Computationally efficient simulation in urban mechanised tunnelling based on multi-level BIM models

Jelena Ninić[a], Hoang-Giang Bui[b], Christian Koch[c], Günther Meschke[b]

[a] *Centre for Structure Engineering and Informatics, The University of Nottingham, UK. Email: jelena.ninic@nottingham.ac.uk*
[b] *Institute for Structural Mechanics, Ruhr University Bochum, Germany*
[c] *Chair of Intelligent Technical Design, Bauhaus-Universität Weimar, Germany*

Abstract

The design of complex underground infrastructure projects involves various empirical, analytical or numerical models with different levels of complexity. The use of simulation models in current state-of-the-art tunnel design process can be cumbersome when significant manual, time-consuming preparation, analysis and excessive computing resources are required. This paper addresses the challenges connected with minimising the user workload and computational time, as well as enabling real-time computations during the construction. To ensure a seamless workflow during design and to minimise the computation time of the analysis, we propose a novel concept for BIM-based numerical simulations, enabling the modelling of the tunnel advance on different levels of detail in terms of geometrical representation, material modelling and modelling of the advancement process. To ensure computational efficiency, the simulation software has been developed with special emphasis on efficient implementation, including parallelisation strategies on shared and distributed memory systems. For real-time on-demand calculations, simulation-based meta models are integrated into the software platform. The components of the BIM-based multi-level simulation concept are described and evaluated in detail by means of representative numerical examples.

1. Introduction and related work

With increasing urbanisation and demand for efficient and environmentally friendly urban mobility, the use of the underground space for tunnel facilities often is the only solution. For the planning of large infrastructure projects such as inner-city subway tracks, engineers have to conduct investigations and develop models with consideration of different scales (from the kilometre scale for the general routing of the track down to the centimetre scale for detailed design of connection points). Evidently, in the planning and design phase adequate models need to be generated to perform computational simulations and visualization considering the different scales involved in such projects.

In order to come up with an optimal design, different design alternatives need to be created, assessed and compared. Assessing the effects of various design alternatives for urban tunnelling projects considering the existing environment is a complex multi-disciplinary problem. Concepts of Building Information Modelling (BIM) have been used recently to address the challenges of interaction modelling in mechanised tunnelling [28, 27, 45]. While some work has focused on the system modelling of coupled simulations using the System Modelling Language (SysML) [40], others have applied product modelling techniques to model individual components of a tunnelling project, for example the shield tunnel [9, 10], the tunnel boring machine [22] and the soil [48, 34, 18], and to interlink and integrate these parts within a holistic Tunnel Information Modelling framework [22].

The use of numerical methods for the assessment of the mechanised tunnelling design has a number of advantages compared to empirical or analytical solutions, since it provides deeper insight into 3D non-linear soil-structure interaction effects. Since the early 1980s, numerous 2D and 3D numerical models for machine driven tunnelling are proposed (see [46, 20, 24, 14, 26] and references therein). Currently, the process-oriented simulation model EKATE is being developed for arbitrarily curved tunnel alignments in soft, partially and fully saturated soils [35, 2]. The application of numerical models to the analysis of the construction of large project sections, including modelling of the existing infrastructure and taking care of adequate discretisation, implies significant modelling and computational efforts [12, 37]. If such analyses are applied for the investigation of different design alternatives, both modelling and computational efforts might become excessive if proper techniques for improving efficiency are not employed.

In this paper we propose a software platform for simulation of the mechanised tunnelling process that address both important aspects of the computational efficiency - the user and computer workload. To reduce the user workload, the platform supports the acquisition of data and model generation and the visualisation of important results using BIM. Moreover, the user-friendly parametric model generation concept decreases the time needed for learning to use the software. The computational time is minimised by optimising the model size and using parallelisation strategies for the solution on the one hand, and applying the meta models for the real time prediction on the other hand. The remainder of this paper is organised as follows: The challenges in reducing computational efforts in complex numerical simulations and the main research objectives of this paper are presented in Section 2. In the subsequent Section 3 three different aspects of the efficiency in numerical modelling are described: i) multi-level BIM-based parametric modelling in Section 3.1, ii) parallelisation of the finite element (FE) simulation model in Section 3.2, and iii) real-time calculation using simulation-based meta models in Section 3.3. Several examples to demonstrate the main features and the performance of the proposed BIM-based simulation methods are given in Section 4. Finally, in Section 5, we summarise the most imortant findings of this paper and provide conclusions.

2. Problem statement and objectives

The computational efficiency of finite element program has grown following the evolution of computer architecture. To meet the demands of higher accuracy and simulation of larger physical domains, the maximum size of finite element models has risen from approximately 18,000 degrees of freedoms (DOFs) in the 1960s [53] to more than ≈ 10^{13} DOFs nowadays [41]. In addition, the complexity of the underlying physical description has increased with the use of coupled multiphase formulations and sophisticated constitutive laws to reflect the material behaviour under different loading scenarios. This nontrivial complexity introduces enormous challenges to keep the computational efficiency of the finite element program required in tunnelling engineering practice. These challenges are connected in particular with the effectiveness of the solution strategy of the resulting discretised linear system and of the robust integration of the nonlinear constitutive laws.

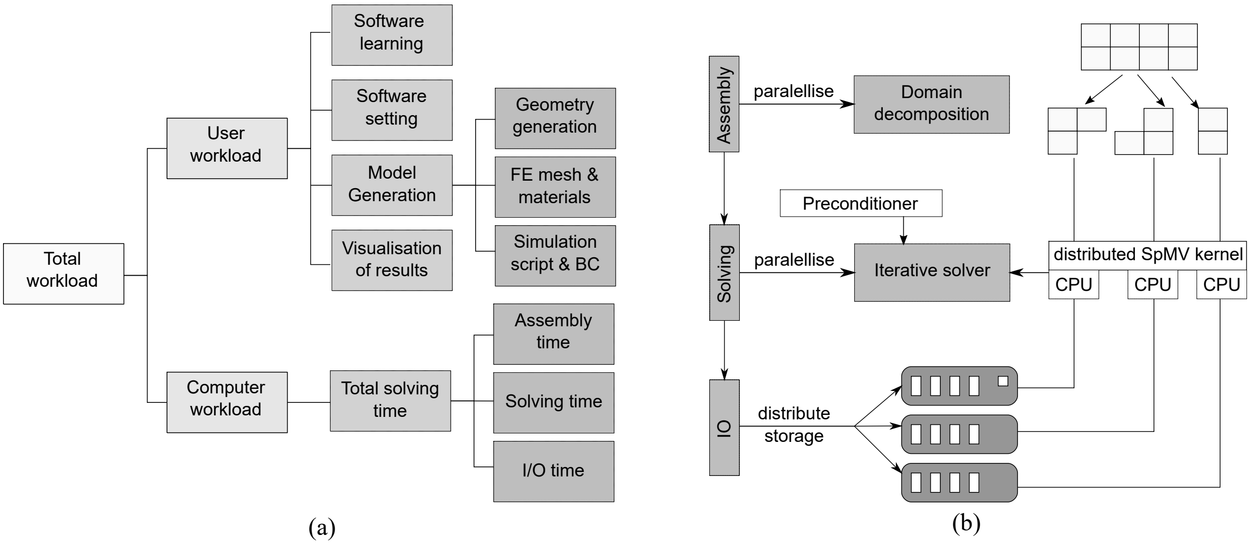


Fig. 1. (a) Total workload for the design assessment of engineering problems using numerical analysis. (b) Parallelisation in each stage of the computational model.

In computational modelling, the total workload can be broadly defined as “the total requests made by users and applications of a system” [25]. The total workload can be divided into user workload and computational workload as illustrated in Figure 1 (a). User workload reflects the time needed for the user to i) learn how to use the software, ii) setup and adjust the software for the particular problem to solve, iii) collect input parameters, iv) generate the computational simulation model, and v) visualise the output of the simulation model in a comprehensible form. The computational workload encompasses the total runtime required for the assembly and solving of the FE system as well as the I/O time (see Figure 1 (a)). The schematic diagram in Figure 1 (b) explains how the parallel processing is done in different stages of the computation: i) domain decomposition to reduce the assembly time, ii) iterative solver using parallelised sparse matrix-vector multiplication and preconditioner to reduce the linear solving time, and iii) the results are distributed and stored in each computing node to reduce I/O time. In addition, the memory consumption can also be seen as part of the computational workload. Therefore, the overall efficiency of computational modelling has to be assessed on the level of software design and implementation, where each level requires an appropriate approach on how to quantify and improve the efficiency.

The resulting user workload highly depends on the problem to be solved. In case of a parametric study, which is often applied to parameter identification problems, optimisation, or sensitivity analysis, a user executes the same model with different parameters many times. The most efficient way to solve this problem would be to once generate the model and the algorithms (software) for the assignment of different sets of parameters and for the automatic execution of simulations. In this case, the total workload would be mostly dependent on the computational workload and could be reduced by an efficient implementation of the solution strategies. Another problem is the investigation of different design alternatives. In that case, each design alternative requires an individual numerical model, and the generation of these numerical models becomes a user workload overhead [38]. In the first scenario, efficient modelling requires fast execution of each simulation and a robust and fail-safe performance of the simulation software, while in the second scenario, it is crucial to automate the generation of different numerical models. In both cases, the parallelisation of the solver contributes significantly to computational efficiency.

Parallelisation of numerical simulations plays a crucial role to speed-up the simulation of a particular type of problem. For small and medium-sized problems, which fit in the computer memory, the shared-memory paradigm is usually employed, particularly for its ease of adoption. Nevertheless, for large scale problems, the distributed memory paradigm must be used to enable storing the problem-dependent data in different computing nodes. Designing a scalable algorithm for distributed computing platforms, i.e. an algorithm that makes the computing time converge asymptotically to Amdahl's rule, is not a trivial task. In practice, one generally focusses on the slowest path of the program, i.e. the hot-spot, and optimise it to improve responding run-time. In typical finite element applications, the most time-consuming tasks are I/O operations and linear system solving. In alternative computational methods, such as the finite cell method, the computing time shifts towards the intensive assembly operation due to the excessive number of integration points. For details on the finite cell method the reader is referred to [43].

