Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXXVI, págs. 1959-1970 (artículo completo)
José G. Etse, Bibiana M. Luccioni, Martín A. Pucheta, Mario A. Storti (Eds.)
San Miguel de Tucumán, 6-9 Noviembre 2018

PROJECTION METHODS FOR SOIL STRUCTURE INTERACTION NUMERICAL SIMULATIONS WITH NON-MATCHING MESHES.

Gustavo Ríos Rodriguez[a] **and Marco Schauer**[b]

[a]*Centro de Investigación de Métodos Computacionales, CIMEC (UNL - CONICET), Predio CONICET-Santa Fe, Colectora RN 168, El Pozo, 3000 Santa Fe, Argentina*

[b]*Technische Universität Braunschweig, Institut für Statik, Beethovenstraße 51, 38106 Braunschweig, Germany*

Keywords: Soil Structure Interaction, Projection Methods, Non-Matching Meshes, FEM-SBFEM Coupling.

Abstract. Nowadays, it is of great interest to perform numerical simulations of soil structure interaction (SSI) since they are applied to a wide range of engineering problems. These include the construction of reliable earthquake resistant structures in seismic active areas, or to increase the comfort of buildings by decoupling them from surrounding emissions like vibrations induced by traffic. To analyse SSI problems taking unbounded soil domains into account a numerical implementation of a coupled finite element method (FEM) and scaled boundary finite element method (SBFEM) approach is used in this work. This approach fulfills the Sommerfeld's radiation condition. The FEM is used to discretise the so called near-field, i.e. the structure and its surrounding soil, while the infinite half-space or so called far-field is realised by the SBFEM. Both methods are coupled at a common interface, where specific information like nodal velocities and forces are required to be exchanged. Since computation of far-field solution demands more effort than for the near-field, it is desirable to use fewer elements in the far-field discretization than in the near-field. Therefore, a projection algorithm is needed to exchange information (i.e. nodal forces and velocities) at the interface between the non-matching meshes. A projection method borrowed from the fluid structure interaction framework is used in this work, namely the Nearest Neighbour projected method. It is wrapped in a library which is called during the coupled FEM-SBFEM solution process. A settlement SSI problem is considered to validate the implementation, as well as to evaluate the savings in memory and computation time. Matching mesh results are taken as reference.

1 INTRODUCTION

A numerical solution strategy to solve soil-structure interaction (SSI) problems through unbounded domains is presented in this work. SSI appears, for example, due to wave-propagation induced by vibrations or impulse loads, usually caused by traffic or earthquakes. To analyse this kind of problems both the structure and the surrounding soil must be considered (Meskouris et al., 2007). In most practical situations there is no analytical solution available. Therefore, approximated solutions are computed, usually by the finite element method (FEM). On the other hand, the solution computation for the infinite soil surrounding the structure has to be treated with numerical methods which satisfy the Sommerfeld's radiation condition, like the boundary element method (BEM) or the scaled boundary finite element method (SBFEM) (Beer, 2001). Like FEM, the SBFEM method does not require a fundamental solution and the coefficient matrices which appear in the mathematical formulation are symmetric and can also be added to the FEM matrices without changing their size. Besides, in linear analysis the SBFEM matrices do not change even when the setup of the near-field is modified, therefore they are computed once. In non-linear analysis this does not hold and the coefficient matrices have to be computed as the simulation evolves. Due to this combination of advantages, the SBFEM method is chosen to solve the far-field problem in this work. Then the solution strategy here presented couples both the FEM and the SBFEM (Wolf, 2003) methods in a monolithic fashion (Hübner et al., 2004; Walhorn et al., 2005). The numerical solution in the near-field (i.e. the structure, its foundation and part of the surrounding soil) is computed with the FEM and the SBFEM is used to approximate the solution in the far-field (i.e. the unbounded soil). In the rest of the manuscript, Ω will denote the near-field and Γ will be the interface between the near- and the far-field.

The discretizations of the FEM and SBFEM subproblems are usually constrained to have matching meshes at the interface Γ between subdomains. This is acceptable as long as the geometrical complexity of the foundation and the surrounding soil are simple since the number of degrees of freedom (DoF) is small and the problem can be solved in a standard desktop PC. However, as the size and complexity of the discrete problem at the interface grow, and if accurate results are needed, since both the memory size and time complexity requirements of the SBFEM are square and cubic on the number of DoF, it becomes almost impossible to analyze complex problems, even in a computer cluster. But if the matching discretizations constraint at the interface Γ is removed, then the near-field can be modeled using a fine mesh and the far-field can be modeled with a coarse mesh to keep the computation cost reduced. Taking this into account, the main contribution of the present work is the introduction of computational techniques which enable to solve SSI problems with a coupled FEM-SBFEM parallel code, using different discretizations for the subdomains. The meshes will be non-coincident at the near / far-field interfaces, so there appears the need to use algorithms to transfer the solution from one subdomain boundary to its neighbor, as well as to track the displacement of their boundaries. The algorithm is borrowed from the FSI framework, and it is based on a fast Nearest Neighbor projected (NNproj) search.

The manuscript is organized as follows: in section 2, the governing equations of the SSI problem are described, as well as the FEM-SBFEM monolithic solution scheme that is solved. In section 3, the NNproj method to perform the solution projection is described. Then, in section 4 it is mentioned how the projection method is introduced in the solution procedure. Later in section 5, the coupled FEM-SBFEM method together with the projection method are applied to solve a simple SSI benchmark problem where the FEM and the SBFEM discretizations do not match at Γ, namely an hemispheric settlement problem. This is aimed to evaluate the

advantages of using the non-conforming mesh projection method in terms of accuracy and computational effort. Results are compared to those obtained with the same coupled FEM-SBFEM methodology but with coincident meshes. Finally, conclusions are made on the results and future research is outlined.

