

# On the influence of far-field model reduction techniques using a coupled FEM-SBFEM approach in time domain

Marco Schauer<sup>a\*</sup>

<sup>a</sup>Technische Universität Braunschweig, Institut für Statik, Beethovenstraße 51, 38106 Braunschweig, Germany

\*Corresponding author E-mail: [m.schauer@tu-braunschweig.de](mailto:m.schauer@tu-braunschweig.de)

## Article Info

**Keywords:** Finite element method, Geometrical decoupling, Model reduction, Scaled boundary finite element method, Soil-structure interaction, Sub-structuring, Unbounded domain

**2010 AMS:**

**Received:** 20 June 2018

**Accepted:** 21 August 2018

**Available online:** 30 September 2018

## Abstract

To analyse soil-structure-interaction problems, often unbounded domain has to be taken into account. Since the finite element method (FEM) does not provide open boundary itself the scaled boundary finite element method (SBFEM) which fulfils the radiation condition for wave propagation to infinity is used. The coupling of FEM and SBFEM in time domain is very time and memory consuming, due to the almost fully populated acceleration unit-impulse matrices and the convolution integral, which has to be solved at every time step. This paper studies ways to overcome this drawback and describes the influence of different model reduction techniques: like extrapolated acceleration unit-impulse response matrices, geometric far-field decoupling and sub-structured far-fields which can be applied to the far-field and also their combination. The different techniques for a FEM-SBFEM coupling in time domain are evaluated in terms of accuracy and computational effort.

## 1. Introduction

The major motivations in the analysis of soil-structure interaction are to construct reliable earthquake-resistant structures and to isolate a building from surrounding emissions to enhance its comfort or usability. The second one becomes increasingly important in our urban society. In both cases, it is vital to analyse the structure and also to take the surrounding into account [1, 23, 25].

Numerical soil-structure interaction analysis in time domain is challenging since wave propagation towards infinity has to be taken into account. Due to the phenomenon, two significantly different types of mechanic have to be addressed. First, there is the region of interest, the near-field, typically the structure itself and second, there is the infinite half-space, the far-field, surrounding the structures foundation. The common finite element method (FEM) is not directly applicable to such problems, since it does not fulfill radiation condition implicitly. The FEM can be supplemented by transmitting boundary conditions [22], absorbing boundary conditions [12, 21] or other types of transmitting boundaries like infinite elements [2, 7]. Those types of boundary conditions are reflected in [13, 14].

A more precise alternative is to discretize the half-space with the help of either the boundary element method (BEM) [4, 6] or the scaled boundary finite element method (SBFEM) [31, 32]. The BEM is based on the boundary integral representation of the physical problem and uses its fundamental solution which fulfills the radiation condition exactly. Here, the SBFEM is used because it combines the advantages of FEM and BEM. The spatial dimension is reduced by one and the radiation conditions are satisfied exactly as it is in the BEM. Just like FEM, SBFEM does not require a fundamental solution and the coefficient matrices are symmetric and can be added to the FEM matrices without changing their dimension [31, 32]. The major advantages of the SBFEM in comparison to the BEM are the symmetric matrices, which can be exploited in storage and solving process and the absence of singular integrals, which need a very special mathematical and numerical treatment.

Such numerical schemes have been applied to different two and three dimensional applications successfully. It is especially challenging to address 3D applications since computational time and memory consumption are increasingly significant. The original formulation of a coupled FEM SBFEM approach is global in space and time; hence, all degrees of freedom are coupled at the common interface and a convolution integral has to be solved [31, 32]. To solve the convolution integral acceleration unit-impulse response matrices  $\mathbf{M}_t^\infty$  have to be provided for each time step  $t$ . How the  $\mathbf{M}_t^\infty$  matrices are computed, is explained in [29] in detail.

With advance in time, the acceleration unit-impulse response matrix  $\mathbf{M}^\infty$  grows from time step  $t_i$  to  $t_{i+1}$  with a constant increment [34, 33]. Hence,  $\mathbf{M}^\infty$  can be extrapolated at later time steps, assuming a piece-wise constant approximation of  $\mathbf{M}^\infty$  at each time step. Based on this

approach, a recursive algorithm to speed up the convolutions integral computation significantly has been proposed [17, 18, 19]. A high performance SBFEM using different time scales applied to near field and far field is also discussed [27]. Instead of solving the convolution integral high-order local approaches based on continued-fractions have been proposed as well [3, 5, 9].

If real-world problems are to be investigated, this often results in very complex models with a large number of degrees of freedom. In order to be able to calculate these complex models within a reasonable time and, if necessary, to carry out additional parameter studies, it is important to have efficient models and algorithms. Therefore, this article discusses different ways of model reduction.

Here, the influence of model reduction techniques, like extrapolated acceleration unit-impulse response matrices, geometrical decoupling and sub-structured far-field, is investigated. The FEM-SBFEM coupling in time domain is analyzed in terms of accuracy and computational effort. These model reductions are applied to the original formulation of SBFEM. The influence is shown by conducting a numerical settlement simulations and taking different mesh refinements and sets of material parameters into account.

## 2. FEM-SBFEM coupling in time domain

Near-field and far-field are discretized by FEM and SBFEM, respectively. The near-field represents the region of interest, including structure foundation and parts of the surrounding if needed. The far-field represents the infinite half-space. Both methods are coupled at a defined common interface  $\Gamma$ , as shown in fig. 2.1.

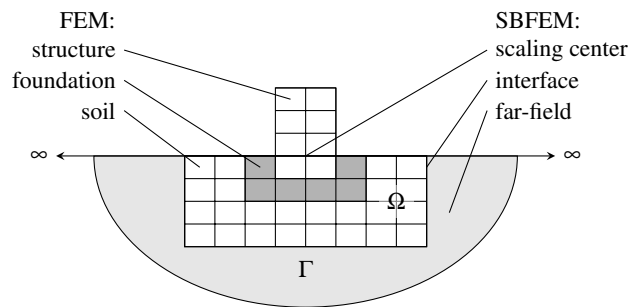


Figure 2.1: Problem definition.

The equation of motion for the displacement-based FEM is given by

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{p}, \quad (2.1)$$

in which  $\mathbf{M}$  is the mass matrix and  $\mathbf{K}$  the stiffness matrix. The vector  $\mathbf{u}$  and its second derivative in time represent displacement and acceleration. Assuming that the period  $T$  can be split up into  $n$  time steps of the same size yields to a time step length  $\Delta t = \frac{T}{n}$ . Applying the Generalized- $\alpha$  scheme [10] to equation (2.1) leads to

$$(1 - \alpha_m)\mathbf{M}\ddot{\mathbf{u}}_{n+1} + \alpha_m\mathbf{M}\ddot{\mathbf{u}}_n + (1 - \alpha_f)\mathbf{K}\mathbf{u}_{n+1} + \alpha_f\mathbf{K}\mathbf{u}_n = \mathbf{p}_{n+1} - \alpha_f\Delta t\mathbf{p}_n. \quad (2.2)$$

To advance in time, update rules for displacement

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta t\dot{\mathbf{u}}_n + \left(\frac{1}{2} - \beta\right)\Delta t^2\ddot{\mathbf{u}}_n + \beta\Delta t^2\ddot{\mathbf{u}}_{n+1}, \quad (2.3)$$

and velocity

$$\dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_n + (1 - \gamma)\Delta t\ddot{\mathbf{u}}_n + \gamma\Delta t\ddot{\mathbf{u}}_{n+1} \quad (2.4)$$

are introduced. Thus, for  $\alpha_m = 0$ , the Generalized- $\alpha$  scheme is equal to the HHT- $\alpha$  [16] scheme and for  $\alpha_m = 0$  and  $\alpha_f = 0$  equal to the Newmark integration method [24]. The parameters  $\alpha_f$ ,  $\alpha_m$ ,  $\beta$ , and  $\gamma$  of the time step integration scheme should be set as follows

$$\alpha_m \leq \alpha_f \leq \frac{1}{2}, \quad \beta \geq \frac{1}{4} + \frac{1}{2}(\alpha_f - \alpha_m) \quad \text{and} \quad \gamma = \frac{1}{2} - \alpha_m + \alpha_f. \quad (2.5)$$

In order to couple FEM and SBFEM, the entries of the matrices in equation (2.1) have to be reordered

$$\begin{bmatrix} \mathbf{M}_{\Omega\Omega} & \mathbf{M}_{\Omega\Gamma} \\ \mathbf{M}_{\Gamma\Omega} & \mathbf{M}_{\Gamma\Gamma} \end{bmatrix} \ddot{\mathbf{u}} + \begin{bmatrix} \mathbf{K}_{\Omega\Omega} & \mathbf{K}_{\Omega\Gamma} \\ \mathbf{K}_{\Gamma\Omega} & \mathbf{K}_{\Gamma\Gamma} \end{bmatrix} \mathbf{u} = \begin{bmatrix} \mathbf{p}_{\Omega\Omega} \\ \mathbf{p}_{\Gamma\Gamma} \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ \mathbf{p}_b \end{bmatrix} \quad (2.6)$$

so that the block with the subscript " $\Omega\Omega$ " contains all nodes located in the near-field while the block with subscript " $\Gamma\Gamma$ " contains all nodes at the far-field-interface. The blocks with the subscripts " $\Omega\Gamma$ " and " $\Gamma\Omega$ " include the coupling information of near-field and far-field nodes. The vector  $\mathbf{p}_b$  acts on the boundary  $\Gamma$  only. This additional force describes the response of the infinite half-space and can be applied to the near-field as a load.

