d$ and on Gram's determinant.

Special care has to be taken if the collocation point $\boldsymbol{x}$ is located on the element to be integrated over, since the involved kernels are singular when the distance r between $\boldsymbol{x}$ and integration point $\boldsymbol{y}$ in (2.12) tends to zero. Hence, the integrant diverges to infinity. Integrals of weakly singular kernels, i.e. diagonal entries of $\mathbf{V}$ are called improper, and their values are finite. Thus, special techniques are necessary to tackle this problem and the proposal of Lachat and Watson [65] is implemented. More complex is the integration of strongly singular kernels such as in the diagonals in $\mathbf{K}$. They exist as a Cauchy principal value. Several evaluation techniques exist, i.e. the general strategy of Guiggiani and Gigante [49], which has been taken for this work. Note that the integral free jump term (2.15) is already included in the diagonal entries of $\mathbf{K}$ on the left hand side of (3.8). Due to intended collocation points on $\Gamma_{D,h}$ and the fact that trial functions for the approximation of $\mathbf{g}_D$ are in S_h, the jump term on the right hand side is distributed to continuous nodes for entries in $\mathbf{K}_{DD}$.

For details on numerical treatment of any integral, the reader is referred to [64]. A good introduction of the integration technique for collocation BEM in elastostatics is given in [91], in [12] for elasto-plastic problems. The treatment of singular domain integrals in this context has been presented in [39].

3.3 Equation Solver

The system matrices presented in (3.8) are fully populated and not symmetric. That's why the application of solvers to the system of equations is more complex. Two different approaches are usually found in applications with BEM: direct solvers by means of factorizations and iterative solvers, basically Krylov subspace methods [100]. In the present formulation, these solvers can be applied to the whole system of equations or treated by means of a Schur complement. However, if Γ_D or Γ_N compared to Γ are small, the former approach has been applied in this work.

LU Factorization In section 2.3 the non-linear system of equation for elasto-plastic problems and its linearization has been introduced. Since the involved system matrices for the applied explicit approach never change, a one-off block *LU* factorization [44] might be affordable. This depends on the size n of the problem and on the iterative procedure for the plastic algorithm. In that case, the number of changes of the right hand side is defined

by the number of iterations for the algorithm to converge. The block factorization is

$$\mathbf{M} = \begin{pmatrix} \mathbf{V}_{DD} & -\mathbf{K}_{DN} \\ \mathbf{V}_{ND} & -\mathbf{K}_{NN} \end{pmatrix} = \mathbf{LU} = \begin{pmatrix} \mathbf{L}_{11} & 0 \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{U}_{11} & \mathbf{U}_{12} \\ 0 & \mathbf{U}_{22} \end{pmatrix} \tag{3.13}$$

with

$$\begin{aligned} \mathbf{V}_{DD} &= \mathbf{L}_{11}\mathbf{U}_{11}, \\ -\mathbf{K}_{DN} &= \mathbf{L}_{11}\mathbf{U}_{12}, \\ \mathbf{V}_{ND} &= \mathbf{L}_{21}\mathbf{U}_{11} \qquad \text{and} \\ -\mathbf{K}_{NN} &= \mathbf{L}_{21}\mathbf{U}_{12} + \mathbf{L}_{22}\mathbf{U}_{22}. \end{aligned} \tag{3.14}$$

The calculations are carried out in a three step procedure where first a classical *LU* factorization of $\mathbf{V}_{DD}$ is carried out. Second, with the resulting factors the matrices $\mathbf{U}_{12}$ and $\mathbf{L}_{21}$ are determined by means of forward- and backward substitutions. Finally, to obtain $\mathbf{L}_{22}\mathbf{U}_{22}$ the matrix $-\mathbf{K}_{NN} - \mathbf{L}_{21}\mathbf{U}_{12}$ is subject to a *LU* factorization again. With this representation the system of equations $\mathbf{Mx} = \mathbf{g}$ is solved for any given right hand side $\mathbf{g} = \begin{pmatrix} \mathbf{g}_D & \mathbf{g}_N \end{pmatrix}$ by means of block-wise forward and backward substitutions

$$\mathbf{Lz} = \mathbf{g} \quad \text{and} \quad \mathbf{Ux} = \mathbf{z}$$

with the result $\mathbf{x} = \begin{pmatrix} \mathbf{t}_D & \mathbf{u}_N \end{pmatrix}$.

This factorization is a $\mathcal{O}(n^3)$ process and hence not affordable for rather large system matrices. It will be seen in chapter 4, that with the there described matrix format the computational effort is reduced significantly.

Iterative Solvers However, most often iterative solvers are used since it decreases computational complexity by the order of one. That's because only matrix-vector products with the system matrix $\mathbf{M}$ are needed. For the fully populated and not symmetric matrices of (3.8) the generalized minimal residual (GMRES) [94] method or the biconjugate gradiend stabilized (BiCGSTAB) method [108] are suitable. Both are Krylov subspace methods. Usually, such kind of solvers try to minimize the residual in iteration step i

$$\mathbf{r}_i = \mathbf{Mx}_i - \mathbf{g}$$

by predicting the new approximate solution

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{p}_i$$

in step $i+1$ with a calculated search direction $\mathbf{p}$ and a relaxation parameter α. The convergence of these iterative solution procedures depends heavily on the condition number

$$\kappa(\mathbf{M}) == \frac{\lambda_{max,\mathbf{M}}}{\lambda_{min,\mathbf{M}}} = \|\mathbf{M}\|_2 \cdot \|\mathbf{M}^{-1}\|_2$$

of the system matrix. The condition of sub matrices in (3.8) mainly depend on n and the location of collocation points [93]. Most often, $\kappa \to \infty$ for $n \to \infty$ and therefore the number of iterations for GMRES and BiCGSTAB will increase with the matrix size. To avoid this, the system of equations needs to be preconditioned by a matrix $\mathbf{P}$ spectrally equivalent to $\mathbf{M}$ [100] so that

$$\mathbf{P}^{-1}\mathbf{M}\mathbf{x} = \mathbf{P}^{-1}\mathbf{g} \quad \text{with} \quad \mathbf{P}^{-1}\mathbf{M} \approx \mathbf{I} \tag{3.15}$$

in case of a left preconditioner. Finding proper preconditioners can be a cumbersome task. The most simple approach is taking $\mathbf{P} = diag(\mathbf{M})$ so that the application of $\mathbf{P}^{-1}$ correlates to a scaling of $\mathbf{M}$ by its diagonal entries. The interested reader is referred to [100] and [101] for different approaches in order to find reasonable preconditioners. In a nutshell, the objective is to find a preconditioner so that $\kappa(\mathbf{P^{-1}M})$ can be bound by a constant term independently of n. Figure 3.2(a) shows the convergence of an exterior Neumann problem in three dimensions by means of GMRES and BiCGSTAB solver and their diagonal preconditioned versions P-BiCGSTAB and P-GMRES.exemplarily. Generally, the

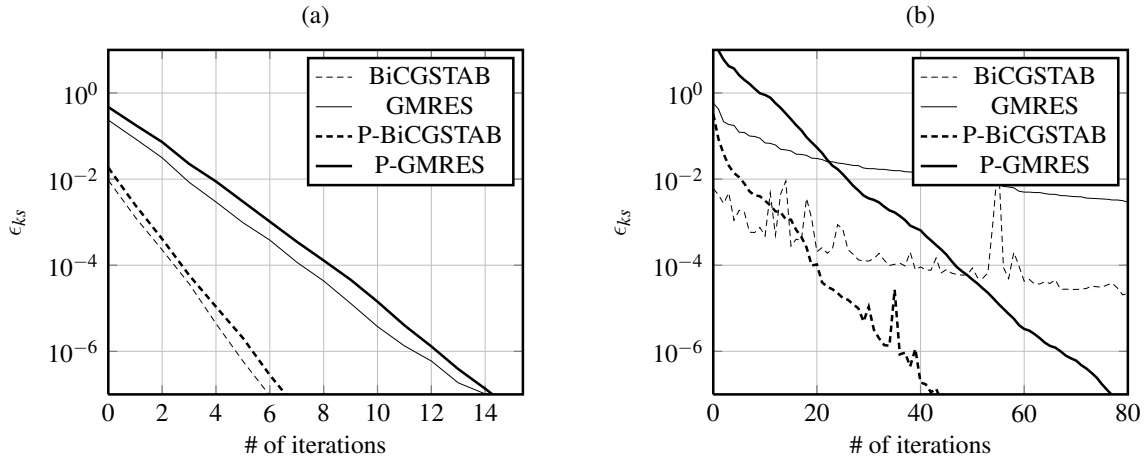

Figure 3.2: Convergence of iterative solvers for an exterior Neumann problem (a) and mixed boundary conditions (b)

double layer potential is better conditioned [30, 101] and a solution within a tolerance of $\epsilon_{ks} = 10^{-7}$ is found in several steps for $n \approx 12000$ degrees of freedom. Therefore, preconditioning is not necessary in this case. In Figure 3.2(b) the convergence of a problem with mixed boundary conditions is shown. Without taking a preconditioner, convergence

is possibly never achieved. By taking the previously described, rather simple diagonal preconditioner the behavior has been improved significantly.

However, the BiCGSTAB method converges faster in all examples, but shows disturbing oscillations due to its bivalent search directions. Without a preconditioner, the oscillations are rather disturbing. In contrary, GMRES does not show this behavior. Generally, the latter method is more robust and should at least converge in n iteration steps. A drawback is that GMRES needs to store its search directions temporarily and it might also reach a plateau in the convergence graph. The latter has been avoided by restarting the algorithm after several iteration steps. All in all, GMRES solver are considered to be robust for solving non-symmetric systems of equations, and is primarily used for most numerical examples in chapter 6.

Schur Complement The Schur complement of (3.8),

$$\mathbf{S} = \mathbf{V}_{ND}\mathbf{V}_{DD}^{-1}\mathbf{K}_{DN} - \mathbf{K}_{NN} \tag{3.16}$$

is found by substituting the first row into the second. It represents a Dirichlet-to-Neumann map which transforms surface displacements $\mathbf{u}_N$ on $\Gamma_{N,h}$ to tractions. From this substitution, the corresponding right hand side is

$$\mathbf{g}_S = \left(\mathbf{K}_{ND} - \mathbf{V}_{ND}\mathbf{V}_{DD}^{-1}\mathbf{K}_{DN}\right)\mathbf{g}_D + \left(\mathbf{V}_{ND}\mathbf{V}_{DD}^{-1}\mathbf{V}_{DN} - \mathbf{V}_{NN}\right)\mathbf{g}_N + \left(\mathbf{N}_N - \mathbf{V}_{ND}\mathbf{V}_{DD}^{-1}\mathbf{N}_D\right)\sigma_p.$$

The resulting system of equations

$$\mathbf{S}\mathbf{u}_N = \mathbf{g}_S \tag{3.17}$$

can be solved by direct solvers or by means of a two-step iterative procedure [101] where first the inner solution for the representation of (3.16) is sought and then the outer solution for the unknown displacements in (3.17). The unknown traction $\mathbf{t}_D$ is found by backward calculation after resolving $\mathbf{u}_D$. The Schur complement in terms of a Dirichlet-to-Neumann map is an important representation for coupling procedures (c.f. [92]). Regarding to that, coupled problems are solved without specifying the solution schemes taken for the subregions.

4 Hierarchical Matrices

As shown in section 3.2, the system matrices are fully populated and, in case of a collocation method, non-symmetric. Latter property influences the solution procedures to be chosen which will be described in chapter 5. However, major implications are self-evident from the fact that the required storage for full matrices scales quadratic with the problem size n. For a matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ with asymptotically $n \to \infty$ the required storage and the number of elementary[1] operations for a matrix vector product are

$$\mathsf{St}(n) = \mathcal{O}\left(n^2\right) \quad \text{and} \quad \mathsf{Op}(n) = \mathcal{O}\left(n^2\right). \tag{4.1}$$

$\mathcal{O}(\cdot)$ denotes the Landau symbol describing the complexity by dropping lower order terms if an upper or lower bound is investigated. Naturally, the user of an algorithm wants to perform simulations with maximum number of unknowns n_{max} in acceptable time. Assuming for the moment that maximum available storage and maximum possible operations double in a time period Δt, then

$$\mathsf{St}_{max}^{(t+\Delta t)} = 2 \cdot \mathsf{St}_{max}^{(t)} \quad \text{and} \quad \mathsf{Op}_{max}^{(t+\Delta t)} = 2 \cdot \mathsf{Op}_{max}^{(t)}.$$

Logically, it is desired to double the problem size

$$n_{max}(t+\Delta t) = 2n_{max}(t)$$

in the same period as well. But with the complexities of (4.1) the required storage at time $t+\Delta t$ is

$$\mathsf{St}(n_{max}(t+\Delta t)) = \mathsf{St}\left(2n_{max}(t)\right) \xrightarrow{\mathcal{O}(n^2)} 4 \cdot \mathsf{St}(n_{max}(t)).$$

Normalized to the available storage $2\mathsf{St}_{max}^{(t)}$, that implies an increase of $4/2 = 2$. In fact the same holds for the number of operations. Hence, the calculation time is also doubled at $t+\Delta t$. To conclude, algorithms showing quadratic or generally polynomial complexity $\mathcal{O}(n^p)$ with $p \geq 2$ are problematic for future applications in terms of large scale computing. In contrary, the computational effort for algorithms with linear complexity stays constant in the sense of the example above. As a consequence for the previously described BEM formulation, rather large problems will cause shortage of storage and the calculation time might become incceptable. This is immediately recognizable in Figure 4.1. To overcome

[1] Usually, floating point operations like multiplication or division are considered as *elementary* in this context

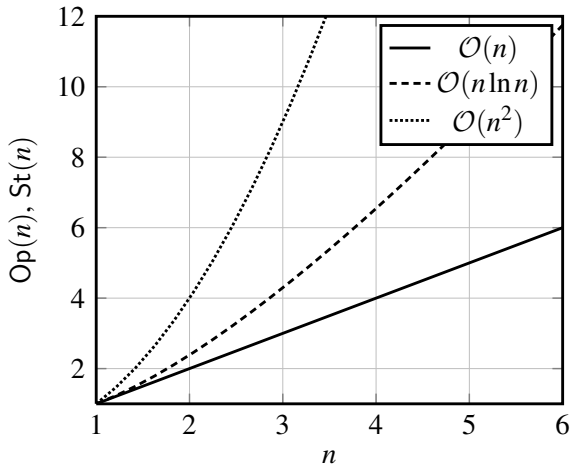

Figure 4.1: Required storage and computational effort for algorithms with different complexities

the quadratic complexity of many different numerical procedures, fast summation methods have been developed. A brief overview of those methods is given in section 1.2. In this chapter the hierarchical matrix ($\mathcal{H}$-matrix) technique is described, which is subsequently applied to the BEM for elasto-plastic problems presented in chapter 3. This matrix format provides almost linear complexity[2] $\mathcal{O}(n \log^{\alpha} n)$ in terms of storage and matrix operations. For problems within a not too distant observation period the logarithmic term is acceptable.

In terms of the the described BEM formulation, different approximation errors have been introduced. First, the approximations introduced by discretization of the BIE, where the residual is minimized in a finite number of collocation points, and the errors by carrying out integrals numerically. Secondly, approximation of the geometry, Cauchy data and field values. The material model is an approximation too, and its parameters are determined by experiments or even predictions in case of underground construction. Finally, the residual of the nonlinear solution procedure ϵ_σ is allowed to have a certain tolerance. Consequently, it is reasonable to approximate the system of equations itself with a similar magnitude of error $\epsilon_{\mathcal{H}}$.

Among the essential journal papers, several textbooks exist concerning $\mathcal{H}$-matrices. A detailed description including proofs and complexity analysis of the matrix operations is read in the book of Hackbusch [52] written in German. Extended details on $\mathcal{H}^2$-matrices in English can be read in the book of Börm [21]. In the lecture notes of Bebendorf [7] emphasis is given to elliptic boundary value problems and the ACA. For a more practical point of view and many numerical results for different problem types carried out by the boundary element methods the reader is referred to Rjasanow and Steinbach [89].

[2] Linear complexity up to a logarithmic factor

4.1 Low Rank Approximation

The near linear complexity of the matrix format described in this chapter stems from the data-sparse representation of low rank matrices ($\mathcal{R}_k$-matrices). The rank $k = \operatorname{rank}(\mathbf{M})$ of a matrix is the number of linear independent column or row vectors. It holds that

$$\operatorname{rank}(\mathbf{M}) \leq \min\{r, c\}, \quad \mathbf{M} \in \mathbb{R}^{r \times c}. \tag{4.2}$$

A $\mathcal{R}_k$-matrix is constructed by different methods and factorized to a data-sparse storage format.

To simplify the explanation of the procedure, we consider a circular geometry. The problem is shown in Figure 4.2 and the entries of the fully populated matrix $\mathbf{M}$ origin from

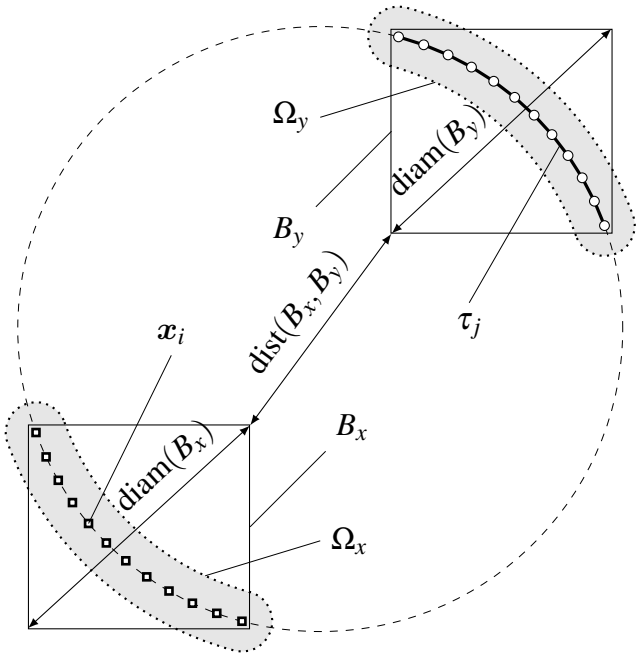

Figure 4.2: Circular tunnel geometry, partly discretized with well separated domains Ω_x and Ω_y

the discretization of the scalar valued single layer potential

$$(\mathcal{V}_\ell p)(\boldsymbol{x}) = \int_\Gamma \mathsf{U}_\ell(\boldsymbol{x}, \boldsymbol{y}) p(\boldsymbol{y}) \, \mathrm{d}s_{\boldsymbol{y}} \qquad \text{with} \qquad \mathsf{U}_\ell(\boldsymbol{x}, \boldsymbol{y}) = -\frac{1}{2\pi} \log |\boldsymbol{x} - \boldsymbol{y}| \tag{4.3}$$

and double layer potential

$$(\mathcal{K}_\ell q)(\boldsymbol{x}) = \int_\Gamma \mathsf{T}_\ell(\boldsymbol{x},\boldsymbol{y}) q(\boldsymbol{y}) \,\mathrm{d}s_{\boldsymbol{y}} \quad \text{with} \quad \mathsf{T}_\ell(\boldsymbol{x},\boldsymbol{y}) = -\frac{1}{2\pi|\boldsymbol{x}-\boldsymbol{y}|^2}(\boldsymbol{x}-\boldsymbol{y})\cdot\mathbf{n}(\boldsymbol{y}) \quad (4.4)$$

from the Laplace equation in two dimensions [101] evaluated for some collocation points $\boldsymbol{x}$ and a part of the discretized boundary $\boldsymbol{y} \in \Gamma_h$. The field values are discretized by piece wise linear trial functions. The points $\boldsymbol{x}$ and $\boldsymbol{y}$ are located in two disjoint domains Ω_x and Ω_y where the desired kernel function $\Phi(\boldsymbol{x},\boldsymbol{y})$ is non-singular. The separation is determined by the *admissibility condition*

$$\min(diam(\Omega_x), diam(\Omega_y)) \leq \eta\, dist(\Omega_x,\Omega_y), \quad \eta > 0. \quad (4.5)$$

The lower the admissibility parameter η is set, the larger the distance and better the approximation will be. However, one would find less regions which are admissible. Since the computational costs are too high for an exact evaluation of distance and diameter, condition (4.5) is defined as an estimation

$$\min(diam(B_x), diam(B_y)) \leq \eta\, dist(B_x,B_y) \quad (4.6)$$

on axis parallel *bounding boxes B* where

$$diam(\Omega) \leq diam(B) \quad \text{and} \quad dist(\Omega_x,\Omega_y) \leq dist(B_x,B_y).$$

With these boxes the calculation of the diameter and the distance

$$diam(B) = \sqrt{\sum_{i=1}^{d} (b_i - a_i)^2} \text{ and}$$

$$dist(B_x,B_y) = \sqrt{\sum_{i=1}^{d} dist([a_{i,x},b_{i,x}]_x, [a_{i,y},b_{i,y}])}$$

is done quickly. Now the evaluation of condition (4.6) is straightforward but may introduce non optimal results. This is illustrated in Figure 4.2, where the distance between the bounding boxes is much smaller than that of the original domains. For admissibility criterion different to (4.6), the reader is referred to [21] and [52].

4.1.1 Singular Value Decomposition

Let the matrix $\mathbf{M} \in \mathbb{R}^{r\times c}$ with $n = \operatorname{rank}(\mathbf{M}) = \min\{r,c\}$ be the discrete single or double layer potential with r collocation points and c supports of trial functions in Γ_h of Figure 4.2.

With a singular value decomposition (SVD) [43] the matrix is factorized in the form

$$\mathbf{M} = \mathbf{U} \cdot \mathbf{\Sigma} \cdot \mathbf{V}^\mathsf{T}, \quad \mathbf{U} \in \mathbb{R}^{r \times n}, \mathbf{\Sigma} \in \mathbb{R}^{n \times n}, \mathbf{V} \in \mathbb{R}^{c \times n} \tag{4.7}$$

where the column vectors of $\mathbf{U}$ and $\mathbf{V}$ form an orthonormal basis and $\mathbf{\Sigma}$ is a diagonal matrix. The diagonal entries $\sigma_i = \Sigma_{ii}$ denote the singular values of $\mathbf{M}$ and are considered to be sorted such that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$. Hence, the factorization can be written as a sum

$$\mathbf{M} = \sum_{i=1}^{n} \sigma_i \mathbf{u}_i \mathbf{v}_i^\mathsf{T}$$

with the basis vectors $\mathbf{u}$ and $\mathbf{v}$. Figure 4.3 shows the distribution of singular values σ_i of

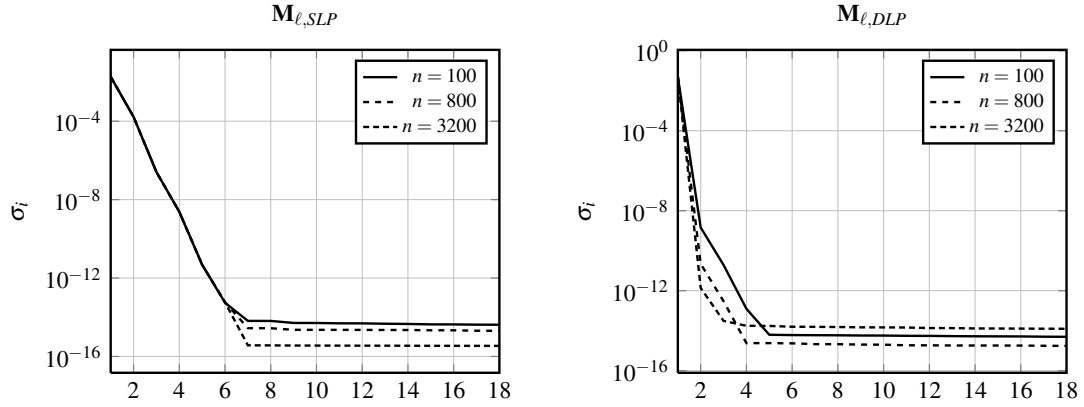

Figure 4.3: Distribution of the first 18 singular values of single layer (SLP) and double layer potential (DLP) for Laplace equation

the discrete single and double layer potential of Laplace equation for the given problem . It can be seen that they decay exponentially and the behavior is only slightly influenced by the full size n of the matrix. Higher singular values do not have much influence on matrix entries and it is considered to simply neglect them. What follows is an approximation of $\mathbf{M}$ by a truncated, compressed SVD

$$\mathbf{M} \approx \mathbf{R}_k = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^\mathsf{T} = \tilde{\mathbf{U}} \cdot \tilde{\mathbf{\Sigma}} \cdot \tilde{\mathbf{V}}^\mathsf{T}. \tag{4.8}$$

The $\mathcal{R}_k$-matrix $\mathbf{R}_k$ now has reduced rank $k < n$. To check the desired accuracy of the $\mathcal{R}_k$-matrix approximation, the error is determined in matrix norms. A convenient measure is the *Frobenius norm* [52]

$$\|\mathbf{M}\|_F^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} m_{ij}^2 = \sum_{i=1}^{\min\{r,c\}} \sigma_i \tag{4.9}$$

which is easy to implement in this context. A measure in the spectral norm $\|\mathbf{M}\|_2 = \sigma_1$ is also possible. Hence, to achieve the desired accuracy $\epsilon_{\mathcal{R}_k}$ the relative error bound i.e. by means of the Frobenius norm

$$\|\mathbf{M} - \mathbf{R}_k\|_F \leq \epsilon_{\mathcal{R}_k} \|\mathbf{M}\|_F \tag{4.10}$$

needs to be fulfilled. The required rank for a certain approximation accuracy is shown in

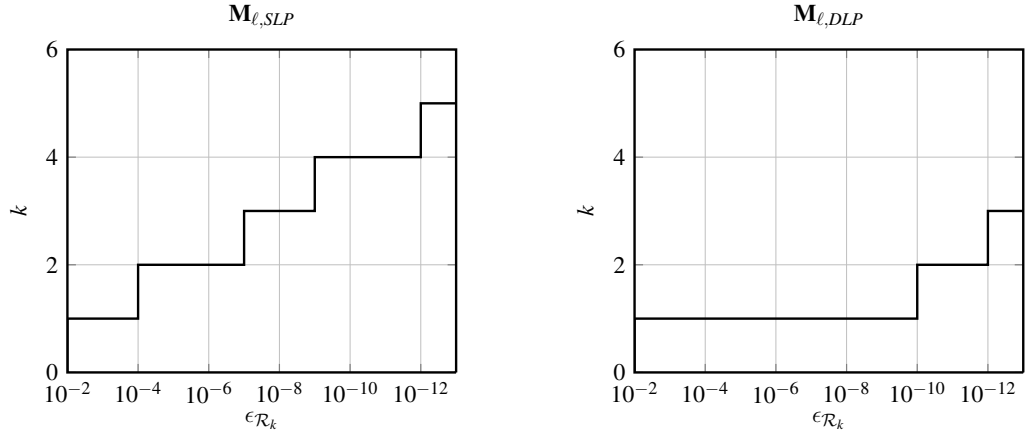

Figure 4.4: Rank distribution for the desired approximation accuracy

Figure 4.4. It can be seen that a rank of $k = 2$ for the single layer and $k = 1$ for the double layer potential is sufficient for a relative error $\epsilon_{\mathcal{R}_k} = 10^{-5}$, and for that particular example, independently of n.The storage requirements and operations for a matrix-vector product

$$\mathsf{St}_{SVD}(\mathbf{R}_k) = k(r+c+1)$$
$$\mathsf{Op}^{matmul}_{SVD}(\mathbf{R}_k) = 2k(r+c) - r + k$$

for the truncated SVD are significantly lower than for the full representation

$$\mathsf{St}_{full}(\mathbf{M}) = \mathsf{Op}^{matmul}_{full}(\mathbf{M}) = rc$$

if $k \ll n$. For example, with the circular geometry shown in Figure 4.2 with $n = 400$ and $\epsilon_{\mathcal{R}_k} = 10^{-5}$ the storage requirement is reduced by a factor of 100. Although SVD is the best possible rank revealing method, its application to matrix compression is not affordable, since the original and fully populated matrix needs to be computed in advance and the SVD procedure itself requires $\mathcal{O}(n^3)$ operations.