2 COUPLED FEM-SBFEM APPROACH

At an arbitrary time step the equation of motion in a displacement-based FEM can be written as

$$\mathbf{M}\frac{d^2\mathbf{u}}{dt^2} + \mathbf{C}\frac{d\mathbf{u}}{dt} + \mathbf{K}\mathbf{u} = \mathbf{p}, \tag{1}$$

where the matrices $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}$ denotes mass, damping and stiffness, respectively. Here, vector $\mathbf{u}$ and its derivatives in time $\frac{d\mathbf{u}}{dt} = \dot{\mathbf{u}}$ and $\frac{d^2\mathbf{u}}{dt^2} = \ddot{\mathbf{u}}$ represent the nodal displacement, velocity and acceleration. The vector $\mathbf{p}$ represents the applied nodal forces. Consider the time period T divided into n constant size time steps $\Delta t = \frac{T}{n}$; then the application of the implicit Newmark time integration scheme yields (Newmark, 1959)

$$\mathbf{M}\ddot{\mathbf{u}}_{t_{n+1}} + \mathbf{C}\dot{\mathbf{u}}_{t_{n+1}} + \mathbf{K}\mathbf{u}_{t_{n+1}} = \mathbf{p}_{t_{n+1}}\,. \tag{2}$$

To couple the FEM with the SBFEM, the entries of the matrices in eq. (1) have to be split into the near-field and far-field, which results in the following equation

$$\begin{bmatrix} \mathbf{M}_{\Omega\Omega} & \mathbf{M}_{\Omega\Gamma} \\ \mathbf{M}_{\Gamma\Omega} & \mathbf{M}_{\Gamma\Gamma} \end{bmatrix} \ddot{\mathbf{u}} + \begin{bmatrix} \mathbf{C}_{\Omega\Omega} & \mathbf{C}_{\Omega\Gamma} \\ \mathbf{C}_{\Gamma\Omega} & \mathbf{C}_{\Gamma\Gamma} \end{bmatrix} \dot{\mathbf{u}} + \begin{bmatrix} \mathbf{K}_{\Omega\Omega} & \mathbf{K}_{\Omega\Gamma} \\ \mathbf{K}_{\Gamma\Omega} & \mathbf{K}_{\Gamma\Gamma} \end{bmatrix} \mathbf{u} = \begin{bmatrix} \mathbf{p}_{\Omega\Omega} \\ \mathbf{p}_{\Gamma\Gamma} \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ \mathbf{p}_b \end{bmatrix} \tag{3}$$

Blocks with subscript "$\Omega\Omega$" contain nodes only at the near-field while blocks with subscript "$\Gamma\Gamma$" comprise nodes only at the far-field. The coupling of near- and far-field nodes is represented in those blocks marked with the subscripts "$\Omega\Gamma$" and "$\Gamma\Omega$". Vector $\mathbf{p}_b$ acts only at the boundary Γ and denotes the response of the infinite half space, so that the far-field influence on the near-field can be applied to the FEM subdomain as a load.

The second part of this substructuring approach is the far-field represented by the SBFEM. The forces $\mathbf{p}_b$ at the interface are given by the convolution integral

$$\mathbf{p}_b(t) = \int_0^t \mathbf{M}^{\infty}(t-\tau)\ddot{\mathbf{u}}(\tau)d\tau \tag{4}$$

where $\mathbf{M}^{\infty}(t)$ is the acceleration unit-impulse response matrix, also known as the influence matrix. To solve the convolution integral equation in time domain, a piecewise constant influence matrix is assumed. Hence eq. (4) can be rewritten in discrete form as

$$\mathbf{p}_b(t_n) = \sum_{j=1}^{n} \mathbf{M}^{\infty}_{n-j} \int_{(j-1)\Delta t}^{j\Delta t} \ddot{\mathbf{u}}(\tau)d(\tau). \tag{5}$$

This equation is then transformed using Newmark time integration scheme into

$$\mathbf{p}_b(t_n) = \gamma\Delta t\mathbf{M}^{\infty}_0\ddot{\mathbf{u}}_n + \sum_{j=1}^{n-1} \mathbf{M}^{\infty}_{n-j}\,(\dot{\mathbf{u}}_j - \dot{\mathbf{u}}_{j-1}). \tag{6}$$

where γ is a parameter of the Newmark scheme. The coupling of FEM and SBFEM is done at the interface of both subdomains with the SBFEM part given by eq. (6) being added to the sorted FEM part in eq. (3) (Wolf and Song, 1996):

$$\begin{bmatrix} \mathbf{M}_{\Omega\Omega} & \mathbf{M}_{\Omega\Gamma} \\ \mathbf{M}_{\Gamma\Omega} & \mathbf{M}_{\Gamma\Gamma} + \gamma\Delta t \mathbf{M}_0^{\infty} \end{bmatrix} \ddot{\mathbf{u}} + \begin{bmatrix} \mathbf{C}_{\Omega\Omega} & \mathbf{C}_{\Omega\Gamma} \\ \mathbf{C}_{\Gamma\Omega} & \mathbf{C}_{\Gamma\Gamma} \end{bmatrix} \dot{\mathbf{u}} + \begin{bmatrix} \mathbf{K}_{\Omega\Omega} & \mathbf{K}_{\Omega\Gamma} \\ \mathbf{K}_{\Gamma\Omega} & \mathbf{K}_{\Gamma\Gamma} \end{bmatrix} \mathbf{u} = \begin{bmatrix} \mathbf{p}_{\Omega\Omega} \\ \mathbf{p}_{\Gamma\Gamma} - \sum_{j=1}^{n-1} \mathbf{M}_{n-j}^{\infty} \left(\dot{\mathbf{u}}_j - \dot{\mathbf{u}}_{j-1} \right) \end{bmatrix}. \tag{7}$$

In this manner, the discretization is completely described. The formulation of the SBFEM in time domain as well as the computation of the influence matrices $\mathbf{M}^{\infty}(t)$ needed in equations (4) to (7) are fully described in Schauer et al. (2012).