The forces acting at the interface of near-field and far-field are given by the convolution integral

$$\mathbf{p}_b(t) = \int_0^t \mathbf{M}^\infty(t - \tau)\ddot{\mathbf{u}}(\tau)d\tau, \quad (2.7)$$

in which  $\mathbf{M}^\infty(t)$  is the unit-impulse matrix. To solve the convolution integral (2.7), the unit-impulse matrices  $\mathbf{M}_i^\infty$  are assumed to be constant within the time step  $\Delta t$ ,

$$\mathbf{M}^\infty(t) = \begin{cases} \mathbf{M}_0^\infty & t \in [0; \Delta t], \\ \mathbf{M}_1^\infty & t \in [\Delta t; 2\Delta t], \\ \vdots & \vdots \\ \mathbf{M}_{n-j}^\infty & t \in [(n-j-1)\Delta t; (n-j)\Delta t], \\ \vdots & \vdots \\ \mathbf{M}_{n-1}^\infty & t \in [(n-2)\Delta t; (n-1)\Delta t], \\ \mathbf{M}_n^\infty & t \in [(n-1)\Delta t; n\Delta t]. \end{cases} \quad (2.8)$$

Due to this assumption and when applying the time step integration scheme, equation (2.7) can be rewritten as

$$\mathbf{p}_b(t_n) = \gamma \Delta t \mathbf{M}_0^\infty \ddot{\mathbf{u}}_n + \sum_{j=1}^{n-1} \mathbf{M}_{n-j}^\infty (\dot{\mathbf{u}}_j - \dot{\mathbf{u}}_{j-1}). \quad (2.9)$$

The coupling of FEM and SBFEM is realized by simply adding equation (2.9) to the resorted FEM (2.6)

$$\begin{bmatrix} \mathbf{M}_{\Omega\Omega} & \mathbf{M}_{\Omega\Gamma} \\ \mathbf{M}_{\Gamma\Omega} & \mathbf{M}_{\Gamma\Gamma} + \gamma \Delta t \mathbf{M}_0^\infty \end{bmatrix} \ddot{\mathbf{u}} + \begin{bmatrix} \mathbf{K}_{\Omega\Omega} & \mathbf{K}_{\Omega\Gamma} \\ \mathbf{K}_{\Gamma\Omega} & \mathbf{K}_{\Gamma\Gamma} \end{bmatrix} \mathbf{u} = \begin{bmatrix} \mathbf{P}_{\Omega\Omega} \\ \mathbf{p}_{\Gamma\Gamma} - \sum_{j=1}^{n-1} \mathbf{M}_{n-j}^\infty (\dot{\mathbf{u}}_j - \dot{\mathbf{u}}_{j-1}) \end{bmatrix}, \quad (2.10)$$

so the FEM-SBFEM coupling is fully described mathematically.

### 3. Numerical example

For the purpose of validation, a simple settlement problem is chosen and the numerical solution will be compared by analytical solutions. Therefore, an infinite half-space is loaded on a square region of  $152.4 \times 152.4\text{m}^2$  by a constant load  $q = 70.0\text{kNm}^{-2}$  as depicted in figure 3.1. The SBFEM scaling centre is located in the centre of the loaded area. The distance between scaling centre and boundary  $\Gamma$  is given by the radius  $r = 190.2\text{m}$ . Loosely deposited sand is chosen, the material parameters are set as follows: Young's modulus  $E = 37150.0\text{kNm}^{-2}$ , Poisson's ratio  $\nu = 0.48$  and mass  $\rho = 1800.0\text{kgm}^{-3}$ .

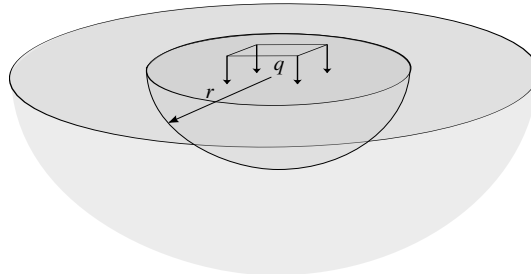


Figure 3.1: Settlement problem configuration: Near-field with radius  $r = 190.2\text{m}$  and distributed load  $q = 70\text{kNm}^{-2}$  on an area of  $152.4 \times 152.4\text{m}^2$ .

#### 3.1. Analytical solution

The settlement and stresses underneath a loaded area at an infinite, isotropic and elastic half space can be evaluated analytically. For the given problem, the settlement at the corner of the applied load is evaluated by solving a semi-analytical approach suggested by Harr [15]. This approach can be simplified when the loaded area is perfectly square

$$s(z) = \frac{qb}{2\pi E} (1 - \nu^2) \left( 2 \ln \left( \frac{m+1}{m-1} \right) - n \frac{1-2\nu}{1-\nu} \arctan \left( \frac{1}{nm} \right) \right) \quad (3.1)$$

with  $m = \sqrt{2+n^2}$  and  $n = \frac{z}{b}$ . The corresponding stresses and shear stresses along the  $z$ -direction are given by analytic solutions [26], which can also be simplified, since the loaded area is perfectly square. Due to the symmetry of the given example  $\sigma_x = \sigma_y$  and  $\tau_{yz} = \tau_{xz} = \tau_{xy} = 0.0$  in the very centre of the loaded area, so that for validation purpose

$$\sigma_x(z) = \frac{q}{2\pi} \left( \arctan \left( \frac{b^2}{z\sqrt{2b^2+z^2}} \right) - \frac{b^2 z}{(b^2+z^2)\sqrt{2b^2+z^2}} \right), \quad (3.2)$$

$$\sigma_z(z) = \frac{q}{2\pi} \left( \arctan \left( \frac{b^2}{z\sqrt{2b^2+z^2}} \right) + \frac{2b^2 z}{(b^2+z^2)\sqrt{2b^2+z^2}} \right), \quad (3.3)$$

$$\tau_{xy} = 0.0 \quad (3.4)$$

as well as the von Mises stress

$$\sigma_v(z) = \sqrt{\sigma_x(z)^2 - 2\sigma_x(z)\sigma_z(z) + \sigma_z(z)^2} \quad (3.5)$$

are taken into account. Thereby,  $b$  describes the physical dimension of the loaded area. To evaluate the settlement and stresses in the centre of the loaded area, the loaded region has to be subdivided into four squares of the same size and has to be evaluated separately. The outcome result of these four subregions must be superposed to get the state variables of settlement and stresses.

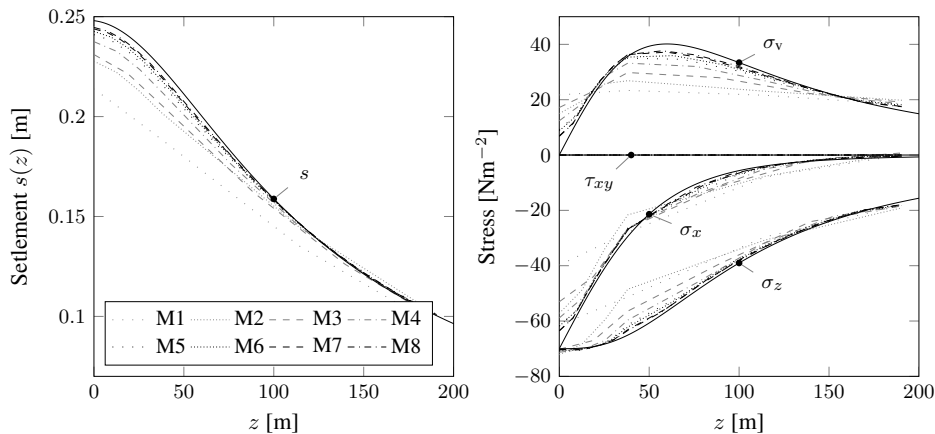
### 3.2. Numerical solution

In order to show the accuracy of the coupled FEM-SBFEM method, the meshes of near-field and far-field are refined several times, so that the geometry becomes smoother with each step of refinement. This also leads to an increasing number of degrees of freedom (DoF), as shown in Table 1. To speed up the computation, all meshes have been optimized by renumbering the containing DoF [11].