4.1.2 Interpolation

One approach to find a low rank approximation of a matrix a priori is to separate the variables $\boldsymbol{x}$ and $\boldsymbol{y}$ in the kernel function $\Phi(\boldsymbol{x},\boldsymbol{y})$. Therefore, $\Phi(\boldsymbol{x},\boldsymbol{y})$ needs to be *asymp-*

totically smooth [51]

$$|\partial_{\boldsymbol{y}}^{\alpha_i}\Phi(\boldsymbol{x},\boldsymbol{y})| \leq c(|\alpha_i|)\frac{\Phi(\boldsymbol{x},\boldsymbol{y})}{\|\boldsymbol{x}-\boldsymbol{y}\|^{|\alpha_i|}} \quad \forall \boldsymbol{y} \in \mathbb{R}^d \setminus \boldsymbol{x}, \quad i \in \{1,\ldots,d\} \tag{4.11}$$

by means of the α_i-th derivation with respect to y_i and a constant c. All fundamental solutions used in the described BEM have this property. The approximation is done finding a degenerated kernel by means of Taylor expansion, multipole expansion by spherical harmonics or similar approaches. For elastostatic problems this has been shown in [57] for the panel clustering method, from which the $\mathcal{H}$-matrix technique originates, and in [78] for the FMM. In the latter publication, the integral operators are reduced to the ones taken for scalar Laplace problems and then expanded by taking spherical harmonics. Since these methods eventually involve higher order derivatives, which may imply a kernel dependent development of a recursive functions, a simpler choice is to take kernel interpolation by means of polynomials. Hackbusch and Börm [53] presented the low rank approximation by interpolation in the context of $\mathcal{H}$-matrices. Basic fundamentals and proofs can be read in the textbooks [52] and [21].

The integral kernel in (4.3) is now interpolated in d dimensions by means of Lagrange polynomials

$$\mathsf{U}_\ell(\boldsymbol{x},\boldsymbol{y}) \approx \sum_{\nu=1}^{k}\sum_{\mu=1}^{k} L_\nu(\boldsymbol{x})\mathsf{U}_\ell(\bar{\boldsymbol{x}}_{\boldsymbol{\nu}},\bar{\boldsymbol{y}}_{\boldsymbol{\mu}})L_\mu(\boldsymbol{y}) \tag{4.12}$$

with k support points defined on each of a d-dimensional box $[a,b]^d = [a_1,b_1]\times\cdots\times[a_d,b_d]$ around $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively. The interpolation functions L_ν and L_μ are represented by the tensor product of the Lagrange polynomials in one dimension L_{ν_i}

$$L_\nu(\boldsymbol{x}) = \prod_{i=1}^{d} L_{\nu_i}(x_i) = \prod_{i=1}^{d}\prod_{\mu=0,\mu\neq\nu_i}^{k_i}\frac{x_i-\bar{x}_{\mu_i}}{\bar{x}_{\nu_i}-\bar{x}_{\mu_i}} \tag{4.13}$$

with

$$L_{\nu_i}(x) = \begin{cases} 1 & \text{if } \nu_i = \mu_i \\ 0 & \text{else} \end{cases} \quad \forall \nu_i,\mu_i \in \{0,\ldots,k_i\}.$$

In (4.13) the index i denotes the spatial direction. Hence, $\bar{x}_{\nu_i}$ is the i-coordinate of the ν-th support point. To get the best approximation quality for the integral kernel, roots of Chebyshev polynomials of the first kind are chosen. This avoids the Runge phenomenon, which leads to diverging approximations, i.e. when taking equally distant interpolation points [70]. Practically, these roots

$$\bar{\xi}_{\nu_i} = \cos\left(\frac{\left(\nu_i-\frac{1}{2}\right)\pi}{k_i}\right) \quad \forall \nu \in \{1,\ldots,k_i\}$$

are defined in the interval $[-1,1]$ and need to be mapped to $[a_i,b_i]$ by the function

$$\bar{x}_{\nu_i} = \mathcal{X}_L(\xi_{\nu_i}) = \frac{a_i+b_i}{2} + \frac{b_i-a_i}{2}\bar{\xi}_{\nu_i}.$$

For practical reasons the interval for the Chebyshev interpolation (4.13) is set to fit the bounding box B taken for the simplified measure of distance and diameter. In Figure 4.5 the

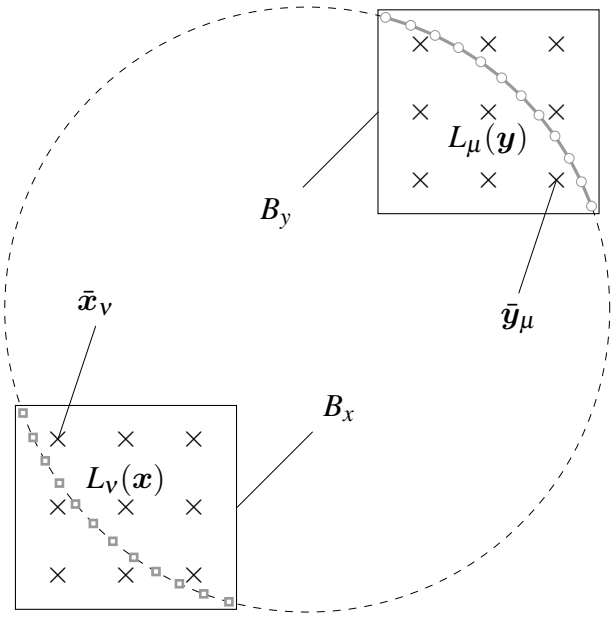

Figure 4.5: Boxes and support points of the interpolating polynomials L_ν and L_μ

bounding boxes B_x, B_y and support points of polynomial order $k_i = 3$ are shown exemplary.

For the separation to work correctly, condition (4.6) is a prerequisite and specially tailored for asymptotically smooth kernels [52]. Finally the interpolated kernel (4.12) is taken for the representation of integral operator $\mathcal{V}_\ell$ in (4.3). Taking constant expressions out of the integral, the approximation is

$$\tilde{\mathcal{V}}_\ell p(\boldsymbol{x}) = \sum_{\nu=1}^{k}\sum_{\mu=1}^{k} L_\nu(\boldsymbol{x})\mathsf{U}_\ell(\bar{\boldsymbol{x}}_{\boldsymbol{\nu}},\bar{\boldsymbol{y}}_{\boldsymbol{\mu}}) \int_\Gamma L_\mu(\boldsymbol{y})p(\boldsymbol{y})\,\mathrm{d}s_{\boldsymbol{y}}.$$

It is seen that the kernel of the boundary integrals is replaced by the interpolation function [53]. After discretization, the resulting low rank representation[3] is given by

$$\mathbf{R}_k = \mathbf{V}\cdot\mathbf{S}\cdot\mathbf{W}^T \quad \mathbf{V}\in\mathbb{R}^{r\times k}, \mathbf{S}\in\mathbb{R}^{k\times k}, \mathbf{W}\in\mathbb{R}^{c\times k}. \tag{4.14}$$

[3]Note that the symbol $\mathbf{V}$ is repeatedly used and is different to that of the factorization by SVD (4.7) and the discrete single layer potential introduced in section 3.2.

Again, $k = k_i^d$ denotes the rank of the matrices of which the entries are given by

$$\begin{aligned}\mathbf{V}[i,\nu] &= L_\nu(\boldsymbol{x_i}),\\ \mathbf{S}[\nu,\mu] &= \mathsf{U}_\ell(\bar{\boldsymbol{x}}_{\boldsymbol{\nu}}, \bar{\boldsymbol{y}}_{\boldsymbol{\mu}}) \text{ and}\\ \mathbf{W}[j,\mu] &= \int_{\Gamma_e} L_\mu(\boldsymbol{y})\varphi_j(\boldsymbol{y})\,\mathrm{d}s_{\boldsymbol{y}}\end{aligned}$$

for all $\boldsymbol{x} \in \Omega_x$ and $\boldsymbol{y} \in \Omega_y$. Since the integral kernel in entries of $\mathbf{W}$ are of polynomial degree p, the integration can be carried out exactly by means of Gaussian quadrature with order $G = 2p-1$. Contrary to the SVD, the rank needs to be chosen a priori and the relative approximation error $\epsilon_{\mathcal{R}_k}$ is known a posteriori only. The computational requirements for this kind of approximation are

$$\begin{aligned}\mathsf{St}_L(\mathbf{R}_k) &= k(r+c+k)\\ \mathsf{Op}_L^{matmul}(\mathbf{R}_k) &= 2k(r+c) - r + k^2\end{aligned}$$

Special care has to be taken if the integral kernel depends on normal derivatives like the fundamental solution $\mathsf{T}(\boldsymbol{x},\boldsymbol{y})$ in (2.10). Naturally, such functions are not smooth everywhere and a naive interpolation is likely to produce unwanted errors. Therefore, the conormal derivative is shifted to the polynomial and the interpolated double layer potential becomes

$$\mathcal{K}_\ell q(\boldsymbol{x}) = \sum_{\nu=1}^{k}\sum_{\mu=1}^{k} L_\nu(\boldsymbol{x})\mathsf{U}_\ell(\bar{\boldsymbol{x}}_{\boldsymbol{\nu}}, \bar{\boldsymbol{y}}_{\boldsymbol{\mu}}) \int_{\Gamma} \mathcal{T}_\ell L_\mu(\boldsymbol{y}) p(\boldsymbol{y})\,\mathrm{d}s_{\boldsymbol{y}}.$$

It is remarkable that for both, the discrete single layer and double layer potentials, the kernel evaluations and evaluation of L_ν at the collocation points $\boldsymbol{x_i}$ stay the same. So do the matrices $\mathbf{V}$ and $\mathbf{S}$. For the Laplace problem $\mathcal{T}_\ell L_\mu(\boldsymbol{y}) = \nabla L_\mu(\boldsymbol{y}) \cdot \mathbf{n}$ holds. It can be envisaged that the implementation of the conormal derivative for elastostatic problems (2.4) is not straightforward task.

For the model case shown in Figure 4.5 the approximation quality $\epsilon_{\mathcal{R}_k}$ is presented in Figure 4.6. Starting from an interpolation order $k_i = 3$ ($k = 9$) for the single layer potential a good approximation with a relative error $\epsilon_{\mathcal{R}_k} = 10^{-5}$ is reached and higher order polynomials increase the rank without improving the approximation too much. For the double layer potential, a higher order has to be chosen for the same accuracy because of the application of the conormal derivative.

4.1.3 Adaptive Cross Approximation

Developed by Bebendorf [6], the ACA has become a popular approximation technique in many fields of application by means of the boundary element methods. I.e. in [9, 61, 66] it is applied to elastostatic, in [71] to elasto-dynamic problems. The implementation is uncomplicated since the algorithm demands only some entries of the original matrix to

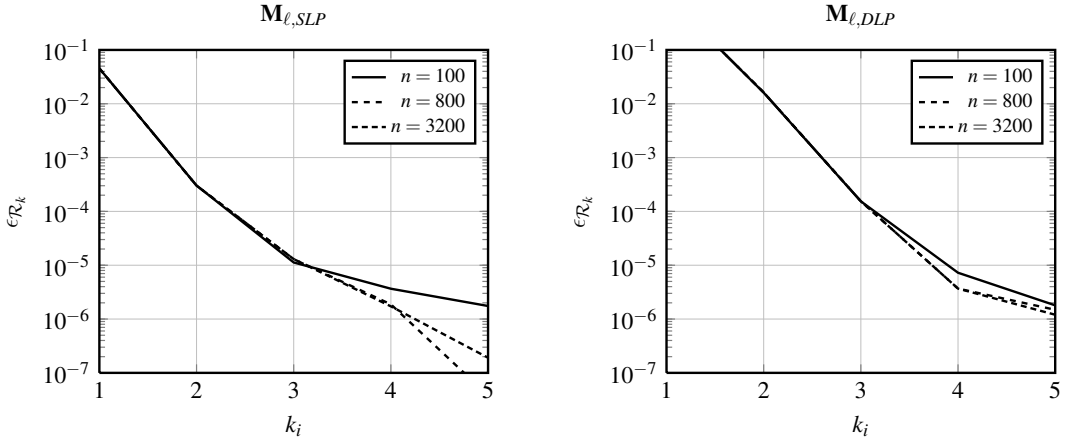

Figure 4.6: Achieved accuracy depending on interpolation order k_i

construct the low rank approximation. Contrary to interpolation, the construction of a $\mathcal{R}_k$-matrix is a pure black-box method. The application to collocation matrices has been shown in [10] and is adopted to elasto-plastic problems in chapter 5.

For the low rank representation carried out by this procedure the outer product format

$$\mathbf{R}_k = \mathbf{A}\mathbf{B}^\mathsf{T}$$

with $\mathbf{A} \in \mathbb{R}^{r\times k}$ and $\mathbf{B} \in \mathbb{R}^{c\times k}$ is taken. The idea is to find the $\mathcal{R}_k$-matrix by minimizing the reminder $\mathbf{F}_k$ of the approximation $\mathbf{M} = \mathbf{R}_k + \mathbf{F}_k$. With the accepted tolerance ϵ the relation

$$\|\mathbf{F}_k\|_F \leq \|\mathbf{F}_{k+1}\|_F \leq \frac{\epsilon}{1+\epsilon}\|\mathbf{R}_k\|_F$$

leads to the recursive stopping criterion

$$\|\mathbf{a}_{k+1}\|_F \cdot \|\mathbf{b}_{k+1}\|_F \leq \epsilon_{\mathcal{H}}\|\mathbf{R}_k\|_F \quad \text{with} \quad \mathbf{R}_k = \sum_{\nu=1}^{k} \mathbf{a}_\nu \mathbf{b}_\nu^T \tag{4.15}$$

for the algorithm. In (4.15) the scaled vectors $\mathbf{a}_{k+1}$ and $\mathbf{b}_{k+1}$ generate columns of $\mathbf{A}$ and $\mathbf{B}$ and denote the row and column of a matrix-cross in $\mathbf{R}_k$ which is determined by finding a pivot entry therein. In order to compute the crosses $\mathbf{a}_k$ and $\mathbf{b}_k$ only the corresponding row and column of the original matrix $\mathbf{M}$ needs to be evaluated which reduces the computational time. Figure 4.7 and Algorithm 2 depict the construction of a low rank representation by means of ACA. Only k original matrix rows and columns are calculated by the user-provided methods FILLROW and FILLCOL. The corresponding pivot points (r_k, c_k) are marked as black spots in Figure 4.7. The choice of pivots is crucial for the accuracy of the algorithm. In order to evaluate original matrix entries, the quadrature order G has to be fixed per block. Moreover, the accuracy for the numerical integration needs to be higher

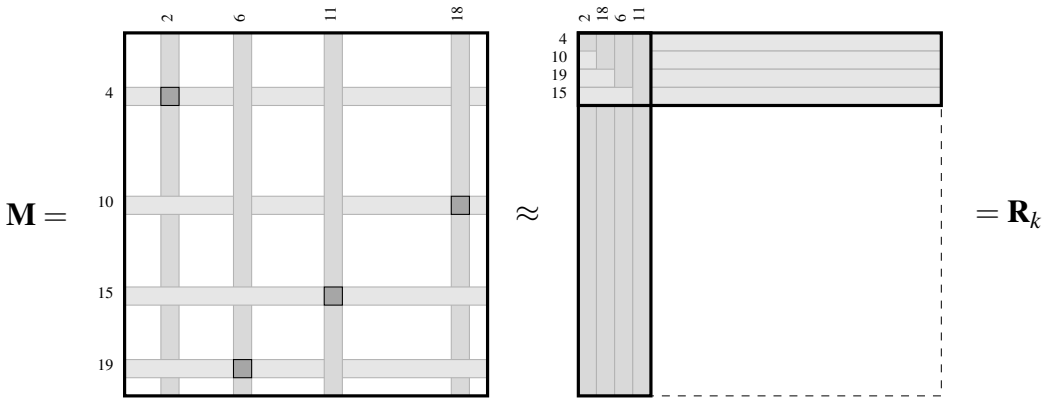

Figure 4.7: Construction of a $\mathcal{R}_k$-matrix by ACA

Algorithm 2 Adaptive cross approximation

procedure ACA
  $k = 1$
  initial pivot row index r_1 (c.f. [7])
  repeat
    $\mathbf{b}_k \leftarrow$ FILLROW(r_k)
    $\mathbf{b}_k := \mathbf{b}_k - \sum_{\nu=1}^{k-1} \mathbf{a}[r_k]\mathbf{b}_\nu$
    pivot column index $c_k = \text{argmax}(|\mathbf{b}_k[r]|)$
    $\mathbf{b}_k = \frac{1}{\mathbf{b}_k[c_k]}\mathbf{b}_k$
    $\mathbf{a}_k \leftarrow$ FILLCOL(c_k)
    $\mathbf{a}_k := \mathbf{a}_k - \sum_{\nu=1}^{k-1} \mathbf{b}[c_k]\mathbf{a}_\nu$
    pivot row index $r_{k+1} = \text{argmax}(|\mathbf{a}_k[c]|)$, $r_{k+1} \neq r_k$
    $k := k + 1$
  until criterion (4.15) $= true$
end procedure

than the desired approximation accuracy in order to avoid perturbations. A more detailed description and different versions of the algorithm as well as strategies to find the right pivot points can be read in [7].

The algorithm converges if the integral kernel is asymptotically smooth with respect to at least one variable (4.11). In case of the collocation method, this is $\boldsymbol{x}$ since the outward normal $\mathbf{n}(\boldsymbol{y})$ involved in the traction operator $\mathcal{T}$ does not have this property for non-smooth surfaces. Hence, the discrete double layer potential of the model example $\mathbf{K}_\ell$ does not impose any problems or differences in the implementation of the algorithm when the right pivoting strategy is chosen [7]. For the model case, Figure 4.8 shows that the rank revealed by ACA is almost optimal compared to SVD in Figure 4.4. On that account, the computational costs $\mathcal{O}(n)$ for the algorithm are rather small. The effort in terms of storage and for

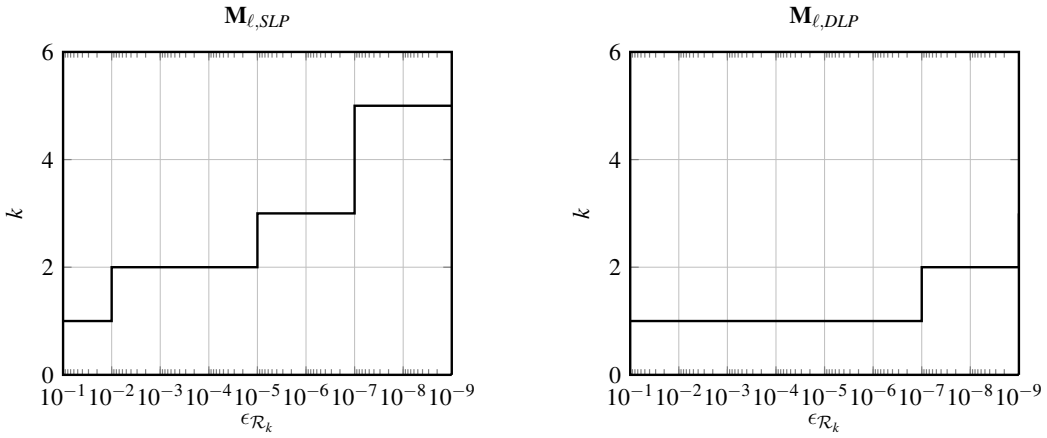

Figure 4.8: Rank distribution for the desired approximation accuracy by ACA

the matrix-vector product are

$$\mathsf{St}_{ACA}(\mathbf{R}_k) = \mathsf{Op}^{matmul}_{ACA}(\mathbf{R}_k) = k(r+c).$$

It is essential to stress that ACA only works with scalar problems. Therefore the matrix valued kernels for the system matrices in (3.8) and (3.11) have to be decomposed beforehand. The implications on the algorithm are described in chapter 5.

4.1.4 Recompression

One of the main advantages of ACA is the variable rank of the $\mathcal{R}_k$-matrix. For the approximation by degenerated kernels the rank is determined by the number of interpolation points fixed by the user or mutable determined by a criterion [25]. Most often the chosen rank is higher than needed. Developed by Börm and Grasedyck [23], the hybrid cross approximation (HCA) takes advantage of both, interpolation and ACA and can avoid this problem. Since the transformation matrix $\mathbf{S}$ in (4.14) consists of point wise evaluations of the fundamental solution, ACA could be applied in order to get a low rank representation. Then the resulting pivot points are taken to construct the interpolation, where the fundamental solution itself serves as interpolation function. Hence, the construction of $\mathbf{W}$ in (4.14) is done by integrating the original kernel. Nevertheless, no original matrix entries can be used like in ACA.

An important approach to further reduce the rank is the recompression of $\mathcal{R}_k$-matrices. The following procedure has been presented in [46] and is applied to both, approximation by kernel interpolation and ACA. The basic concept is again based on SVD but now applied to the outer product format of $\mathbf{R}_k$. This is efficiently achieved by means of a two QR decomposition and one SVD of a $k \times k$-matrix [47]. In that case, both factors of the

$\mathcal{R}_k$-matrix are decomposed to

$$\begin{aligned} \mathbf{A} &= \mathbf{Q}_A \mathbf{R}_A \quad \text{with} \quad \mathbf{Q}_A \in \mathbb{R}^{r\times k}, \mathbf{R}_A \in \mathbb{R}^{k\times k} \quad \text{and} \\ \mathbf{B} &= \mathbf{Q}_B \mathbf{R}_B \quad \text{with} \quad \mathbf{Q}_B \in \mathbb{R}^{r\times k}, \mathbf{R}_A \in \mathbb{R}^{k\times k} \end{aligned} \tag{4.16}$$

where $\mathbf{Q}_A$ and $\mathbf{Q}_B$ are orthogonal, $\mathbf{R}_A$ and $\mathbf{R}_B$ upper triangular matrices. Finally, for a small rank k a SVD of the product $\mathbf{R}_A\mathbf{R}_B$ is affordable and is done as described in section 4.1.1 resulting in $\mathbf{U}_{AB}, \mathbf{V}_{AB} \in \mathbb{R}^{k\times k'}$. Out of it the new factorization with usually lower rank k' is

$$\mathbf{R}_{k'} = \mathbf{A}' \cdot \mathbf{B}' = \mathbf{Q}_A \mathbf{U}_{AB} \cdot \mathbf{Q}_B \mathbf{V}_{AB}. \tag{4.17}$$

The recompressed matrix $\mathbf{R}_{k'}$ is now the *best approximation* with respect to the norm used for the truncated SVD. The overall complexity for this operation is $\mathsf{Op}_{recomp} = 6k(2r + c) + 23k^3$ and yet affordable for $k \ll \min(r,c)$.

To demonstrate the efficiency, the described technique is now applied to the low rank approximation of $\mathbf{M}$ constructed by interpolation. By taking $\epsilon = 10^{-5}$ for the single layer potential, which corresponds to an interpolation with order three or rank $k = 9$ respectively, the final rank is reduced to $k' = 3$ for all tested discretization $n = 100, 800, 3200$. In case of the double layer potential, $k = 25$ is reduced to $k' = 1$. Both values correspond to the optimal rank revealed by SVD and ACA.

Remark: It is noteworthy to say that the double layer potential on a circular geometry has a certain property which results to a rank one representation depending on how good the circular geometry is approximated. The results in this section show that behavior.

4.2 Geometrical Bisection

The example shown in Figure 4.2 handles a separated region of the boundary and hence just a part of the discretized operator. The system matrices in equations (3.8) and (3.11) as a whole cannot be approximated by a single $\mathcal{R}_k$-matrix since no separation with respect to both variables in the sense of (4.6) can be found. The ACA would probably lead to a full rank representation and kernel interpolation to completely wrong results. Almost every fundamental solution in boundary element methods behaves like $\Phi(\boldsymbol{x},\boldsymbol{y}) = \|\boldsymbol{x}-\boldsymbol{y}\|^{-\alpha}$ with $\alpha \in \mathbb{N}^+$. Therefore, matrix entries need to get separated into a *near field* where the distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ is small, and a *far field*. Near field entries are evaluated straightforward, by working out entries like in every standard BEM, whereas far field entries are subject to approximation techniques which have been described in section 4.1. Naturally, the splitting is done block wise, as illustrated by the introductory example in section 1.3. For general situations, this simple approach cannot be applied and the splitting of the system matrix needs a more distinct strategy. To obtain well separated regions as displayed in Figure 4.2, the matrix indices are collected in *clusters* [55] based on geometrical information. Resting upon clusters of row- and column indices, a *block cluster* is constructed

which defines the structure of the $\mathcal{H}$-matrix with well separated blocks.

4.2.1 Cluster Tree

Almost every fast summation method deploys a tree to represent the partition of matrices with general structure. The *cluster tree* in context of $\mathcal{H}$-matrices is a binary tree and created by splitting the geometry recursively. The construction is explained using the example of the circular geometry in section 4.1, but now for the full set of equations on the whole geometry. As shown in Figure 4.9, the indices i and j are assigned to *characteristic points*

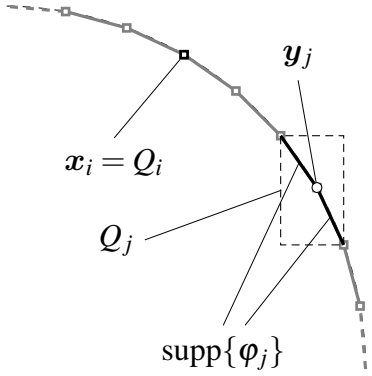

Figure 4.9: Characteristic points with local bounding boxes Q

$\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ with local axis parallel bounding boxes Q_i and Q_j. Row indices of the system matrices in (3.8) and (3.11) correspond to collocation points. Therefor Q_i reduces to the characteristic point. In case of column indices, Q_j defines a bounding box around the support of the trial function. All indices are collected to the index sets I and J. A cluster is the union of one or more indices of a set including additional information stored in a label. For each set, a labeled binary cluster tree T is constructed. The nodes are clusters where $r = t_0 = \text{root}(T)$ denotes the root and is labeled by all indices i.e. I and their associated positions $\boldsymbol{x}_i$. By splitting the geometry once, r gets exactly two children - the clusters t_1^1 and t_2^1. The superscript denotes the level ℓ in T. For r the level is set to $\ell = 0$. Algorithm 3 is a general implementation of the cluster tree creation procedure. The splitting is continued recursively until a stopping criterion

$$\text{size}(t) = \#t \leq n_{min} \tag{4.18}$$

is fulfilled which is characterized by the minimum leaf size n_{min} which denotes the minimal amount of indices in a cluster. In Figure 4.10 the general structure of a binary tree with clusters is shown. If a cluster does not have any child, it is called a leaf. In that example this is the case for t_1^2 and t_3^2.