The time required by I/O operations generally is limited by system resources and hardware capabilities. It can be improved by using faster hardware (e.g. memories, system network). In the post-processing stage, the results are stored on separated computing nodes in order to reduce the time to access the memory, as can be seen in Figure 1(b). The results can be merged and used for visualisation after the computation is finished. In addition, however, I/O time can be significantly reduced by extracting only the useful information from the simulation data, since most of the data is not of particular interest for further analysis.

The solution of the linear systems of equations arising during the iterative solution of nonlinear problems is more relevant in regards to computational efforts and needs to be optimised, since the solution time depends on the employed solution algorithm. The optimisation of linear solvers has been the topic of intensive research for decades [42]. For large systems, iterative linear solvers are typically employed to solve the linear system up to the prescribed accuracy, rather than its direct solver counterpart, mostly due to its memory requirement characterised by a linear increase of the order $O(n)$, where n is the size of the linear problem. For direct solvers, the memory requirement grows with $O(n^3)$. While the theoretical properties of iterative linear solvers depends are attractive for large scale problems, their convergence properties depend on the quality of the preconditioner. A suitable preconditioner is therefore crucial for the performance of iterative solvers and needs to be designed specifically for the problem at hand.

When simulation models are used during tunnel construction, the model parameters have to be updated according to the actual monitoring data and the actual geological conditions in already constructed sections to ensure reliable predictions in forthcoming advancement steps. In this case, even if all aforementioned techniques for the optimisation of software efficiency are applied, the required continuous model updating during the tunnel drive would not be feasible in practice when using 3D Finite Element (FE) models discretised in space and time, as it requires an inverse identification of the model parameters by performing a large number of (several hundred) realizations.

In order to address the aforementioned problems and to improve the existing modelling approach in terms of efficiency, the following research question has to be answered:

- How to devise and implement an integrated platform for the efficient design, analysis and visualisation of mechanised tunneling projects in urban environments? This includes the following sub-questions:
  1. How to reduce the user workload when creating numerical simulations?
  2. How to reduce computational workload?
  3. How to enable real-time calculations?

3. Methodology: Efficiency aspects in numerical analysis of urban mechanised tunnelling

3.1. Reducing user workload and model size: BIM-supported parametric modelling using a multi-level approach

3.1.1. BIM-based numerical modelling

Nowadays, the design and process information of complex, integrated, multi-disciplinary systems as encountered in mechanised tunnelling are often managed and stored in BIM [55, 9]. The underlying information needed for numerical analysis is therefore available in a BIM, however the translation from this information to the format accepted by numerical design tools currently is still a largely manual process.

In this paper, we propose the integrated design-analysis framework SATBIM, where numerical models are automatically generated based on the geometry and semantics stored in BIM design tools (see Figure 2). In the SATBIM platform, an automatic link between a parametric BIM for tunnelling, denoted as Tunnel Information Model (TIM)

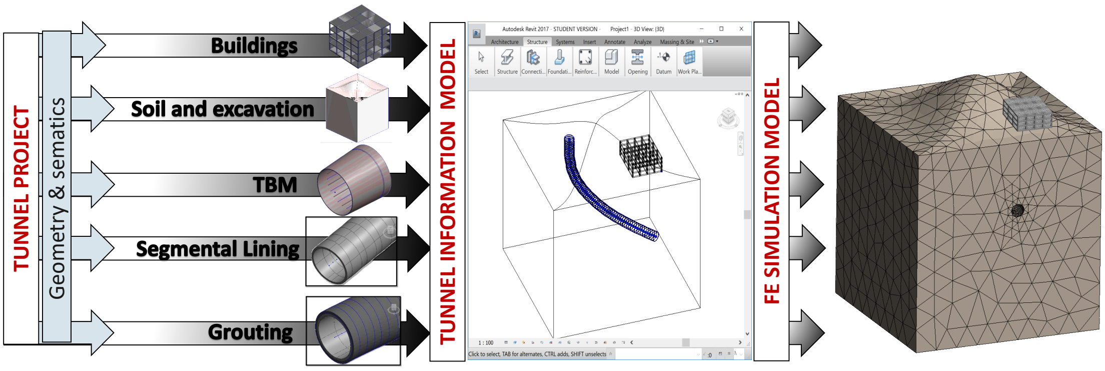


Fig. 2. Generation of numerical simulation based on design data stored in a BIM.

and an adequate multi-level simulation model is developed to enable a consistent and efficient assessment of the design (or of several design alternatives) by means of numerical analysis. SATBIM is an integrated, open-source platform for information modelling, structural analysis and visualisation of the mechanised tunnelling process. Within this platform, industry-standard tools (Autodesk Revit) are employed for the design of the tunnel structure and the surrounding infrastructure with consideration of different Levels of Detail (LoDs) for all system components. As opposed to the concept of Level of Development (LOD), or Level of Model Definition (LOMD), that has been introduced by the American Institute of Architects (AIA) in collaboration with the American BIMforum [1, 8], in this paper we refer to Level of Detail (LoD). According to [8], LoD defines how much detail is included in the model element, whereas LOD defines the degree to which the element's geometry and semantic information has been thought through in the development process. LOD, in that sense, specifies the reliability and maturity of information in the model. In conclusion, in this paper we do not focus on the model development process, but on the degree of detail (for both geometry and information) that are captured in each of the system components. Based on the parametric TIM, numerical models for each component are generated, considering proper representations of geometric properties as well as material data, interfaces between components and the construction process. The numerical models are then fully automatically instantiated and executed based on the workflow of the BIM model for mechanised tunnelling. Finally, the simulation outputs are read back and visualised within Revit [36].

Using this approach, the user workload is minimised in terms of model generation as well as time and effort needed for learning how to operate the software. The BIM model for mechanised tunnelling implemented in the design tool Autodesk Revit is fully parametrised, using the Dynamo plug-in. This provides a visual programming environment that allows to make use of external libraries to enable data processing, generating geometries and manipulating BIM objects at runtime. The user provides the model design parameters (geometry, semantics, external geometries etc.) through the user-friendly interface of Dynamo, and generates the tunnel design model based on parametric Revit objects (instances of so-called “families”). Subsequently, the user executes an automatic `SatBimModeller` (explained below) to generate numerical models based on this BIM design model.

3.1.2. Multi-level approach

In the planning phase of tunnelling projects, decisions have to be made on the tunnel project, which cover different spatial scales: At the kilometre scale the tunnel alignment needs to be optimised, while the design of the TBM and the tunnel structure, including e.g. details of joints, are related to the centimetre scale (see Figure 3a).

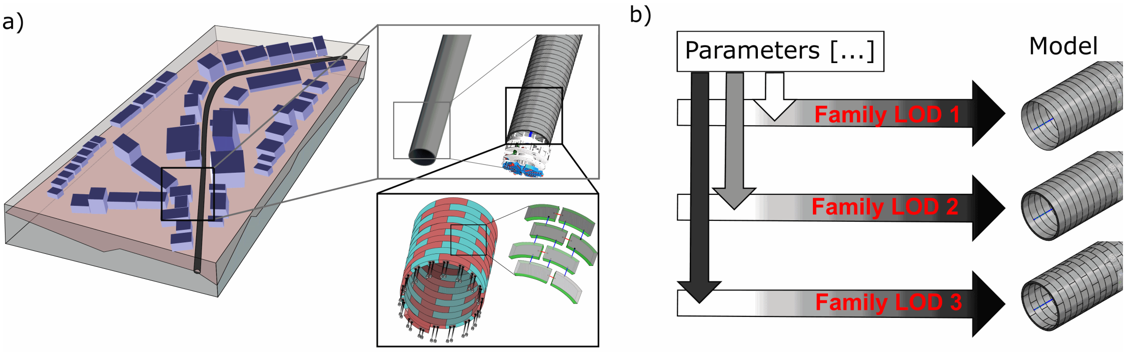


Fig. 3. a) Multi-level representation of geometry; b) parameter consistency between different LoDs for individual components

Each scale is characterised by a certain amount of information (detail of description) and corresponding numerical models. Those scales have to be consistently connected in order to avoid conflicts and a lack of compatibility in numerical analysis. Moreover, in different design phases, a different type and amount of information is available. Hence, the complexity of an adequate numerical assessment changes accordingly.

To have consistency between different sub-models throughout different phases in design and analysis, characterised by using building information and computational simulation approaches, in SATBIM we propose a multi-level approach. For each component of the shield tunnelling process described in Section 3.1.4, three LoDs are defined: *Low*, *Medium* and *High*, also referred to as LoD 1, 2, and 3 in the following.

In general, on the lowest LoD (LoD 1), a non-volumetric representation of the component is assumed, since in the corresponding numerical models, the respective component is not represented with actual structural models, but instead with an empirical model realised through a set of boundary conditions. On the medium LoD (LoD 2), a volumetric representation is established for each component, where the component is “occupying” the exact volume, however, the geometry is simplified, i.e. approximated. Finally, on the highest LoD (LoD 3), we include more details about the actual geometry of the individual component (see Section 3.1.4).

For each component and on each LoD, a so-called “Revit family” for the corresponding component is created. In order to keep consistency between different LoDs, parametric consistency between families is defined as shown in

Figure 3b. The full set of parameters defining a component is needed for definition of the highest LoD, while only a subset of the parameter list is used on lower LoDs. This way of handling parameters allows one to automatically preserve the consistency of the multi-scale model.

Selecting the "optimal" complexity of the model with respect to (w.r.t.) the objective of the analysis leads to the optimal (reduced) size of the numerical models, and consequently, reduced computational efforts. For example, minimising the overall risk of damage to buildings needs high LoD for structures and low-detail modelling of the lining ring installation, while estimating stresses in the lining structure needs high LoD for lining, while low LoD for buildings is sufficient.