**Table 1:** Mesh discretization with different number of DoF. The meshes correspond to [29].

Mesh	M1	M2	M3	M4	M5	M6	M7	M8
DoF <sub>FEM</sub>	396	738	1314	3096	6030	9033	14655	19818
DoF <sub>SBFEM</sub>	123	219	291	480	843	1083	1515	1827

The longitudinal wave speed is  $c_p = \sqrt{\frac{E(1-\nu)}{\rho(1+\nu)(1-2\nu)}} = 425.779\text{ms}^{-1}$ , so that the critical time step length is given by  $\Delta t_{\text{crit}} = \frac{r}{30c_p} = 0.0149\text{s}$  as it is suggested by Borsutzky [8]. The numerical solution is carried out for a period of 6.25s with a time step length  $\Delta t = 0.0125\text{s} \leq \Delta t_{\text{crit}}$ , so that 500 time steps have to be computed. Since the settlement problem is solved in time domain, the chosen period must assure that the system reaches its steady-state within this period [29]. The parameters of the Generalized- $\alpha$  time step integration scheme can be defined by a dissipation parameter within the range  $0 \leq \rho_\infty \leq 1$  ( $\rho_\infty = 0$  asymptotic annihilation,  $\rho_\infty = 1$  no dissipation). Here,  $\rho_\infty = 0.6$  is chosen; hence, numerical damping is introduced by the time step integration scheme, this yields to integration parameters  $\alpha_m = \frac{1}{8}$ ,  $\alpha_f = \frac{3}{8}$ ,  $\beta = \frac{25}{64}$ , and  $\gamma = \frac{3}{4}$ .



**Figure 3.2:** Analytic (solid black line) and numerical solutions for settlement and stresses.

In figure 3.2, the analytical solutions for settlement (eq. (3.1)) and stresses (eq. (3.2)–(3.5)) are shown. The analytical solutions and the numerical results at steady-state of the meshes M1 to M8 are depicted. Refining the meshes leads to a better accuracy of the numerical results. This is confirmed by the graphs.

## 4. Model reduction techniques

Computing the  $\mathbf{M}^\infty$  matrices and solving the convolution integral requires a high computational effort. To simulate  $n$  time steps, for each time step a matrix which includes the  $N$  DoF at the interface  $\Gamma$  has to be stored in computers memory. Since the matrices are usually almost fully populated,  $N \times N \times n$  values have to be stored. Additionally, there are  $N \times n$  values for the nodal velocities at the interface to store. Another disadvantage is the convolution integral itself, since its computation requires more and more time with increasing time steps to conduct the matrix vector multiplications. These two disadvantages of this coupled approach cannot be cured completely, but significantly improved. To overcome the high time and memory consumption of the coupled FEM-SBFEM approach in time domain, different model reduction techniques are discussed and evaluated in terms of accuracy and computational effort. The following sections pay attention on how the results are influenced by applying the following model reduction techniques: extrapolated acceleration unit-impulse response matrices 4.1, geometric far-field decoupling 4.2 and sub-structured far-fields 4.3 which can be applied to the far-field. Additionally, combinations of model reduction techniques are discussed 5.

### 4.1. Extrapolation of $\mathbf{M}^\infty$

Analysing the evolution of the  $\mathbf{M}^\infty$  entries exhibit that they grow constantly after a certain time step. The simplest and fastest way to check the constant growth is to evaluate the behaviour of one arbitrary single matrix entry to find the time step  $t_m$  from which linear behaviour of

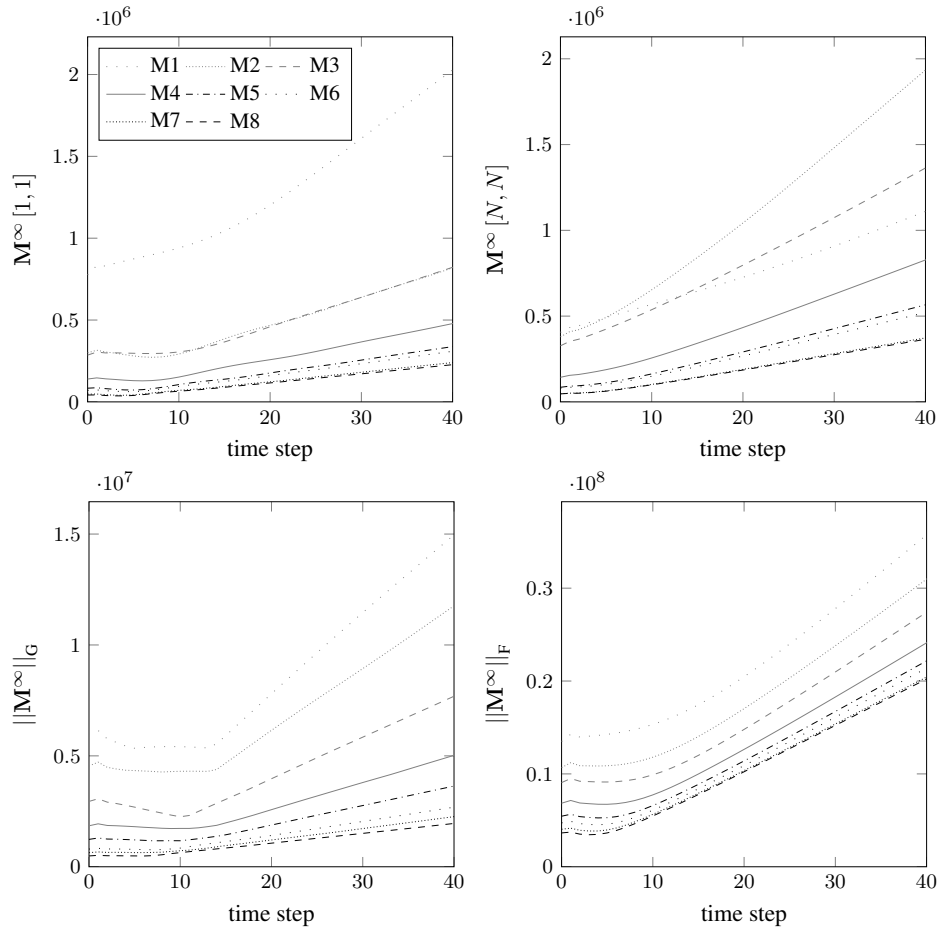


Figure 4.1: The time dependent behaviour of  $M^\infty$  evaluated by different approaches.

the matrix is assumed. Alternatively, a matrix norm can be used to check constant growth of the matrices, like the total norm

$$\|M^\infty\|_G = N \max_{i,j=1,\dots,N} |m_{ij}| \tag{4.1}$$

or the Frobenius norm

$$\|M^\infty\|_F = \sqrt{\sum_{i=1}^N \sum_{j=1}^N m_{ij}^2}. \tag{4.2}$$

All three approaches lead to remarkably different data, which has to be reviewed. This also results in significantly different assumptions of the time step  $t_m$ , starting from which constant growth can be assumed and extrapolation is started for the coupled computation. The data of the first 40 time steps is shown in figure 4.1. Reviewing the graphs in figure 4.1 consequently leads to the following points:

- (i) The values inside the matrix do not behave in the same manner. This is clearly depicted by the graphs  $M_n^\infty[1; 1]$  and  $M_n^\infty[N; N]$ . From this, it follows that the chosen matrix entry has an influence on the time step  $t_m$  and, thus, on the coupled problems solution.
- (ii) The total norm  $\|M^\infty\|_G$  provides usable values which are adverse for numerical treatment. Checking constant growth by algorithms might lead to improper time starting from where constant growth is assumed and consequently to wrong solutions.
- (iii) The Frobenius norm  $\|M^\infty\|_F$  shows the best behaviour to check constant growth. This is doubtless due to the consideration of all matrix entries. Unfortunately, this approach also is the one with the highest computational effort.