3 PROJECTION METHOD

Since the objective of this work is to use independent discretizations of the far-field and near-field subdomains, both meshes will be non-coincident at the interface. Therefore, the forces computed with the SBFEM method at the boundary of the far-field (Γ_{SBFEM}) have to be projected to the boundary of the near-field (Γ_{FEM}). Also, the velocities have to be projected in the other way. Here, the so called Nearest Neighbor projected (NNproj) method is used with this end and will be described next. Subscripts 1 and 2 will be used to make reference either to the near-field or the far-field.

3.1 Nearest Neighbor projected method

Let $\mathbf{u}$ be a continuous scalar or vector field on domain Ω_1, being Γ_1 its approximation to boundary Γ, $\hat{\mathbf{u}}_1$ the discrete approximation of $\mathbf{u}$ on Ω_1, and $\hat{\mathbf{u}}_{\Gamma_1}^{(i)}$ its value at vertex-i on Γ_1 (i.e., the data of the projection problem). Let q be a vertex of the mesh on Γ_2, with coordinates $\mathbf{x}_q$ and q_p its normal projection on Γ_1, with coordinates $\mathbf{x}_{q_p}$. The discrete approximation of $\mathbf{u}$ at point q can be computed by interpolation, considering the values of the usual FEM shape functions $N_1^{(j)}(\mathbf{x}_{q_p})$ and the nodal values $\hat{\mathbf{u}}_{\Gamma_1}^{(j)}$ corresponding to the vertices of the element in Γ_1 which hosts q_p (e.g. element e_6 in fig. 1).

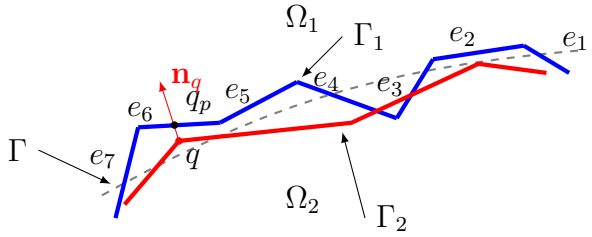


Figure 1: Nearest Neighbor projected scheme in 2-D.

To apply this projection method, it is required to compute the element on Γ_1 which hosts the normal projection of point q, hereafter called "host element". A procedure for its solution is described next.

3.1.1 Host element problem

The host element problem can be efficiently solved in two steps, as it is proposed in (de Boer, 2008) or (Löhner, 2008). The first step consists in building a list l_q of m-nearest neighbors and *probable* host elements to query point q, sorted them by increasing distance to q. This is accomplished in the present work in approximate fashion, i.e. admitting a small tolerance in the search and thus significantly reducing the time complexity if compared to an exact search, using the ANN (Approximate Nearest Neighbor) library (Arya and Mount, 2010). The second step finds the host element iterating on l_q. In 2-D, for each query point, each iteration consists in solving the intersection problem between the straight line that passes through both vertices of the element e_s, for $s = 1, 2, \ldots, m$, and the boundary normal at the query point ($\mathbf{n}_q$) to find its normal projection q_p. If intersection exists, then it is determined whether it falls or not in e_s applying the following condition

$$\min(N_1^{(i)}(\mathbf{x}_{q_p}), 1 - N_1^{(i)}(\mathbf{x}_{q_p})) \geq 0, \quad \forall v_i \in \mathcal{V} \tag{8}$$

where $N_1^{(i)}(\mathbf{x}_{q_p})$ is the shape function at vertex v_i of e_s, evaluated at q_p, and $\mathcal{V}$ is the set of e_s vertices. If condition given by eq. (8) is satisfied, then e_s hosts q_p and $\hat{\mathbf{u}}_{\Gamma_2}^{(q)}$ is computed as follows

$$\hat{\mathbf{u}}_{\Gamma_2}^{(q)} = \sum_{v_i \in \mathcal{V}} N_1^{(i)}(\mathbf{x}_{q_p}) \hat{\mathbf{u}}_{\Gamma_1}^{(i)} \,. \tag{9}$$

In 3-D, the plane π that passes through all the vertices of e_s is considered. If intersection exists between the normal $\mathbf{n}_q$ and π, then both q_p and e_s are projected onto the *most parallel* coordinate plane to π. Then, eq. (8) is applied like in 2-D, as it is suggested in O'Rourke (1998). For triangles, barycentric coordinates are used to ascertain that the point is inside the element with condition given by eq. (9). If the point is outside of triangle e_s, the algorithm has to iterate on the next element in l_q. On the other hand, inverse bilinear mapping method presented in Buss (2003) is used to compute $N_1^{(i)}(\mathbf{x}_{q_p'})$ together with eq. (9) for quadrangles.