For the considered fast method based on $\mathcal{H}$-matrices different splitting strategies can be taken. In the following paragraphs, strategies which replace the call to CLUSTERSTRATEGY

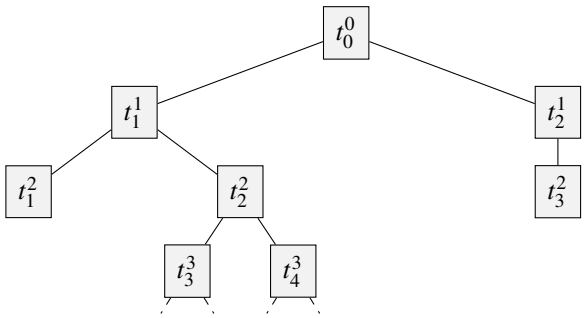

Figure 4.10: General structure of a cluster tree up to level $\ell = 3$

Algorithm 3 Creation of a cluster tree

```
1:  procedure ClusterTree(t, B)
2:      if #t > n_min then
3:          {t_1,...,t_i},{B_1,...,B_i} ← ClusterStrategy(t,B)
4:          for all (t_i, B_i) do
5:              if number of children t_i > 0 then
6:                  ClusterTree(t_i, B_i)
7:              end if
8:          end for
9:      end if
10: return t
11: end procedure
```

in Algorithm 3 are described and discussed on the example of a circle for arbitrarily distributed collocation points $\boldsymbol{x}_i$.

Geometrically Regular Cluster Tree The geometry is subdivided into axis parallel bounding boxes $B_n^\ell = [a_m, b_m]_{m=1}^d$ with equal edge length $b_m - a_m = const$. Here, n denotes the number of the box in the ℓ-th level of subdivision. Furthermore, an initial bounding box B_0^0 is constructed covering all local boxes Q. In this example $Q_i = \boldsymbol{x}_i$ and $B_0^0 = [a,b]^d$ for $\ell = 0$ with

$$a_m = \min\{\boldsymbol{x}_i[m]\} \quad \text{and} \quad b_m = \max\{\boldsymbol{x}_i[m]\} \quad \forall m \in \{1,\dots,d\} \tag{4.19}$$

and for all indices $i \in \{1,\dots,I\}$ as depicted in Figure 4.11. In the next step, B_0^0 is halved by a cut plane defined with respect to the bounding boxes' direction of largest extend. The result is now two boxes

$$B_1^1 = [a_1, {}^{a_1+b_1}/_2] \times [a_2, b_2] \quad \text{and} \quad B_2^1 = [{}^{a_1+b_1}/_2, a_1] \times [a_2, b_2]$$

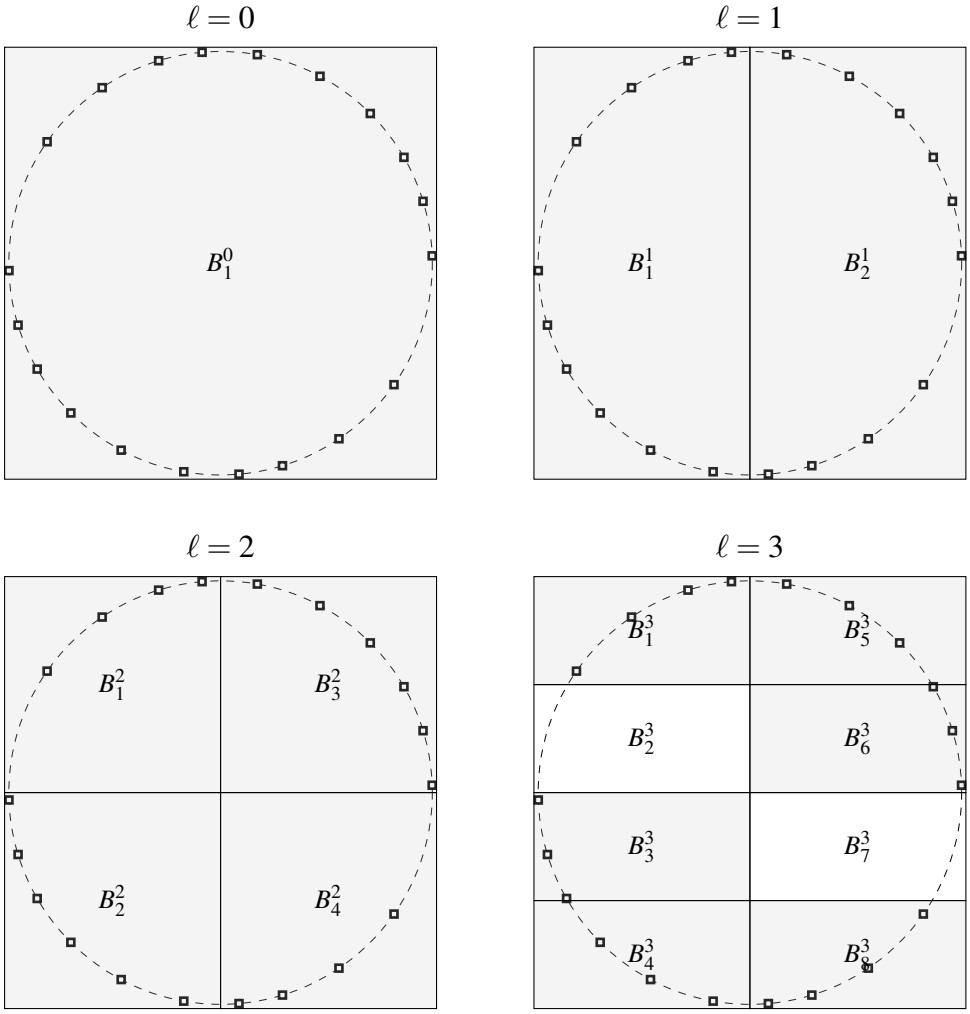

Figure 4.11: Geometrically regular clustering

in level $\ell = 1$. The point indices i covered by those boxes are collected in the index set $\hat{t}_n^\ell \subseteq I$ which is associated with t_n^ℓ. The procedure is repeated for each box until the minimal leaf size (4.18) is reached. In the example depicted in Figure 4.11 the minimum leaf size $n_{min} = 3$ and it can be seen, that there are empty sets $\hat{t}_3^3 = \hat{t}_7^3 = \emptyset$ for B_3^3 and B_7^3 in the third level. Hence, they are also not represented in the corresponding cluster tree shown in Figure 4.12. Moreover, B_1^3 and B_8^3 are to be split again in level 4, which is indicated by dashes.

This kind of clustering strategy produces equally sized bounding boxes for each level. This is advantageous for interpolation schemes and if there should be only one cluster structure for different geometries. The support points of the interpolation polynomial L_ν are located at the same local positions in the bounding boxes. Since the integral kernels are translational invariant, the amount of kernel evaluations could be reduced. A drawback

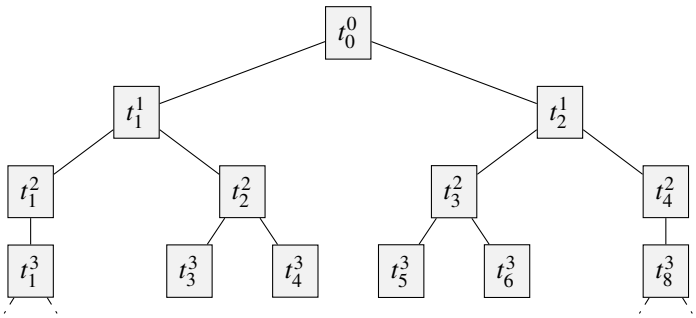

Figure 4.12: Clustertree for geometric clustering

is that a child cluster could be empty, like t_3^3 and t_7^3 and thus clusters t_1^2 and t_4^2 have just one child.

Remark: The FMM, which is also based on geometrical bisection, usually takes cubes as bounding boxes. They are bisected by perpendicular planes in every direction of coordinates per level. Thus, the tree structure becomes a quad tree in two dimensions or an octree in three. Theoretically, the same subdivision could be used for $\mathcal{H}$-matrices, too.

By taking a closer look on the clustering of level three it is visualized that the bounding box B_5^3 could be smaller while still covering all indices $\hat{t}_3^3$. This is not optimal and has been taken into account for the following clustering strategy.

Geometrically Balanced Cluster Trees In the admissibility criterion (4.6), the diameter of the bounding box plays an important role. Hence, it is endeavored to minimize the measure. The subdivision steps are almost the same for a geometrically regular T, but with the following differences: after every subdivision the bounding boxes of t are minimized so that their largest extend is determined by the characteristic points $\boldsymbol{x}_i$ of their indices $i \in \hat{t}_n^\ell$. This means that $B_n^\ell = [a,b]^d$ is determined by (4.19) in every level which is illustrated in Figure 4.13. It is realized that the diameters of B shrink quickly with increasing level. In general, the structure of T is not the same as in geometrically regular clustering and shown in Figure 4.15.

Geometrically balanced cluster trees are favored if the underlying grid is anisotropic. Still, clusters of the same level can have diverging sizes $\#t$.

Cardinality Balanced Cluster Trees The latter two approaches are based purely on geometry and neglect the cardinality $\#t$ of the clusters. Now the axis oriented cut plane is sought by balancing the number of characteristic points in B, optimally such that the size of children of t_n^ℓ

$$\#t_1^{\ell+1} - \#t_2^{\ell+1} \leq 1.$$

After bisection, the minimal bounding box is determined again. For the model case this has been illustrated in Figure 4.14. This splitting algorithm leads to a minimal depth of

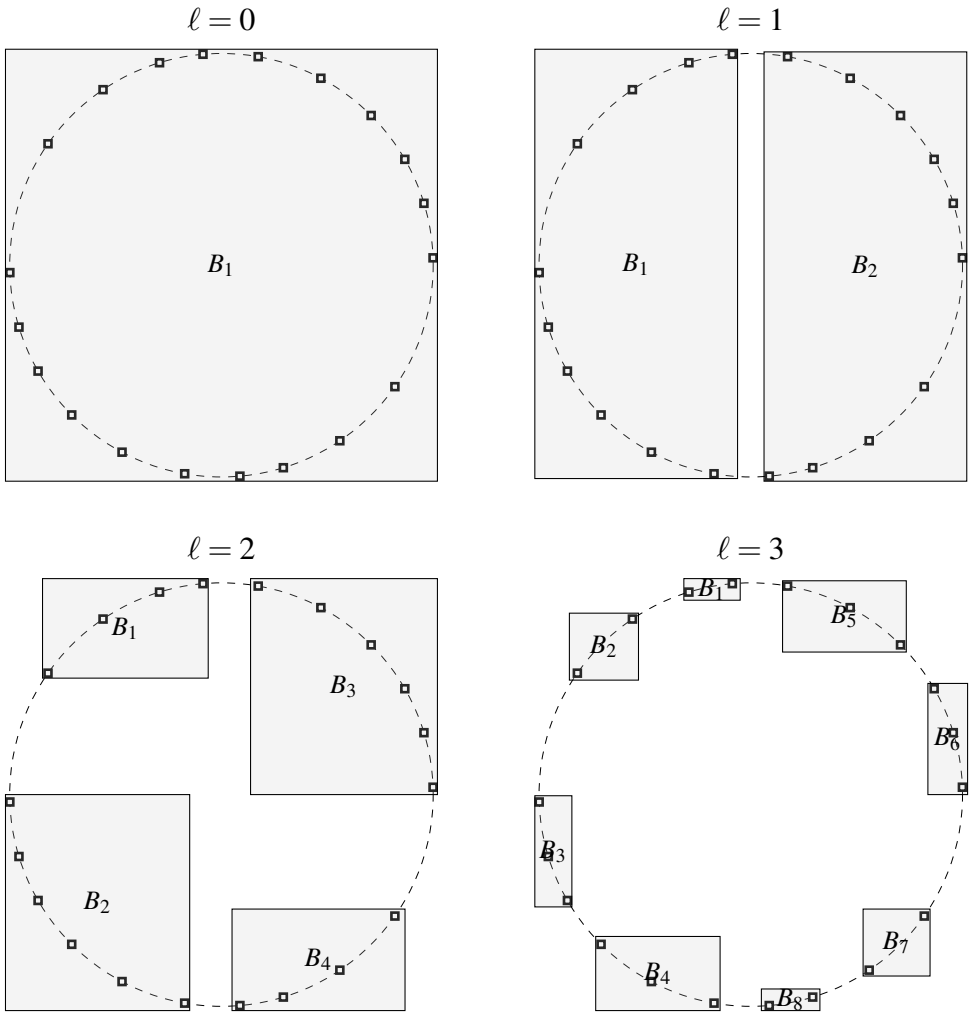

Figure 4.13: Geometrically balanced clustering

T with respect to n_{min}. A balanced cluster tree is shown in Figure 4.15, which coincides with T geometrical balance for the considered example. However, in general this is not the case. Keeping $\#t$ as high as possible is advantageous for ACA, thus taking a cardinality balanced cluster tree is a good choice.

In [7], an alternative splitting strategy called principal component analysis (PCA) is proposed. It produces cardinality balanced cluster trees T by splitting with arbitrary planes determined by the principal direction of the set of characteristic points.

Summarized, the choice of the right clustering strategy depends on the approximation technique for the $\mathcal{R}_k$-matrices as well as on the underlying grid. If indices correlate to a quasi-uniform discretization, geometric binary splittings are a good choice [95], for anisotropic or graded meshes cardinality balanced trees [47].

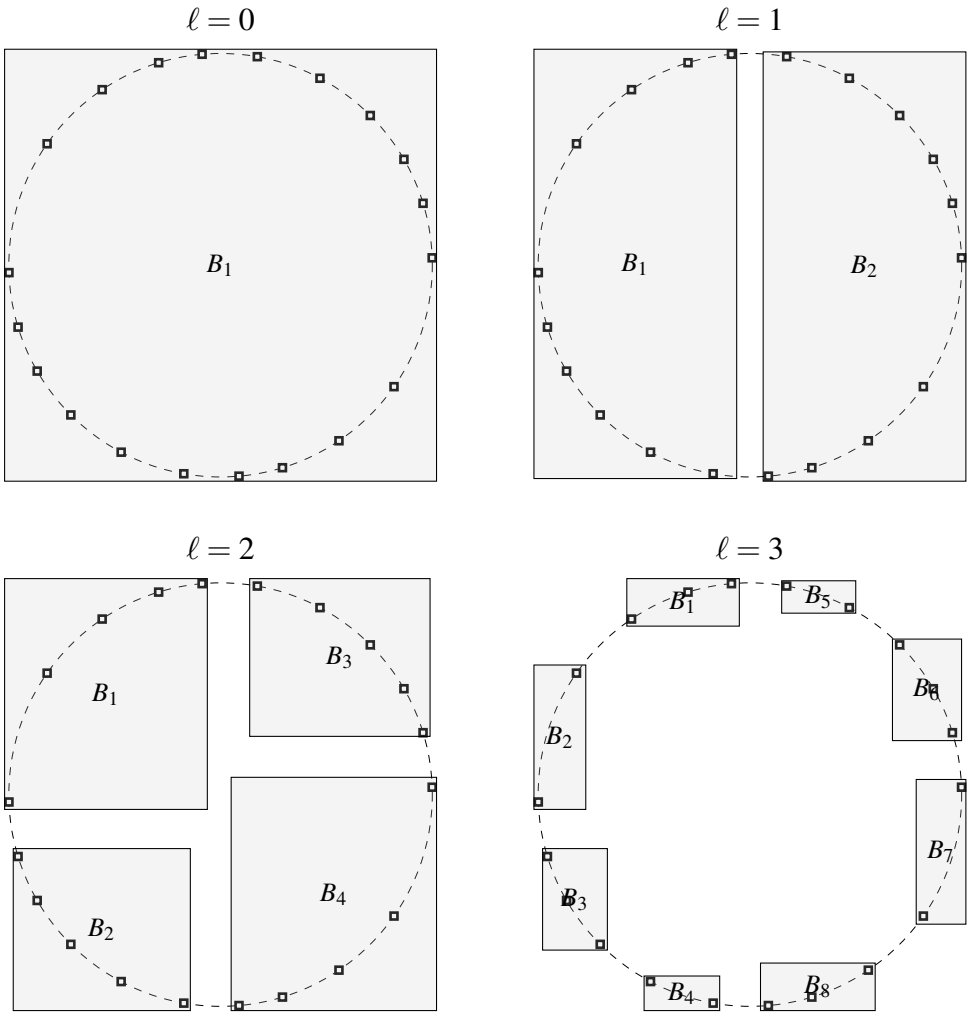

Figure 4.14: Cardinality balanced clustering

4.2.2 Block Cluster Tree

This subsection is devoted to the task of finding admissible pairs of clusters which define a matrix block subject to approximation as described in section 4.1. Therefor the clusters t related to collocation nodes $\boldsymbol{x}_i$ and s related to the supports $\operatorname{supp}(\varphi_j)$ of the trial functions on boundary elements are considered. Their associated index sets are

$$\hat{t}_n^\ell \subseteq I, \quad I = \{1,\ldots,r\} \quad \text{and} \quad \hat{s}_n^\ell \subseteq J, \quad J = \{1,\ldots,c\}.$$

The clusters are nodes of the column cluster tree T_I and the row cluster tree T_J. Since testing all possible combinations $t \times s$ needs $\mathsf{Op}^{t\times s} = \mathcal{O}(n^2)$, only clusters of the same level $t^\ell \times s^\ell$ in T_I and T_J respectively are taken. By considering nodes $t \in T_I$ and $s \in T_J$ in the same cluster tree level ℓ, an admissible pair can be found by checking condition (4.6)

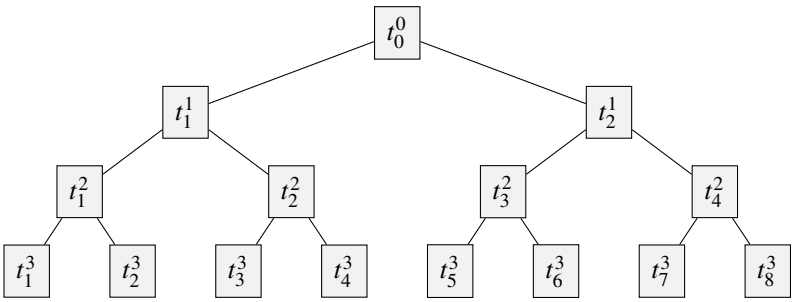

Figure 4.15: Balanced clustertree

with their associated bonding boxes B_t and B_s. The results are stored in a level consistent *block cluster tree* $T_{I\times J}$ whose nodes

$$b = t \times s, \quad b \in T_{I\times J}$$

form a partition $P = T_{I\times J}$. A level consistent bock cluster tree is a quad tree. Half of the tree is shown in Figure 4.16. From now on, the distinction between cluster nodes, i.e. t and

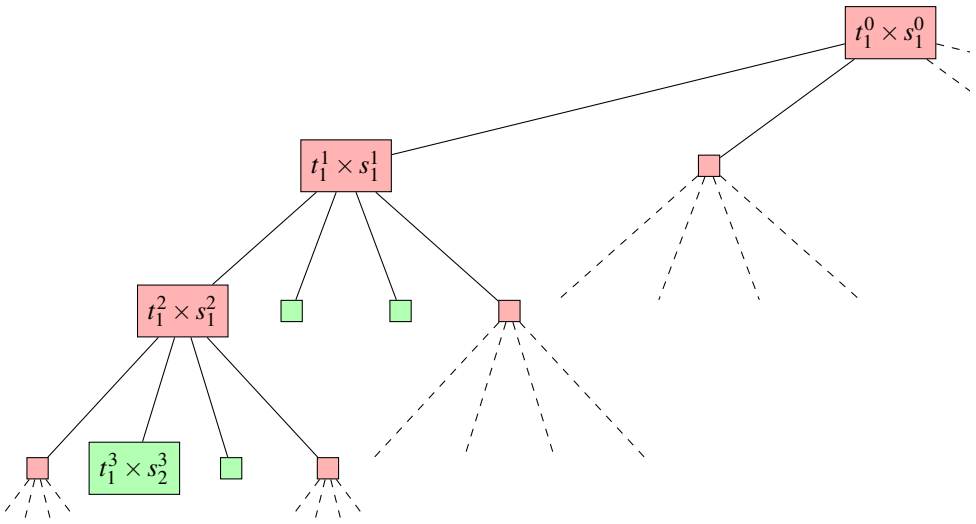

Figure 4.16: Block cluster tree structure

their corresponding index sets $\hat{t}$ is neglected and thus $b \subseteq I \times J$ can be used. If the partition $t^\ell \times s^\ell$ is admissible or if t and s are leafs then b is a leaf in $T_{I\times J}$. In Figure 4.16 admissible leafs are green and inadmissible leafs are red.

The procedure is depicted in Figure 4.17 by taking the example of geometrically balanced cluster trees as shown in Figure 4.13 and by setting $T_I = T_J$ for simplicity. The admissibility parameter $\eta = 1$ for condition (4.6). Starting from the root partition $b^0 = t^0 \times s^0$

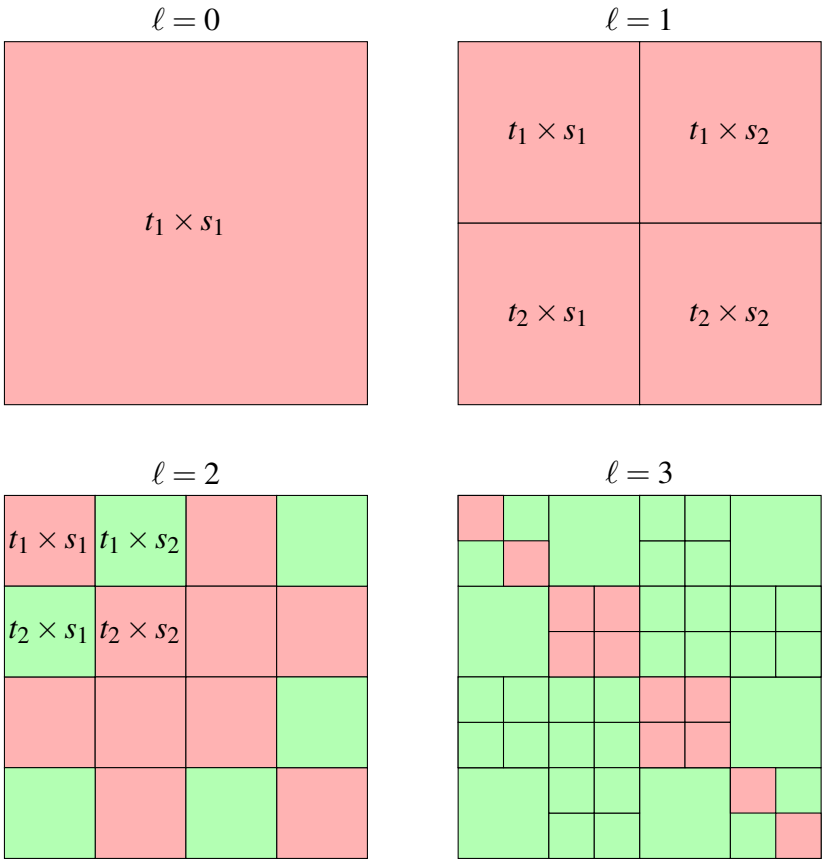

Figure 4.17: Partition defined by the block cluster tree

it is envisaged that for level $\ell = 0$ no admissible partitions are possible. For $\ell = 1$ the distance between B_1 and B_2 is small compared to their diameters, hence (4.6) does not hold. Proceeding to $\ell = 2$, the first admissible partition $t_1^2 \times s_4^2$ is identified and depicted in Figure 4.17.

For the considered level consistent example, it can be envisaged that the number of indices $\#t \approx \#s$. This is not always the case, especially if characteristic points of one cluster are located in the domain so that $\#t \ll \#s$ like for Newton potentials. In that case it is expedient to violate the level consistency and proceed with the traverse of s until $\#t^\ell \approx \#s^{\ell_s}$ with $\ell_s > \ell$ resulting in an inhomogeneous block cluster tree.

All admissible pairs of clusters are taken together denoting the far field $P^+ \subset T_{I\times J}$, all others denote the near field $P^- = T_{I\times J} \setminus P^+$. If $b \in P^+$ defines a matrix partition $\mathbf{M}|_b$ in the far field it can be represented by a $\mathcal{R}_k$-matrices. Blocks in P^-, represented by full matrices, are evaluated like in the classical method. The block cluster tree directly defines the structure of the $\mathcal{H}$-matrix which is treated in the next section.

4.2.3 Clustering Volume Discretizations

Usually, when solving inhomogeneous boundary value problems, volume discretizations are involved. In order to explain the concept, the circular example is extended by a volume mesh as shown in Figure 4.18. The bounding boxes $B_{\tilde{x}}$ and B_z are related to a cluster

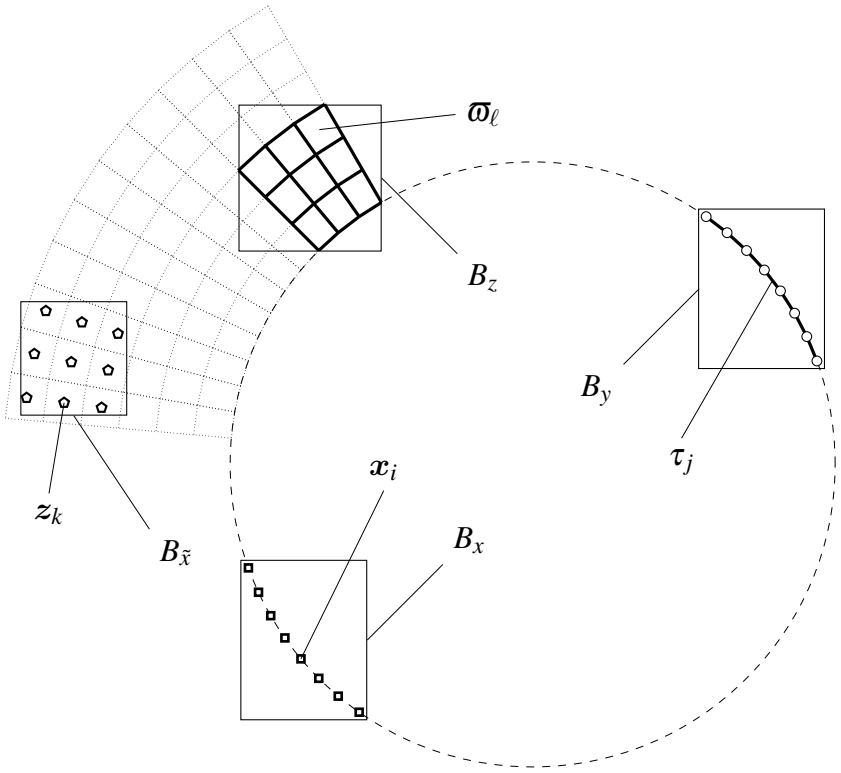

Figure 4.18: Clustering of a domain discretization

containing stress points z and cell elements ϖ respectively. It is visualized that many boxes B_x containing collocation points $x \in \Gamma$ will get admissible to B_ϖ. Hence, P^+ will dominate the partition related to domain term in (3.8). The same holds for relations between boxes $B_{\tilde{x}}$ and B_y.