#### 4.1.1. Algorithm

The algorithm discussed is based on [17]. For the purpose of linearising the unit-impulse matrices, these matrices have to split up for every time step  $t_n$  with  $n > m$ , that yields

$$M_n^\infty = T^\infty t_n + C^\infty. \tag{4.3}$$

Here,  $C^\infty$  is a constant matrix and  $T^\infty$  describes the gradient of the unit-influence matrix  $\left(\frac{\Delta M^\infty}{\Delta t}\right)$ , which is

$$T^\infty = M_{m+1}^\infty - M_m^\infty \tag{4.4}$$

for an equidistant time step size  $\Delta t$ . Equation (2.9) has to be split up into two subtotals. The first subtotal considers the non-linear matrices for the time steps  $t_n$ ,  $1 \leq n \leq m$ , the second subtotal represents the linear matrices for all time steps  $t_n$ ,  $n > m$ . This yields to

$$\begin{aligned} \mathbf{p}_b(t_n) &= \gamma \Delta t \mathbf{M}_0^\infty \ddot{\mathbf{u}}_n + \mathbf{p}_b^1(t_n) + \mathbf{p}_b^{\text{nl}}(t_n), \\ &= \gamma \Delta t \mathbf{M}_0^\infty \ddot{\mathbf{u}}_n + \sum_{j=1}^{n-m+1} \mathbf{M}_{n-j+1}^\infty (\dot{\mathbf{u}}_j - \dot{\mathbf{u}}_{j-1}) + \sum_{j=n-m+2}^{n-1} \mathbf{M}_{n-j+1}^\infty (\dot{\mathbf{u}}_j - \dot{\mathbf{u}}_{j-1}). \end{aligned} \quad (4.5)$$

Solving the non-linear terms  $\mathbf{p}_b^{\text{nl}}(t_n)$  is analogous to equation (2.9). The linear term  $\mathbf{p}_b^1(t_n)$  is transferred into a recursive algorithm. Inserting equation (4.3) into the linear term of equation (4.5) leads to

$$\begin{aligned} \mathbf{p}_b^1(t_n) &= \sum_{j=1}^{n-m+1} (\mathbf{T}^\infty t_{n-j+1} + \mathbf{C}^\infty) (\dot{\mathbf{u}}_j - \dot{\mathbf{u}}_{j-1}), \\ &= (\mathbf{T}^\infty t_m + \mathbf{C}^\infty) (\dot{\mathbf{u}}_{n-m+1} - \dot{\mathbf{u}}_{n-m}) + \sum_{j=1}^{n-m} (\mathbf{T}^\infty t_{n-j+1} + \mathbf{C}^\infty) (\dot{\mathbf{u}}_j - \dot{\mathbf{u}}_{j-1}), \\ &= \mathbf{M}_m^\infty (\dot{\mathbf{u}}_{n-m+1} - \dot{\mathbf{u}}_{n-m}) + \sum_{j=1}^{n-m} (\mathbf{T}^\infty t_{n-j+1} + \mathbf{C}^\infty) (\dot{\mathbf{u}}_j - \dot{\mathbf{u}}_{j-1}). \end{aligned} \quad (4.6)$$

Inside the recursive algorithm

$$\mathbf{p}_b^1(t_{n-1}) = \sum_{j=1}^{n-m} (\mathbf{T}^\infty t_{n-j} + \mathbf{C}^\infty) (\dot{\mathbf{u}}_j - \dot{\mathbf{u}}_{j-1}) \quad (4.7)$$

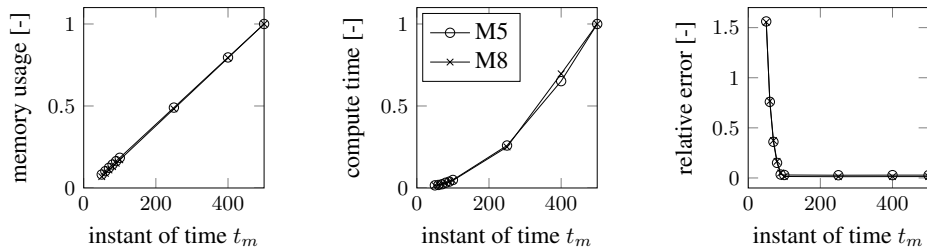
is needed, hence,  $n$  has to be substituted by  $n-1$  in equation (4.6). Applying the difference from  $\mathbf{p}_b^1(t_n)$  to  $\mathbf{p}_b^1(t_{n-1})$  results in the recursive formula

$$\mathbf{p}_b^1(t_n) = \mathbf{p}_b^1(t_{n-1}) + \mathbf{M}_m^\infty (\dot{\mathbf{u}}_{n-m+1} - \dot{\mathbf{u}}_{n-m}) + \mathbf{T}^\infty (\dot{\mathbf{u}}_{n-m} - \dot{\mathbf{u}}_0). \quad (4.8)$$

Each following time step can be computed in the same manner. Based on this linearisation,  $N \times N \times m$  values are to be stored instead of the previous  $N \times N \times n$  values, with  $m \leq n$ . The number of interface velocities  $N \times n$  to be stored does not change.

#### 4.1.2. Numerical results

The influence of matrix extrapolation to the solution of a given problem is discussed next. Therefore, two meshes (M5 and M8) from section 3 are analyzed, assuming that the unit-impulse response matrix grows constantly after a defined instant of time  $t_m$ . When constant growth of  $\mathbf{M}^\infty$  is assumed, only  $m$  matrices are computed, all other matrices are extrapolated (see eq. (4.4)).



**Figure 4.2:** Influence of matrix extrapolation to the settlement problem: Normalized memory usage, normalized compute time and corresponding relative error.

Figure 4.2 summarizes the most important results of the examination. The normalized memory consumption, normalized compute time and relative error of the computed displacement  $s(z=0)$  in the centre of the loaded area with respect to the instant of time  $t_m$  from where constant growth of  $\mathbf{M}^\infty$  is assumed are shown. The memory needed to provide the  $\mathbf{M}^\infty$  matrices can be reduced significantly by assuming that they grow with constant increment. In order to show the reduction of memory usage for each instant of time  $t_m$ , the memory usage is related to the memory usage when no model reduction is done and all matrices are provided, hence  $t_m = 500$ .

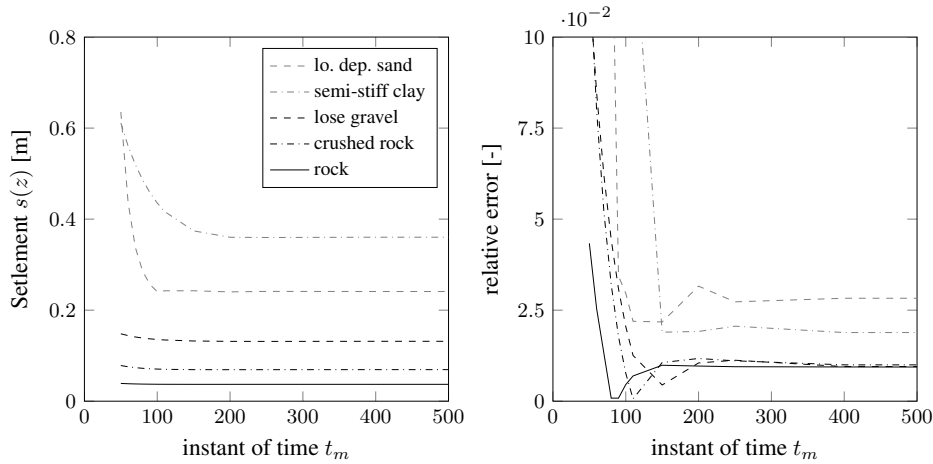
By reducing the number of  $\mathbf{M}^\infty$  matrices, the computational effort is reduced as well. The corresponding normalized computing times are shown for both meshes. In both cases, the computation time of  $t_m = 500$  has been used for normalization. As long as the instant of time  $t_m$  is chosen at a permissible time step, the relative error of the numerical simulation stays constant. If the number of provided matrices is too little (extrapolation it started too early), the relative error of the computation increases obviously.

Here, 100 matrices are sufficient to solve the given problem without increasing the relative error. This means a reduction of memory consumption of 81.6% and at the same time a reduction of the computational effort of 95.1% in case of mesh M5. The reductions for mesh M8 are even greater with 82.8% and 95.4%. When too few matrices are provided, consequently, the far-field leaks stiffness and the computed settlement becomes too large. If the number of matrices is reduced further, the simulation may become unstable and the near-field might float into direction of the applied forces due to insufficient far-field response.