4 NEAR-FIELD FAR-FIELD INTERFACE IMPLEMENTATION

The coupling of the near-field and far-field is done at the interface Γ. The nodal information at Γ has to be extracted from the near-field and the far-field (see eq. (7)). Here ndof is the number of degrees of freedom, which is the number of nodes (nnod) at the subdomain times the number of unknowns at the nodes. Therefore, ndof defines the size of the subdomains and so the size of their vectors and matrices. In the following ndof_{FEM}, $\text{ndof}_{\text{SBFEM}}$ and ndof_{Γ} are used to describe the size of near-field, far-field and interface. The near-field velocity

$$\dot{\mathbf{u}}_{\Gamma} = \dot{\mathbf{u}}_{\Gamma,\text{FEM}} \tag{10}$$

is needed to compute the response of the infinite half space

$$\mathbf{p}_{b,\Gamma} = \mathbf{p}_{b,\text{SBFEM}} \tag{11}$$

at the interface. The basic algorithms to conduct the time domain analysis using matching or non-matching meshes are introduced next.

4.1 Matching Interface

In this case, the main procedure is shown in alg.(1) where time step size Δt and total simulation time T are defined as input. At the beginning the FEM-SBFEM interface is initialized (alg. 2). Afterwards the computation of the coupled problem is performed within a time loop, $0 \leq t \leq T$, where the response of the far-field has to be computed (alg. 3) before the time integration scheme is applied to calculate the state variables of the next time step. Then, the velocities have to be updated at the interface (alg. 4) and the loop can proceed with the next time step.

Algorithm 1 (Time domain analysis)**.**

Input: time step size Δt, maximum time T
Output: $\mathbf{x}(t)$, $\sigma(t)$, etc.
[1.1] initialize interface (Algorithm 2)
[1.2] **while** $t < T$; $t+ = \Delta t$
[1.3] compute far-field response $\mathbf{p}_b(t)$ (Algorithm 3)
[1.4] solve $\mathbf{M\ddot{u}} + \mathbf{C\dot{u}} + \mathbf{Ku} = \mathbf{p} + \mathbf{p}_b$
[1.5] update state $\dot{\mathbf{u}}(t)$ at interface (Algorithm 4)

When matching meshes are used, the far-field nodes at the interface are coincident with those of the near-field, therefore $\text{nnod}_{\Gamma,\text{SBFEM}} = \text{nnod}_{\Gamma,\text{FEM}}$. As a consequence, the correspondence at the interface of the FEM nodes with those of the SBFEM has to be determined only once, when the interface is initialized, and it is shown in alg.(2) which identifies the matching nodes of FEM and SBFEM.

Algorithm 2 (Interface_init)**.**

Input: FEM, SBFEM nodes and connectivities
Output: mapping
[2.1] **Loop** $\text{node}^{(k)}_{\Gamma,\text{SBFEM}}$ $k = 0, ..., \text{nnod}_{\Gamma,\text{SBFEM}}$
[2.2] **Loop** $\text{node}^{(\ell)}_{\text{FEM}}$ $\ell = 0, ..., \text{nnod}_{\text{FEM}}$
[2.3] **if** ($\text{node}^{(k)}_{\Gamma,\text{SBFEM}}$==$\text{node}^{(\ell)}_{\text{FEM}}$)
[2.4] **then** mapping[k] = $\text{node}^{(\ell)}_{\text{FEM}}$.id

---

The far-field response $\mathbf{p}_{b,\Gamma}(t)$ given by eq. (6) is calculated before the time integration scheme is applied to the equation of motion. In order to compute the response the interface velocity $\dot{\mathbf{u}}_\Gamma(t)$ and the interface acceleration $\ddot{\mathbf{u}}_\Gamma(t)$ are required. Both can be obtained from the near-field state by applying the mapping information of alg.(2). Since $\mathbf{p}_{b,\Gamma}(t)$ has the same size as the far-field, the information has to be mapped to the near-field, by again applying the mapping information ($\mathbf{p}_{b,\Gamma}(t) \rightarrow \mathbf{p}_{b,\text{FEM}}(t)$).

After the time step integration of the equation of motion is done, the interface velocity has to be updated and stored for each time step to provide this information to solve the convolution integral (see eq. (6)) at any further time step. This is shown in alg.(4), where the interface velocity $\dot{\mathbf{u}}_\Gamma(t)$ is extracted from the near-field velocity $\dot{\mathbf{u}}_{\text{FEM}}(t)$ and stored into a vector of vectors $\dot{\mathbf{u}}_n$, which has size $\text{nnod}_{\text{SBFEM}} \times \frac{T}{\Delta t}$.

Algorithm 3 (Interface_response)**.**

Input: $\dot{\mathbf{u}}_\Gamma$,$\ddot{\mathbf{u}}_{\text{FEM}}(t)$, $\mathbf{M}_n^\infty$, mapping
Output: $\mathbf{p}_{b,\Gamma}(t_n)$
[3.1] extract $\ddot{\mathbf{u}}_\Gamma(t)$ out of $\ddot{\mathbf{u}}_{\text{FEM}}(t)$ with mapping
[3.2] compute $\mathbf{p}_{b,\Gamma}$ (see equation (6))

Algorithm 4 (Interface_update)**.**

Input: $\dot{\mathbf{u}}_{\text{FEM}}(t)$, mapping
Output: $\dot{\mathbf{u}}_\Gamma(t)$
[4.1] extract $\dot{\mathbf{u}}_\Gamma(t)$ out of $\dot{\mathbf{u}}_{\text{FEM}}(t)$ with mapping
[4.2] store $\dot{\mathbf{u}}_\Gamma(t)$ at interface