3.1.3. Automatic model generation

One of the most important aspects of efficient modelling is automation in model generation. The SATBIM platform integrates different tools for BIM, (re-)meshing and numerical analysis into a user-friendly platform with a high level of automation to enable a seamless integration of the of design, analysis and assessment tasks. As mentioned, the geometry and the semantics of the design are generated and stored using the industry-standard BIM tools Autodesk Revit [3]. For pre-processing and solving the numerical model, we employ GiD [29] and the open-source FE framework KRATOS [13], respectively. `SatBimModeller` enables the model generation, setup and execution of the simulation model by connecting the multi-level Tunnel Information Model with the simulation software KRATOS and the GiD pre/post-processor. The core of the `SatBimModeller` is a Python program, which reads all defined specifications of the model from ACIS files together with model, material and semantic parameters, and automatically generates the FE simulation model and a simulation script for KRATOS (see Figure 4).

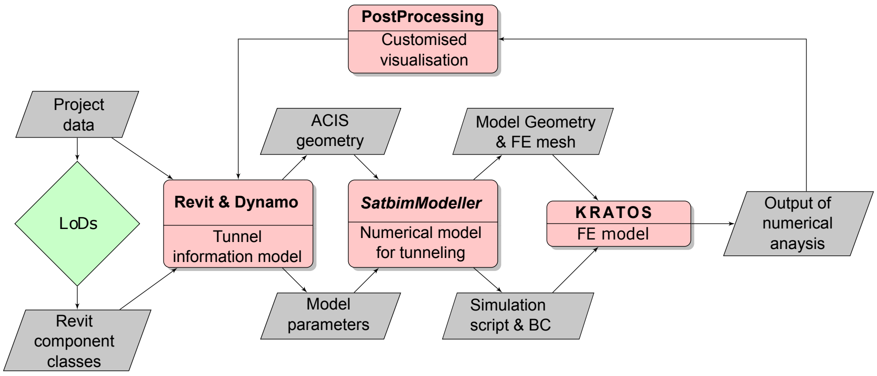


Fig. 4. Data flow in SATBIM.

The SATBIM software is organised in a highly modular way to provide high flexibility not only for further extensions, but also to allow for adaptations in KRATOS. Moreover, `SatBimModeller` allows one to generate simulation output in a form which can be imported back into Revit for the purpose of visualisation of the computational results. Visualisation of simulation results within the TIM is essential to enable a comprehensive, intuitive and quick understanding of effects of design variants on relevant parameters such as soil settlements or the risk of damage of the existing environment (see Figure 4).

3.1.4. Multi-level simulation model for machine driven tunnel advancement

A realistic model to be applied during construction and in the design phase has to represent all components of the tunnelling process relevant for the prognosis of the response of the surrounding soil during excavation. These components include: i) the soil and the excavation domain, ii) the segmental lining with the support measures applied at the tunnel face and at the tail void, iii) the TBM and iv) existing infrastructure. In the SATBIM project, for each of these components a multi-level representation of the information models and corresponding numerical models is developed. To this end, the main components and their LoDs are illustrated in Figure 5 and described below:

- Soil model:
  - LoD 1: In the information model, the soil is not described with geometry, but only via parametrically assigned semantics of the material properties. The equivalent numerical representation is the subgrade reaction model, where the soil is modelled with infinitely thin, uncoupled linear springs, described with subgrade reaction modulus K_s (see [50, 23]), neglecting the soil-structure interaction and the weight of the excavated soil. The material model is the simplest possible soil model, a linear-elastic model.
  - LoD 2: In the information model, the geometry, determined as a rectangularly bounded box, is used to delimit the volume occupied by soil layers. The corresponding numerical representation is a 3D structural model discretized with tetrahedral elements and described as a two-phase fully saturated material, according to the theory of porous media (see [35] for details). The applied material models are different elasto-plastic constitutive models that are available in KRATOS, including the Mohr-Coulomb and the Drucker Prager model, which are preferably used for sandy soils.
  - LoD 3: The 3D geological model consisting of geotechnical layers is represented with a single or multiple geometric volume and material descriptions for each corresponding layer. The geometry is defined using the actual CAD geometry containing soil or rock layers, their boundaries, and their geotechnical properties in a standard format for tunnel ground models. In terms of numerical modelling, the same FE representation of the soil as for LoD 2 is employed here. The material model is the more general Clay and Sand Model (CASM), characterised by non-associative plasticity and Lode-angle dependent yield surfaces [52].
- Lining model:
  - LoD 1: The information model is represented by a shell, representing the outer boundary of the lining

structure. Numerically, the confinement and support provided by the lining structure is represented with a volume loss method and a prescribed volume loss coefficient $V_l = \left(V_0 - V_{def}\right)/V_0 \cdot 100\,\%$.

  - LoD 2: In the information model, the lining tube is represented by a set of lining rings described with ring geometry and material properties. Numerically, a structural 3D model is applied, where the ring volume elements, discretised by linear hexahedral finite elements, are activated during the simulated tunnel advance.
  - LoD 3: In the information model, each segment is represented as a single volume. In the numerical representation, a structural 3D model of segmented lining is used. The longitudinal and circumferential joints between segments are modelled in a discrete manner. The joints are modelled by introducing bolts represented by beam elements and a surface-to-surface normal contact condition between segments and transversal joins of the lining rings. In the simulation of the tunnel construction process, the segment-wise installation of the tunnel lining is modelled.
  - Grouting: A structural volumetric model is discretised with tetrahedral elements. To account for the pressurization of the grouting mortar, a two-phase formulation for fully saturated media is adopted, and the hydration is described by time-dependent material properties for both the strength characteristics and the permeability (for details refer to [30]).

- TBM model:
  - LoD 1: The information model is represented by a cylindrical shell, representing the outer boundary of the TBM volume. This model corresponds to a non-structural model, where the shield is described by boundary conditions [17]. In this approach, the shield is represented by a set of n segments with length L_r with uniformly defined boundaries in terms of radial displacements that approximate the conical surface of the shield, where $n = L_{TBM}/L_r$ and L_{TBM} is the total length of the machine.
  - LoD 2: The TBM is represented by a shield volume considering the thickness of the shield, its tapered geometry and material. In the numerical model, this volume is discretised with tetrahedral finite elements and is used to represent the TBM as a deformable body moving through the soil and interacting with the soil through surface-to-surface contact.
  - LoD 3: In the information model, details of the TBM such as the shield body, the shield wall (for abutment of hydraulic jacks) and hydraulic jacks are modelled. In the numerical representation, these elements are discretised with tetrahedral (shield body and wall) and linear truss (hydraulic jacks) finite elements. In the simulation model, the TBM advance is accounted for by pushing forward the TBM by elongation of hydraulic jacks [2].

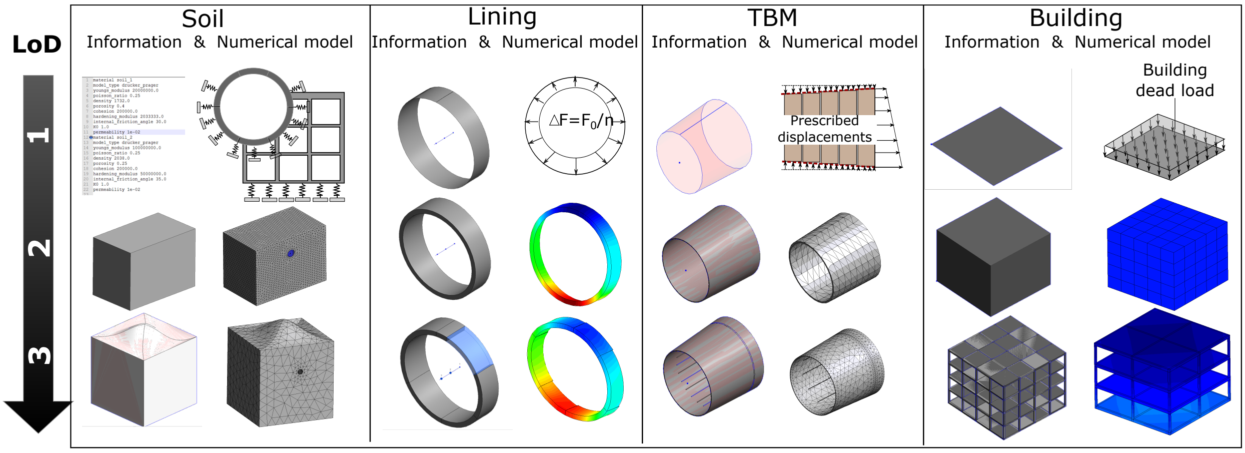
  - Face support: For the two-phase elastoplastic soil models (LoD 2 and LoD 3), a surface pressure is applied to the tunnel face by prescribing total stresses at the heading face with a linear distribution (a minimum value at the tunnel crown and a maximum value at its bottom) according to design recommendations.

- Model for existing Buildings:

  - LoD 1: In the information model, a building is represented with the building footprint and semantics describing the building material. The corresponding numerical model is a non-structural representation, where the building is substituted by a dead load from the building weight acting on the soil surface.
  - LoD 2: The information model is a 3D geometry approximated with a cube volume that the building is "occupying" and the semantics describing the building material. In the corresponding numerical model, a building is described with a substitute elastic stiffness E, height H and weight ρ computed according to an approach proposed in [44]. The volume is discretised with isotropic volume tri-linear hexahedral elements.
  - LoD 3: The information model is a full 3D structural frame model with material description. In the numerical model, the columns and floors are both modelled by means of isotropic volume hexahedral elements.
  - Foundation-soil interactions (Building (LoD 2, LoD 3) with soil (LoD 2, LoD 3)): The interaction is accomplished through a mesh-independent surface-to-solid tying algorithm using a penalty approach, which prevents the penetration of the foundation further into the soil. Different mechanisms of soil-structure interaction corresponding to "sagging" and "hogging" modes are considered by deactivating the tying links between building and soil contact pairs when the tensile stress in tying links is larger than the prescribed threshold.