Further numerical studies using different sets of material parameters, as summarised in table 2, have been conducted using the mesh M5. The time step  $\Delta t$  and the integration scheme parameters stay unchanged. All conducted simulations lead to similar results regarding the following parameters: memory consumption, compute time and relative error. Figure 4.3 shows the settlement at time step  $t = 500$ , 6.25s at the surface in the centre of the loaded area, when only  $t_m = \{50, 60, 70, 80, 90, 100, 110, 150, 200, 250, 400, 500\}$   $\mathbf{M}^\infty$  matrices are provided and all further matrices are extrapolated as discussed before. Additionally, the corresponding relative error with respect to equation (3.1) is shown up to 10%. It is obvious that the error increases when too few matrices are provided and extrapolating of the far-fields' influence starts too early.

**Table 2:** Material sets.

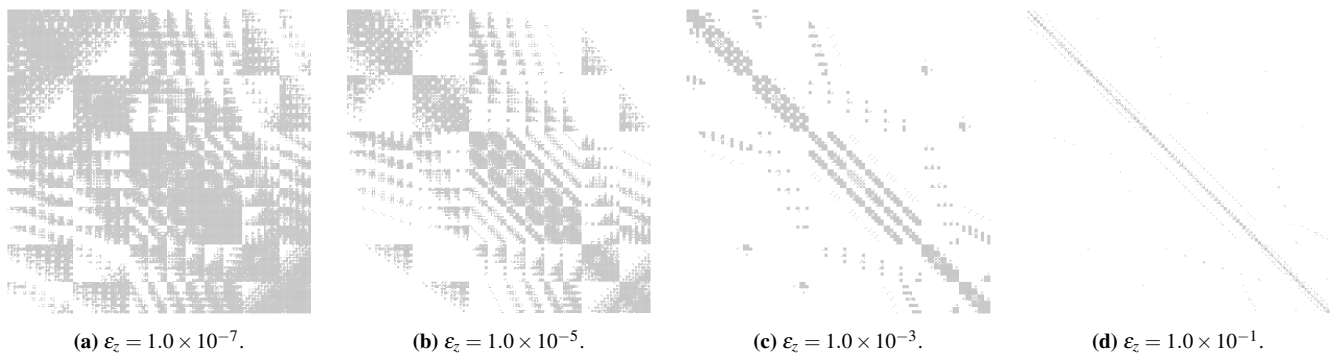
	$E$ kNm <sup>-2</sup>	$\nu$	$\rho$ kgm <sup>-3</sup>	$c_p$ ms <sup>-1</sup>	$c_s$ ms <sup>-1</sup>
loosely deposited sand	37150.0	0.48	1800.0	425.779	83.502
lose gravel	82000.0	0.30	1600.0	262.660	140.398
semi-stiff clay	26000.0	0.45	2200.0	211.725	63.838
crushed rock	150000.0	0.35	2100.0	338.583	162.650
rock	300000.0	0.25	2400.0	387.299	223.607



**Figure 4.3:** Settlements' steady-state and relative error at 6.25s, when  $t_m$  is chosen.

### 4.2. Far-field decoupling

The matrices are fully populated, so that all nodes at the near-field far-field interface are coupled with each other. Having a closer look to the matrix entries lead to the awareness that the geometrical distance of two nodes at the interface  $\Gamma$  influences the matrix entry strongly. The position of two nodes within the matrix has no influence. How strong the influence of two node turns out is determined by the corresponding entry in  $\mathbf{M}^\infty$  matrices. The smaller the distance of two node (the closer they are), the bigger is this matrix entry. The bigger the distance of two nodes, the smaller is the matrix entry. Assuming that the influence of two very far nodes has no recognizable influence to the simulation lead to introduce a threshold  $\epsilon_z$ , which defines the minimum value considered in the  $\mathbf{M}^\infty$  matrices. The threshold is applied to the matrix entries after its computation, so it is some kind of a post process. Figure 4.4 shows the assignment of  $\mathbf{M}_0^\infty$  at time step  $t = 0$ , all non-zero entries are pictured by a small dot. Entries containing value of zero or a value smaller then  $\epsilon_z$  are not pictured. Here, the threshold varies within the range  $1.0 \times 10^{-7} \leq \epsilon_z \leq 1.0 \times 10^{-1}$ . It is evident how the introduced threshold influences the matrix appearance. The bigger  $\epsilon_z$  is chosen to be, the less information is stored in the matrix.

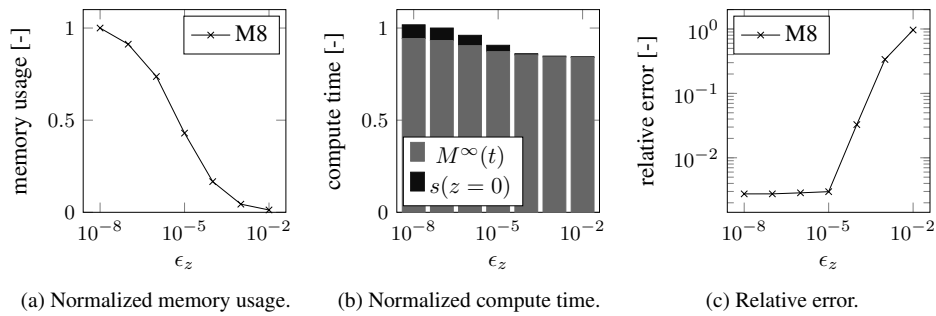


**Figure 4.4:** M3 matrix assignment of  $\mathbf{M}_0^\infty$  using geometrical decoupling.

The pictured matrices are generated by examining the example M3 (see Table 1). If no threshold is set, the matrices have  $291 \times 291 = 84681$  entries, and are fully populated. Introducing threshold  $\epsilon_z = 1.0 \times 10^{-7}$  results in 49855 non-zero values and sparsity of 58.87%. When  $\epsilon_z = 1.0 \times 10^{-5}, 1.0 \times 10^{-3}, 1.0 \times 10^{-1}$  is set 28.6%, 8.51% and 1.2% of memory is needed, respectively. Similar reductions have been shown in [8, 20].

Since the magnitude of each matrix entry changes from one time step to another, the number of matrix entries to be stored may change as well. In order to optimize the memory needed, the sparsity pattern is not constant any more, but may change from one time step to another. That is why the implementation of the algorithm has to be done carefully.

The influence of far-field decoupling is analysed by evaluation memory usage, compute time and relative error of final displacement. Therefore, the influence matrices  $\mathbf{M}_i^\infty$  are computed using different geometrical thresholds  $\epsilon_z$  for M8 (see Table 1). The tolerance value lies within the range  $\epsilon_z = 1.0 \times 10^{-8}$  to  $\epsilon_z = 1.0 \times 10^{-2}$ . The results are shown in figure 4.5.



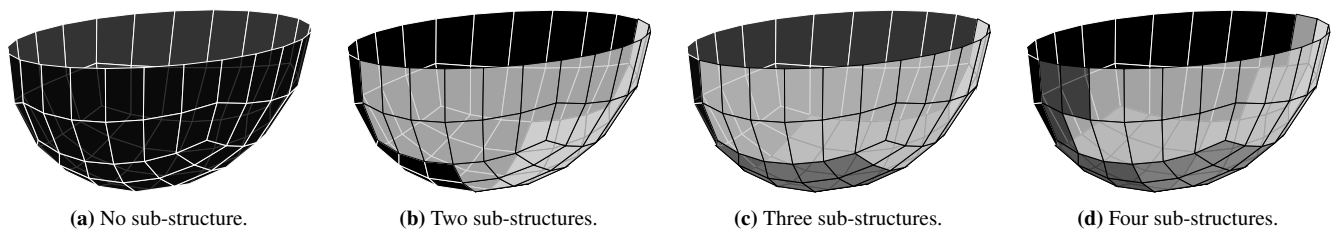
**Figure 4.5:** Normalized memory usage of  $\mathbf{M}_0^\infty$  (a), normalized compute time (b) and relative error of final displacement (c) with respect to the chosen geometrical decoupling threshold  $\epsilon_z$ .