Since the characteristic points are in the same domain Ω_p, relations $B_{\tilde{x}}$ and B_ϖ are troublesome. These boxes form the partition for the domain term in (3.11). Therefore, the clustering strategy for the characteristic points in Ω_p are carefully chosen in order to enlarge the related P^+. The size of B_z depends heavily on the chosen trial functions for the representation of strains and stresses (3.5). If they are chosen to be continuous, $\mathrm{supp}(\theta_\ell)$ will usually cover four cells in two dimensions and lead to an unnecessary large B_z. Moreover, the discretization can get very coarse if the distance to the boundary increases. Keeping supports local to ϖ will reduce bounding box sizes with the cost of a discontinuous strain and stress interpolation. However, for elasto-plastic problem statements or problems with inclusions, stresses and strains can be discontinuous anyway.

4.3 Hierarchical Matrix Format

The set of $\mathcal{H}$-matrices has been introduced by Hackbusch [50] as a general matrix format and to the BEM community in [24]. It allows the storage of fully populated matrices with $\mathcal{O}(n \log n)$ by means of a hierarchical structure and the approximation of admissible blocks by $\mathcal{R}_k$-matrices. Due to the structure, higher matrix operations like *LU* and *QR* factorization are carried out with almost same complexity in a purely algebraic manner. Moreover, information on the underlying physical problem is not needed after construction. By taking ACA for the low rank approximation even the underlying physics is not important. Only the asymptotically smoothness of the fundamental solution has to be guaranteed (see section 4.1).

In [53] $\mathcal{H}^2$-matrices have been introduced and their construction analyzed in [53] and [25]. The textbook of Börm [21] is dedicated to this format. By taking an additional hierarchical structure of nested cluster bases, the computational complexity and storage is reduced optimally. Due to the introduced multilevel transformations, this still algebraic format is comparable to the FMM technique and can provide optimal $\mathcal{O}(n)$ complexity for many operations too [20]. However, this work covers the application of $\mathcal{H}$-matrices and serves as a basis for the application of $\mathcal{H}^2$-matrices to elastostatic and elasto-plastic problems.

In the previous section, row and column cluster trees T_I, T_J as well as the block cluster tree $T_{I\times J}$, determined by the admissibility condition (4.6) have been described using the example of a circular geometry. By using the information stored in $T_{I\times J}$, the corresponding data sparse $\mathcal{H}$-matrix is constructed and every leaf $b = t \times s$ forms a matrix block $\mathbf{M}|_b$. The tree structure of $T_{I\times J}$ is recognizable by the indices of $\mathcal{H}$-matrix rows and columns. The larger the tree path from one node b to the other, the larger their distance with respect to the row and column indices in the matrix will be. Since the clustering is based on geometric information this holds for the spatial distance between their characteristic points too.

Intuitively, the sparsity of the matrix depends on the ratio of the number of full matrix blocks $\#P^+$ in near field and $\mathcal{R}_k$-matrix blocks $\#P^+$ in far field. For detailed definition of the sparsity constant by means of $\mathcal{H}$-matrices, the reader is referred to [52].

For the considered model case in this chapter, Figure 4.19 shows the structure of system matrices in the $\mathcal{H}$-matrix format. Uniform discretizations with n degrees of freedom are considered. Again, red matrix blocks describe P^-, green blocks P^+ and the numbers the rank of the matrix block. For the discrete single layer potential (a) as well as for the double layer potential (b) the far field P^+ increases with growing n. On close examination for matrices with $n = 3200$, it can be seen that the largest admissible matrix block with a size 800×800 is approximated by a single $\mathcal{R}_k$-matrix of rank $k = 4$ in $\mathbf{V}$ and, due to the circular geometry, by $k = 1$ for $\mathbf{K}$. The compression by means of $\mathcal{R}_k$-matrices results in storage requirements of 4% and 3% respectively compared to the original size.

As the name implies, block matrices defined by $T_{I\times J}$ and visualized in Figure 4.19 are stored hierarchically. Thus all matrix operations are implemented recursively [47].

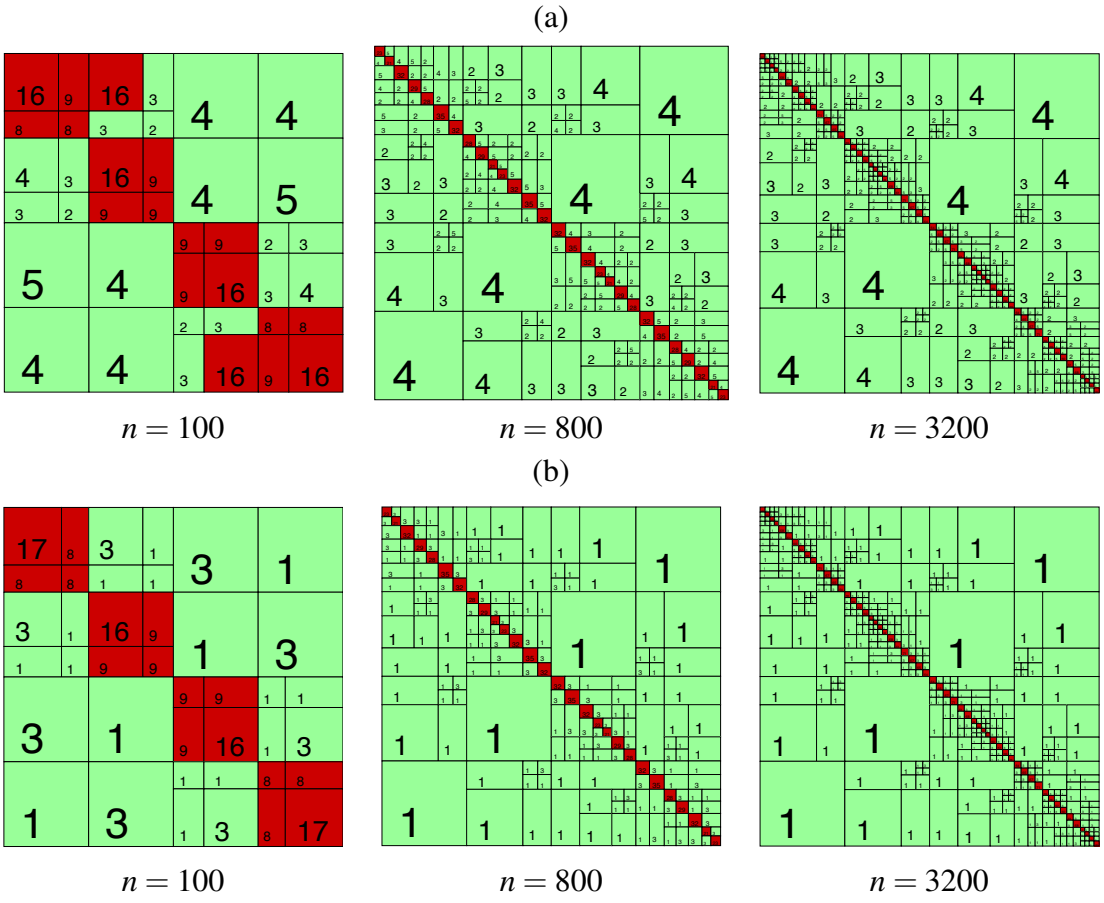

Figure 4.19: $\mathcal{H}$-matrix format of $\mathbf{V} \in \mathbb{R}^{n \times n}$ (a) and $\mathbf{K} \in \mathbb{R}^{n \times n}$ (b)

4.3.1 Operations and Complexity

Compared to other fast methods like FMM or wavelet methods, $\mathcal{H}$-matrices allow many more matrix operations than just the matrix-vector multiplication. It is important to stress that these operations show almost linear complexity, too [47]. Of course the accuracy of results carried out by these operations is determined by the accuracy of the matrix approximation. In Table 4.3.1 some operations and their complexity compared to full and sparse[4] matrices with size $n \times n$ is shown. In case of the latter, $m \geq n$ denotes the number of non-zeros in the matrix. A complete overview of complexities and algorithms for $\mathcal{H}$-matrices is given in the book of Hackbusch [52].

[4]There exist many different sparse matrix formats which are applied to different optimized matrix and matrix-vector operations. Therefore, the table should give just a rough idea of the performance of $\mathcal{H}$-matrices.

operation	$\mathcal{H}$-matrix			sparse	full
$\mathbf{M}\cdot\mathbf{v}$	$4n\log n - n + 2$	$\approx$	$\mathcal{O}(n\log n)$	$\mathcal{O}(m))$	$\mathcal{O}(n^2)$
$\mathbf{M}\cdot\mathbf{M}$	$13n\log^2 n + 65\log n - 51n + 52$	$\approx$	$\mathcal{O}(n\log^2 n)$	$\mathcal{O}(mn)$	$\mathcal{O}(n^{2.37})$
$\mathbf{LU}$	$^{11}/_2\log^2 n + 25n\log n - 57(n-1)$	$\approx$	$\mathcal{O}(n\log^2 n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^{2.38})$
$\mathbf{M}^{-1}$	$13n\log^2 n + 47n\log n - 109n + 110$	$\approx$	$\mathcal{O}(n\log^2 n)$	—	$\mathcal{O}(n^{2.38})$

Table 4.1: Complexity of operations in linear algebra compared between $\mathcal{H}$-, full and sparse matrices

As a fact, sparse and full matrices are a subset of the $\mathcal{H}$-matrix format and thus can be described with any loss of information. This motivated the construction of sparse preconditioners for systems of equations arising from FEM formulations [48]. Among others, clustering strategies have been developed which do not depend on any geometrical information, but only on the matrix graph [8]. Therefor the $\mathcal{H}$-matrix format is a flexible choice for algorithms covering numerical methods with different requirements in linear algebra.

Hierarchical Factorizations As mentioned in section 3.3, the coefficient matrices in (3.8) are ill-conditioned if the number of unknowns gets very large. For the considered applications in terms of large scale computations this is clearly the case. Thus, special care has to be taken for the solution procedure carried out by $\mathcal{H}$-matrices. Table 4.3.1 shows that the inverse and the *LU* factorization shows almost linear complexity and are computed in reasonable time, especially when the tolerated approximation error is set to a relatively high value, i.e. $\epsilon_{\mathcal{H}} = 10^{-1}$. For that reason these factorizations

$$\mathbf{P}_{\mathcal{H}}^{-1} = \mathbf{M}_{\mathcal{H}}^{-1}, \quad \text{or} \quad \mathbf{P}_{\mathcal{H}}^{-1} = (\mathbf{LU})_{\mathcal{H}}^{-1} = \mathbf{U}_{\mathcal{H}}^{-1}\mathbf{L}_{\mathcal{H}}^{-1}$$

can be used to as a spectrally equivalent preconditioner in (3.15). Of course, taking an appropriate accuracy for the hierarchical *LU* factorization the system of equations could be solved directly. This approach might especially be favored in solving multiple right hand sides like for the iterative plastic algorithm represented by (2.31).

By taking a formulation utilizing the CTO, an update of the inverse Jacobi matrix in (2.35) is necessary. As pointed out in [4], this could be achieved with minimal effort by adding low rank terms to the inverse or *LU* factorization subsequently.

4.3.2 Coarsening

The complexity of $\mathcal{H}$-matrix operations is also determined by depth of the underlying block cluster tree. It has been shown by Grasedyck [46], that after construction the structure of $\mathcal{H}$-matrices can be coarsened. This is due to the fact that the admissibility parameter η could be larger than chosen and that condition (4.6) might be sufficient but not necessary. Coarsening yields reduced complexity for matrix operations and less storage requirements as well.

If all children b_i for all $i \in \{1,\dots,4\}$ of a node $b = t \times s \in T_{I\times J}$ are leafs or small[5], then the corresponding matrix block $\mathbf{M}|_b$ is constructed based on the leaf matrix blocks $\mathbf{M}|_{b_i}$. Therefore, all leaf blocks are agglomerated [52] to a single $\mathcal{R}_k$-matrix

$$\bar{\mathbf{M}}|_b = \begin{pmatrix} \mathbf{M}|_{b_1} & \mathbf{M}|_{b_2} \\ \mathbf{M}|_{b_3} & \mathbf{M}|_{b_4} \end{pmatrix} = \begin{pmatrix} \mathbf{M}|_{b_1} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & \mathbf{M}|_{b_2} \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ \mathbf{M}|_{b_3} & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{M}|_{b_4} \end{pmatrix}$$

of rank $k = \sum_{i=1}^{4} \operatorname{rank}(\mathbf{M}|_{b_i})$. By plugging this matrix into the recompression algorithm by means of a QR factorization (4.16) as described in section 4.1.4, matrix $\mathbf{M}|_b$ is attained for a given accuracy, which is now a leaf in the $\mathcal{H}$-matrix and represented by an $\mathcal{R}_k$-matrix.

[5] Small enough in a sense that the representation of these blocks results in too much computational overhead with respect to the hierarchical representation

5 Implementation

As aforementioned, the main application of the resulting numerical software is the simulation of underground construction. In praxis of geotechnical engineering, software packages based on the FEM or finite difference method (FDM) are used. Although the BEM offers many advantages, the method is mainly used by the scientific community. For the last decade the computer software *BEFE++* has been developed at the institution related to the author[1], with the aim of closing this gap. Simulation of two or three-dimensional excavation problems [33], support measures like rock bolts and pipe roofs [87] or shotcrete shells and steel arches [86] have been implemented. Hence, it is possible to simulate the excavation process carried out in terms of the new Austrian tunneling method (NATM) [13]. Moreover, the software is capable of treating dynamic problems in the field of soil-structure and fluid-structure interaction [73, 84] by means of a coupled FEM-BEM solution. In *BEFE++* these numerical methods are also coupled to the particle based discrete element method (DEM). In addition, the BEM part of the software is capable to solve anisotropic elastic and scalar, inhomogeneous potential problems.

The aim of this work is to address the complexity problem of the BEM in the field of elastostatic and elasto-plastic analysis. This should be achieved by introducing an incorporated software solution in *BEFE++* which is reliably running on multi-core computational environments. The previous chapters gave an introduction to the numerical tools whose application is described in the following.

In this chapter the implemented algorithm for the elasto-plastic iteration scheme is presented which is subject to the fast solution procedure based on $\mathcal{H}$-matrices introduced in chapter 4. The approximation schemes presented in section 4.1 are well suited for scalar quantities. Thus the utilization of the numerical tools to elastostatic and elasto-plastic problems with vector valued quantities is specified including matrix operations and the solution procedure.

This chapter describes the implications on the implementation by means of $\mathcal{H}$-matrices. For the clustering, ACA, the matrix representation and for most operations of linear algebra the *HLib* software library [22] has been taken and utilized to fulfill the needs of the method. The setup for vector valued problems, the traverse of the matrix, the integration as well as the low rank approximation by means of interpolation, the block system of equations and its solution have been implemented in *BEFE++*and developed from scratch.

[1] Institute for Structural Analysis, Graz University of Technology, Austria, *www.ifb.tugraz.at*

5.1 Vector Valued Problems

Since ACA only works for scalar problems, the system matrices as well as their corresponding field variables need to get sorted with respect to their components and then treated separately [9]. A compulsory exercise to provide matrix entries originating from an asymptotically smooth kernel when using ACA. Consider a matrix block representing a part of the discrete single layer potential $\mathbf{V}|_{t\times s}$ in two-dimensional elastostatics, assembled like in a standard collocation method. As shown in Figure 5.1 the algorithm finds a pivot

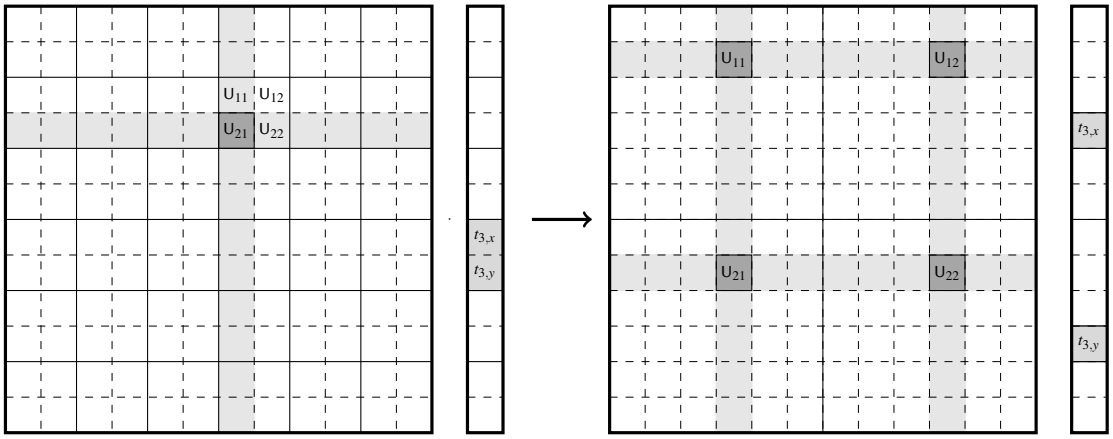

Figure 5.1: ACA and a matrix block with non-smooth (a) and smooth (b) kernel entries

entry related to U_{21} in the fourth row and 7th column. The entries of the row depend on integrals of the 21- *and* 22-components of Kelvin's fundamental solution U. Naturally, these entries differ for same $|\boldsymbol{x}-\boldsymbol{y}|$ and hence jump with respect to the row. The same holds for the components in the column. For sure, ACA will not fail since the algorithm still reveals the rank of the block or at least the full rank. But the singular values might not decay exponentially, and thus the compression rate may be worse. Hence, system matrices are split

$$\mathbf{V}|_{t\times s} = \begin{pmatrix} \mathbf{V}_{xx}|_{t\times s} & \mathbf{V}_{yx}|_{t\times s} \\ \mathbf{V}_{yx}|_{t\times s} & \mathbf{V}_{yy}|_{t\times s} \end{pmatrix}$$

as proposed in the beginning of this section and depicted on the right in Figure 5.1. Now ACA acts on $\mathcal{R}_k$-matrix blocks in every sub-matrix $\mathbf{M}_{ij} = \mathbf{V}_{ij}|_{t\times s}$ consisting of integrals of one component U_{ij} which is guaranteed to be asymptotically smooth (4.11). In this context, the subscripts ij correspond to the components of the matrix valued kernels originating from the fundamental solution tensors.

For all system matrices in (3.8) this leads to a block-wise representation

$$\mathbf{V} = \begin{pmatrix} \mathbf{V}_{xx} & \mathbf{V}_{xy} & \mathbf{V}_{xz} \\ \mathbf{V}_{yx} & \mathbf{V}_{yy} & \mathbf{V}_{yz} \\ \mathbf{V}_{zx} & \mathbf{V}_{zy} & \mathbf{V}_{zz} \end{pmatrix}, \quad \mathbf{K} = \begin{pmatrix} \mathbf{K}_{xx} & \mathbf{K}_{xy} & \mathbf{K}_{xz} \\ \mathbf{K}_{yx} & \mathbf{K}_{yy} & \mathbf{K}_{yz} \\ \mathbf{K}_{zx} & \mathbf{K}_{zy} & \mathbf{K}_{zz} \end{pmatrix} \tag{5.1}$$

and

$$\mathbf{N} = \begin{pmatrix} \mathbf{N}_{x11} & \mathbf{N}_{x22} & \mathbf{N}_{x33} & \mathbf{N}_{x12} & \mathbf{N}_{x13} & \mathbf{N}_{x23} \\ \mathbf{N}_{y11} & \mathbf{N}_{y22} & \mathbf{N}_{y33} & \mathbf{N}_{y12} & \mathbf{N}_{y13} & \mathbf{N}_{y23} \\ \mathbf{N}_{z11} & \mathbf{N}_{z22} & \mathbf{N}_{z33} & \mathbf{N}_{z12} & \mathbf{N}_{z13} & \mathbf{N}_{z23} \end{pmatrix}, \tag{5.2}$$

for the matrices in (3.11) to

$$\mathbf{V}' = \begin{pmatrix} \mathbf{V}'_{11x} & \mathbf{V}'_{11y} & \mathbf{V}'_{11z} \\ \mathbf{V}'_{22x} & \mathbf{V}'_{22y} & \mathbf{V}'_{22z} \\ \mathbf{V}'_{33x} & \mathbf{V}'_{33y} & \mathbf{V}'_{33z} \\ \mathbf{V}'_{12x} & \mathbf{V}'_{12y} & \mathbf{V}'_{12z} \\ \mathbf{V}'_{13x} & \mathbf{V}'_{13y} & \mathbf{V}'_{13z} \\ \mathbf{V}'_{23x} & \mathbf{V}'_{23y} & \mathbf{V}'_{23z} \end{pmatrix} \quad \text{and} \quad \mathbf{N}' = \begin{pmatrix} \mathbf{N}'_{11,11} & \mathbf{N}'_{11,22} & \cdots & \mathbf{N}'_{11,23} \\ \mathbf{N}'_{22,11} & \mathbf{N}'_{22,22} & & \\ \vdots & & \ddots & \\ \mathbf{N}'_{23,11} & & & \mathbf{N}'_{23,23} \end{pmatrix}. \tag{5.3}$$

This has to be done for all field variables too, resulting in system vectors

$$\begin{aligned} \mathbf{u} &:= \left([u_{x,n}]_{n=1}^{N_u} \quad [u_{y,n}]_{n=1}^{N_u} \quad [u_{z,n}]_{n=1}^{N_u} \right)^{\mathsf{T}} \\ \mathbf{t} &:= \left([t_{x,n}]_{n=1}^{N_t} \quad [t_{y,n}]_{n=1}^{N_t} \quad [t_{z,n}]_{n=1}^{N_t} \right)^{\mathsf{T}} \\ \boldsymbol{\sigma} &:= \left([\sigma_{11,n}]_{n=1}^{N_\sigma} \quad [\sigma_{22,n}]_{n=1}^{N_\sigma} \quad [\sigma_{33,n}]_{n=1}^{N_\sigma} \quad [\sigma_{12,n}]_{n=1}^{N_\sigma} \quad [\sigma_{13,n}]_{n=1}^{N_\sigma} \quad [\sigma_{23,n}]_{n=1}^{N_\sigma} \right)^{\mathsf{T}} \end{aligned} \tag{5.4}$$

in three dimensions. Here, $N_\cdot$ denotes the number of points where these values exist corresponding to the correlated system matrix. The structure of $\mathbf{K}'$ is similar to $\mathbf{V}'$ and therefore omitted. Each block is represented by an $\mathcal{H}$-matrix. Thus, the construction of cluster and block cluster trees for the sub-matrices is an important issue. Since the clustering is based on characteristic points (see section 4.2), the corresponding trees T_I and T_J as well as $T_{I\times J}$ are the same. This fact leads to an optimized integration procedure which is described later on. However, if boundary conditions of an element differ with respect to the spatial coordinates like for roller bearings or symmetry boundary conditions, the characteristic points per direction are different for matrices $\mathbf{M}_{ij}$ in (5.1) and so is the structure of $T_{I\times J}$. Such boundary conditions are preferably formulated in terms of a saddle point problem, a byproduct of the proposed coupling strategy for $\mathcal{H}$-matrices in section 5.3.

5.2 Matrix Construction

Frequently, the construction of the system matrices is the most time consuming part in the analysis. For that matter, only parts of the matrices on the right hand side correlated to non-zero boundary conditions of the integral equation (3.8) are assembled. Therefore, free surfaces with zero tractions or regions of fixed support are not considered in the cluster trees determining these matrices.

Traverse of Tree Structure Usually, the number of degrees of freedom n_Γ determined by the collocation points on the boundary differs from the number of stresses n_Ω to be evaluated at stress points. As a result, the matrices $\mathbf{N} \in \mathbb{R}^{n_\Gamma \times n_\Omega}$, $\mathbf{V}' \in \mathbb{R}^{n_\Omega \times n_\Gamma}$ and $\mathbf{K}' \in \mathbb{R}^{n_\Omega \times n_\Gamma}$ are of rectangular shape. On that account, the number of indices in the same level of the row and column clusters may differ widely and the resulting matrix partition reveals rectangular matrix blocks too. Therefore, the approximation may be improved by checking admissible blocks in different levels of T_I and T_J resulting to a block inhomogeneous $T_{I\times J}$. Naturally, the tree construction time is increased. For matrices carried out in the results of chapter 6, the additional construction time is negligible compared to integration time.

Because of the separation of matrix entries in blocks, information is lost about their geometrical association without any countermeasures. In order to reunite the trees, the traversing of the matrix structure has been implemented in a particular way. For all associated sub-matrices $\mathbf{M}_{ij}$ the block cluster tree $T_{I\times J}$ is the same and so are their leafs $b_\ell \in T_{I\times J}$. Traversing the correlated trees $T_{I\times J,ij}$ simultaneously, they are linked together and stored in a group of associated leafs $\tilde{b}_\ell = \{b_\ell : b_\ell = b \in T_{I\times J,ij}\}$.

Coarsening and Recompression Related to the traverse of the block cluster trees is the application of a three-stage recompression technique too. First, every admissible matrix block $\mathbf{M}_{ij}|_{b_\ell} \in \mathbb{R}^{r\times c}$ is subject to a truncated SVD as described in section 4.1.4. Second, every node $b_c \in T_{I\times J}$ whose children are all admissible leafs, is a candidate for the coarsening procedure described in section 4.3.2. Third, if the rank of the resulting $\mathcal{R}_k$-matrices is in the magnitude of the block size $n = \min\{r,c\}$ where $k \gtrsim n/2$, then the representation as a full matrix is more efficient and cheaper in terms of storage. Thus the factorization is expanded. All these recompression techniques happen in place to avoid storage overheads. As a consequence, the traverse algorithm collects all inadmissible leafs b_f, nodes b_c, all nodes b_ℓ whose fathers are not of type b_c and connect them to $\tilde{b}_f$, $\tilde{b}_c$ and $\tilde{b}_\ell$. The corresponding implementation is illustrated in Algorithm 4.