3.2. Reducing computational time: Parallelisation

In this paper, parallelisation of the numerical analysis of the tunnel advancement process is applied to reduce the computing time w.r.t. several aspects. Firstly, the system domain is decomposed into sub-domains using state-of-the-art domain decompositioning packages [19], where each contribution of the sub-domains to the global linear system is generated by separate computing processes. Secondly, a specialised preconditioner is used to speed-up the convergence of the iterative solver. This reduces the computing time while maintaining the accuracy of the results. Special attention needs to be paid to the specific structure of the tangent matrices (see Eq. 3) arising from coupled hydro-mechanical problems as present in tunnel construction in partially or fully saturated soils (see e.g. [51, 7] for applications of block preconditioners to multiphase problems). This is particularly addressed in Section 3.2.2.



Soil
Lining
TBM
Building
LoD
Information & Numerical model
Information & Numerical model
Information & Numerical model
Information & Numerical model
1
2
3
ΔF=F0/n
Prescribed displacements
Building dead load

```
material soil_1
model_type drucker_prager
youngs_modulus 20000000.0
poisson_ratio 0.25
density 1732.0
porosity 0.4
cohesion 200000.0
hardening_modulus 2033333.0
internal_friction_angle 30.0
K0 1.0
permeability 1e-02
material soil_2
model_type drucker_prager
youngs_modulus 100000000.0
poisson_ratio 0.25
density 2038.0
porosity 0.25
cohesion 200000.0
hardening_modulus 50000000.0
internal_friction_angle 35.0
K0 1.0
permeability 1e-02
```

Fig. 5. Representation of the soil, lining, TBM and buildings as information and equivalent numerical models on different LoDs.

3.2.1. Governing equations and finite element discretisation

The governing equations describing the underlying physics of machine-driven tunnel advancement in partially or fully saturated soils are presented in [35] and are not repeated here in detail. The weak form of the balance equations (i.e. equilibrium of forces and the mass balance of the groundwater) and the conditions related to the frictional contact between the tunnel boring machine and the soil is characterised by four terms:

$$\delta W = \delta W^s + \delta W^w + \delta W^c + \delta W^t, \tag{1}$$

where δW^s and δW^w represent the weak form of equilibrium of forces and the mass balance of the groundwater, respectively, δW^c represents the virtual work of the contact forces arising from the frictional contact between the tunnel boring machine and the soil [11] and δW^t is concerned with the contribution of the interaction forces between the buildings and the soil, which result from the tying of the domains of the existing buildings and the soil surface via the

penalty method:

$$\delta W^t = \int_{\Gamma^t} \rho_t (\delta \mathbf{u}^{(1)} - \delta \mathbf{u}^{(2)}) \cdot (\mathbf{u}^{(1)} - \mathbf{u}^{(2)}) dV. \tag{2}$$

This allows for a nonconforming, independent discretisation of the existing surface structures and the soil, which is an important ingredient in terms of efficiency of model generation.

Eq. (1) involves the displacements $\mathbf{u}^s$ and the water pressure p^w the main unknowns. After linearisation of Eq. (1) and spatial discretisation using finite elements, the resulting coupled system of equations arising during the iterative solution within the time interval $[t_n, t_{n+1}]$ can be written in block-form:

$$\begin{bmatrix} \Delta u^s \\ \Delta p^w \end{bmatrix}_{n+1}^{i} = \begin{bmatrix} A_{uu} & A_{uw} \\ A_{wu} & A_{ww} \end{bmatrix}^{-1} \begin{bmatrix} R^u_{ext} - R^u_{int} \\ R^w_{ext} - R^w_{int} \end{bmatrix}^{i} . \tag{3}$$

The subscripts u and w refer to the mechanical and the fluid transport related subproblem, Δu^s and Δp^w are the incremental vectors of nodal displacements and pore pressures to be computed in the i-th iteration, and the vectors $R_{ext} - R_{int}$ denote the residuals of the external and internal force vectors. To furnish the spatial discretisation of Eq. 1 compatible with the LBB conditions [54], the water pressure is approximated by a functional space of lower order as compared to the approximation space of the displacements. In the numerical examples presented in Section 4, tetrahedral element, with 10 nodes are used for the approximation of the displacements, while 4-node tetrahedral elements are employed for the pressure.

3.2.2. Computation and parallelisation strategy

The domain decomposition strategy for numerical simulations in mechanised tunnelling must account for the fact that the contact surfaces between the TBM and the surrounding soil continuously change during the advancement of the TBM. The contact surfaces are determined by a contact search algorithm based on Bounding Volume Hierarchy (BVH). The TBM is denoted as the slave body and the surrounding soil as the master body. When using a domain decomposition strategy, it can happen that when the TBM passes through the boundary between two computational domains, the contact search algorithm may fail since some of the master segments do not belong to the current working process. To cope with this issue, the master segments are distributed a priori to all computational domains as ghost segments. Figure 6 illustrates the distributed ghost master segments related to the domain of soil elements containing the contact surface between the soil and the TBM (vertical column slab of finite elements) .

In order to solve the block system (3) using iterative solvers effectively, the block preconditioner is employed. It is well known from the literature that the Incomplete LU (ILU) preconditioner [6] is not suitable for multiphase problems, as the LU preconditioner does not scale well when the size of the problem grows and due to the coupling characteristic of the multiphase matrix resulting from the fact, that the solid block has much larger magnitude as compared to the pressure block. This imposes an additional challenge for the ILU preconditioner to compute an appropriate inverse. According to [7], block preconditioners are the preferred choices that can be used to speed up the convergence of the iterative solver for multiphase problems. Several variants of block preconditioners exist, including the Block

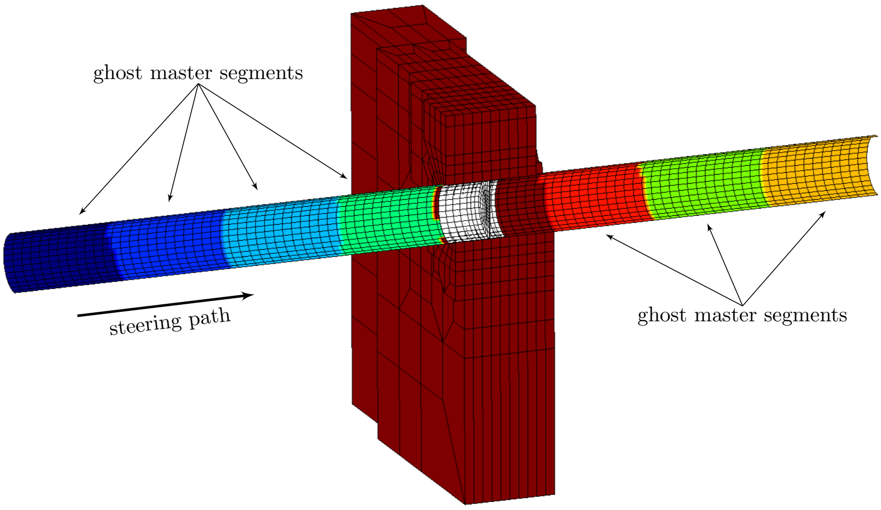


Fig. 6. Domain decomposition methods in mechanised tunnelling: Ghost master segments to account for TBM-soil contact in multiple computational domains.

Gauss-Seidel preconditioner [4, 5],

$$P_{\text{gs}}^{-1} = \begin{bmatrix} I & 0 \\ 0 & -A_{ww}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -A_{wu} & I \end{bmatrix} \begin{bmatrix} A_{uu}^{-1} & 0 \\ 0 & I \end{bmatrix}, \tag{4}$$

and a class of block preconditioners based on the Schur complement, which is abbreviated as Schur preconditioners [51, 7],

$$P_{\text{sf}}^{-1} = \begin{bmatrix} I & -A_{uu}^{-1}A_{uw} \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{uu}^{-1} & 0 \\ 0 & S^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -A_{wu}A_{uu}^{-1} & I \end{bmatrix} \tag{5}$$

$$P_{\text{sl}}^{-1} = \begin{bmatrix} A_{uu}^{-1} & 0 \\ 0 & S^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -A_{wu}A_{uu}^{-1} & I \end{bmatrix} \tag{6}$$

$$P_{\text{su}}^{-1} = \begin{bmatrix} I & -A_{uu}^{-1}A_{uw} \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{uu}^{-1} & 0 \\ 0 & S^{-1} \end{bmatrix}, \tag{7}$$

where A_{uu}, A_{ww} denote the matrix representing the solid and fluid phase, and A_{wu}, A_{uw} are the coupling matrices. S is the Schur complement matrix, which is defined as $S{=}A_{ww} - A_{wu}\text{diag}(A_{uu})^{-1}A_{uw}$.

The quality of this block preconditioner depends substantially on the quality of the sub-preconditioners, which are used to invert the sub-blocks A_{uu}^{-1} and A_{ww}^{-1} (S^{-1}). In the numerical example in Section 4, the multigrid preconditioner [15] is chosen as the sub-preconditioner. The multigrid preconditioner is well known to work effectively for elliptic problems (i.e. thermal problems, elastic problems). Hence, it is expected that the multigrid preconditioner provides an appropriate inversion for the solid block A_{uu}, where it has a much larger contribution to the system matrix as compared to the pressure block A_{ww}. For a detailed investigation of the performance of the block preconditioners in the multiphase problems, the reader is referred to [11]. Figure 7 provides an overview of the simulation software used for parallel computation.

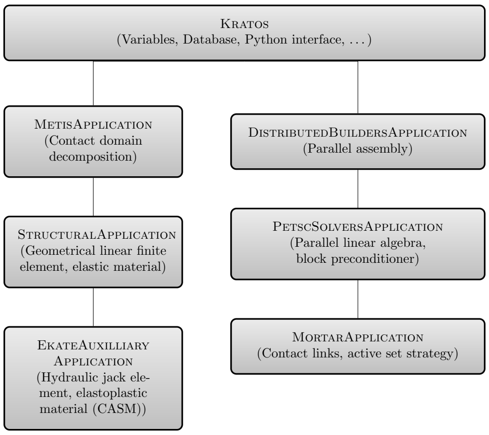

Kratos
(Variables, Database, Python interface, . . .)
MetisApplication
(Contact domain decomposition)
DistributedBuildersApplication
(Parallel assembly)
StructuralApplication
(Geometrical linear finite element, elastic material)
PetscSolversApplication
(Parallel linear algebra, block preconditioner)
EkateAuxilliary Application
(Hydraulic jack element, elastoplastic material (CASM))
MortarApplication
(Contact links, active set strategy)

Fig. 7. Structure of the simulation software.