Memory usage and compute time are normalized by the chosen reference solution using  $\epsilon_z = 1.0 \times 10^{-8}$ . The relative error shows the computed displacement  $s(z=0)$  compared to the given analytic solution in equation (3.1). It is obvious that increasing the  $\epsilon_z$  decreases memory usage strongly and somewhat compute time, as shown in figure 4.5a and figure 4.5b, respectively. In this case, for small  $1.0 \times 10^{-8} \leq \epsilon_z \leq 1.0 \times 10^{-5}$  the relative error stays almost constant, and the advantage of less memory usage (down to 16%) should be used. When  $\epsilon_z$  becomes too large the error increases and the user has to decide up to which error the result is still acceptable (see figure 4.5c). The other meshes M1 to M7 show very similar behaviour.

Far-field decoupling only reduces the amount of memory needed, since the entire far-field is discretized. Therefore, the far-field decoupling has no relevant influence to compute effort and time. As one can see in figure 4.5b, almost 85% of compute time is needed compared to the reference simulation also for large tolerances  $\epsilon_z \geq 1,0 \cdot 10^{-4}$ . The reason for this behaviour is also obvious, since the system size at the interface stays constant and so the complexity of equations to solve. Only the time needed for memory access does change since matrices may be more sparse due to the far-field decoupling.

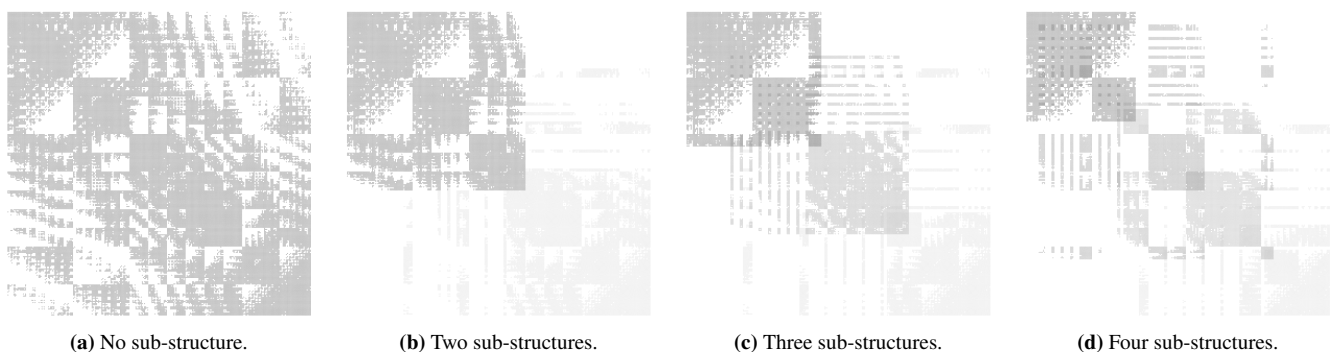
### 4.3. Far-field sub-structuring

As already mentioned, the geometrical far-field decoupling has no relevant influence on the compute time. Alternatively to the model reduction in post process, as described in previous section, it is possible to do a model reduction in a preprocess, before the matrices  $\mathbf{M}^\infty$  are computed. For this purpose, the far-field has to be sub structured [32]. When the far-field is decomposed in sub structures artificial boundaries are introduced. These boundaries decouple nodes with large distance, which have small interaction, as described before. Each sub structure contains only a fraction of all nodes located at the interface  $\Gamma$ . For each sub structure  $\mathbf{M}^\infty$  can be computed independently. In figure 4.6a, the interface of mesh M3 is shown. The figures 4.6b to 4.6d show sub-structured interfaces examples.



**Figure 4.6:** M3 subdivided in different numbers of sub-structures.

This is justified by the coupling of FEM and SBFEM, since the substructures have common interface nodes. This common interface nodes couple the different substructures on the boundary. Along the line from scaling center to these nodes towards infinity the substructures stay uncoupled. The coupling of sub-structured influence matrices, here  $\mathbf{M}_0^\infty$ , is shown in the figures 4.7b to 4.7d. The pattern of the full matrix 4.7a is also present in the pattern of sub-structured matrices. The color of matrix entries in figure 4.7 correspond to the color of substructures in figure 4.6.



**Figure 4.7:** M3 matrix assignment of  $\mathbf{M}_0^\infty$  using sub-structuring.



For each sub-structure,  $\varepsilon_z$  is set to  $1.0 \times 10^{-7}$  when  $\mathbf{M}_0^\infty$  is computed, so that the figures 4.7a and 4.4a are identical in comparison of related figures. In contrast to the geometrical decoupling, the pattern of the influence matrix changes in a totally different way when sub-structuring is applied. The reduction of memory usage is possible and can be derived by figure 4.7 easily. It is also obvious that with an increasing number of sub-structures the size of each matrix is reduced and the ratio of matrix entries to allocated values increases. This indicates that the interaction of the nodes within one sub-structure is relatively strong. With increasing number of sub-structures the number of non-zero values within a matrix increases as well, since the nodes within a sub-structure are geometrically very close to each other. The size of  $\mathbf{M}_0^\infty$  corresponding to the sub-structured domain pictured in figures 4.6 as well as the allocated number of entries and the resulting ratio compared to the full matrix without sub-structuring is summarized in table 3.

**Table 3:** Size and allocation of sub-structured  $\mathbf{M}_0^\infty$  matrices.

sub-structures	matrix size	entries <sub>sub</sub>	allocated <sub>tot</sub>	ratio [%]
1	291 × 291	84681	84681	100.00
2	171 × 171	29241	58482	69.06
	171 × 171	29241		
3	129 × 129	16641	48411	57.17
	123 × 123	15129		
	129 × 129	16641		
4	105 × 105	11025	42255	49.90
	102 × 102	10404		
	99 × 99	9801		
	105 × 105	11025		

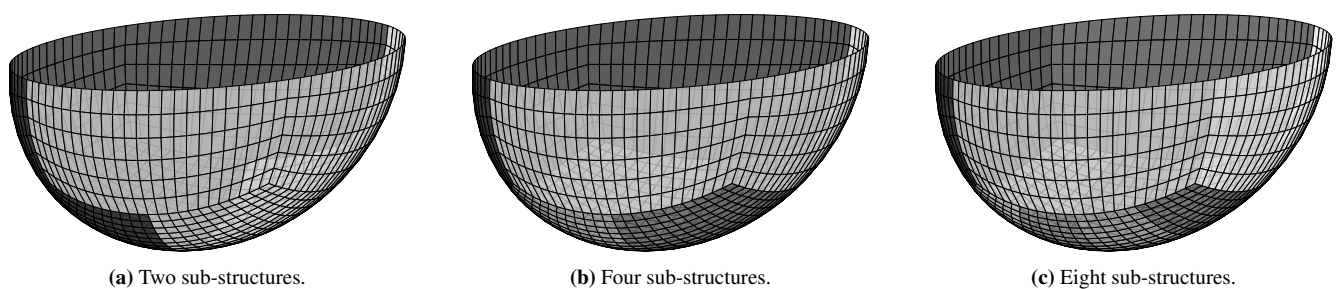
The artificial boundary does not only decouple far distant nodes, but also direct neighbor nodes, who naturally strongly interact. However, this approach leads to good results if the topology of element mesh is taken into account during sub-structuring process [30]. Theoretically, it is also possible to do the sub-structuring based on nodes instead of elements. This would reduce the amount of data even further; hence, interface nodes would appear only once. This certainly implies that information of some elements is lost, so that the influence of the half-space is not fully considered. Overlapping of sub-structures is impossible, since there are no common nodes. If the size of the neglected elements is small, this approach leads to a comparable approach. Another possibility is to use patches as discussed in [28]. The reduction of memory usage is obvious as well and can be derived by figure 4.7 easily.

In order to analyze the influence of sub-structuring technique to the solution, computations with different number of sub-structures are performed. Therefore, the far-field-interface of M8 is decomposed based on elements into  $n$  sub-structures. As already mentioned, each sub-structure represents only a part of the entire far-field and can be solved separately. By doing so, the system size of each sub-structure is reduced compared to the initial configuration, in which the entire interface is taken into account. When the given problem is computed FEM and SBFEM are coupled (cf. eq. (2.6)) and consequently the sub-structures as well:

$$\begin{bmatrix} \mathbf{M}_{\Omega\Omega} & \mathbf{M}_{\Omega\Gamma} \\ \mathbf{M}_{\Gamma\Omega} & \mathbf{M}_{\Gamma\Gamma} \end{bmatrix} \ddot{\mathbf{u}} + \begin{bmatrix} \mathbf{K}_{\Omega\Omega} & \mathbf{K}_{\Omega\Gamma} \\ \mathbf{K}_{\Gamma\Omega} & \mathbf{K}_{\Gamma\Gamma} \end{bmatrix} \mathbf{u} = \begin{bmatrix} \mathbf{p}_{\Omega\Omega} \\ \mathbf{p}_{\Gamma\Gamma} \end{bmatrix} - \sum \begin{bmatrix} \mathbf{0} \\ \mathbf{p}_b^{\text{sub}} \end{bmatrix}, \quad (4.9)$$

when  $\mathbf{p}_b^{\text{sub}}$  is evaluated for each sub-structure.