4.2 Non-matching interface

The previously described methodology is no longer applicable in this situation. To avoid a redevelopment of the existing code the projection library is introduced into the existing code interface adding a few lines to the original implementation. Since now the far-field nodes are not in coincidence with those of the near-field, the interface nodes have to be determined and used for the mapping of the FEM nodes. This yields to different numbers of degrees of freedom for each domain at the common interface. The sizes of the near-field and far-field are given by $\text{nnod}_{\Gamma,\text{FEM}} \neq \text{nnod}_{\text{FEM}}$, $\text{nnod}_{\Gamma,\text{SBFEM}} = \text{nnod}_{\text{SBFEM}}$, respectively, but $\text{nnod}_{\Gamma,\text{FEM}} \neq \text{nnod}_{\Gamma,\text{SBFEM}}$ since Γ_{FEM} and Γ_{SBFEM} do not coincide. This circumstance demands to project the state from one domain boundary to its neighbor and vice versa. In alg.(2) a new object, which is responsible for the state projection is introduced, as well as the two domain boundaries Γ_{FEM} and Γ_{SBFEM}.

Algorithm 2 (Interface_init_mod)**.**

Input: FEM, SBFEM nodes and connectivities
Output: mapping
[2.1] new stateProjection
[2.2] Γ_{FEM} = stateProjection.domain1
[2.3] Γ_{SBFEM} = stateProjection.domain2
[2.4] **Loop** $\text{node}^{(k)}_{\Gamma,\text{FEM}}$ $k = 0, ..., \text{nnod}_{\Gamma,\text{FEM}}$
[2.5] **Loop** $\text{node}^{(\ell)}_{\text{FEM}}$ $\ell = 0, ..., \text{nnod}_{\text{FEM}}$
[2.6] **if** ($\text{node}^{(k)}_{\Gamma,\text{FEM}}$==$\text{node}^{(\ell)}_{\text{FEM}}$)
[2.7] **then** mapping[k] = $\text{node}^{(\ell)}_{\text{FEM}}$.id

Algorithms 3mod and 4mod stay nearly untouched, except for the state projection calculations. Since the nodal velocities $\dot{\mathbf{u}}$, accelerations $\ddot{\mathbf{u}}$ and forces $\mathbf{p}_b$ can not be mapped directly from one domain to the other, a state projection needs to be carried out, applying the NNproj presented in sections 3.1.

Algorithm 3 (Interface_response_mod)**.**

Input: $\dot{\mathbf{u}}_{\Gamma,\text{SBFEM}}$, $\ddot{\mathbf{u}}_{\text{FEM}}(t)$, $\mathbf{M}_n^{\infty}$, mapping
Output: $\mathbf{p}_{b,\text{FEM}}(t_n)$
[3.1] extract $\ddot{\mathbf{u}}_{\Gamma,\text{FEM}}(t)$ out of $\ddot{\mathbf{u}}_{\text{FEM}}(t)$ with mapping
[3.2] project $\ddot{\mathbf{u}}_{\Gamma,\text{FEM}}(t)$ to $\ddot{\mathbf{u}}_{\Gamma,\text{SBFEM}}(t)$ ($\Gamma_{\text{FEM}} \rightarrow \Gamma_{\text{SBFEM}}$)
[3.3] compute $\mathbf{p}_{b,\text{SBFEM}}$ (see equation (6))
[3.4] project $\mathbf{p}_{b,\text{SBFEM}}$ to $\mathbf{p}_{b,\text{FEM}}$ ($\Gamma_{\text{SBFEM}} \rightarrow \Gamma_{\text{FEM}}$)

Algorithm 4 (Interface_update_mod)**.**

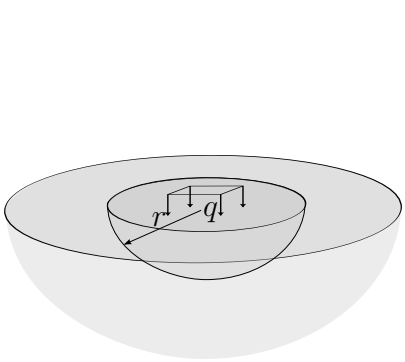
Input: $\dot{\mathbf{u}}_{\text{FEM}}(t)$, mapping
Output: $\dot{\mathbf{u}}_{\Gamma,\text{SBFEM}}(t)$
[4.1] extract $\dot{\mathbf{u}}_{\Gamma,\text{FEM}}(t)$ out of $\dot{\mathbf{u}}_{\text{FEM}}(t)$ with mapping
[4.2] project $\dot{\mathbf{u}}_{\Gamma,\text{FEM}}(t)$ to $\dot{\mathbf{u}}_{\Gamma,\text{SBFEM}}(t)$ ($\Gamma_{\text{FEM}} \rightarrow \Gamma_{\text{SBFEM}}$)
[4.3] store $\dot{\mathbf{u}}_{\Gamma,\text{SBFEM}}(t)$ at interface

5 APPLICATION

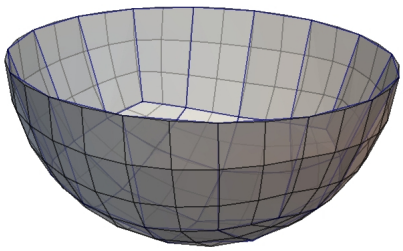
To show the benefits of employing the state projection method within the FEM-SBFEM coupling, the proposed approach is applied to a simple settlement problem. The savings in memory and computation time with respect to the matching meshes coupled solution are evaluated, as well as the accuracy of the proposed methodology. The problem also introduces difficulties such as a curved interface, with parts of the near-field intersecting those of the far-field due to different discretizations, as well as a non-plane wave propagating through the interface. The computations were carried out on a compute server with two Intel(R) Xeon(R) CPU E5-2699A v4 @ 2.40GHz and 256 GB RAM running on Linux CentOS with kernel 3.10.0-514.21.2.el7.x86_64.