3.3. Minimising time for forward calculations: Meta models for real-time prediction

In situations where quick decisions need to be made based upon numerical simulations in the design (investigation of large number of the design alternatives) or in the construction stage (process steering) of a tunnelling project, numerical predictions have to be provided within a few seconds. This time frame is impossible to achieve by using direct finite element computations, even if highly efficient parallelisation schemes are employed. Such extremely short response times can only be achieved by using computationally efficient meta (surrogate) models instead of the original FE models. Meta models are a compact representation of simulation models able to return the response of complex FE models for predefined output parameters, e.g. of tunnelling-induced surface settlements, almost instantaneously [37]. To enable real-time predictions, meta models are trained a priori from the process-oriented simulation model generated from the multi-level TIM using `SatBimModeller`. Besides settlements, output parameters may include the lining stresses, pore pressures or damage in existing buildings.

Figure 8 illustrates the use of simulation-based meta models for real-time predictions. Based on different design alternatives of the tunnel project, characterised by particular design parameters, simulation models are generated and executed using the `SatBimModeller`. The simulation results are stored in a format suitable for meta model training. In SATBIM several algorithms are implemented to train the simulation data set and select an optimal meta model based on different methods for data training:

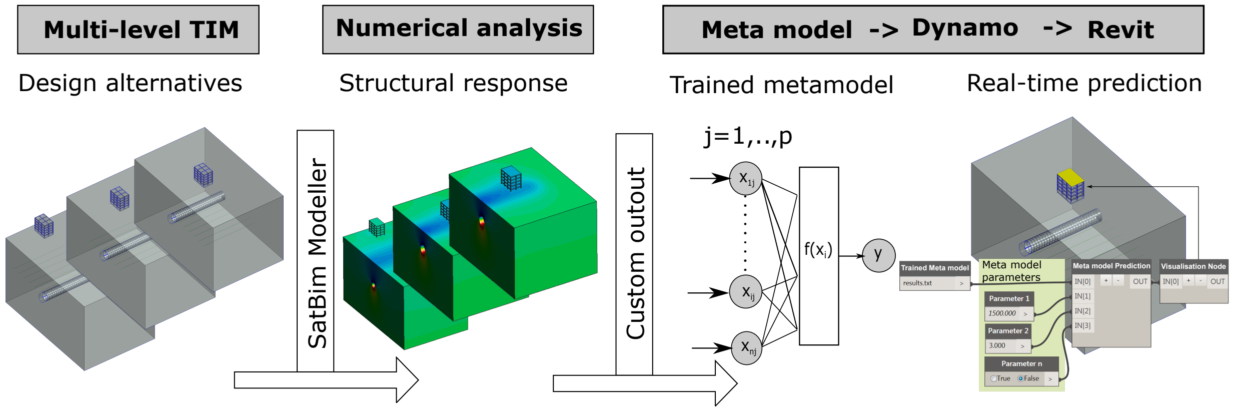


Fig. 8. Workflow of real-time predictions for design optimisation within the Tunnelling Information Model.

- Polynomial Regression (PR),
- Support Vector Regression with Radial Basis Function kernel (SVR-RBF) and Polynomial kernel (SVR-Poly),
- Artificial Neural Networks (ANNs).

In all training methods, the parameters x_i of the simulation model are taken as input variables for the meta model training, while the settlements caused by the tunnel advancement or the risk of damage or other parameters relevant for the design are target parameters $\mathbf{y}$. Meta models are then trained to predict the output for given input values, minimising the error between the target and output values. For the generation of the meta model, the data set is randomly split into data set for training, testing and validation. The meta model is trained with the training set, evaluating the training performance based on testing set, and it prediction capabilities are finally validated with the validation data set.

Considering the different dimensions of different parameters, all input-output pairs are mapped to the interval $(0.1, 0.9)$ using a data normalization algorithm. For a parameter V, the normalised value V_{norm} is obtained from:

$$V_{\text{norm}} = \frac{V - V_{min}}{V_{max} - V_{min}} \left(\bar{V}_{\text{max}} - \bar{V}_{\text{min}}\right) + \bar{V}_{\text{min}}. \tag{8}$$

V_{max} and V_{min} are the maximal and minimal values of the variable V, and $\bar{V}_{\text{max}}$ and $\bar{V}_{\text{min}}$ are the maximal and minimal values of the variable V after normalization, defined as 0.1 and 0.9.

Polynomial regression. For given a data set $\{y_i, x_{i1}, \ldots, x_{ip}\}_{i=1}^{n}$ of n inputs and p patterns, a polynomial regression model is a form of linear regression in which the relationship between the independent variable $\mathbf{x}$ and the dependent variable y is modelled as a nth degree polynomial in x [32]. In its general form, the polynomial regression model can be formulated as:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_m x_i^m + \varepsilon_i \;\; (i = 1, 2, \ldots, n), \tag{9}$$

where the "error variable" ε_i is an unobserved random variable that adds noise to the relationship between the dependent variable and the inputs. For the implementation used in this paper, second-order polynomials are used:

$$y_i\left(\beta,\ x\right) = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \sum_{i=1}^{n} \beta_{ii} x_i^2 + \sum_{i=1}^{n} \sum_{j>i}^{n} \beta_{ij} x_i x_j = \mathbf{X}^T \boldsymbol{\beta} + \varepsilon_i. \tag{10}$$

Support Vector Regression. In the SVR method, the support vector determines the approximation function [49]. The SVR technique searches the multivariate regression function $f\left(x\right)$ based on the input data set, i.e., the training set x to predict the output data:

$$f\left(x\right) = \sum_{x_i \in SV} \left(\alpha_i - \alpha_i^*\right) K\left(x,\ x_i\right) + b, \tag{11}$$

where K is a kernel, b is an offset parameter of the model (equivalent to ε_i in regression models), SV is the support vector set, and $\alpha_i,\ \alpha_i^* \neq 0$ are Lagrange multipliers of the primal-dual formulation of the problem solved by quadratic programming. Polynomials, splines and radial basis functions are examples of kernel functions. The kernel K is, in general, a non-linear mapping from an input space onto a characteristic space through a dot product of the non-linear kernel function $\phi\left(x\right)$ as :

$$K\left(x, x_i\right) = \phi\left(x\right) \cdot \phi\left(x_i\right) \tag{12}$$

The kernel functions ϕ used in this paper are:

- Polynomial: $(\gamma\langle x, x'\rangle + r)^d$, with d as the polynomial degree and r as an independent coefficient with default value 0.
- Radial basis function: $\exp\left(-\gamma\left|x - x'\right|^2\right)$, where γ is a coefficient which must be greater than 0.

Artificial Neural Networks. In the ANN model, the output $y_k\left(x,\ w\right)$ is calculated according to Eq. (13), where w_{ij} are connecting weights from the input to the hidden layer, w_{jk} are connecting weights from the hidden to the output layer, θ_i (θ_j) is a bias and $f()$ is the activation function used to transform the incoming values and transfer them to the next layer:

$$y_k\left(x,\ w\right) = f\left(\sum_{j=1}^{m}(w_{jk} + +\theta_j)\, f\left(\sum_{i=1}^{n} w_{ij} x_i + \theta_i\right)\right). \tag{13}$$

The objective in ANN training is to determine the free parameters ($\boldsymbol{w}$ and $\boldsymbol{\theta}$) to minimise the objective function which represents the error E between the output and the target values:

$$E = \sum_{k=1}^{m} (y_k - t_k)^2. \tag{14}$$

Some of the aforementioned machine learning methods are based upon parameters which influence the training performance. In order to obtain robust meta models, those parameters are optimised with the Particle Swarm Optimisation (PSO) method [21] as illustrated in Figure 9. The algorithm for robust meta model training is implemented in Python, following the main idea described in Figure 9 and applying the machine learning methods described above. For the implementations, the Python library `scikit-learn` for supervised learning was used [39]. This toolkit contains implementation of Regression models, ANNs and SVR, however, for robust learning, in addition PSO was implemented to optimise training parameters of different machine learning models. The training performance and hence meta model prediction capabilities are therefore enhanced by i) random data split for training, test and validation; ii) normalisation of the data, and iii) an optimised hybrid training approach. Moreover, methods for the reading and preprocessing the data, the selection of the optimal model and forward calculations based on the trained meta models were implemented.

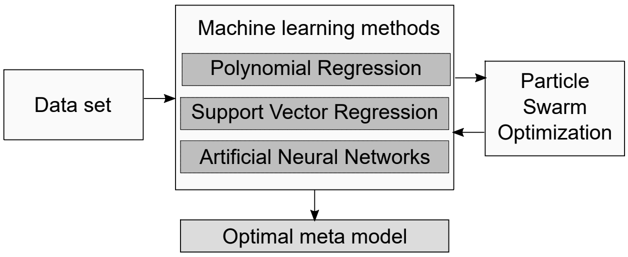

Data set
Machine learning methods
Polynomial Regression
Support Vector Regression
Artificial Neural Networks
Particle Swarm Optimization
Optimal meta model

Fig. 9. Selection of robust meta models by using Particle Swarm Optimisation for selecting optimal training parameters.

4. Numerical examples

In all numerical examples presented in this section, the multilevel simulation framework described in Section 3.1 is used for the generation of large tunnel sections on different LoD as described in Section 3.1.4. Based on this Revit model, a finite element simulation model is generated using `SatBimModeller`. The time needed for the model generation is dominated by the time needed to set up the parameters in the Dynamo interface, and the software runtime (for the generation of the information and the numerical models), which, depending on the size of the problem can take from seconds to minutes on a personal computer with 256 GB of memory. In the numerical examples below, the following aspects in regard to efficiency are discussed: i) choosing the optimal LoD of different model components w.r.t. the analysis scenario, ii) near-instant forward calculations using simulation-based meta models, and iii) computational efficiency by employing parallelisation strategies.