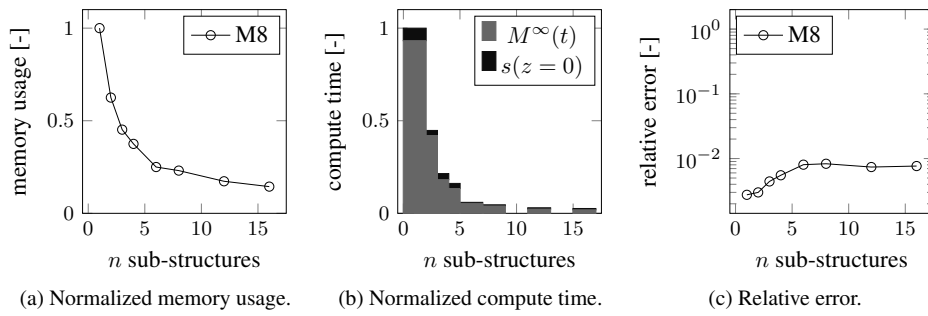
In figure 4.8, the interface of mesh M8 is shown, divided into two, four and eight structures.



**Figure 4.8:** M8 subdivided into different number of sub-structures.

The influence of the introduced artificial boundaries is depicted in figure 4.9. With an increasing number of sub-structures, memory usage and compute time are reduced significantly as shown in 4.9a and 4.9b. This behavior is expected, since reducing size and so the number of degrees of freedom at the interface reduces the complexity of the problem and so the time to compute it [29].

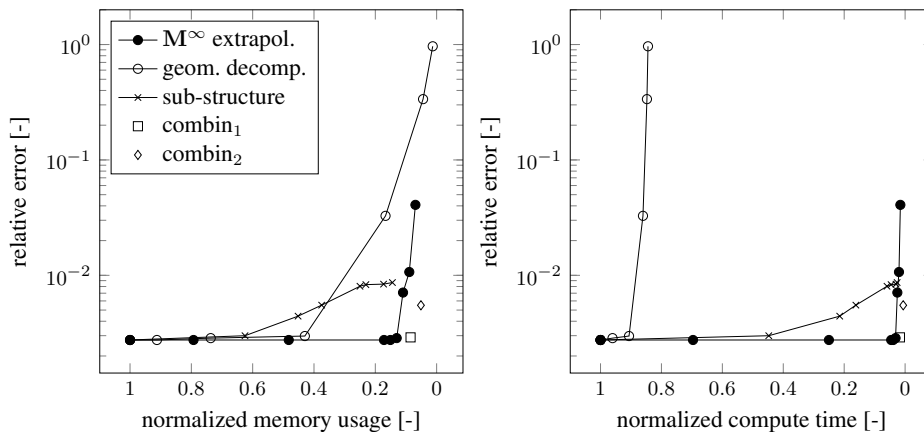
Figure 4.9c shows the relative error of the conducted settlement simulation with respect to number of sub-structures. Compared to the previous approach in section 4.2, the relative error is growing faster. The error increases clearly when 2 to 6 sub-structures are introduced, thereafter the relative error stays almost constant. If additional artificial boundaries are introduced, the far-field loses stiffness and the settlement is overestimated. From the engineering point of view, this approach leads to conservative results. The big advantage of this approach is that sub-structuring reduces memory consumption and also compute time massively (see fig. 4.9a and 4.9b). Such a massive reduction can neither be realized by extrapolation of  $\mathbf{M}_0^\infty$  nor by using a far-field decomposition technique.



**Figure 4.9:** Normalized memory usage of  $\mathbf{M}_0^\infty$  (a), normalized compute time (b) and relative error of final displacement (c) with respect to the number of sub-sections  $n$ .

## 5. Combination of model reduction techniques

Next, combinations of the presented model reduction techniques are discussed. Artificial boundaries are introduced as discussed in section 4.3 when the domain is sub-structured. This has a direct influence on the computational result, due to decoupling of neighbor nodes which normally have a strong interaction. Figure 5.1 shows a reduction of memory consumption and compute time to 62% and 45%, respectively when far-field is decomposed into three sub-structures. In this case, the relative error only increases slightly. The geometrical decoupling as discussed in section 4.2 is able to reduce memory consumption significantly, in figure 5.1 one can see that the memory usage can be reduced to 45% without increasing the relative error. Its influence to compute time is of minor importance (93%), since the problem size and so the complexity stays constant. The speed up arises from the reduction of amount of data. When extrapolation of  $\mathbf{M}^\infty$  at  $t_m$  is done, as discussed in section 4.1, memory usage and computational effort can be reduced down to less than 20% and 5%, respectively without increasing the relative error.



**Figure 5.1:** Relative error of  $\mathbf{M}^\infty$  extrapolation, geometrical decoupling and sub-structuring in relation to normalized memory usage (a) and normalized compute time (b).

The strong effect of far-field sub-structuring to reduce computational time significantly is shown. Hence, it is promising to combine the far-field sub-structuring with the geometrical decoupling. Geometrical decoupling has no significant influence on the relative error but on the memory consumption cf. section 4.2. With increasing number of sub-structures, the error does not rise significantly. When sub-structuring and geometrical far-field decompositions is combined and  $\varepsilon_z$  is chosen large enough, the pattern of sub-structured matrices correlates with matrices of full far-field nearly completely [8]. If  $\varepsilon_z$  is chosen to be small this is not the case. Table 4 shows the same configuration as table 3 but with taking geometric decoupling into account. Here,  $\varepsilon_z$  is chosen to be  $1.0 \times 10^{-7}$ . How big the further reduction is another  $\approx 10$  to  $\approx 40\%$ , depends strongly on the number of sub structures but without increasing the relative error.

If all three model reduction techniques are combined, the memory consumption and computational effort can be reduced even further. This can be shown by the next two examples impressively: if far-field is subdivided into two sub-structures,  $\varepsilon_z$  and  $t_m$  are chosen to be  $1.0 \cdot 10^{-6}$ , 60, respectively. The memory consumption and compute time can be reduced to  $\approx 2$  and  $\approx 9\%$ , respectively, compared to the reference solution of M8 when no model reduction is applied (cf. fig.5.1 combin<sub>1</sub>). The relative error is  $2.9 \times 10^{-3}$  compared to the analytical solution and is dominated by the sub-structuring method, the other two model reductions have no observable influence on the solution. Subdividing the far-field into four instead of two sub-structures and setting up the same parameters for  $\varepsilon_z$  and  $t_m$ , leads to a memory consumption and compute time of  $\approx 1$  and  $\approx 5\%$ , respectively, compared to the reference solution (cf. fig.5.1 combin<sub>2</sub>). The relative error increases to  $5.5 \times 10^{-3}$  which is almost twice as high. In this case, the relative error is also clearly dominated by the sub-structuring method. Similar reductions of memory usage and computational effort have been discussed in [30], in which the meshes as well as the reduction have been smaller. This leads to the awareness that the gain of model reduction techniques rises the bigger and the more complex the problems are.

## 6. Conclusion

A coupled approach of FEM and SBFEM is particularly well suited for applications, whenever complex structures have to be analysed and an infinite halfspace has to be taken into account, since the standard FEM can handle complex geometries with different types of material easily

**Table 4:** Size and allocation of sub structured  $\mathbf{M}^\infty$  matrices of M3 with  $\varepsilon_z = 1.0 \times 10^{-7}$ .

sub structures	matrix size	entries	allocation <sub>sub</sub>	allocation <sub>tot</sub>	ratio <sub>sub</sub> [%]	ratio <sub>total</sub> [%]
1	291 × 291	84681	49855	49855	58.87	58.87
2	171 × 171	29241	19601		67.03	
	171 × 171	29241	19605	39206	67.05	46.30
3	129 × 129	16641	11665		70.10	
	123 × 123	15129	11567		76.46	
	129 × 129	16641	11673	34905	70.15	41.22
4	105 × 105	11025	8105		73.51	
	102 × 102	10404	8102		77.87	
	99 × 99	9801	8015		81.78	
	105 × 105	11025	8125	32347	73.70	38.20

and the SBFEM fulfils the radiation condition to infinity exactly. If damping needs to be considered, it can be easily added to equation (2.1), the discussed techniques are still valid.