5.1 Setup of the problem

The settlement problem has been chosen since it has a steady state solution. The results are compared with those of the original (i.e. matching meshes) coupling implementation.



(a) Problem definition.



(b) Interface discretization for non-coincident meshes (SBFEM - Blue, FEM - Black).

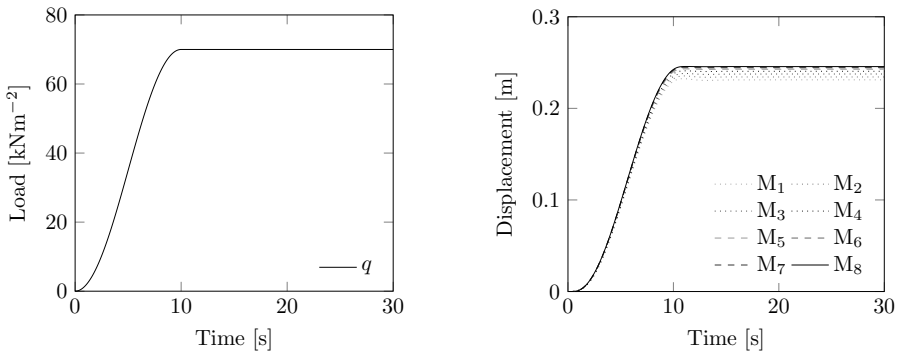
Figure 2: Settlement problem.

A load $q = 70\,[\mathrm{kNm}^{-2}]$ is applied on a square region of $152.4 \times 152.4\,[\mathrm{m}^2]$. The radius from the center to the nodes at the interface is $190.2\,[\mathrm{m}]$ (cf. figure 2a). The infinite half-space material is assumed homogeneous linear elastic with the following properties: Young's modulus $E = 37150.0\,[\mathrm{kNm}^{-2}]$, Poisson's ratio $\nu = 0.48$ and density $\rho = 1800.0\,[\mathrm{kgm}^{-3}]$. The chosen material leads to p-wave and s-wave speed of $c_p = 425.8\,[\mathrm{ms}^{-2}]$ and $c_s = 83.5\,[\mathrm{ms}^{-2}]$, respectively. The time step size $\Delta t = 0.0125\,[\mathrm{s}]$ is chosen so that neither $\Delta t_{\mathrm{krit}}^{\mathrm{FEM}} = \frac{\ell_{\mathrm{min}}}{c_p}$ nor $\Delta t_{\mathrm{krit}}^{\mathrm{SBFEM}} = \frac{r}{30 c_p}$ are exceeded (Borsutzky, 2008). The simulation is carried out for a time period of $30\,[\mathrm{s}]$. During the first $10\,[\mathrm{s}]$ the applied load is increased from $q = 0\ [\mathrm{kNm}^{-2}]$ to $q = 70\,[\mathrm{kNm}^{-2}]$ and then remains constant (cf. figure 3a). The far-field's influence is taken into account by 500 unit acceleration impulse response matrices ($6.25\,[\mathrm{s}]$), which are computed in a preprocess step using the same Δt than in the coupled simulation. All other matrices needed to conduct the full simulation are extrapolated during the calculations (Lehmann, 2005).

Matching meshes In this case, the FE and SBFE nodes are coincident at the interface. To analyze the convergence of the implemented approach, the given setup is refined several times from coarse mesh M_1 to fine mesh M_8. In table 1 the number of degrees of freedom of the near-field $\mathrm{DoF}_{\mathrm{FEM}}$ and the degrees of freedom of the coupled far-field $\mathrm{DoF}_{\mathrm{SBFEM}}$ are summarized.

Table 1: Mesh discretization with different number of DoF. The meshes correspond to Schauer et al. (2012).

Mesh	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
$\mathrm{DoF}_{\mathrm{FEM}}$	396	738	1314	3096	6030	9033	14655	19818
$\mathrm{DoF}_{\mathrm{SBFEM}}$	123	219	291	480	843	1083	1515	1827



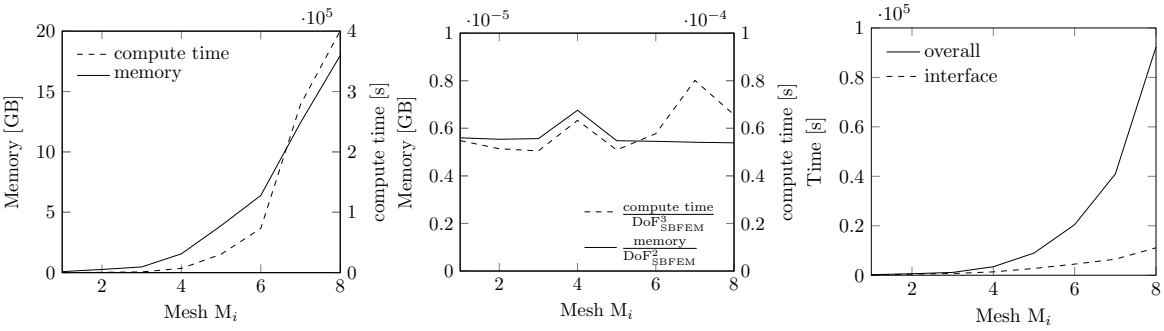
(a) Applied load over time. (b) Settlement over time.

Figure 3: Applied load (q) and settlement (s) over time using different discretizations M_i.