4.1. *Importance of the LoD w.r.t. the objective of the analysis*

In the first example, the importance of the choice of the LoD of certain model components for the accuracy of the results w.r.t. the objective of the analysis is discussed. In this example, the objective is the prediction of the settlements trough induced by tunnel construction. To this end, the effect of different LoDs of soil material models and of the lining model on tunnelling-induced settlements and their effect on the size of the model and computational efficiency are analysed.

Figure 10 shows the steps taken for the generation of the simulation model based on the information model using `SatBimModeller`. The model is used for investigations of the influence of the selected LoD of structural components and material models. The soil geometry is modelled on the medium level of detail (LoD 2), and the lining structure is varied from low to high complexity (LoD 1, LoD 2 and LoD 3). For medium and most detailed representation of the lining structure, the TBM is modelled on LoD 2, while for the volume loss representation of the lining structure support, the TBM is modelled on LoD 1. In this demonstration example, 27 lining rings of 2.5 m length and 10 m excavation radius are excavated under 15 m of soil overburden.

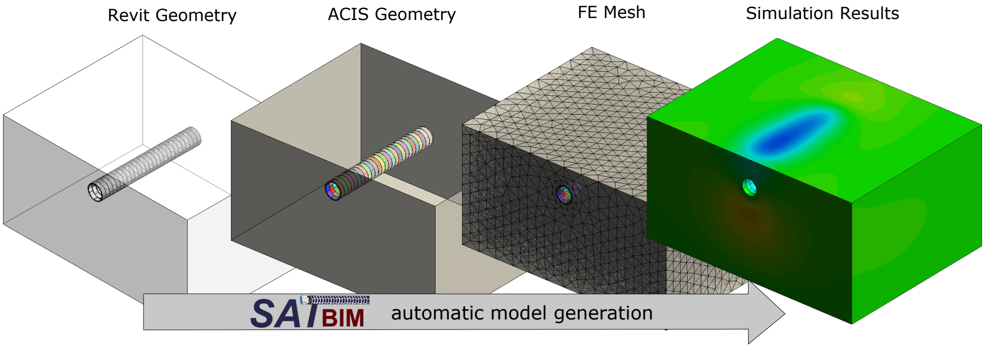


Fig. 10. Generation of the finite element simulation model based on tunnel information model generated in Revit using `SatBimModeller`.

The multi-level approach is also applied to the selection of the material model. Three different material models are considered: a linear elastic model (LE) on LoD 1, the Mohr Coulomb model (MC) on LoD 2, and the CASM model on LoD 3. The soil material parameters used for this example are summarised in Table 1.

4.1.1. *Ring-wise vs. segment-wise installation*

Figure 11 shows the induced settlements trough for different lining models. Here, $V_l = 0.8\%$ is applied for modelling of the lining on LoD 1. It can be observed that the selected value of soil confinement leads to slightly larger settlements and a more narrow shape of the settlement trough for the same volume loss as compared to higher LoDs (rel. error= 9.5 %). The choice between lining on LoD 2 and LoD 3 does not make a significant difference when calculating tunnelling-induced settlements (rel. error=3.2 %) as shown in Figure 11. Assuming that the most

Constitutive law	LE	MC	CASM
Young modulus [MPa]	80	80	80
Poisson ratio	0.25	0.25	0.25
Density [kg/m^3]	1732	1732	1732
Porosity	0.4	0.4	0.4
Cohesion [kPa]	-	200	-
Hardening modulus [MPa]	-	58.3	-
Friction angle [°]	-	30	-
Dilatancy angle [°]	-	30	-
Permeability [m/s]	0.001	0.001	0.001
Slope of the unload/reload curve in the (v — ln p') space κ	-	-	0.001
Slope of the normal compression curve in the (v — ln p') space λ	-	-	0.01
Spacing ratio r	-	-	0.2
Shape parameter of the yield surface n	-	-	2.0
Slope of the critical state line under triaxial compression M	-	-	1.08
Initial preconsolidation mean stress for saturated soil P_0 [kN/m^2]	-	-	10^{15}

Table 1. Material parameters for LE, MC and CASM material model

detailed model (LoD 3) leads to the highest accuracy in the prediction, the general conclusion is that a high accuracy in modelling of the lining structure does not contribute significantly to the accuracy of the settlement prediction. In general, if multiple runs are required for parametric studies, the model with the highest LoD for the lining structure could be used for the determination of the volume loss parameter, and the subsequent simulation models could be generated with the lowest LoD (volume loss) to reduce the model size and computational costs.

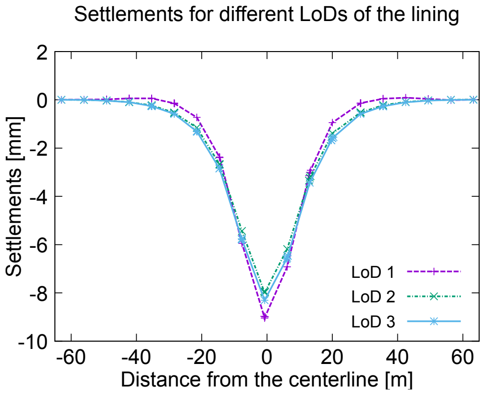


Fig. 11. Tunnelling-induced surface settlements trough for tunnel linings with LoD 1, LoD 2 and LoD 3 for the CASM soil model.

However, for detailed lining design assessment in terms of its structural response, detailed modelling of lining rings is required, and therefore the choice of accuracy level for modelling lining rings makes a significant difference. In the volume loss method (LoD 1), the lining structure is not explicitly modelled and therefore cannot be assessed. In Figure 12, the difference in distribution of the total longitudinal, radial and tangential stresses in lining rings for LoD 2 and LoD 3 is shown. The accuracy in modelling is paid for with higher computational costs (see Table 2), which is further discussed below.

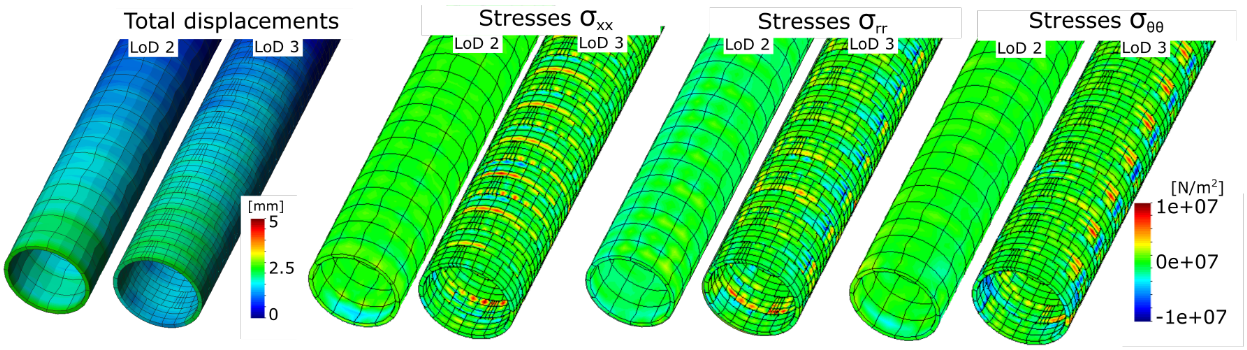


Fig. 12. Comparison of the numerical models for the lining structure on LoD 2 and LoD 3 in terms of total displacements and total stresses in longitudinal σ_{xx}, radial σ_{rr} and tangential $\sigma_{\theta\theta}$ direction.

Lining LoD / Soil model	1	2	3 sw (rw)
LE	12	27	178 (41)
MC	22	31	201 (57)
CASM	54	73	240 (79)

Table 2. Time [min] needed for calculation of different LoDs w.r.t. the material model and the representation of the lining

4.1.2. Effect of the LoD w.r.t the material model

Next, the influence of the material model is evaluated. In general, the size of the model is unchanged here, however, consideration of the non-linear soil behaviour and complexity of the soil constitutive law increases the computation time. Considering the representation of the lining with LoD 3 and adopting three different material models (LE, MC and CASM) leads to the settlement trough changes as depicted in Figure 13. Table 2 summarizes the total calculation

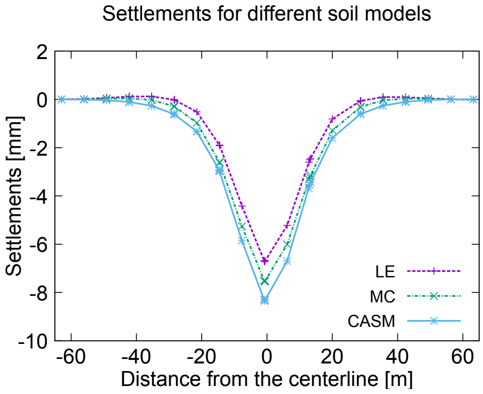


Fig. 13. Tunnelling-induced surface settlements trough for three different material models for the soil (LE, MC and CASM). The lining is modelled as a segmented structure with interfaces between segments and segment-wise installation (LoD 3).

time for all previously mentioned combinations of the lining model and the material model of the soil. The third column gives the total time of the simulation run for both segment-wise (sw) and ring-wise (rw) installation of the segmental lining ring. All analyses were executed on a shared memory system with a direct solver (here we employ the PARDISO solver) using 32 processes on a single computer, employing an Intel Xeon processor with 64 cores, each core running at 2.67 GHz, and 256 GB of RAM. From this table, it becomes evident, that increasing the accuracy of geometric or material modelling increases the computational cost. However, the optimal setup of the computational model depends on the objective of the analysis. As shown in Figure 11 for the analysis of the surface settlements, the model with the highest complexity is not always necessary.