Since the far-field information can be computed before near-field and far-field are coupled to conduct the actual analysis, it is recommended to reduce the far-field model as discussed. When constant growth of  $\mathbf{M}^\infty$  is observed starting from an allowed instant of time  $t_m$ , it is possible to reduce the need of memory storage and computational effort significantly. Here, it has been shown that the memory usage and the computational effort can be reduced down to less than 20% and 5%, respectively. This can be achieved without increasing the relative error of the numerical simulation. The proposed algorithms' formulation implies that the more degrees of freedom are taken into account and the more time steps have to be computed, the larger the advantage of this approach is.

Currently, the data of  $\mathbf{M}^\infty$  is analysed to choose the time set  $t_m$ . It is revealed that picking an arbitrary matrix entry and check constant growth is not sufficient. A single entry can not state for the entire matrix and its time dependent behaviour, the entire matrix has to be taken into account. A determination of  $t_m$  a priori is challenging and will be of interest in future research. Being able to do so could allow to determine computational cost and memory consumption before running the far-field computation. Secondly, the checking of constant growth could be skipped and additional compute time could be saved to make the approach even more effective.

Using far-field decoupling or decomposition technique is relatively simple. The geometrical decoupling technique is simple to implement, since only one algorithm is needed which assures that only values bigger than  $\varepsilon_z$  are taken into account and the matrix size is adapted to new number of non-zero values. An efficient usage of sub-structuring technique is more complex since algorithms are needed to couple an arbitrary number of far-field sub-structures to the near-field. Furthermore, the far-field has to be decomposed by the user before computing the far-fields'  $\mathbf{M}^\infty$  matrices. Consequently, this leads to a higher effort in mesh generation. This higher effort is worthwhile since memory consumption and computation time can be reduced significantly if introducing an additional error is allowable.

It is to conclude that the studied model reduction is generally applicable to soil mechanics analysis, as long as linear material models are sufficient to describe the given problem. Hence, the different chosen sets of material parameters cover a broad range of geotechnically relevant materials. Further, it can be assumed that this approach is universally valid.

## References

- [1] Antes H, Spyarakos C. 1997 *Soil-Structure Interaction*, eds. Beskos DE, Anagnostopoulos SA. *Computer Analysis and Design of Earthquake Resistant Structures* Computational Mechanics Publications, Southampton.
- [2] Astley RJ. 2000. *Infinite elements for wave problems: a review of current formulations and a assessment of accuracy* 49(7):951–976 *International Journal for Numerical Methods in Engineering*.
- [3] Bazyar MH, Sing C. 2008. *A continued fraction based high order transmitting boundary for wave propagation in unbounded domains of arbitrary geometry* 74(2):209–237 *International Journal for Numerical Methods in Engineering*.
- [4] Beer G. 2001. *Programming the Boundary Element Method* Wiley & Sons, Chichester, U.K.
- [5] Birk C, Prempramote S, Song C. 2012. *An improved continued-fraction-based high-order transmitting boundary for time-domain analyses in unbounded domains* 89(3):269–298 *International Journal for Numerical Methods in Engineering*.
- [6] Brebbia CA, Telles JCF, Wrobel LC. *Boundary Element Techniques - Theory and Applications in Engineering* Springer, Berlin, Heidelberg.
- [7] Bettess P. 1992. *Infinite Elements* Penshaw Press, Sunderland, U.K.
- [8] Borsutzky R. 2008. *Seismic Risk Analysis of Buried Lifelines* 63 *Mechanik-Zentrum Technische Universität Braunschweig*.
- [9] Chen D, Birk C, Song C, Du C. 2013 *A high-order approach for modelling transient wave propagation problems using the scaled boundary finite element method* 97(13):937–959 *International Journal for Numerical Methods in Engineering*.
- [10] Chung J, Hulbert GM. 1993. *A Time Integration Algorithm for Structural Dynamics with Improved Numerical Dissipation: The Generalized- $\alpha$  Method* 60:371–375 *Journal of Applied Mechanics*.
- [11] Cuthill E, McKee J. 1969. *Reducing the Bandwidth of Sparse Symmetric Matrices* 157–172, *Proceedings of the 1969 24<sup>th</sup> National Conference*. ACM.
- [12] Engquist B, Majda A. 1977. *Absorbing boundary conditions for the numerical simulation of waves* 31(139):629–651 *Mathematics of Computation*.
- [13] Givoli D. 1991 *Non-reflecting Boundary Conditions* 94:1–29 *Journal of Computational Physics*.
- [14] Givoli D. 1992 *Numerical Methods for Problems in Infinite Domains* Elsevier Science Limited, Amsterdam.
- [15] Harr ME. 1966. *Foundations of Theoretical Soil Mechanics* McGraw-Hill Book Company.
- [16] Hilber H, Hughes T, Taylor R. 1977. *Improved numerical dissipation for time integration algorithms in structural dynamics* 5:283–292, *Earthquake Engineering & Structural Dynamics*.
- [17] Lehmann L. 2003. *Schnelles Verfahren zur Berechnung der Baugrund-Bauwerk-Interaktion im Zeitbereich* 22:6–9, D-A-CH Mitteilungsblatt.
- [18] Lehmann L, Antes H, Schanz M. 2004. *Transient analysis of soil-structure interaction problems: An effective FEM/SBFEM approach* 99–116, *Advanced Numerical Analyses of Solids and Structures, and Beyond*, Graz, Institute for Structural Analysis, Verlag der Technischen Universität Graz.
- [19] Lehmann L. 2005. *An effective finite element approach for soil-structure analysis in the time-domain* 21(4):37–50, *Structural Engineering and Mechanics*.
- [20] Lehmann L. 2006. *Wave Propagation in Infinite Domains*, Springer, Berlin / Heidelberg.
- [21] Liao ZP, Wong HL. 1984. *A transmitting boundary for the numerical simulation of elastic wave propagation* 3(4):174–183 *Soil Dynamics and Earthquake Engineering*.
- [22] Lysmer J, Kuhlmeyer RL. 1969. *Finite dynamic model for infinite media* 95:859–875 *Journal of Engineering Mechanics*.
- [23] Meskouris K, Hinzen KG, Butenweg C, Mistler M. 2007. *Bauwerke und Erdbeben - Grundlagen - Anwendung - Beispiele*, Vieweg+Teubner Verlag, Wiesbaden.
- [24] Newmark N. 1959. *A method of computation for structural dynamics* 85:67–94, *Journal of Engineering Mechanics Division*.

- [25] Petersen C. 2000. *Dynamik der Baukonstruktionen* Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig/Wiesbaden.
- [26] Poulos HG, Davis EH. 1974. *Elastic Solutions for Soil and Rock Mechanics* John Wiley & Sons, INC.
- [27] Radmanović B, Katz C. 2010. *A High Performance Scaled Boundary Finite Element Method* **10** IOP Conf. Series: Material Science and Engineering.
- [28] Schauer M, Lehmann L. 2009. *Large Scale Simulation with Scaled Boundary Finite Element Method* **9**(4) 103–106, roceedings in Applied Mathematics and Mechanics
- [29] Schauer M, Roman JE, Quintana-Ortí ES, Langer S. 2012. *Parallel Computation of 3-D Soil-Structure Interaction in Time Domain with a Coupled FEM/SBFEM Approach* **52**:446–467, Journal of Scientific Computing.
- [30] Schauer MM. 2015. *Ein effizienter gekoppelter FEM-SBFEM Ansatz zur Analyse von Boden-Bauwerk-Interaktionen im Zeitbereich* Dissertation, Technische Universität Braunschweig.
- [31] Wolf J, Song C. 1996. *Finite-Element Modelling of Unbounded Media* John Wiley & Sons, Chichester.
- [32] Wolf J. 2003. *The Scaled Boundary Finite Element Method* John Wiley & Sons, Chichester.
- [33] Yan J, Zhang C, Jin F. 2004. *A coupling procedure of FE and SBFEM for soil-structure interaction in the time domain* **59**1453–1471, Int. J. Numer. Meth. Engng.
- [34] Zhang X, Wegner JL, Haddow JB. 1999. *Three-Dimensional Dynamic Soil-Structure Interaction Analysis in the Time Domain* **28**(10):1501–1524, Earthquake Engineering and Structural Dynamics.