Integration Another impact of subdivided system matrices is the fact that directional data $|\boldsymbol{x}-\boldsymbol{y}|$ needs to be calculated multiple times if implemented in a naive way. This is optimized by putting all $\tilde{b}_f$, $\tilde{b}_c$ and $\tilde{b}_l$ to a stack, which can be treated in parallel by assigning a group of associated leafs to one single processor. For every $\tilde{b}$ the directional data requested by ACA, hence the complete matrix based kernel, is stored temporarily in

Algorithm 4 The traverse of block cluster tree $T_{I\times J}$

```
1: procedure TRAVERSE(b̃)
2:     if all children of all b_i ∈ b̃ are leafs and (4.6) holds then
3:         INTEGRATECOARSEN(b̃)
4:     else
5:         b̃_ℓ = ∅, b̃_f = ∅, b̃_i = ∅
6:         for all children of b_i ∈ b̃ do
7:             if b_i is a leaf then
8:                 if (4.6) holds for b_i then
9:                     add b_i to b̃_ℓ
10:                else
11:                    add b_i to b̃_f
12:                end if
13:            else
14:                add b_i to b̃_i
15:            end if
16:        end for
17:        TRAVERSE(b̃_i)
18:        INTEGRATERECOMPRESSLOWRANK(b̃_ℓ)
19:        INTEGRATEFULL(b̃_f)
20:    end if
21: end procedure
```

a container. If the ACA algorithm requests rows or columns corresponding to the same characteristic points in different sub-blocks, they are instantly taken from that container. In case of a pivoting strategy where the first pivot point is chosen to be the barycenter of the bounding box B, this is the case at least once. It has been observed, that the pivots overlap in many instances and therefore the procedure is accelerated significantly.

For the construction of $\mathcal{R}_k$-matrices by kernel interpolation the application is straightforward. Since kernel evaluations are carried out only between the interpolation points $|\bar{x}_\nu - \bar{y}_\mu|$ for matrix $\mathbf{S}$ in the low-rank representation $\mathbf{VSW}^\mathsf{T}$ (4.14), the entries are spread to the corresponding $\mathbf{M}_{ij}$ instantly. It is also interesting to see that for the discrete operators which do not involve the $\mathcal{T}$ (2.4), the entries of $\mathbf{W}$ are the same for all $\mathbf{M}_{ij}$. However, pivots of $\mathbf{S}$ come into play if HCA is taken to construct $\mathcal{R}_k$-matrices and for the proposed matrix structure the stack of $\tilde{b}$ is needed again. Notably, if the low rank approximations are set up by means of kernel interpolation alone, the proposed matrix partition would not be necessary. For entries of $\mathbf{W}^\mathsf{T}$, the integration of the interpolant is carried out exact with the proper choice of Gauss rule.

As a consequence of the issues addressed in this section, the implemented approach seems to be the most flexible one. Thus, the construction of $\mathcal{R}_k$-matrices with ACA or by kernel interpolation is easily exchangeable with respect to the implemented structure of $\mathcal{H}$-matrices.

5.3 Matrix Operations

Due to the matrix representation presented in section 5.1, an additional stage of nesting is introduced for the block solution procedures of section 3.3. The same holds for many other operations in terms of linear algebra. In the realization of these operations, the flexibility of the *HLib* is greatly exploited. An additional tree structure representing $\mathbf{M}_{ij}$ is put over, thus the block procedures already implemented in the software library are taken and adopted.

Kriemann [63] presented the parallelization of the most important $\mathcal{H}$-matrix operations for shared-memory systems. However, this implementation is only available in the commercial version of the *HLib*. Thus, a parallel version of the matrix-vector product has been utilized exploiting the sub-matrix structure of the described system matrices.

Remark: It has been observed that the performance of linear algebra operations with the *HLib* might slow down if a parallel *LAPACK* [1] implementation is used. This is due to the fact, that blocks $\mathbf{M}|_b$ of an $\mathcal{H}$-matrix are usually relatively small. Thus the computational overhead preparing threads undoes the benefit of using more central processing units (CPUs).

Schur Complement In order to couple the presented methodology to other numerical methods, a coupling strategy by means of Lagrange multipliers [92] is suitable. The proposed approach works well if boundary and finite element methods are used simultaneously. However, for a pure BEM coupling the all-floating boundary element tearing and

interconnecting (BETI) method [77] is the method of choice. Especially unbounded domains are problematic with domain decomposition approaches, thus a combination with with the proposed fast formulation treating such domains suggests itself. In this formulation non-linear behavior is only present in form of additional terms on the right hand side of the system of equations. Thus, the application to BETI is straightforward. By using the all-floating approach, the Schur complement (3.16) simply yields to

$$\mathbf{S} = \mathbf{V}^{-1}\mathbf{K}$$

since $\Gamma_D = \emptyset$ for all sub-domains. Dirichlet boundary conditions like roller bearings are treated like special sub-domains and taken into account by the global solver. Therefor, problems with mixed boundary conditions are treated without destructing the proposed block structure of system matrices. The described method has been implemented in *BEFE++* and is current field of research of different authors.

5.4 Explicit Elasto-Plastic Algorithm

With the system matrices in $\mathcal{H}$-matrix format in hand, the elasto-plastic analysis is started. As described in section 2.3 different numerical approaches are possible to solve the non-linear, inhomogeneous system of equations. However, an explicit initial stress iteration technique with constant Jacobi matrix J_0 is taken, accepting lower convergence rates. Implementing (2.29) and (2.30) directly is not necessary and would imply elaborate operations with system matrices. Hence, the needed quantities are calculated in place using the discrete system of equations and the RRA. The latter is used as a black box and for the simulation of soil and rock, models described in [42, 86] are taken.

The step-wise solution of linearized problems is incorporated as follows. First, the total external load $\mathbf{g} = \begin{pmatrix}\mathbf{g}_D & \mathbf{g}_N\end{pmatrix}$ is applied and a purely elastic solution is carried out (3.8) by setting $\boldsymbol{\sigma}_p = \mathbf{0}$. With (3.11) the resulting stresses are calculated at all stress points in the domain or on the boundary by means of stress recovery. The stress state is compared with the yield stress of the material, resulting in a load factor which determines the factorization of the load increments $\mathbf{g}^s$ per load step s [40]. Depending on the robustness of the taken material law, the number of load increments S might be set individually or even to $S = 1$. For all load steps $s \in \{1, \ldots, S\}$ the following procedure is the same.

The algorithm is starting with the initialization denoted by the iteration step $n = 0$, which occurs as a subscript in the involved quantities. While still $\boldsymbol{\sigma}_{p,0} = \mathbf{0}$, a linear solution $\mathbf{t}_0$ and $\mathbf{u}_0$ is found initially. Using (3.11) again, the total stress state $\boldsymbol{\sigma}_0$ for the current load step is determined and checked against the yield criterion (2.28) returning an admissible stress state. By using (2.19) and (2.32) the first initial stress state $\dot{\boldsymbol{\sigma}}_{p,1}$ at iteration step $n = 1$ is obtained.

What follows is an iterative process where (3.8) and (3.11) are used in their incremental

form

$$\begin{pmatrix} \mathbf{V}_{DD} & -\mathbf{K}_{DN} \\ \mathbf{V}_{ND} & -\mathbf{K}_{NN} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{t}}_{D,n} \\ \dot{\mathbf{u}}_{N,n} \end{pmatrix} = \begin{pmatrix} \mathbf{N}_{DP} \\ \mathbf{N}_{NP} \end{pmatrix} \dot{\boldsymbol{\sigma}}_{p,n} \tag{5.5}$$

and

$$\dot{\boldsymbol{\sigma}}_n = \mathbf{V}'\dot{\mathbf{t}}_n - \mathbf{K}'\dot{\mathbf{u}}_n + \mathbf{N}'\dot{\boldsymbol{\sigma}}_{p,n}. \tag{5.6}$$

In (5.5) the known Cauchy data is not present anymore since the equilibrating forces arise from initial stresses only. With $\dot{\boldsymbol{\sigma}}_{p,n}$ on the right hand side the full set of incremental Cauchy data $\dot{\mathbf{t}}_n$ and $\dot{\mathbf{u}}_n$ is determined and used in (5.6) to find the stress increment $\dot{\boldsymbol{\sigma}}_n$. Now the elastic trial stress (2.20) is calculated and with (2.28) the new correcting stress increments $\dot{\boldsymbol{\sigma}}_{p,n+1}$. The procedure is repeated until

$$\frac{\|\dot{\boldsymbol{\sigma}}_{p,n}\|}{\|\dot{\boldsymbol{\sigma}}_{p,1}\|} < \epsilon_\sigma \tag{5.7}$$

where ϵ_σ denotes the allowed tolerance for equilibrium. The initial stress iteration technique is summarized in Algorithm 5.

Algorithm 5 Initial stress iteration

procedure PLASTICITERATION
  determine S and w_s $\forall s \in \{1,\dots,S\}$
  $\boldsymbol{\sigma}_p^{s=1} = \mathbf{0}$, $\boldsymbol{\sigma}^1 = \mathbf{0}$
  for all $s \in \{1,\dots,S\}$ **do**
    calculate $\mathbf{g}^s$
    $n := 0$
    $\mathbf{t}_n, \mathbf{u}_n \leftarrow$ (3.8) with $\boldsymbol{\sigma}_{p,0} = \boldsymbol{\sigma}_p^s$
    $\boldsymbol{\sigma}_0 \leftarrow$ (3.11)
    $\dot{\boldsymbol{\sigma}}_{p,1} \leftarrow \mathrm{RRA}(\boldsymbol{\sigma}_0, \boldsymbol{\sigma}^s)$
    $n := 1$
    while condition (5.7) $= false$ **do**
      $\dot{\mathbf{t}}_n, \dot{\mathbf{u}}_n \leftarrow$ (5.5)
      $\dot{\boldsymbol{\sigma}}_n \leftarrow$ (5.6)
      $\dot{\boldsymbol{\sigma}}_{e,n} \leftarrow$ (2.20)
      $\dot{\boldsymbol{\sigma}}_{p,n+1} \leftarrow \mathrm{RRA}(\dot{\boldsymbol{\sigma}}_{e,n}, \boldsymbol{\sigma}_n)$
      $\mathbf{t}_n := \mathbf{t}_n + \dot{\mathbf{t}}_n$, $\mathbf{u}_n := \mathbf{u}_n + \dot{\mathbf{u}}_n$, $\boldsymbol{\sigma}_n := \boldsymbol{\sigma}_n + \dot{\boldsymbol{\sigma}}_n$, $\boldsymbol{\sigma}_p^s := \boldsymbol{\sigma}_p^s + \dot{\boldsymbol{\sigma}}_{p,n}$
      $n := n + 1$
    end while
    $\boldsymbol{\sigma}^s = \boldsymbol{\sigma}_n$
  end for
end procedure

6 Numerical Results

The following numerical results show the solution of elastic homogeneous and inhomogeneous as well as elasto-plastic problems taking the fast BEM by means of $\mathcal{H}$-matrices as described in the previous chapters. To test the efficiency, all calculations are carried out with the software package *BEFE++* and the implementation described in chapter 5. The algorithm is crosschecked by two academic examples in two and three dimensions. Statically loaded by means of vertical and internal forces, a beam is analyzed and verified with analytic solutions. The successful application to the initial stress iteration technique as described in section 5.4 is demonstrated by means of a circular tunnel excavation. Finally, the effectivity of the matrix approximation is studied with the examples of a tunnel cross passage and a pile foundation in a half-space.

All tests are realized on a *SunFire X4600 M2* computer with 8 *AMD Opteron* CPUs running at 2.4 GHz core speed under *Debian Linux*. The *LAPACK* [1] implementation has been taken from the *AMD Core Math Library* (*ACML*) [76]. Although most parts of the code use shared memory parallelization implemented by means of the *OpenMP* specification [80], the calculations are done in serial with one CPU if not otherwise noted. This guarantees a better comparison related to speedup factors of matrix construction and operations.

To measure the performance of the $\mathcal{H}$-matrix algebra, the compression factor

$$c_{\mathcal{H}} = \frac{\mathsf{St}(\mathbf{M}_f)}{\mathsf{St}(\mathbf{M}_{\mathcal{H}})} \tag{6.1}$$

and the speedup factor

$$s_{\mathcal{H}} = \frac{t_f}{t_{\mathcal{H}}} \quad \text{with} \quad t_f = t(\mathsf{Op}(\mathbf{M}_f)), \quad t_{\mathcal{H}} = t(\mathsf{Op}(\mathbf{M}_{\mathcal{H}})), \tag{6.2}$$

are introduced. The subscripts denote quantities originating from fully populated matrices $\mathbf{M}_f$ and data sparse $\mathcal{H}$-matrices $\mathbf{M}_{\mathcal{H}}$ respectively. For algorithms with optimal linear complexity the factors should behave like $\mathcal{O}(n)$ if $n \to \infty$ for the matrix-vector product and the storage requirements. In this chapter, n denotes the size of the corresponding system matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$. For the purpose of better comparison between the different system matrices, a factor

$$b = n_\Omega / n_\Gamma$$

is defined. The factor denotes the ratio of stress equations n_Ω and the degrees of freedom n_Γ on the boundary mentioned in chapter 5. Therefore, the system matrices $\mathbf{N} \in \mathbb{R}^{n_\Gamma \times n_\Omega}$,

$\mathbf{V}' \in \mathbb{R}^{n_\Omega \times n_\Gamma}$ and $\mathbf{K}' \in \mathbb{R}^{n_\Omega \times n_\Gamma}$ have been treated by a fictitious size setting

$$n = \sqrt{b n_\Gamma^2} \tag{6.3}$$

as if they would be in $\mathbb{R}^{n \times n}$.

6.1 Beam

To verify the accuracy of the constructed matrices as well as to compare different discretization schemes the academic example of a linear elastic beam is investigated in this section. The beam is depicted in Figure 6.1. The geometry, material data as well as the applied forces are given in Table 6.1. Young's modulus E, cross section A and the geo-

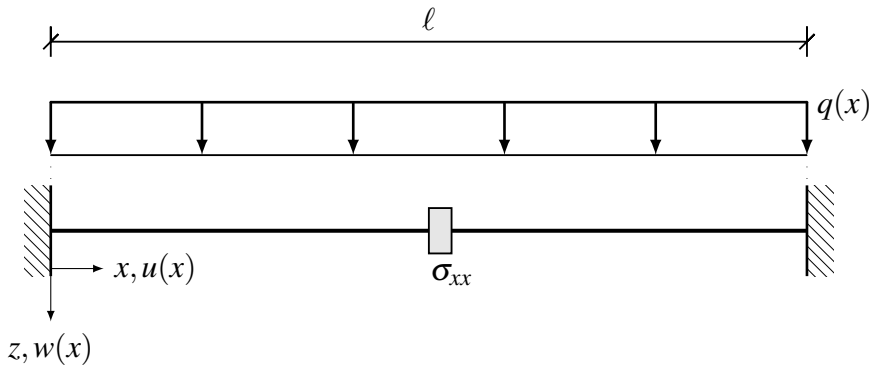

Figure 6.1: Beam subject to vertical loading and under constraint induced by initial stress

length	$\ell = 10\,\mathrm{m}$
cross section	$A = 1 \times 1\,\mathrm{m}^2$
area moment of inertia	$I = 0.0833\,\mathrm{m}^4$
Young's modulus	$E = 2.1 \cdot 10^8\,\mathrm{kN/m^2}$
Poisson's ratio	$\nu = 0$
vertical load	$q_0 = 10\,\mathrm{kN}$
initial stress	$\sigma_{xx} = 50000\,\mathrm{kN/m^2}$

Table 6.1: Geometry and material properties of the beam

metrical moment of inertia as well as the vertical loading $q = q_0$ and the initial stress σ_{xx} are considered to be constant along the beam. Since the ration of height length is $1/10$, the Euler-Bernoulli beam theory is applicable for comparison. Therefore, the problem is

described by the boundary value problem: find the deflection $w(x)$ so that

$$\left(EI\frac{\partial^4 w}{\partial x^4}\right)(x) = q(x) \qquad \forall x \in \Omega = \{0,\dots,\ell\}$$
$$\frac{\partial w}{\partial x}(0) = \frac{\partial w}{\partial x}(\ell) = 0$$
$$w(0) = w(\ell) = 0$$

The behavior in axial direction is decoupled from the beam theory and described by the boundary value problem (1.1). Since the initial stresses are evenly distributed along the beam, no axial displacements u occur and their influence is indicated by constant internal normal forces N and reacting forces on Γ_D. The solution for the considered problem is

$$\begin{aligned}
w(x) &= \frac{q_0\ell}{24EI}\left(\frac{x}{l}\right)^2\left(1-2\frac{x}{\ell}+\left(\frac{x}{\ell}\right)^2\right),\\
u(x) &= 0,\\
N(x) &= A\sigma_{xx},\\
V(x) &= \frac{q_0\ell}{2}\left(1-2\frac{x}{\ell}\right),\\
M(x) &= -\frac{q_0\ell^4}{12}\left(1-6\frac{x}{\ell}+6\left(\frac{x}{\ell}\right)^2\right)
\end{aligned}$$

where V is the shear force and M the bending moment.

The considered problem is modeled in three dimensions as shown in Figure 6.2. Due to

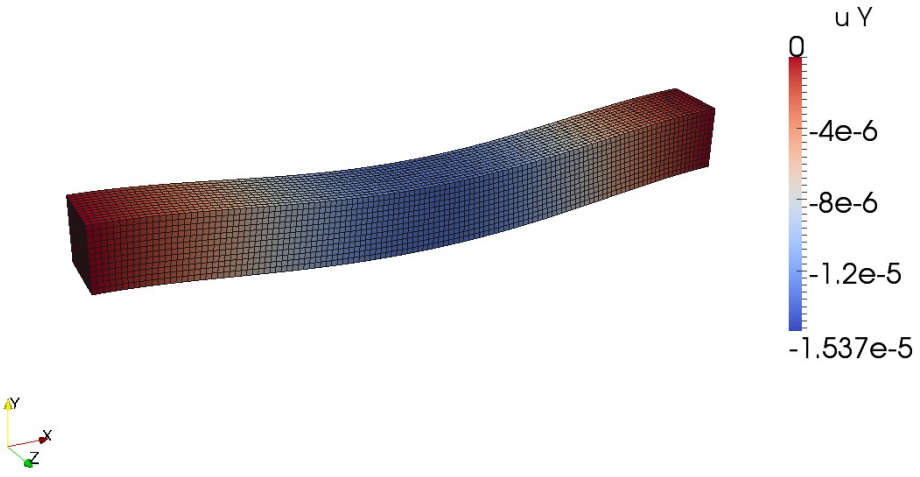

Figure 6.2: Deflected beam by means of quadrilateral and hexahedral elements

the external load and the applied initial stress, all system matrices in (3.8) and (3.11) are used and a mixed boundary value problem shall be solved. Again, attention is invited to the

fact that only matrix parts on the right hand side related to non-zero boundary conditions are assembled. The parameters for the construction of the $\mathcal{H}$-matrix approximation are constant throughout this study and given in Table 6.1. All rectangular shaped system matrices $\mathbf{M} \in \mathbb{R}^{r \times c}$ with $r \neq c$ are based upon a inhomogeneous block cluster tree. Matrices where $r = c$ are constructed by level consistent block cluster trees. The minimal leaf size is set to $n_{min} = 50$ for all examples.

approximation tolerance	$\epsilon_{\mathcal{H}} = 10^{-4}$
minimal leaf size	$n_{min} = 50$
solver accuracy	$\epsilon_{ks} = 10^{-5}$
admissibility factor	$\eta = 2$
clustering strategy	geometrically balanced

Table 6.2: Parameters for $\mathcal{H}$-matrix construction of the beam example

The beam is discretized by quadrilateral or triangular elements for the boundary approximation. In order to apply initial stresses, hexahedral or tetrahedral cells are taken respectively. The deflected beam for both types of discretization is shown in Figure 6.2 and Figure 6.3. Naturally, the order of displacement interpolation is chosen to be linear or

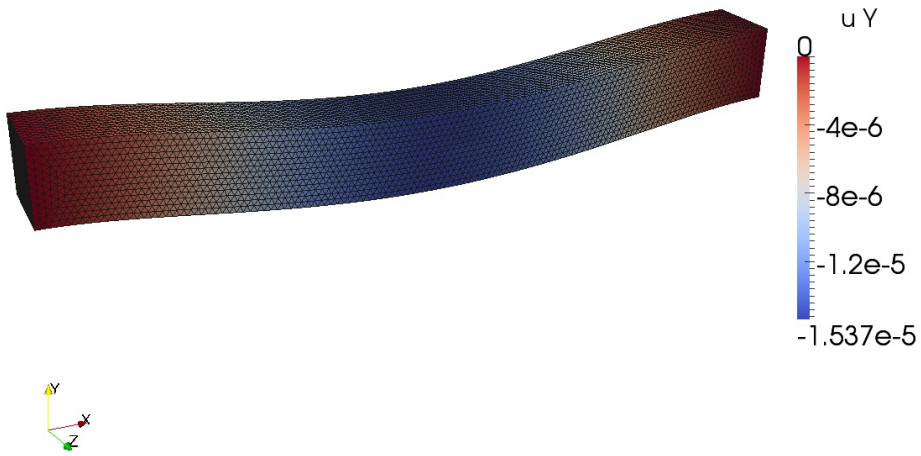

Figure 6.3: Deflected beam by means of triangular and tetrahedral elements

quadratic in S_h. The piecewise interpolation functions for the traction and stress field are always one order less and in S_h^-.

In order to asses the numerical results, the beam deflection along the x-axis is shown in Figure 6.4 for discretizations with quadrilaterals and triangles with linear and quadratic trial functions for u respectively. Obviously, the simulations converge to the Euler-Bernoulli solution with increasing degrees of freedom n. Interestingly, quadrilaterals with quadratic trial functions for u and linear for t, show larger deflections with less elements than the

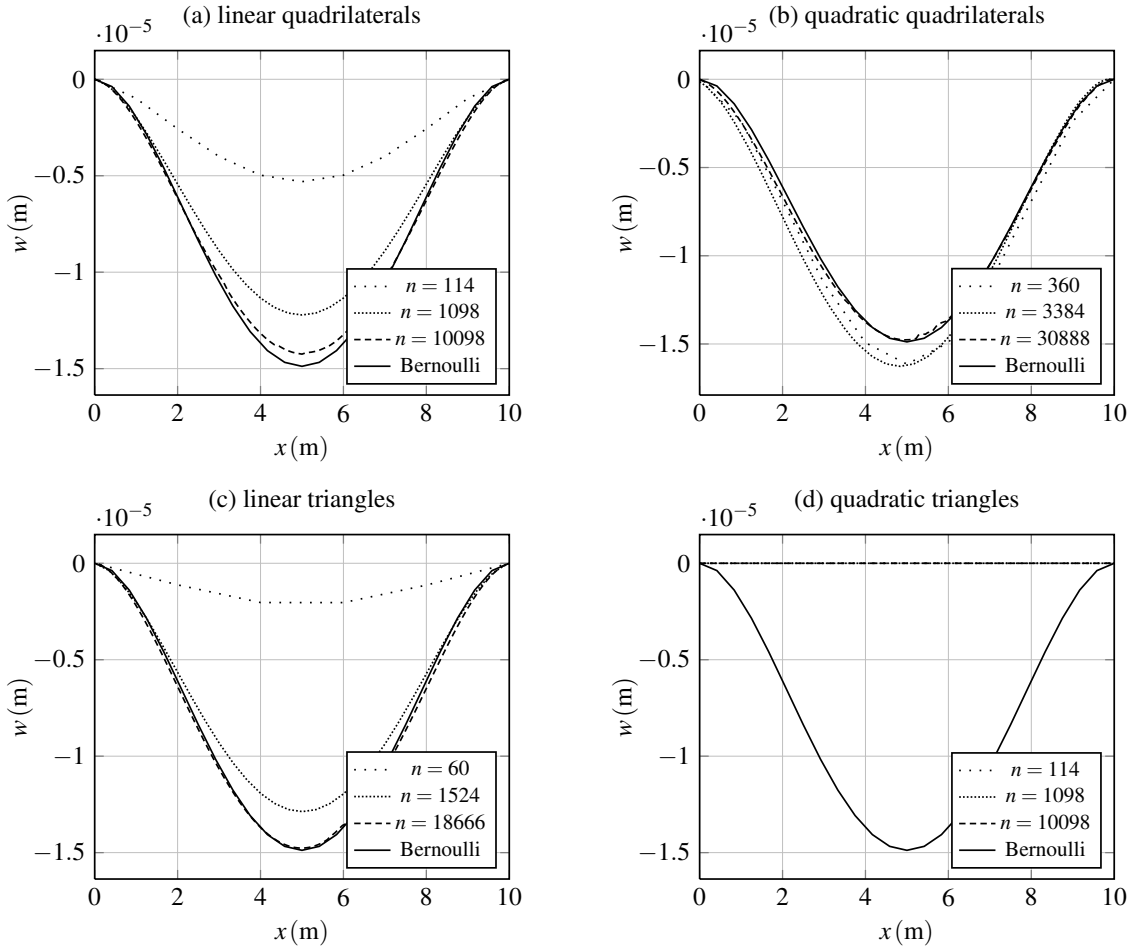

Figure 6.4: Beam deflection $w(x)$ for different discretizations

converged solution as shown in Figure 6.4(b). Internal forces have been verified for constant quadrilateral cells in Figure 6.5. Although constant trial functions for the stresses are taken, the internal forces show good agreement with the Euler-Bernoulli beam. Therefore, the matrix approximation does not affect the total results.

Now, matrix properties are investigated. First, the compression factors (6.1) of the approximated discrete single and double layer potentials by means of ACA are examined. The results for different approximation schemes are shown in Figure 6.6. It is observed, that there are almost no differences between triangular or quadrilateral discretization. Except for linear triangles and the right hand side (RHS) matrix which corresponds to parts of the discrete single layer potential $\mathbf{V}$ in this example. A remarkable better performance is observed and can only be partially explained by the smaller support of their shape functions. As expected due to the larger support, the left hand side (LHS) mainly consisting of double layer potential matrices does not compress as well as the RHS for both, linear

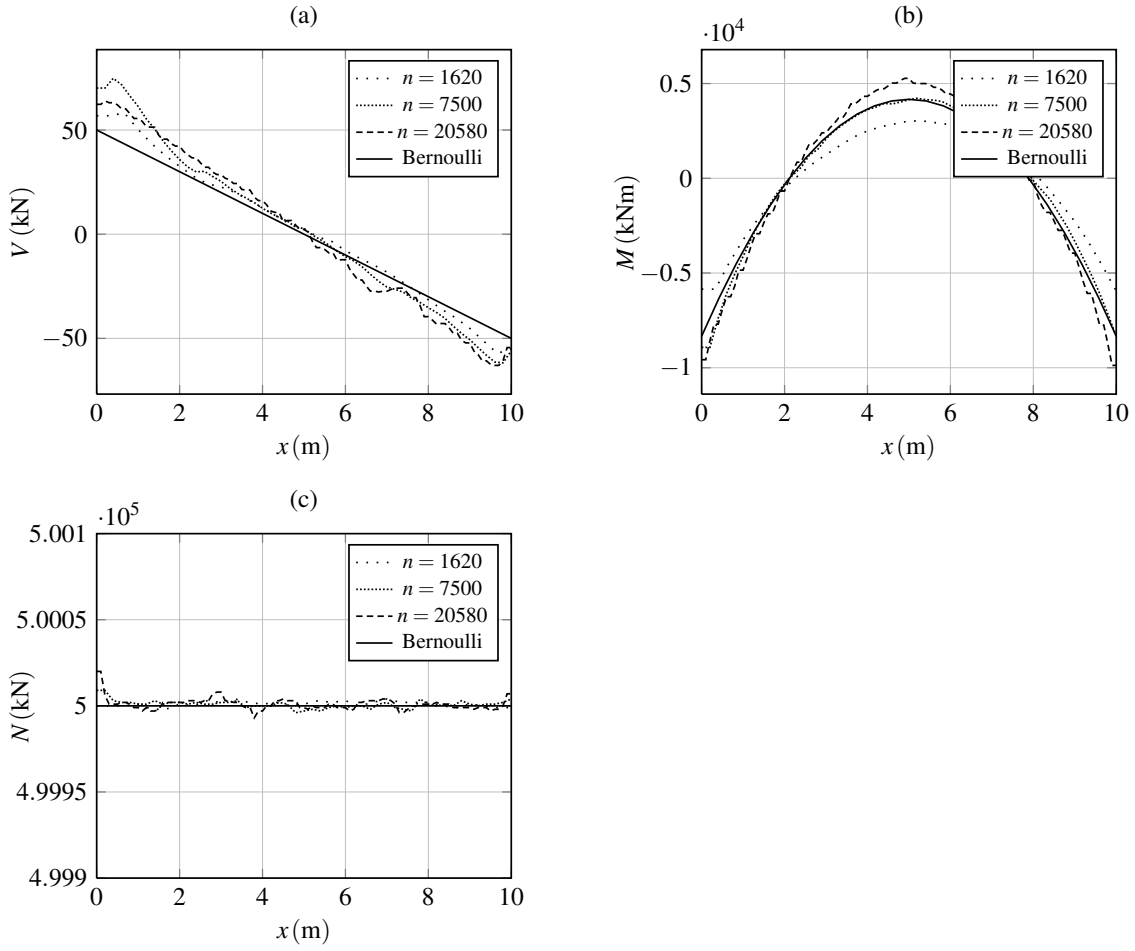

Figure 6.5: Beam internal forces for different numbers of cells along the cut line n_c

and quadratic displacement interpolation respectively. The overall behavior for the smaller matrices in Figure 6.6(a) is good and nearly linear for the larger matrices in Figure 6.6(b) like expected.