4.2. Real-time prediction

SATBIM is further applied for the generation of meta models for real-time predictions of tunnelling-induced settlements of a building. The tunnel is passing in the vicinity of an existing building as illustrated in Figure 14. The design and model parameters investigated in this study are the building LoDs 1, 2 and 3, the building distance from the tunnel in Y direction $(0, 10, 20, 30, 40, 50, 60\,\mathrm{m})$ and the depth of the tunnel crown w.r.t. the foundation of the existing building $(5, 10, 15, 20, 25\,\mathrm{m})$ as shown in Figure 15. For the remaining components, the levels of representation adopted for this model are: Soil model on LoD 2, lining model on LoD 1 and the TBM model on LoD 1. The tunnel diameter D in this example is 10 m, and the construction of 27 rings (steps) of 2.5 m length is analysed. In this example, the Mohr Coulomb model is used as the constitutive relation for the soil. This elasto-plastic constitutive model is assumed associative with linear hardening. The SATBIM platform described in Section 3.1 is used for the generation of 105 tunnelling information models resulting from the combination of those three parameters (see Figure 14 (top)). Based on each of these Revit models, a simulation model is generated using `SatBimModeller`. The tunnelling-induced displacements for the corresponding information models are shown in Figure 14 (bottom). The generation of the design model in Revit and the corresponding FE simulation model took in average 16 seconds and 42 seconds, respectively.

In this numerical experiment, which contains 105 simulations for the construction of 25 tunnel rings with a combination of three parameters (building LoD, overburden and distance), the selected monitoring quantities are the vertical displacements of the building top. This resulted in data sets of 2554 monitoring samples. This data set is used for the generation of a meta model for the prediction of building settlement w.r.t building LoD, overburden and distance. In the next step, this meta model is used for real-time predictions and forward calculations for sensitivity analyses.

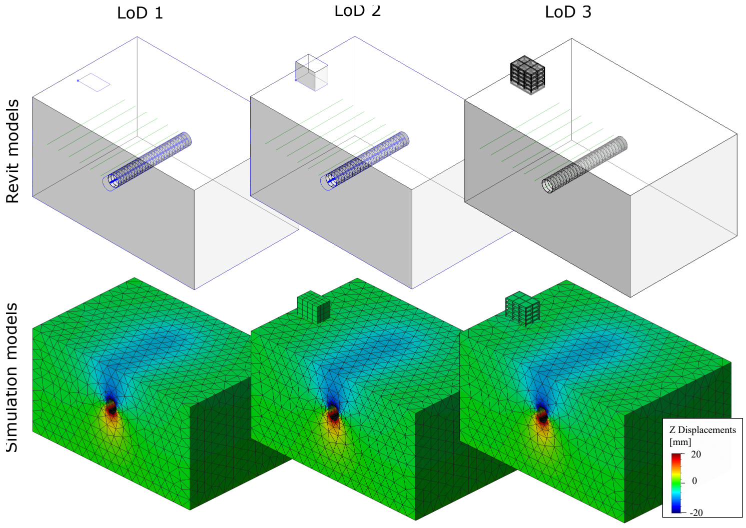


Fig. 14. Top: BIM model for investigation of building LoD sensitivity; bottom: simulation models for the respective BIM models.

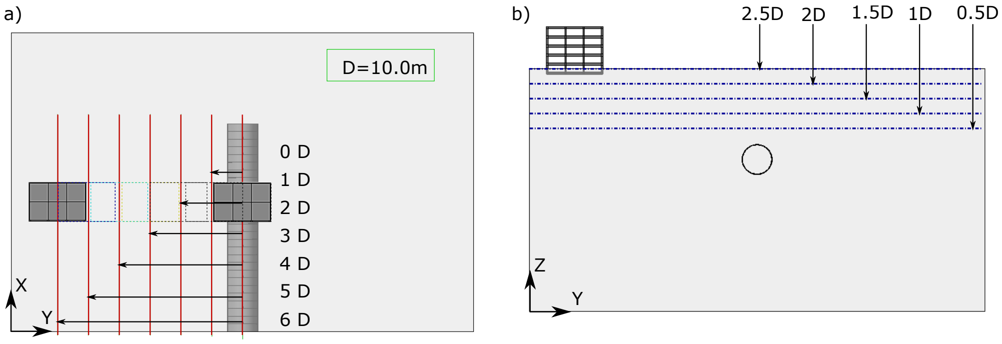


Fig. 15. Parameters for generation of the meta model: a) design alternatives in terms of distance of the tunnel alignment from a building; b) overburden of tunnel w.r.t. foundation of the existing building.

Meta model training for the prediction of building settlements. The procedure and the algorithm described in Figure 9 is applied for the training of the meta model based on the data set of 2554 samples. Here, the data set is randomly divided into portions of 80%, 15% and 5% of the samples for training, testing and validation, respectively. Figure 16a shows the relative Root Main Square Error (rRMSE) for different machine learning techniques applied for the data training. rRMSE is defined as

$$r\text{RMSE} = \sqrt{\frac{1}{p_n}\sum_{1}^{p_n}\left(\frac{y_k - t_k}{t_k}\right)^2}, \tag{15}$$

where y_k is an output of the meta model and t_k is the target value for this output. From Figure 16a, it can be seen that the optimised ANN shows the best performance for the data set training. The Particle Swarm Optimisation algorithm is used to determine the optimum number of neurons in the hidden layers. The optimal solution is reached within less than 20 iterations.

Figure 16b shows the comparison between the data set and meta model predictions for the training, testing and validation set for the best meta model. From this figure, it can be concluded that the optimised ANN meta model has excellent prediction capabilities with an error for both the test and the validation set of less than 3%.

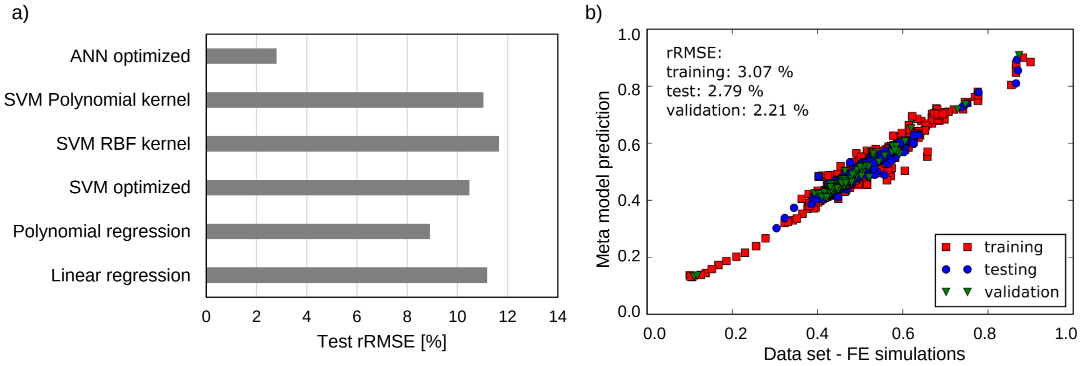


Fig. 16. a) Performance of the training of the meta model using different machine learning techniques; b) comparison between vertical displacements of the building obtained from the FE simulation and the predictions of the trained meta model for the training, test and validation sets for the ANN meta model .

Meta models for real-time design assessment. Having established a robust meta model, we can now apply it for fast predictions of the effect of design variants directly within Revit. In order to do that, we developed a Dynamo node to calculate the forward pass of the meta model based on model parameters that are set in Dynamo. The trained meta model is imported as a text file, while parameters are set using value boxes or sliders as shown in Figure 17a. The comparison of the settlements predicted using simulation-trained meta models and the results obtained from numerical simulations for arbitrarily selected parameters shows excellent agreement (see Figure 17b). The meta model prediction is obtained almost instantaneously (in less than 1 second for the prediction and 5-20 seconds for the regeneration of the model based on new design parameters and visualisation in Revit), while one finite element simulation in this example takes $\approx 3\,\text{h}$ (on the hardware described above). Therefore, the application of meta models for prediction of tunnelling-induced effects is an alternative to provide an efficient forward model.

Application of meta models for sensitivity analysis. In order to assess the importance and need for adopting a certain LoD of the component for the analysis of the soil-structure interaction problem, sensitivity analysis can be conducted. Here, we can use already created meta models for near-instant forward calculation of the tunnelling-induced effects on the building to evaluate the sensitivity between the input parameters and the output, and to estimate its importance. A variance-based global sensitivity analysis [33] has been conducted in order to measure the sensitivity of the model output (settlements at the building top) w.r.t. the aforementioned input parameters. In this methodology, the importance of the input parameter is quantified through the absolute mean μ^* of the elementary effect EE_i (16), representing the

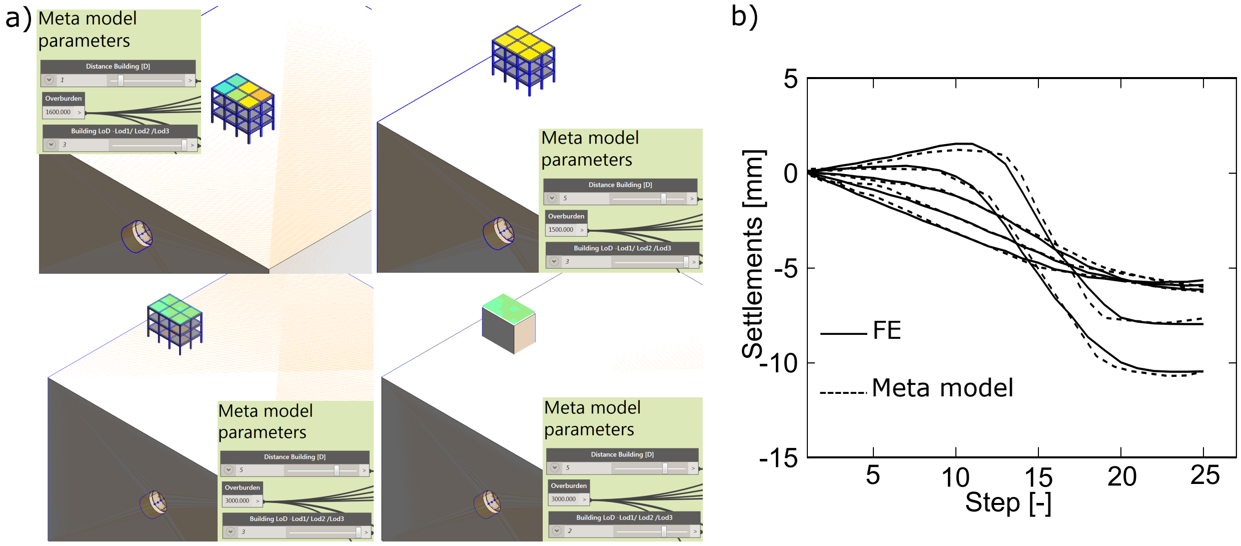


Fig. 17. a) Real-time assessment of soil-structure interaction w.r.t. tunnel alignment, overburden and building LoD in BIM tool Revit; b) comparison of the temporal building settlements predicted by the FE simulation and the trained meta model.