The displacement in time at the center of the applied load is shown in fig.(3b). It can be seen that with increasing number of degrees of freedom the simulation converges to the semi-analytic solution

$$s = \frac{4qb}{\pi E}\left(1-\nu^2\right)\ln\frac{\sqrt{2}+1}{\sqrt{2}-1} = 0.248\,[\mathrm{m}] \tag{12}$$

given by Harr (1966).



(a) (b) (c)

Figure 4: (a) Compute time and memory consumption of far-field computation ($\mathbf{M}_n^\infty$), (b) Time and memory consumption with respect to the $\text{DoF}_{\text{SBFEM}}$ and (c) compute time of coupled FEM-SBFEM computation for different discretizations $\mathbf{M}_i$.

As explained in the introduction the computational effort of the SBFEM is very high compared to that of the FEM. Figure 4a shows the memory consumption to store the $\mathbf{M}^\infty$ matrices as well as the time required to compute them. The ratios of memory consumption and computing time to the number $\text{DoF}^2_{\text{SBFEM}}$ is shown in fig.(4b). It can be seen that the computation of the infinite far-field matrices has memory storage and calculation complexities $\mathcal{O}(\text{DoF}^2_{\text{SBFEM}})$ and $\mathcal{O}(\text{DoF}^3_{\text{SBFEM}})$, respectively. The compute time of the coupled FEM-SBFEM computation to solve the settlement problem is shown in fig.(4c). This does not include the time needed to compute the unit impulse response matrices shown in fig.(4a). The first graph 'overall' includes the total compute time as well as input and output. The second graph 'interface' represents the time needed to solve eq. (6).

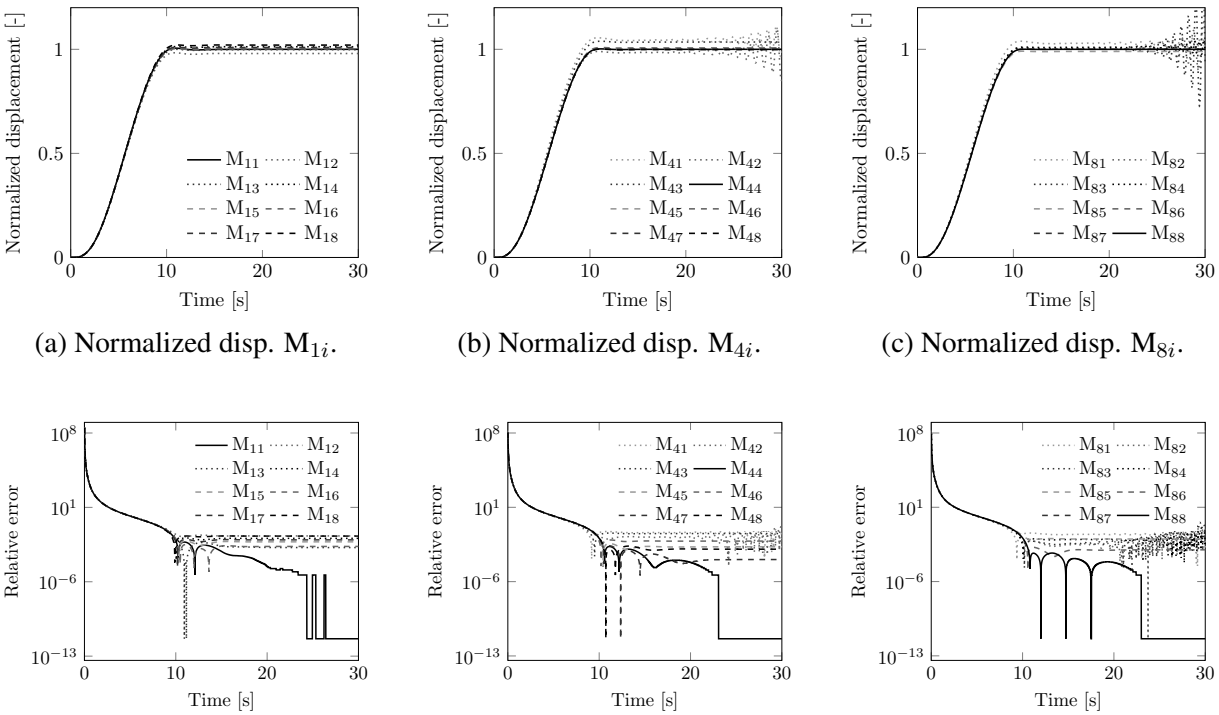
Non-matching meshes As already shown for the matching meshes, the computation of the $\mathbf{M}^\infty$ is $\mathcal{O}(\text{DoF}^3_{\text{SBFEM}})$. As a consequence, it is important to keep the number of far-field nodes as low as possible. Therefore, SBFEM discretizations coarser than those of FEM are used next (see fig.(2b)).

Table 2: Non-matching mesh discretization with different number of DoF in near-field and far-field.

Mesh	DoF_{FEM}	$\text{DoF}_{\text{SBFEM}}$							
$\mathbf{M}_{1i}$	396	**123**	219	291	480	843	1083	1515	1827
$\mathbf{M}_{2i}$	738	123	**219**	291	480	843	1083	1515	1827
$\mathbf{M}_{3i}$	1314	123	219	**291**	480	843	1083	1515	1827
$\mathbf{M}_{4i}$	3096	123	219	291	**480**	843	1083	1515	1827
$\mathbf{M}_{5i}$	6030	123	219	291	480	**843**	1083	1515	1827
$\mathbf{M}_{6i}$	9033	123	219	291	480	843	**1083**	1515	1827
$\mathbf{M}_{7i}$	14655	123	219	291	480	843	1083	**1515**	1827
$\mathbf{M}_{8i}$	19818	123	219	291	480	843	1083	1515	**1827**

Only results for M_{1i}, M_{4i} and M_{8i} configurations (see table 2) are discussed since they are similar for the others. Also, displacements are normalized with respect to the numerical solution computed with the corresponding matching meshes, denoted with boldface in table 2.