The compression factors of system matrices related to the stress field or the stress equation (3.11) are investigated in Figure 6.7 for different element types and the previously explained combination of linear and constant trial functions. The plot for the discrete Newton potential $\mathbf{N}$ and volume potential $\mathbf{N}'$ are shown in Figure 6.7(a). The former show the expected compression rates. In comparison, the compression of the latter is reduced. For the geometry of this example, the performance of $\mathbf{V}'$ and $\mathbf{K}'$ are in between the Newton and volume potential.

Summarizing the sparsity of the system matrices carried out by means of ACA, the discrete boundary integral operators of the first equation show the expected performance and so does the Newton potential. Since the collocation points are in the considered do-

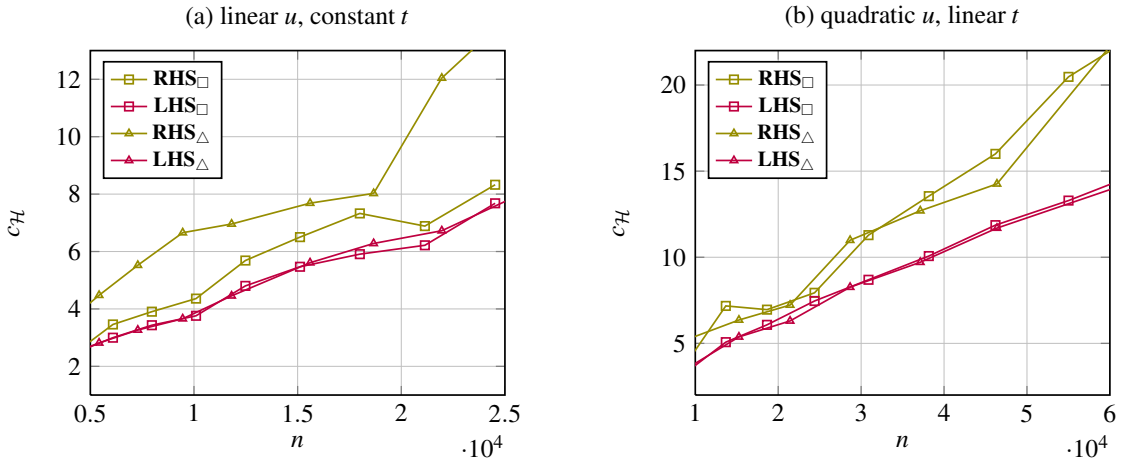

Figure 6.6: Compression factors for system matrices with different discretization schemes

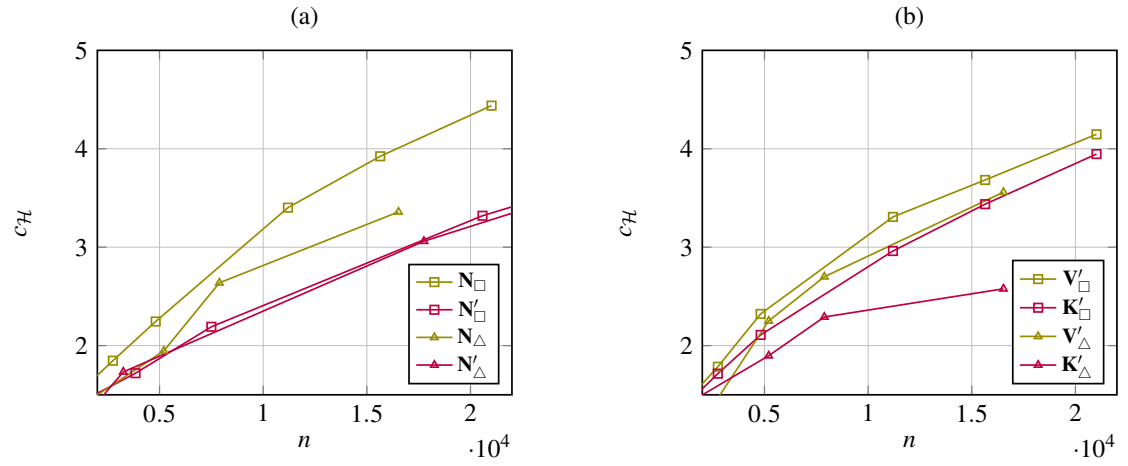

Figure 6.7: Compression factors for domain matrices with different discretization schemes

main and one characteristic point is associated to 6 stress values, the compression of all matrices in (3.11) is reduced whereas the volume potential $\mathbf{N}'$ ends up with the weakest performance.

The kernel interpolation technique is now applied to the same task. The numerical results are based on linear and constant interpolation for physical quantities. The order of kernel interpolation is set to 3, resulting to a fixed rank $k = 3^3 = 27$ of the approximated matrix blocks beforehand. The recompression and coarsening technique described in section 4.3.2 and section 5.2 has been applied to both, ACA and kernel interpolation. The outcome is shown in Figure 6.8 for comparison. Although both approaches show similar capabilities for matrices related to the boundary only, kernel interpolation seems to perform better if domain terms are involved. The most significant differences are observed

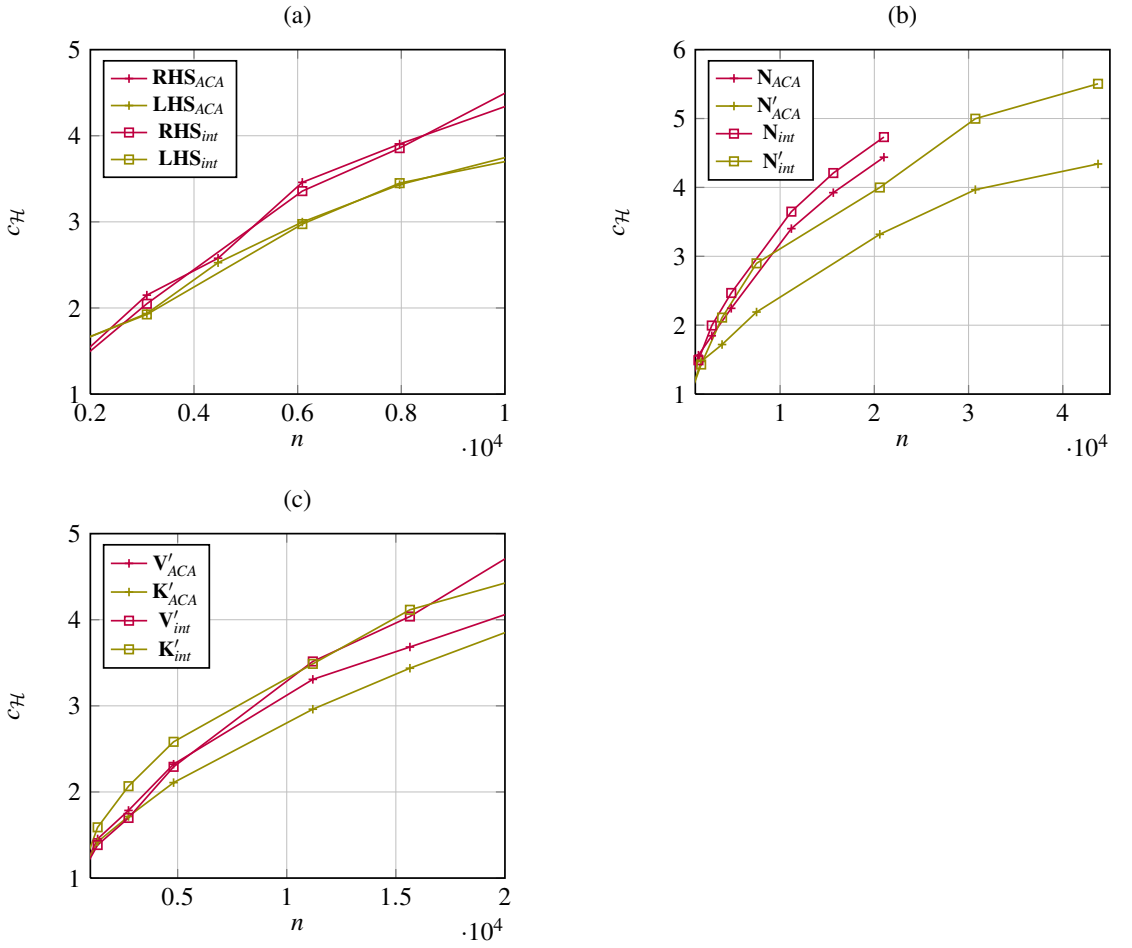

Figure 6.8: Comparison of compression factors between ACA and kernel interpolation

for $\mathbf{V}'$ and $\mathbf{N}'$ respectively. A weakness of kernel interpolation is the rank which is fixed beforehand. The results of a study on the influence of the interpolation order is shown in Figure 6.9. The relative approximation error of the matrices compared to the full matrices is measured in the Euclidean norm and calculated by equation (6.4).

$$\epsilon = \frac{\|\mathbf{M}_f - \mathbf{M}_{\mathcal{H}}\|}{\|\mathbf{M}_f\|} \tag{6.4}$$

Although the tolerance of ACA is set to be $\epsilon_{\mathcal{H}} = 10^{-4}$, the approximation error is better than that, at least for the considered model geometry. An interpolation order of 2 is clearly insufficient to reach the desired accuracy for the system matrices in (3.8) but might be acceptable for matrices of stress equation (3.11). For the volume potential where the compression factors are better than for ACA, the kernel interpolation of order 2 performs well also in terms of accuracy. Surprisingly, the desired accuracy for the approximation

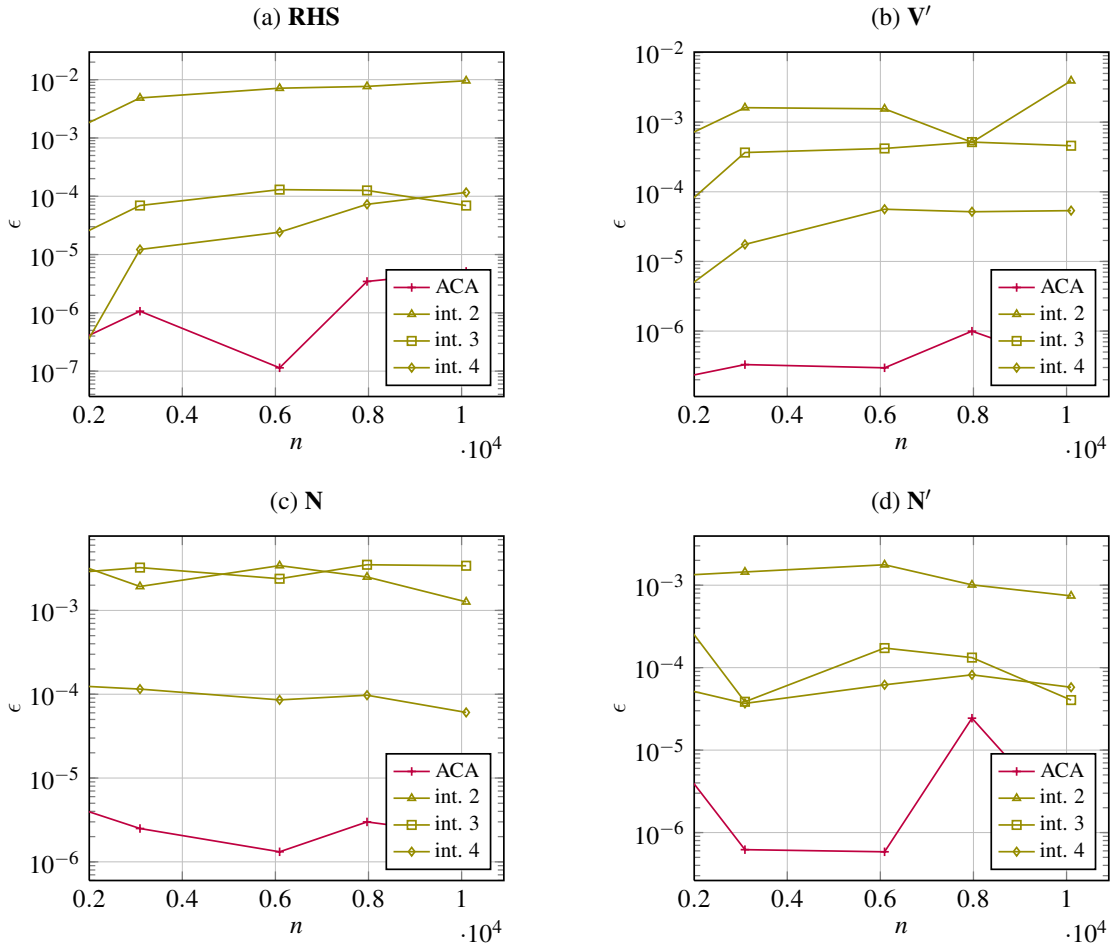

Figure 6.9: Approximation accuracy for ACA and kernel interpolation

of the Newton potential is achieved only by interpolation order 4. A better behavior is expected if the order of interpolation might be variable per block. But clearly, the implementation lacks of a criterion for the interpolation order depending on the distance of the cluster bounding boxes.

6.2 Circular Tunnel

To verify the fast initial stress iteration technique, a circular two-dimensional plane strain tunnel excavation in an infinite domain is simulated. Linear trial functions for u, t and σ are taken. The discretization of the interior, potentially plastic zone extends to r_e as shown in Figure 6.10. The input data for this example are given in Table 6.2. The traction loading

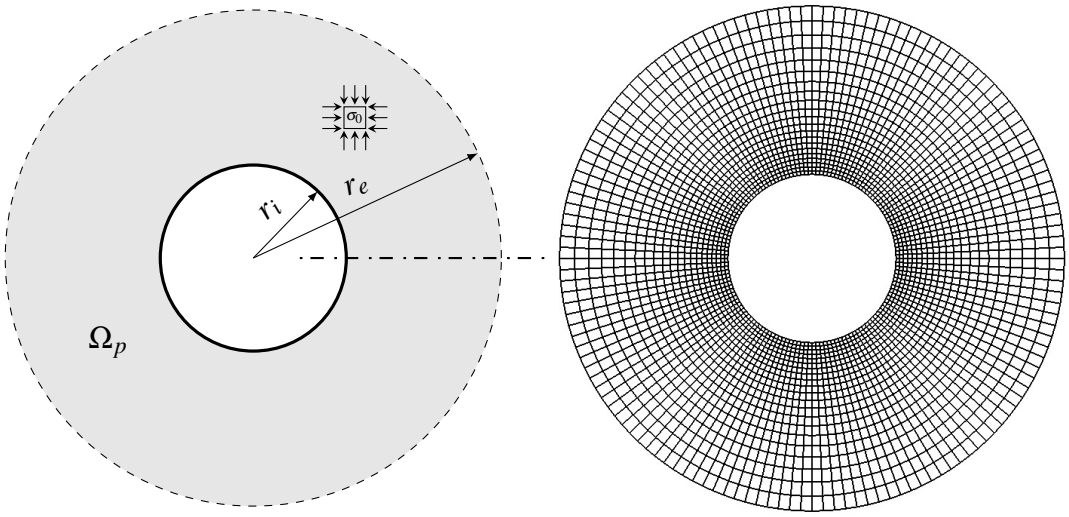

Figure 6.10: Circular tunnel excavation and its discretization

on the surface is derived from a chosen virgin stress state σ_0 by multiplying the outward normal to the boundary. A Hyperbolic Mohr-Coulomb Material model [42] has been used in such a way that it represents the Drucker-Prager yield criterion.

inner radius	$r_i = 6\,\mathrm{m}$
outer radius	$r_e = 18\,\mathrm{m}$
Young's modulus	$E = 28000\,\mathrm{MPa}$
Poisson's ratio	$\nu = 0.32$
cohesion	$c = 15.0\,\mathrm{MPa}$
friction angle	$\phi = 28.5^\circ$
virgin stress	$\sigma_{0,yy} = -54\,\mathrm{MPa}$
	$\sigma_{0,xx} = -27\,\mathrm{MPa}$
	$\sigma_{0,zz} = -27\,\mathrm{MPa}$
	$\sigma_{0,ij} = 0, \forall i \neq j$ with $i, j \in \{x, y, z\}$

Table 6.3: Input data for the deep tunnel excavation

The calculations are compared with a coupled FEM-BEM reference solution where the region of plasticity $r_i \leq r \leq r_e$ is covered by FEM and the outer, purely elastic domain

$r > r_e$ by BEM. In the first study the parameters for the $\mathcal{H}$-matrices given in Table 6.2 are kept constant and one load step is applied. For matrix approximation, ACA has been used.

approximation tolerance	$\epsilon_{\mathcal{H}} = 10^{-4}$
minimal leaf size	$n_{min} = 50$
admissibility factor	$\eta = 1$
solver accuracy	$\epsilon_{ks} = 10^{-5}$
nonlinear residual tolerance	$\epsilon_{\sigma} = 10^{-3}$
clustering strategy	geometrically balanced

Table 6.4: Parameters for simulation of circular tunnel

The results for the radial and tangential stress along a horizontal cut line are shown in Figure 6.11(a) and Figure 6.11(b) respectively. The algorithm converges with increasing

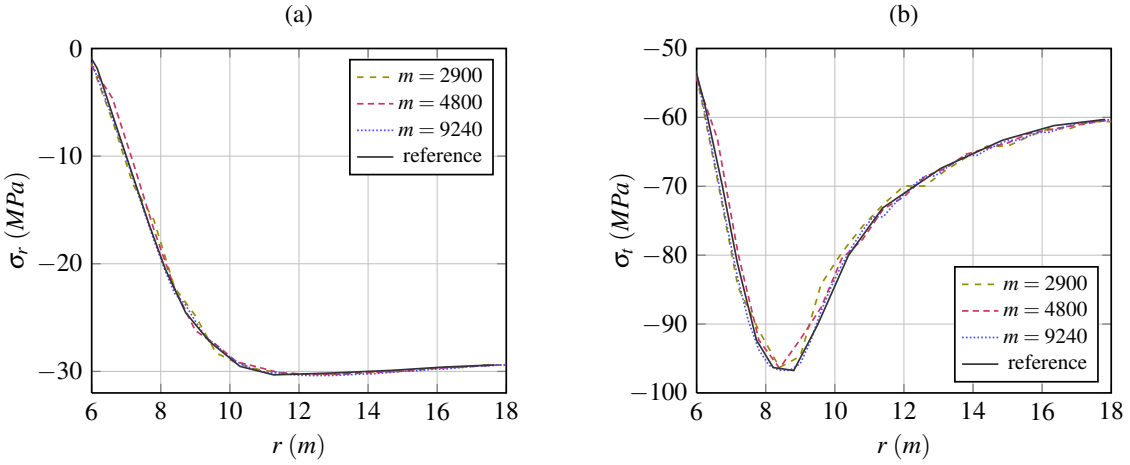

Figure 6.11: Radial stress (a) and tangential stress (b) along a horizontal cut-line with different number of stress points m for the circular deep tunnel excavation

degrees of freedom on the boundary and stress points m in the interior. For this two-dimensional example, the size of the stress vector in equation (3.8) is $n_\Omega = 3m$. The stress distribution for two different discretizations is shown in Figure 6.12 exemplarily. It is realized that the usage of the $\mathcal{H}$-matrix algebra does not introduce any instabilities or inaccuracies to the converged solution. In Table 6.2 the required storage per stress equation, the compression rates and speedup factors for the discrete volume potential $\mathbf{N}' \in \mathbb{R}^{n_\Omega \times n_\Omega}$ are shown. Like for the beam example in section 6.1 the volume potential $\mathbf{N}'$ appears to be the most problematic discrete operator for both benchmarks. Starting with an approximate matrix size of $n_\Omega = 20000$ the used storage per stress value increases only moderately. The speedup for the creation of the system matrix behaves almost linear.

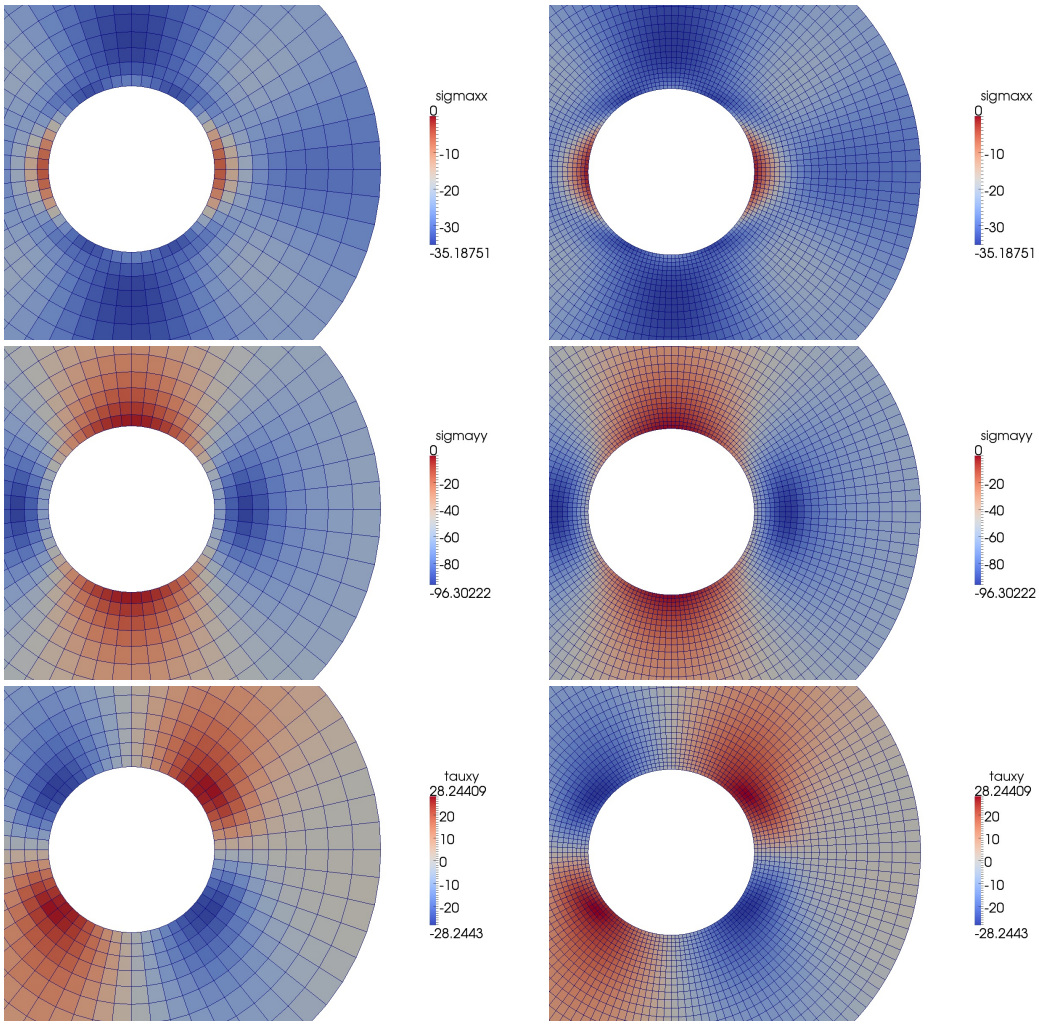

Figure 6.12: Stress distribution around the circular tunnel for (a) 504 and (b) 2900 cells

Finally, a discretization with $n_\Omega = 24132$ stress equations has been chosen to investigate the influence of the approximated volume potential's accuracy $\epsilon_{\mathcal{H},N'}$. The outer radius is reduced to $r_e = 10\,\text{m}$ so that many matrix blocks correspond to the plastic zone. The tolerated error for all other matrices is set to $\epsilon_\mathcal{H} = 10^{-6}$. In Table 6.2 the influence on the important settlements of the tunnel as well as the compression rate and the matrix construction time is shown. It can be seen that at least for that example the accuracy can be set to a lower level while dealing with the displacements u_{max}. This is admissible as long as (3.8) and the boundary integrals in (3.11) are accurate enough. There is a factor 1.7 gain in matrix-compression but only a slight speedup in construction time compared to an accuracy of 10^{-4}.

The investigation is concluded with Table 6.2 where again n_Ω denotes the number of

n_Ω	KB/n_Ω	$c_\mathcal{H}$	$s_\mathcal{H}$
1512	10.53	1.25	0.99
4200	14.58	2.41	1.14
8820	20.06	3.60	1.77
14400	24.08	4.83	2.55
19200	27.38	5.65	3.10
23760	27.05	7.07	4.94
27720	28.86	7.73	5.12
31080	29.87	8.33	5.35
35100	30.82	9.12	6.80
38400	31.04	9.90	-
45000	32.51	11.03	-

Table 6.5: Storage requirement, compression rate $c_\mathcal{H}$ and speedup $s_\mathcal{H}$ for different sizes n_Ω of the approximated discrete volume potential $\mathbf{N}'$ for the circular tunnel excavation

$\epsilon_{\mathcal{H},N'}$	u_{max} (m)	$c_\mathcal{H}$	$s_\mathcal{H}$
10^{-1}	0.020183702	21.46	8.39
10^{-2}	0.020345300	14.99	7.73
10^{-3}	0.020346800	11.34	7.33
10^{-4}	0.020346900	9.08	6.79
10^{-5}	0.020346900	7.42	6.33

Table 6.6: Influence of the approximation-accuracy of $\mathbf{N}'$ on the surface settlements for the circular tunnel excavation

stresses to be calculated in the interior, $\#iter$ the number of iterations needed for the algorithm to converge. As seen in chapter 5 this includes the solution of (3.8), the evaluation of (3.11), the valid stresses and the residual per step. The speedup for the iteration and for the whole calculation is shown. It is interesting to see, that the number of initial stress iterations stays almost constant for finer discretizations. This suggests, that in this example the effect on the accuracy of the nonlinear algorithm is rather small at a certain level of refinement.

n_Ω	$\#iter$	$s_{\mathcal{H},iter}$	$s_{\mathcal{H},total}$
1512	59	1.06	0.99
4200	66	1.43	1.12
8820	69	2.52	1.72
14400	71	3.42	2.46
19200	72	3.93	2.94
23760	73	5.61	4.74
27720	74	5.97	5.53
31080	74	6.02	5.91
35100	74	6.28	6.41

Table 6.7: Speedup of the nonlinear iteration and the complete algorithm for the circular tunnel excavation

6.3 Tunnel Cross Passage

To show the robustness of the implemented matrix approximation technique, tunnels with a cross-passage in three dimensions by means of a real world example are tested. The behavior of the $\mathcal{H}$-matrix arithmetic is investigated. The discretized problem is shown in Figure 6.13. The tunnel surface has been meshed with flat triangles (green) and is

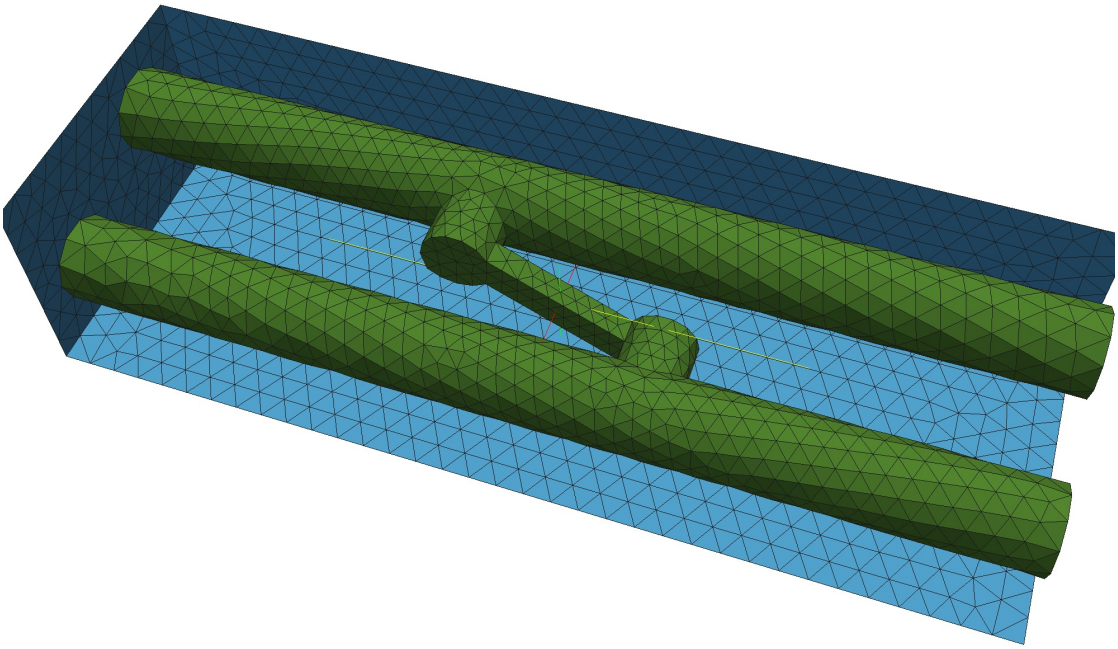

Figure 6.13: Discretized tunnels with cross passage

surrounded by a box which is discretized with tetrahedral elements (blue). For such a geometry it is nearly impossible to generate a cell mesh consisting of hexahedral elements. Linear trial functions are chosen for the displacement interpolation, constant ones for the tractions and stresses. The parameters for the simulation are found in Table 6.3. The minimal leaf size n_{min} varies depending on the mesh size.

approximation tolerance	$\epsilon_{\mathcal{H}} = 10^{-4}$
minimal leaf size	$n_{min} = \{40, \ldots, 100\}$
admissibility factor	$\eta = 1$
solver accuracy	$\epsilon_{ks} = 10^{-6}$
clustering strategy	geometrically balanced

Table 6.8: $\mathcal{H}$-matrix parameters for the simulation of a tunnel cross passage

The $\mathcal{H}$-matrix structure of $\mathbf{V}_{xx}$ with a matrix size of 1598 (4794 degrees of freedom in total) is shown in Figure 6.14. Again, the numbers reveal the rank of the corresponding

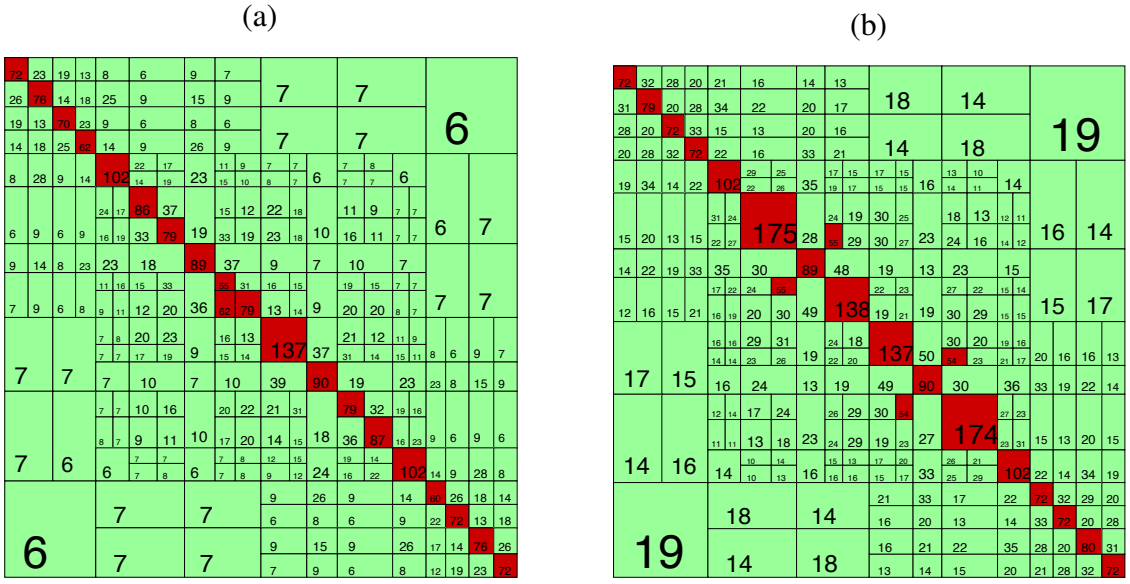

Figure 6.14: $\mathcal{H}$-matrix structure of $\mathbf{V}_{xx}$ (a) and $\mathbf{K}_{xx}$ (b) for the tunnel cross passage with 4795 degrees of freedom

matrix block. The rank of the blocks of the other sub-matrices $\mathbf{V}_{xy}$, $\mathbf{V}_{yx}$ and $\mathbf{V}_{yy}$ does not show any significant differences. A geometrically balanced clustering scheme has been taken. Although the structure of both matrices is similar, the double layer potential $\mathbf{K}$ shows an increased rank in its blocks compared to the single layer potential $\mathbf{V}$.

The construction time and the storage requirements of the system matrices have been investigated. In Figure 6.16 it is shown that for the discrete single and double layer potential an almost linear behavior in terms of storage can be observed. The compression of $\mathbf{N}'$ suffers again from its 6×6 block structure. The weaker performance is realized by Figure 6.15 depicting the structure of the sub-matrix $\mathbf{N}'_{11,11}$. The near field P^- is much larger than observed for other system matrices. Since the degrees of freedom on the boundary n_Γ and the number of stress points n_Ω diverge, the matrices $\mathbf{N}$, $\mathbf{V}'$ and $\mathbf{K}'$ are of a stretched rectangular shape. In that case the strategy for the block cluster tree influences the compression rate and an inhomogeneous clustering scheme as described in section 4.2.3 provides the best results. The admissibility between the row- and column-clusters is checked also on different levels. The storage per degree of freedom, compression rate and the overall construction time for the system matrices $\mathbf{N}$, $\mathbf{V}'$ and $\mathbf{N}'$ is listed in Table 6.3, Table 6.3 and Table 6.3 respectively. Although the ratio b increases, the compression rate stays almost the same. The required storage per fictitious degree of freedom n (6.3) and the construction time increase only moderately.

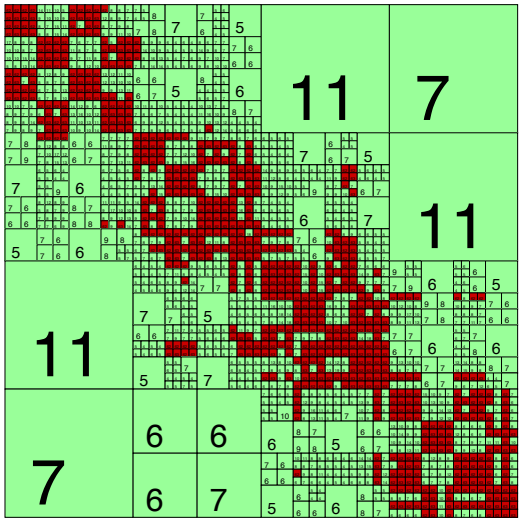

Figure 6.15: $\mathcal{H}$-matrix structure of the volume potential sub-matrix $\mathbf{N}'_{11,11} \in \mathbb{R}^{(n_\Omega/6)\times(n_\Omega/6)}$ for $n_\Omega = 24030$

n_Ω	b	KB/n	$c_\mathcal{H}$	$t_\mathcal{H}$ (s)
24030	14.8	27.65	1.76	238.9
40476	24.9	35.58	1.78	454.5
44292	23.9	37.99	1.86	573.3
52800	23.0	45.13	1.90	797.7
65166	28.4	49.69	1.88	1205.2

Table 6.9: Storage requirement and construction time of $\mathbf{N}$ for the tunnel cross passage

n_Ω	b	KB/n	$c_\mathcal{H}$	$t_\mathcal{H}$ (s)
24030	7.5	31.81	2.15	499.4
40476	12.7	39.59	2.24	1138.5
44292	12.1	41.01	2.42	1588.0
52800	11.7	48.75	2.48	2140.4
65166	14.4	53.38	2.51	2753.1

Table 6.10: Storage requirement and construction time of $\mathbf{V}'$ for the tunnel cross passage

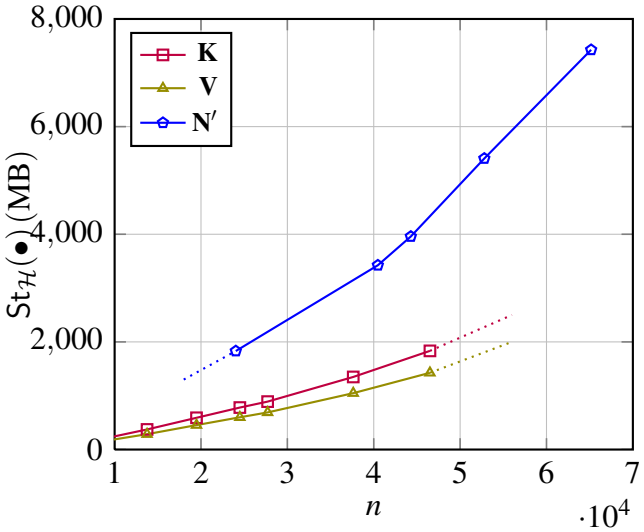

Figure 6.16: Sizes of **V**, **K** and **N**′ for different discretizations of the tunnel cross passage

n_Ω	b	KB/n	$c_\mathcal{H}$	$t_\mathcal{H}$ (s)
24030	14.8	30.12	1.62	1488.9
40476	24.9	38.91	1.63	3757.2
44292	23.9	41.64	1.70	6425.9
52800	23.0	52.50	1.64	7278.4
65166	28.4	58.08	1.65	11416.8

Table 6.11: Storage requirement and construction time of **K**′ for the tunnel cross passage

6.4 Pile Foundation

Pile foundations are a classical exercise for boundary element formulations since they are usually embedded in a half-space. An example is shown in Figure 6.17. Forces from build-

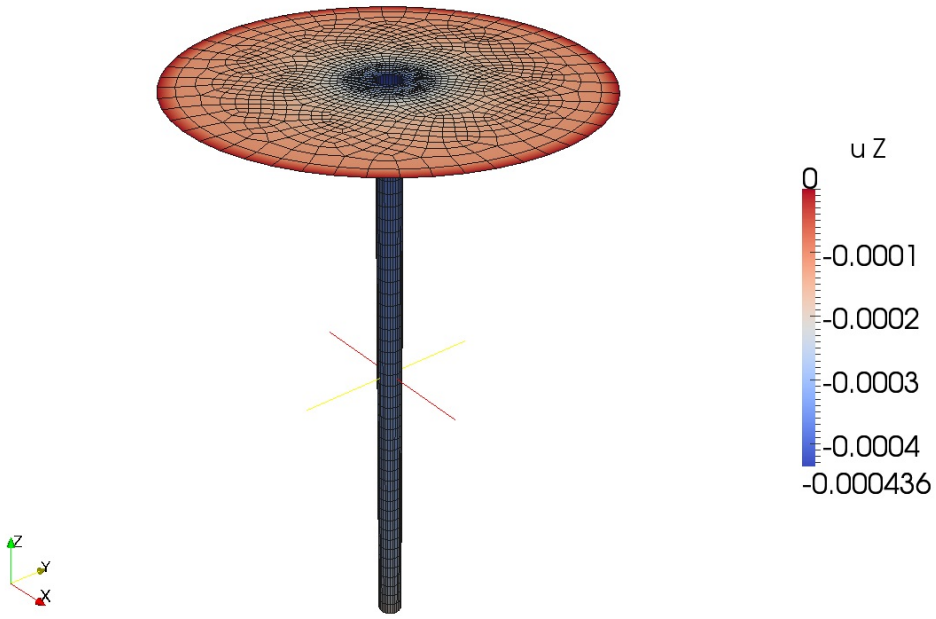

Figure 6.17: Settlements of a pile discretized by means of quadrilaterals

ings are transmitted through the foundation and imposed along the pile into the ground. To avoid the need of the discrete pile, constant tractions are imposed as boundary conditions along the pile surface for simplicity. In this section, different clustering strategies are tested and their performance is compared. The results are only considered in terms of the result vector $\mathbf{u}$ whose accuracy is measured in the Euclidean norm

$$\epsilon = \frac{\|\mathbf{u}_f - \mathbf{u}_{\mathcal{H}}\|}{\|\mathbf{u}_f\|} \quad \text{with} \quad \|\mathbf{u}\|^2 = \sum_{i=1}^{n} (\mathbf{u}[i])^2.$$

As described in section 4.2, different clustering strategies are tested: geometrical regular, geometrical balanced, cardinality balanced as well as clustering by means of PCA and regular boxes like in the FMM. The pile is discretized by 6257 quadrilateral boundary elements. Linear trial functions are used to describe the displacement field, constant ones for the traction interpolation. Hence, $n = 18699$ degrees of freedom are identified. Table 6.4 shows the chosen parameters. The calculations are carried out by one CPU. The $\mathcal{R}_k$-matrices are constructed by both, ACA and kernel interpolation. First, compression rates are investigated and the results shown in Table 6.4. Like in the example with the deflected beam in section 6.1, ACA seems not to find the best overall approximation.

approximation tolerance	$\epsilon_{\mathcal{H}} = 10^{-4}$
order of interpolation	$k_i = 3$
minimal leaf size	$n_{min} = 60$
admissibility factor	$\eta = 2$
solver accuracy	$\epsilon_{ks} = 10^{-5}$

Table 6.12: Parameters for the simulation of a pile foundation

For the chosen parameters, the accuracy is better than expected and therefore the computational effort higher than needed. By taking kernel interpolation of fixed order $k_i = 3$, the approximation seems just fine. The compression factors are slightly better for almost all chosen clustering strategies. Proposed in literature, cardinality balanced and PCA based clustering is well suited for low rank approximations by means of ACA. In this study and for the particular geometry, this is only partly reproduced. Geometrically balanced and PCA trees perform best for both approximation techniques. Geometrical regular clustering is not suited for the $\mathcal{H}$-matrix approximation by the described implementation. As

	cardinality	geometric	pca	regular	regularbox
ACA					
$\epsilon_{\mathcal{H},V}$	$3.9 \cdot 10^{-6}$	$4.17 \cdot 10^{-6}$	$4.73 \cdot 10^{-5}$	$4.13 \cdot 10^{-6}$	$4.13 \cdot 10^{-6}$
$c_{\mathcal{H},V}$	7.91	8.2	9.0	1.89	1.89
$\epsilon_{\mathcal{H},K}$	$5.2 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$5.29 \cdot 10^{-4}$	$1.32 \cdot 10^{-4}$	$1.32 \cdot 10^{-7}$
$c_{\mathcal{H},K}$	6.41	6.91	7.40	2.63	2.63
kernel interpolation					
$\epsilon_{\mathcal{H},V}$	$1.94 \cdot 10^{-4}$	$1.46 \cdot 10^{-4}$	$1.66 \cdot 10^{-4}$	$1.26 \cdot 10^{-4}$	$1.26 \cdot 10^{-4}$
$c_{\mathcal{H},V}$	8.06	8.48	9.13	1.88	1.88
$\epsilon_{\mathcal{H},K}$	$\cdot 10^{-4}$	$\cdot 10^{-4}$	$\cdot 10^{-4}$	$\cdot 10^{-4}$	$\cdot 10^{-7}$
$c_{\mathcal{H},K}$	6.72	7.19	7.79	2.72	2.72

Table 6.13: Accuracy and compression rates for different clustering schemes

far as calculation times concern, ACA is supposed to perform better at least for boundary integrals. This becomes clear in Table 6.4. The computation time of kernel interpolation for all clustering techniques is significantly higher than for ACA. For the right hand side which correlates to the single layer potential on the pile boundary with non-zero tractions, geometrical balanced clustering is superior to all other schemes. On the other hand, PCA performs best for the discrete double layer $\mathbf{K}$ on the whole boundary which correlates to the left hand side. To visualize different clustering strategies applied to the considered geometry, Figure 6.19 shows the $\mathcal{H}$-matrix structure of the LHS and Figure 6.18 of the RHS carried out by ACA.

	cardinality	geometric	pca	regular	regularbox
setup T_I, $T_{I\times J}$	0.87	0.78	0.76	2.23	2.29
ACA $[s]$					
integration $\mathbf{V}$ $[s]$	527.3	523.5	529.8	927.1	926.7
integration $\mathbf{K}$	2452.4	2374.9	2255.7	3082.4	3117.1
solution	36.4	22.8	25.6	74.1	74.7
total	3038.3	2946.3	2825.7	4152.2	4187.9
kernel interpolation $[s]$					
integration $\mathbf{V}$ $[s]$	790.5	699.1	753.4	1396.4	1402.5
integration $\mathbf{K}$	13082.8	15151.9	11892.0	26771.4	26831.4
solution	35.3	22.4	25.3	73.6	74.9
total	13931.3	15899.8	12687.3	28317.9	28388.5

Table 6.14: Total running time and timings for setup, integration and solution

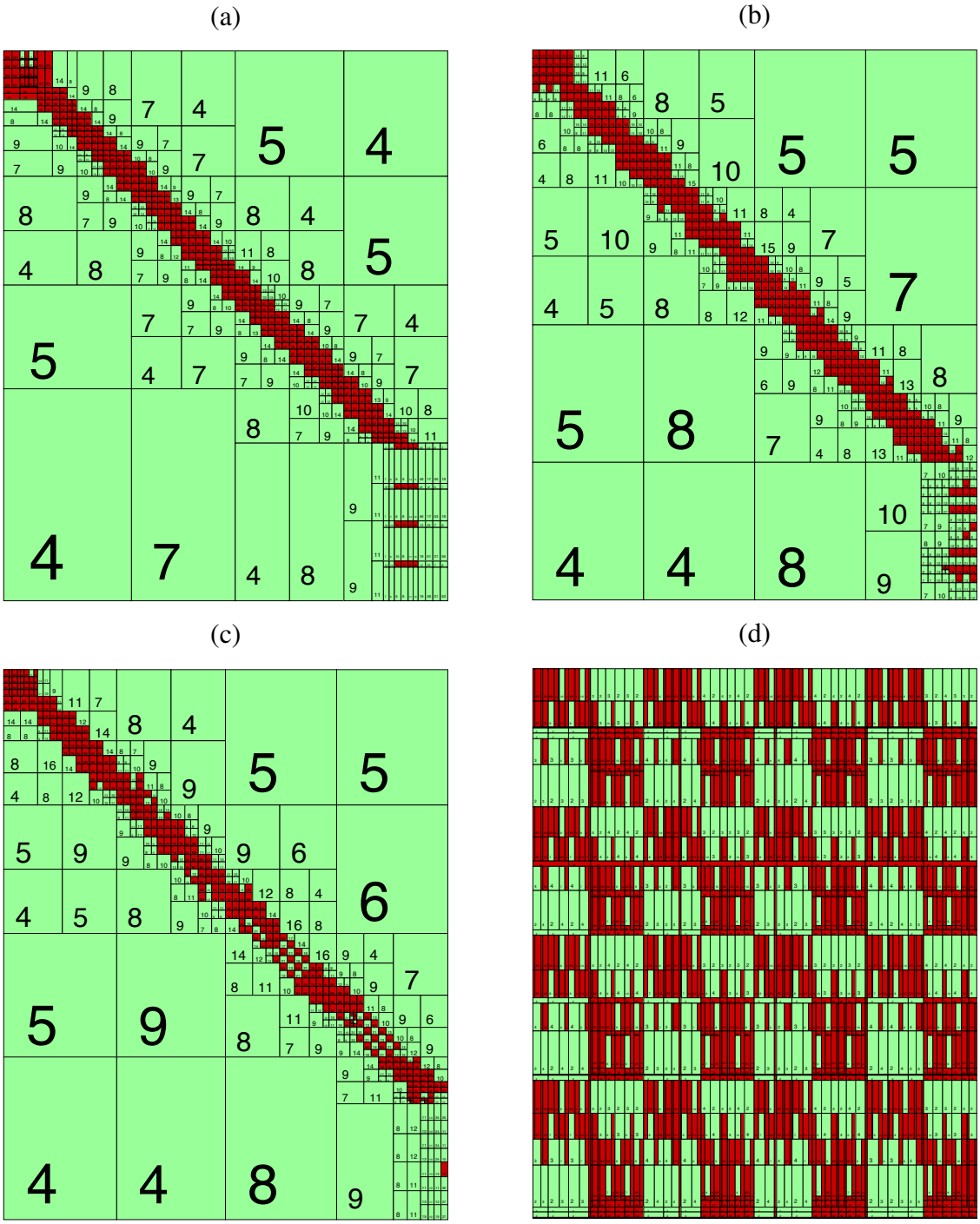

Figure 6.18: $\mathcal{H}$-matrix structure of $\mathbf{V}_{xx}$ by means of geometrical balanced (a), cardinality balanced (b), PCA (c) and geometric regular (d) clustering

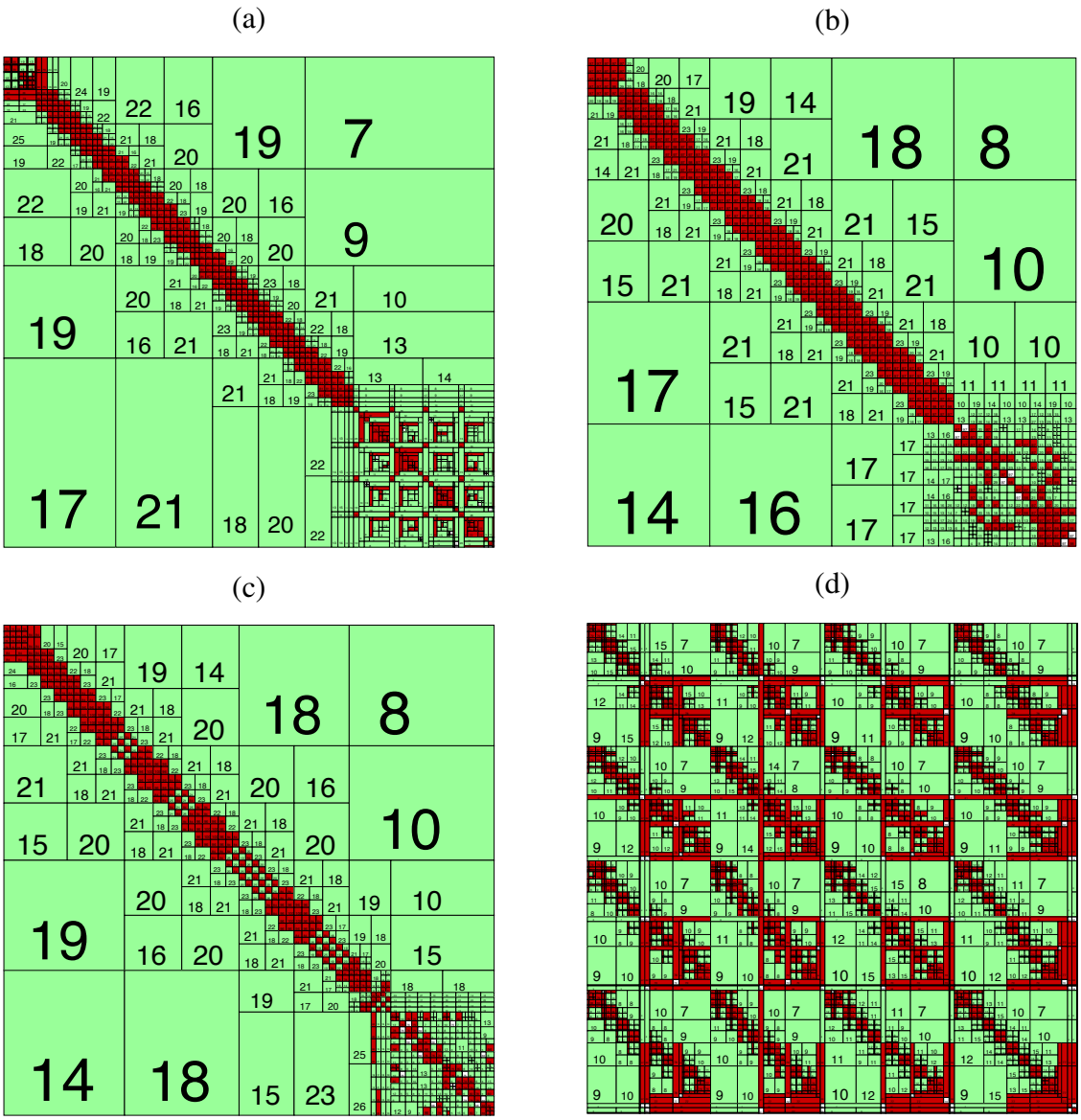

Figure 6.19: $\mathcal{H}$-matrix structure of $\mathbf{K}_{xx}$ by means of geometrical balanced (a), cardinality balanced (b), PCA (c) and geometric regular (d) clustering

7 Conclusion

In the previous chapters, a fast boundary element method (BEM) to solve linear elastic and elasto-plastic problems has been presented. By using domain discretizations, the method allows the solution of inhomogeneous problems in terms of body forces or internal stresses. The implementation is tailored to solve tasks in underground construction, where regions of plasticity are localized near the tunnel face or inclusions in situ. This fact permits the usage of the BEM only instead of a coupled approach with the finite element method (FEM). An initial stress iteration technique has been utilized in order to solve the nonlinear plasticity problem. To achieve nearly linear complexity, the concept of hierarchical matrices ($\mathcal{H}$-matrices) has been deployed to the formulation. The implementation is to be seen as a step towards the aim to provide a fast and reliable, boundary based simulation tool for the field of geotechnics. To obtain a robust formulation, the following features have been super-induced to the simulation software *BEFE++*.