Figures 5a to 5c show the normalized displacement over time. On the other hand, the relative error is shown in figures 5d to 5f. The reference curves are shown in solid black. For all M_{1i} simulations, a steady state solution is attained, with relative error ranging from 0.1% to 2%. In case of M_{4i} and M_{8i} not all near-field / far-field couplings lead to a steady state solution. It can be seen that for M_{41} and M_{42} as well as M_{81}, M_{82} and M_{83} solutions start to oscillate after 20 [s]. The relative error of the steady state solutions is within the same range as before.



(a) Normalized disp. M_{1i}. (b) Normalized disp. M_{4i}. (c) Normalized disp. M_{8i}.

(d) Relative error M_{1i}. (e) Relative error M_{4i}. (f) Relative error M_{8i}.

Figure 5: Normalized displacement and relative error over time for discretizations M_{1i}, M_{4i} and M_{8i}.

It is likely that the increasing oscillations for some near- / far-field mesh configurations is due to the fact that a too coarse far-field discretization is not able to properly transmit to infinity the high frequencies which are incorporated by the relatively more refined near-field. Consequently, it seems that Sommerfeld's radiation condition is fulfilled only for the lower frequencies but not for the higher ones.

6 CONCLUSIONS

It is shown that the state projection strategy coupled to the FEM-SBFEM approach can achieve accurate results. The projection strategy is introduced by modifying how the interface velocities and forces are evaluated within the coupled approach (cf. section 4). The use of projection allows to get computational and memory costs benefits, with few changes to the original interfaces of the FEM and SBFEM codes. The biggest reduction in time and memory consumption comes from the computation of the influence matrices. Finally, it is found in the settlement problem that if the SBFEM mesh is too compared to that of FEM, then it is not able to properly transmit the high frequencies to infinity and the Sommerfeld's radiation condition is not fully satisfied.

ACKNOWLEDGMENTS

This work has received financial support from Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET, Argentina, grant PIP 11220150100588CO), Universidad Nacional del Litoral (UNL, Argentina, grant PROMAC 2016), Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT, Argentina, grants PICT 2660–14, PICT-2015-2904), Red CYTED 2015 CAD-ING 516RT0512.

This work was performed with *Free Software Foundation/GNU-Project* resources like GNU–Linux OS, GNU–GCC, GNU–Octave, as well as other Open Source resources such as ParaView and LaTeX.

REFERENCES

Arya S. and Mount D. *ANN: A Library for Approximate Nearest Neighbor Searching*. Department of Computer Science, University of Maryland, 2010. Https://www.cs.umd.edu/~mount/ANN.

Beer G. *Programming the Boundary Element Method*. Wiley & Sons, Chichester, UK, 2001.

Borsutzky R. *Braunschweiger Schriften zur Mechanik - Seismic Risk Analysis of Buried Lifelines*, volume 63. Mechanik-Zentrum Technische Universität Braunschweig, 2008.

Buss S.R., editor. *3-D Computer Graphics: A Mathematical Introduction with OpenGL*. Cambridge University Press, New York, 2003. ISBN 978-05-218-2103-2.

de Boer A. *Computational Fluid-Structure Interaction. Spatial Coupling, Coupling Shell and Mesh Deformation*. Ph.D. thesis, Technische Universiteit Delft, 2008.

Harr M.E. *Foundations of Theoretical Soil Mechanics*. McGraw-Hill Book Company, New York, 1966.

Hübner B., Walhorn E., and Dinkler D. A monolithic approach to fluid-structure interaction using finite elements. *Computational Methods of Applied Mechanics in Engineering*, 193:2087–2104, 2004.

Lehmann L. An effective finite element approach for soil-structure analysis in the time-domain. *Structural Engineering and Mechanics*, 21:437, 2005.

Löhner R. *Applied Computational Fluid Dynamics Techniques - An Introduction Based on Finite Element Methods*. John Wiley and Sons, Chichester, second ed. edition, 2008.

Meskouris K., Hinzen K.G., Butenweg C., and Mistler M. *Bauwerke und Erdbeben - Grundlagen - Anwendung - Beispiele*. Vieweg+Teubner Verlag, Wiesbaden, 2007.

Newmark N. A method of computation for structural dynamics. *Journal of Engineering Mechanics Division*, 85:67, 1959.

O'Rourke J., editor. *Computational Geometry in C, Second Ed.* Cambridge University Press, New York, 1998. ISBN 052-16-4010-5.

Schauer M., Roman J.E., Quintana-Ortí E.S., and Langer S. Parallel computation of 3-d soil-structure interaction in time domain with a coupled FEM/SBFEM approach. *Journal of Scientific Computing*, 52:446–467, 2012.

Walhorn E., Kölke A., Hübner B., and Dinkler D. Fluid-structure coupling within a monolithic model involving free surface flows. *Computers and Structures*, 83:2100–2111, 2005.

Wolf J. *The Scaled Boundary Finite Element Method*. John Wiley & Sons, Chichester, 2003.

Wolf J. and Song C. *Finite-Element Modelling of Unbounded Media*. John Wiley & Sons, Chichester, 1996.