The trial functions on the discrete boundary for the primal and dual variable respectively can be chosen individually and independent of the geometry description. In order to employ a block system of equations, the collocation points are chosen carefully. To allow discontinuous traction distribution the points are located inside the element on the Dirichlet boundary. At the region subject to inhomogeneity, the discretization is chosen completely independent of boundary elements. Moreover, domain variables may be interpolated discontinuous or continuous, disregarding the geometry description, too. To avoid hypersingular integral kernels, the stress points are located inside the domain. As a consequence all system matrices consist of entries originating from the same fundamental solution. This is a preliminary for the applied fast method in order to guarantee asymptotically smooth kernels.

The sparsity of $\mathcal{H}$-matrices is based on approximations of matrix blocks by the bias of low rank matrices ($\mathcal{R}_k$-matrices). For the purpose of matrix separation, cluster trees which are based on geometric information have been employed. The block partition of matrices is carried out by means of different strategies which have been outlined. Furthermore, two major strategies were shown in order to find an approximating $\mathcal{R}_k$-matrix for admissible matrix blocks. Namely, the adaptive cross approximation (ACA) algorithm as well as the concept of kernel interpolation. Although ACA is considered to be less intrusive in terms of software implementation, the application to vector valued problems like elasticity or elasto-plasticity demands some special arrangements which have been established. Nevertheless, the algorithm produces a rather good approximation with adaptive rank. By taking kernel interpolation the rank is fixed beforehand and usually too large. With the aid of

appropriate recompression algorithms the rank is reduced significantly. In order to solve the system of equations once and in case of elasto-plastic analysis even multiple times, an iterative solver is implemented. To narrow down the number of iterations, a data sparse preconditioner based on hierarchical LU factorization is taken which does not constrain the overall complexity.

The implementation has been verified by several examples. The most important aspect is the invariance of the results if the $\mathcal{H}$-matrix arithmetic is applied. By taking an appropriate tolerance for the matrix approximation, i.e. in the same magnitude as for the nonlinear iteration technique, there is no significant change in boundary results, stress results in the domain or the rate of convergence for the initial stress iteration. But with the addition that solution times and storage requirements are remarkably reduced. The system matrices representing boundary integrals show the expected almost linear behavior in terms of storage and complexity of matrix operations for fairly small degrees of freedom. This applies to the discrete Newton potential too. Problems associated with matrix compression have been observed for the volume potential in the stress integral equation. This is partially explained by properties of the corresponding trial functions and their support. Compared to boundary elements, cells show larger diameters and so do the bounding boxes of correlated clusters. Additionally, stress points are not boundary concentrated anymore. As a consequence, less clusters associated to stress points and clusters correlated to support points become admissible. Into the bargain, the discrete volume potential is build upon many sub-matrices. In case of three dimensions even $6 \times 6 = 36$. Hence, even if the system matrix is rather large, the size of sub-matrices allocated for $\mathcal{H}$-matrix construction is reduced by a factor of 6. The results show, that the construction time of this system matrix is remarkably slower than for the others. Promising is the expected overall behavior in terms of complexity, starting from a larger problem size.

By taking the kernel interpolation technique instead of ACA, improvements are found related to compression rates of almost all system matrices. However, the fixed rank results to a computational overhead and construction time is higher than for ACA. The implementation needs a criterion for the order of interpolation. Moreover, since the bounding boxes have different extensions for all spatial directions, an interpolation scheme with different orders per direction would be optimal in order to reduce the produced rank. However, the recompression and coarsening technique reduces the rank after creation and produces approximations with the desired accuracy.

To address the problem concerning the volume potential, a formulation with plastic multipliers as iterative quantities [39] could be used but would prevent a black box usage of different material models in the implementation. Not only related to the described issue but also in general, the application of kernel interpolation techniques must be topic for further investigation. The implementation would benefit from the fact that kernel evaluations for the far field are reduced significantly. This applies also to the evaluation and integration of the corresponding interpolation functions. By taking hybrid approximation techniques

[23] a further improvement in terms of construction time is predictable.

The presented thesis provides a number of starting points for future research work. Still, the main focus is to avoid input data related to the domain. Adaptive generation of triangulation as well as boundary only approaches have been proposed in literature. In context of this work, a rigorous application of a Cartesian grid for the inhomogeneous domain is proposed. There is no need of generating this grid by the user. If the grid is incorporated in the clustering strategy and due to the translationally invariant fundamental solutions, single matrix entries as well as whole matrix blocks are carried out once and may be used multiple times in the whole matrix structure. This will speed up the construction of the affected discrete integral operators. Since single matrix blocks could be set to zero without memory consumption, a task would be to identify regions affected by plasticity and update the corresponding matrix blocks on the fly. Hence, only parts of the system matrices involved in stress calculations are not zero. This has to be embedded into the initial stress iteration.

To further reduce the computational requirements, the application of hierarchical matrices with linear complexity ($\mathcal{H}^2$-matrices) makes sense. A Cartesian grid in conjunction with the kernel interpolation technique would greatly improve the method. $\mathcal{H}^2$-matrices rest upon nested cluster bases which are readily constructed by means of the grid structure. As mentioned before, the application to hybrid methods has not been done yet.

By taking a formulation which uses the consistant tangential operator (CTO) for the iterative solution procedure, faster convergence is achieved. In order to update the resulting Jacobi matrix, a quasi Newton method with low rank updates of the factorizations can be used. This has been shown for the FEM [4] and is an open topic for the BEM. Related to localization effects in plasticity, strain softening effects are successfully treated by the BEM in [98]. Therefore, volume terms by means of non-singular kernels are needed, which are appropriate to be treated by the $\mathcal{H}$-matrix algebra, too and could be incorporated in the method easily.

The sequential excavation process can be simulated by single region analysis with evolving boundaries [33]. In this context, the optimal structure and the mesh-related adaptive update of the $\mathcal{H}$-matrix are research questions to be answered. Adaptive updates of $\mathcal{H}$-matrices has been treated in a different context in [45].

Recapitulating the work in hand, a fast BEM formulation has been created which fulfills the research objective stated in the beginning. The proposed approach shows its advantages if the region of inhomogeneity is considerably small and the affected domain is unbounded. On one hand, for many civil engineering works in underground construction like deep tunnels, pile foundations or caverns this is clearly the case. On the other hand, the implementation is not limited to such applications but also suitable for other problems in elasticity with considered local plastic effects like i.e. contact or crack problems.

References

[1] Anderson, E.; Bai, Z.; Bischof, C.; Blackford, S.; Demmel, J.; Dongarra, J.; Du Croz, J.; Greenbaum, A.; Hammarling, S.; McKenney, A. and Sorensen, D. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, third edition, 1999.

[2] Antes, H. A short course on boundary element methods. Lecture Notes, Institute for Applied Mechanics, 2010. URL *http://www.infam.tu-bs.de/infam2/include/Studium/vorlesungen/bem-short.pdf*.

[3] Banerjee, P.; Henry, D. and Raveendra, S. Advanced inelastic analysis of solids by the boundary element method. *International Journal of Mechanical Sciences*, 31 (4):309–322, 1989.

[4] Bebendorf, M. and Chen, Y. Efficient solution of nonlinear elliptic problems using hierarchical matrices with broyden updates. *Computing*, 81:239–257, 2007.

[5] Bebendorf, M. and Venn, R. Constructing nested bases approximations from the entries of non-local operators. *Numerische Mathematik*, 121:609–635, 2012.

[6] Bebendorf, M. Approximation of boundary element matrices. *Numerische Mathematik*, 86:565–589, 2000.

[7] Bebendorf, M. *Hierarchical Matrices: A Means to Efficiently Solve Elliptic Boundary Value Problems*, volume 63 of *Lecture Notes in Computational Science and Engineering (LNCSE)*. Springer-Verlag, 2008.

[8] Bebendorf, M. and Fischer, T. On the purely algebraic data-sparse approximation of the inverse and the triangular factors of sparse matrices. *Numerical Linear Algebra with Applications*, 18(1):105–122, 2011.

[9] Bebendorf, M. and Grzhibovskis, R. Accelerating galerkin bem for linear elasticity using adaptive cross approximation. *Mathematical Methods in the Applied Sciences*, 29:1721–1747, January 2006.

[10] Bebendorf, M. and Rjasanow, S. Adaptive low-rank approximation of collocation matrices. *Computing*, 70(1):1–24, 2003.

[11] Beer, G. and Watson, J. O. Infinite boundary elements. *International Journal for Numerical Methods in Engineering*, 28:1233–1247, 1989.

[12] Beer, G.; Smith, I. M. and Dünser, C. *The Boundary Element Method with Programming*. Springer Wien - New York, 2008.

[13] Beer, G.; Dünser, C.; Riederer, K.; Glauber, P.; Thöni, K.; Zechner, J. and Stettner, M. *Technology Innovation in Underground Construction*, chapter 7 "Computer Simulation of Conventional Construction", pages 129–161. CRC Press/Balkema, 2300 AK Leiden, The Netherlands, 2010.

[14] Benedetti, I.; Aliabadi, M. and Davi, G. A fast 3d dual boundary element method based on hierarchical matrices. *International Journal of Solids and Structures*, 45: 2355–2376, 2008.

[15] Bettess, P. Infinite elements. *International Journal for Numerical Methods in Engineering*, 11(1):53–64, 1977.

[16] Beylkin, G.; Coifman, R. and Rokhlin, V. Fast wavelet transforms and numerical algorithms. *Communications on Pure and Applied Mathematics*, 44(2):141–183, 1991.

[17] Bonnet, M. *Boundary Integral Equation Methods for Fluids and Solids*. John Wiley & Sons, Ltd, Chinchester, West Sussex, 1995.

[18] Bonnet, M. Boundary integral equation methods for elastic and plastic problems. In Stein, E.; Borst, R.d. and Hughes, T., editors, *Encyclopedia Of Computational Mechanics*, volume 2, chapter 23, pages 719–749. John Wiley & Sons, Ltd, Chichester, 2004.

[19] Bonnet, M. and Mukherjee, S. Implicit bem formulations for usual and sensitivity problems in elasto-plasticity using consistent tangent operator concept. *International Journal of Solids and Structures*, 33(30):4461–4480, 1996.

[20] Börm, S. $\mathcal{H}^2$-matrices – multilevel methods for the approximation of integral operators. *Computing and Visualization in Science*, 7:173–181, 2004.

[21] Börm, S. *Efficient numerical methods for non-local operators. $\mathcal{H}^2$-matrix compression, algorithms and analysis.* Number 14 in EMS Tracts in Mathematics. Zürich: European Mathematical Society (EMS), 2010.

[22] Börm, S. and Grasedyck, L. Hlib – a library for $\mathcal{H}$- and $\mathcal{H}^2$-matrices. http://www.hlib.org, 1999. URL *http://www.hlib.org*.

[23] Börm, S. and Grasedyck, L. Hybrid cross approximation of integral operators. *Numerische Mathematik*, 101:221–249, 2005.

[24] Börm, S.; Grasedyck, L. and Hackbusch, W. Introduction to hierarchical matrices with applications. *Engineering Analysis with Boundary Elements*, 27(5):405 – 422, 2003.

[25] Börm, S.; Löhndorf, M. and Melenk, J. M. Approximation of integral operators by variable-order interpolation. *Numerische Mathematik*, 99:605–643, 2005.

[26] Carrer, J. and Telles, J. A boundary element formulation to solve transient dynamic elastoplastic problems. *Computers and Structures*, 45(4):707–713, 1992.

[27] Chen, G. and Zhou, J. *Boundary Element Methods with Applications to Nonlinear Problems*. Atlantis Press, Amsterdam - Paris, 2010.

[28] Cheng, A. and Cheng, D. Heritage and early history of the boundary element method. *Engineering Analysis with Boundary Elements*, 29(3):268 – 302, 2005.

[29] Davies, T. and Bu, S. Infinite boundary elements for the analysis of halfspace problems. *Computers and Geotechnics*, 19(2):137 – 151, 1996.

[30] Dijkstra, W. and Mattheij, R. The condition number of the bem-matrix arising from laplace's equation. *Electronic Journal of Boundary Elements*, 4(2):67–81, 2006.

[31] Ding, J.; Ye, W. and Gray, L. J. An accelerated surface discretization-based bem approach for non-homogeneous linear problems in 3-d complex domains. *International Journal for Numerical Methods in Engineering*, 63(12):1775–1795, 2005.

[32] Doyle, J. M. Radiation conditions in elasticity. *Zeitschrift für Angewandte Mathematik und Physik*, 16:527–531, 1965.

[33] Dünser, C. and Beer, G. Simulation of sequential excavation with the boundary element method. *Computers and Geotechnics*, 44:157 – 166, 2012.

[34] Eberwien, U.; Dünser, C. and Moser, W. Efficient calculation of internal results in 2d elasticity bem. *Engineering Analysis with Boundary Elements*, 29(5):447 – 453, 2005.

[35] Ewald, P. P. Die berechnung optischer und elektrostatischer gitterpotentiale. *Annalen der Physik*, 369(3):253–287, 1921.

[36] Frangi, A. and Maier, G. Dynamic elastic-plastic analysis by a symmetric galerkin boundary element method with time-independent kernels. *Computational Methods in Applied Mechanical Engineering*, 171:281–308, May 1999.

[37] Gao, X.-W. A boundary element method without internal cells for two-dimensional and three-dimensional elastoplastic problems. *Journal of Applied Mechanics*, 69 (2):154–160, 2002.

[38] Gao, X.-W. and Davies, T. 3-d infinite boundary elements for half-space problems. *Engineering Analysis with Boundary Elements*, 21(3):207–213, 1998.

[39] Gao, X.-W. and Davies, T. G. An effective boundary element algorithm for 2d and 3d elastoplastic problems. *International Journal of Solids and Structures*, 37: 4987–5008, May 2000.

[40] Gao, X. and Davies, T. *Boundary Element Programming in Mechanics*. Cambridge University Press, Cambridge, 2002.

[41] Gaul, L.; Kögl, M. and Wagner, M. *Boundary element methods for engineers and scientists: an introductory course with advanced topics*. Springer-Verlag, Berlin-Heidelberg-New York, 2003.

[42] Gens, A.; Carol, I.; Gonzales, N.; Caballero, A. and Garolera, D. Library of model subroutines: Soil and rock models. *TunConstruct Deliverable 1.1.4*, pages 1, 77, 83–89, 120, 150, 2007.

[43] Golub, G. and Kahan, W. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics Series B Numerical Analysis*, 2(2):205–224, 1965.

[44] Golub, G. and Van Loan, C. *Matrix Computations*. Johns Hopkins University Press, Baltimore, third edition, 1996.

[45] Grasedyck, L.; Hackbusch, W. and Borne, S. L. Adaptive geometrically balanced clustering of $\mathcal{H}$-matrices. *Computing*, 73:1–23, 2004.

[46] Grasedyck, L. Adaptive recompression of $\mathcal{H}$-matrices for bem. *Computing*, 74: 205–223, 2005.

[47] Grasedyck, L. and Hackbusch, W. Construction and arithmetics of $\mathcal{H}$-matrices. *Computing*, 70:295–334, 2003.

[48] Grasedyck, L.; Kriemann, R. and Le Borne, S. Domain decomposition based $\mathcal{H}$-lu preconditioning. *Numerische Mathematik*, 112:565–600, 2009.

[49] Guiggiani, M. and Gigante, A. A general algorithm for multidimensional cauchy principal value integrals in the boundary element method. journal of applied mechanics, 57:906–915, 1990. *Journal of Applied Mechanics*, 57:906–915, 1990.

[50] Hackbusch, W. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. *Computing*, 62: 89–108, 1999.

[51] Hackbusch, W. A sparse matrix arithmetic. part ii: Application to multi-dimensional problems. *Computing*, 62:89–108, 1999.

[52] Hackbusch, W. *Hierarchische Matrizen*. Springer Berlin Heidelberg, 2009.

[53] Hackbusch, W. and Börm, S. $\mathcal{H}^2$-matrix approximation of integral operators by interpolation. *Applied Numerical Mathematics*, 43(1-2):129 – 143, 2002.

[54] Hackbusch, W. and Börm, S. Data-sparse approximation by adaptive $\mathcal{H}^2$-matrices. *Computing*, 69:1–35, 2002.

[55] Hackbusch, W. and Nowak, Z. P. On the fast matrix multiplication in the boundary element method by panel clustering. *Numerische Mathematik*, 54:463–491, 1989.

[56] Hartmann, F. *Methode der Randelemente - Boundary Elements in der Mechanik auf dem PC*. Springer-Verlag, Berlin-Heidelberg-New York, 1987.

[57] Hayami, K. and Sauter, S. A. A formulation of the panel clustering method for three dimensional elastostatics. In *Proceedings of the Annual Meeting of the Japanese Society for Industrial and Applied Mathematics*, pages 218–219. Japanese Society for Industrial and Applied Mathematics (JSIAM), 1996.

[58] Hsiao, S.; Mammoli, A. A. and Ingber, M. S. The evaluation of domain integrals in complex multiply-connected three-dimensional geometries for boundary element methods. *Computational Mechanics*, 32:226–233, 2003.

[59] Ingber, M.; Mammoli, A. and Brown, M. A comparison of domain integral evaluation techniques for boundary element methods. *International Journal for Numerical Methods in Engineering*, 52:417–432, 2001.

[60] Koehler, M.; Yang, R. and Gray, L. Cell-based volume integration for boundary integral analysis. *International Journal for Numerical Methods in Engineering*, pages n/a–n/a, 2012.

[61] Kolk, K.; Weber, W. and Kuhn, G. Investigation of 3d crack propagation problems via fast bem formulations. *Computational Mechanics*, 37:32–40, 2005.

[62] Kolymbas, D. *Tunnelling and Tunnel Mechanics. A Rational Approach to Tunnelling*. Springer-Verlag, Berlin-Heidelberg, 2005.

[63] Kriemann, R. Parallel $\mathcal{H}$-matrix arithmetics on shared memory systems. *Computing*, 74:273–297, 2005.

[64] Krommer, A. R. and Überhuber, C. W. *Numerical Integration on Advanced Computer Systems*. Springer-Verlag, Berlin-Heidelberg, 1994.

[65] Lachat, J. and Watson, J. Effective numerical treatment of boundary integral equations: A formulation for three-dimensional elastostatics. *International Journal for Numerical Methods in Engineering*, 10:991–1005, July 1976.

[66] Maerten, F. Adaptive cross-approximation applied to the solution of system of equations and post-processing for 3d elastostatic problems using the boundary element method. *Engineering Analysis with Boundary Elements*, 34(5):483 – 491, 2010.

[67] Maier, G. and Polizzotto, C. A galerkin approach to boundary element elastoplastic analysis. *Computer Methods in Applied Mechanics and Engineering*, 60(2):175 – 194, 1987.

[68] Manolis, G. D. and Banerjee, P. K. Conforming versus non-conforming boundary elements in three-dimensional elastostatics. *International Journal for Numerical Methods in Engineering*, 23(10):1885–1904, 1986.

[69] Mantic, V. A new formula for the c-matrix in the somigliana identity. *Journal of Elasticity*, 33:191–201, 1993.

[70] Mason, J. C. and Handscomb, D. C. *Chebyshev Polynomials*. Chapman & Hall/CRC, London, 2003.

[71] Messner, M. and Schanz, M. An accelerated symmetric time-domain boundary element formulation for elasticity. *Engineering Analysis with Boundary Elements*, 34(11):944 – 955, 2010.

[72] Milazzo, A.; Benedetti, I. and Aliabadi, M. H. Hierarchical fast bem for anisotropic time-harmonic 3-d elastodynamics. *Computers & Structures*, 96–97(0):9 – 24, 2012.

[73] Moser, W.; Antes, H. and Beer, G. A duhamel integral based approach to one-dimensional wave propagation analysis in layered media. *Computational Mechanics*, 35:115–126, 2005.

[74] Moser, W.; Dünser, C. and Beer, G. Mapped infinite elements for three-dimensional multi-region boundary element analysis. *International Journal for Numerical Methods in Engineering*, 61(3):317–328, 2004.

[75] Nardini, D. and Brebbia, C. A new approach to free vibration analysis using boundary elements. *Applied Mathematical Modelling*, 7(3):157 – 162, 1983.

[76] Numerical Algorithms Group Ltd., . Amd core math library (acml) documentation, 2012. URL *http://developer.amd.com/libraries/acml/downloads/pages/default.aspx*.

[77] Of, G. and Steinbach, O. The all-floating boundary element tearing and interconnecting method. *Journal of Numerical Mathematics*, 17(4):277–298, 2010.

[78] Of, G.; Steinbach, O. and Wendland, W. Applications of a fast multipole galerkin in boundary element method in linear elastostatics. *Computing and Visualization in Science*, 8:201–209, 2005.

[79] Of, G.; Steinbach, O. and Urthaler, P. Fast evaluation of volume potentials in boundary element methods. *SIAM Journal on Scientific Computing*, 32(2):585–602, 2010.

[80] OpenMP Architecture Review Board, . Openmp application program interface, May 2008. URL *http://www.openmp.org/mp-documents/spec30.pdf*.

[81] Ortega, J. M. and Rheinboldt, W. C. *Iterative Solution of Nonlinear Equations in Several Variables*, volume 30 of *Classics in Applied Mathematics*. SIAM, 1970.

[82] Parreira, P. On the accuracy of continuous and discontinuous boundary elements. *Engineering Analysis*, 5(4):205–211, 1988.

[83] Partridge, P.; Brebbia, C. A. and Wrobel, L. C. *The Dual Reciprocity Boundary Element Method*. Computational Mechanics Publications, Southampton, UK, 1992.

[84] Pereira, A. and Beer, G. Fluid-structure interaction by a duhamel-bem/fem coupling. In Manolis, G. D. and Polyzos, D., editors, *Recent Advances in Boundary Element Methods*, pages 339–354. Springer Netherlands, 2009.

[85] Phillips, J. and White, J. A precorrected-fft method for electrostatic analysis of complicated 3-d structures. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 16(10):1059 –1072, oct 1997.

[86] Prazeres, P. G.; Thoeni, K. and Beer, G. Nonlinear analysis of natm tunnel construction with the boundary element method. *Computers and Geotechnics*, 40(0): 160 – 173, 2012.

[87] Riederer, K.; Dünser, C. and Beer, G. Simulation of linear inclusions with the bem. *Engineering Analysis with Boundary Elements*, 33:959–965, 2009.

[88] Rizzo, F. An integral equation approach to boundary value problems of classical elastostatics. *Quarterly of Applied Mathematics*, 25(1):83–95, 1967.

[89] Rjasanow, S. and Steinbach, O. *The Fast Solution of Boundary Integral Equations (Mathematical and Analytical Techniques with Applications to Engineering)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[90] Rokhlin, V. Rapid solution of integral equations of classical potential theory. *Journal of Computational Physics*, 60(2):187 – 207, 1985.

[91] Rüberg, T. *Non-conforming Coupling of Finite and Boundary Element Methods in Time Domain*. PhD thesis, Graz University of Technology, 2007.

[92] Rüberg, T. and Schanz, M. Coupling finite and boundary element methods for static and dynamic elastic problems with non-conforming interfaces. *Computational Methods in Applied Mechanical Engineering*, 2008.

[93] Rüberg, T. and Schanz, M. An alternative collocation boundary element method for static and dynamic problems. *Computational Mechanics*, 44:247–261, 2009.

[94] Saad, Y. and Schultz, M. Gmres: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[95] Sauter, S. A. and Schwab, C. *Randelemente, Analyse und Implementierung schneller Algorithmen*. Teubner-Verlag, Wiesbaden, 2004.

[96] Simo, J. and Hughes, T. *Computational Inelasticity*. Springer, 1998.

[97] Simo, J. and Taylor, R. Consistent tangent operators for rate-independent elastoplasticity. *Computer Methods in Applied Mechanics and Engineering*, 48(1):101–118, 1985.

[98] Sládek, J.; Sládek, V. and Bažant, Z. P. Non-local boundary integral formulation for softening damage. *International Journal for Numerical Methods in Engineering*, 57 (1):103–116, 2003.

[99] Steinbach, O. Fast evaluation of newton potentials in boundary element methods. *East-West Journal of Numerical Mathematics*, 7(3):211–222, 1999.

[100] Steinbach, O. *Lösungsverfahren für Lineare Gleichungssysteme*. Mathematik für Ingenieure und Naturwissenschaftler. Teubner-Verlag, June 2005.

[101] Steinbach, O. *Numerical Approximation Methods for Elliptic Boundary Value Problems*. Springer, New York, NY, USA, 2008.

[102] Sutradhar, A.; Paulino, G. H. and Gray, L. J. *Symmetric Galerkin Boundary Element Method*. Springer-Verlag, Berlin-Heidelberg, 2008.

[103] Swedlow, J. and Cruse, T. Formulation of boundary integral equations for three-dimensional elasto-plastic flow. *International Journal of Solids and Structures*, 7 (12):1673 – 1683, 1971.

[104] Telles, J. C. F. *The Boundary Element Method Applied to Inelastic Problems*, volume 1 of *Lecture Notes in Engineering*. Springer-Verlag, Berlin-Heidelberg-New York, 1983.

[105] Telles, J. C. F. and Brebbia, C. A. On the application of the boundary element method to plasticity. *Applied Mathematical Modelling*, 3:466–470, 1979.

[106] Telles, J. C. F. and Carrer, J. A. M. Implicit procedures for the solution of elasto-plastic problems by the boundary element method. *Mathematical and Computer Modelling*, 15:303–311, 1991.

[107] Telles, J. and Brebbia, C. Boundary element solution for half-plane problems. *International Journal of Solids and Structures*, 17(12):1149 – 1158, 1981.

[108] Vorst, H.V. d. Bi-cgstab: a fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644, 1992.

[109] Wang, P. and Yao, Z. Fast multipole boundary element analysis of two-dimensional elastoplastic problems. *Communications in Numerical Methods in Engineering*, 23: 889–903, 2007.

[110] Watson, J. O. Boundary elements from 1960 to the present day. *Electronic Journal of Boundary Elements*, 1(1):34–46, 2003.

[111] Zhao, Z. and Lan, S. Boundary stress calculation–a comparison study. *Computers & Structures*, 71(1):77 – 85, 1999.