total sensitivity index and a measure of the overall effect of a factor on the output. The elementary effect EE_j^i of the jth parameter X_j in the ith repetition is calculated using Equation 16, where $Y\left(X_j\right)$ is an output and Δ is a predetermined multiple of $1/(p-1)$, with p denoting the number of intervals of X_j and $X_j \leq 1 - \Delta$ [33].

$$EE_j^i = \frac{Y\left(X_1, X_2, ..., X_j + \Delta, .., X_k\right) - Y\left(X_1, X_2, ..., X_j, .., X_k\right)}{\Delta}, \qquad \text{and} \qquad \mu_j^* = \frac{1}{n}\sum_{i=1}^{n}\left|EE_j^i\right| \tag{16}$$

For the forward calculation of the elementary effect EE_i of a certain parameter variation on the output, the previously trained meta model is applied. This offers significant advantages, since the calculation of the elementary effect requires a large number of forward calculations to ensure the uniqueness of the solution [31]. Figure 18 shows the sensitivity

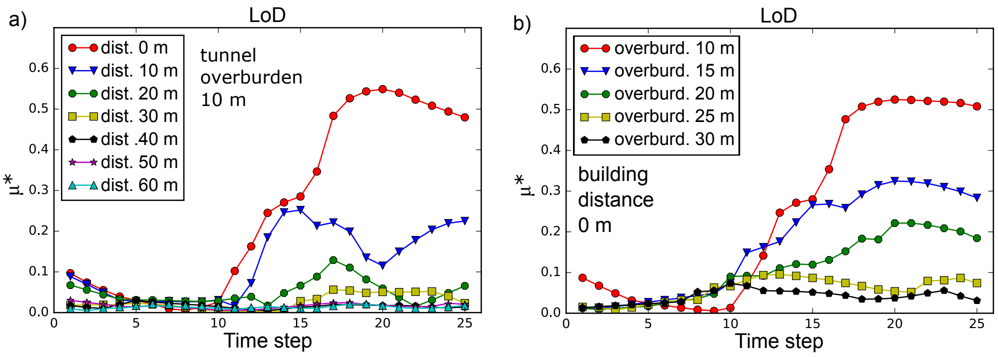


Fig. 18. Sensitivity of the building LoD expressed in terms of the absolute mean of the elementary effect EE_i in regard to a) assumptions regarding the distance of the alignment to the building for a tunnel overburden of 10 m; b) different levels of tunnel overburden for a building distance of 0 m.

measure μ^* for the selected LoD of the building w.r.t. the vertical displacements for different assumptions regarding the distance of the alignment to the building and for different levels of the overburden. From the plots, we can conclude that global sensitivity of the LoD, indicated by μ^*, drops with increasing distance of the building and overburden of the tunnel. It is, for instance, obvious that when the distance of the building from the tunnel is approximately $4D$, the selected building LoD becomes irrelevant. This allows to conclude that for buildings located at a distance of more than $4D$ from the tunnel alignment, the lowest LoD is fully sufficient to reduce computational costs and efforts for model generation.

4.3. Parallelisation of a large model

In this numerical example, the SATBIM framework is used for the efficient generation of a large tunnel section combining components on different LoDs. The simulation efficiency in this example is further increased by employing a parallelisation strategy to perform large-scale simulations of the advancement process in mechanised tunnelling. Finally, the knowledge about the optimal building LoD, for the scenario of shield tunnelling in vicinity of existing infrastructure from the previous example (see Figure 18), is applied to further optimise the size of the model, preserving the accuracy of the solution. In a first simulation all buildings included in the BIM model of the investigated tunnel section are modelled with the highest LoD (see Figure 19a), while in a second numerical analysis, only buildings having a high sensitivity w.r.t. the LoD are modelled with high accuracy, while the rest is modelled with LoD 2, which significantly reduces the size of the problem (see Figure 19b).

In both models, LoD 2 is selected for the representation of the tunnel lining structure and the TBM. This model accounts for the shield as a deformable body moving through the soil and interacting with the soil through surface-to-surface contact. The tunnel advance is modelled by means of de-activation of soil elements and installation of the lining rings and grouting elements. Tunnelling-induced deformations are controlled by applying the face support pressure and the grouting pressure at the tunnel face and in the steering gap, respectively. The elasto-plastic Mohr Coulomb model with associative flow rule is used as the constitutive relation between effective stresses and the strains in the fully saturated soil. The groundwater level is assumed at the surface. Selecting the optimal LoD for the buildings, the model size has been reduced by 17% in terms of number of d.o.f, while keeping the accuracy of the numerical solution, as shown in Figure 19a and b. Therefore, for further improvement of the efficiency by means of parallelisation of the simulation model, only the model with the optimised building LoDs is considered.

Type of simulation	Average solution time (s)	Average number of iterations
GMRES (8 processes)	900.25	398.3
GMRES (16 processes)	652.08	385
GMRES (32 processes)	576.51	378.5
GMRES (64 processes)	277.31	377.5

Table 3. Performance of parallel solver in terms of average solution time per advancement step and average number of linear solver iterations.

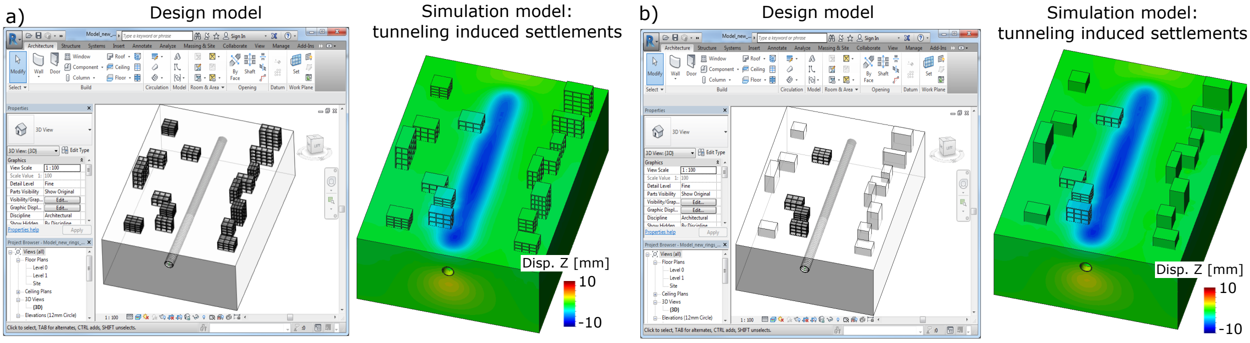


Fig. 19. Revit model of a more than 200 m long tunnel section used for generation of a large-scale simulation model and visualisation of the settlement trough obtained from the numerical simulation generated using `SatBimModeller` for a) highest LoD for representation of the infrastructure and b) optimised LoD for representation of the infrastructure.

In the spatial discretisation of the tunnel section, the buildings are discretised by means of 27-node hexahedral elements. For the soil Taylor-Hood tetrahedral elements (i.e. 10-node elements for the displacements and 4-node elements for the fluid pressure) are used. This amounts to 9410 elements for the buildings and 194,186 elements for the soil. The maximum number of d.o.fs during advancement is 1,130,169. The results of the computation, in terms of speed-up and number of average iterations of the parallel solver, are presented in Table 3 and Figure 20.

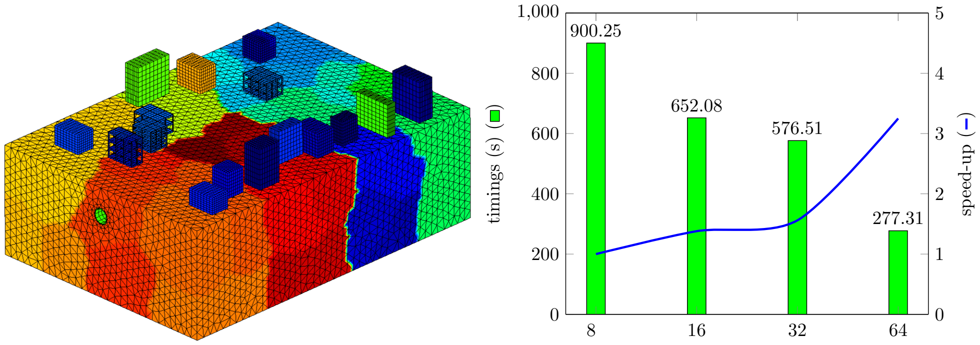


Fig. 20. Left: Decomposition of the model using 64 partitions; Right: Speed-up of parallel solver using 8 to 64 processors.

Using the parallel solver described in Subsection 3.2.2, in particular, the Block Gauss-Seidel preconditioner coupled with the multigrid preconditioner for the solid block and a direct solver to invert the pressure block, the simulation is performed with eight processes first. Subsequently, the speed-up from the computations using a larger number of processes is measured relative to the simulation using eight processes. The simulation of the considered project section takes approximately 15 min for a linear solving step. This reduces to 4.5 min using 64 processors in parallel, which corresponds to a speed-up of approximately 3.3. The fact that the measured speed-up is considerably below the theoretical level (which is 8) is attributed to the overhead for synchronisation of data across processes and the cost for the preconditioner. This can be improved by choosing a better approximation for the Schur complement matrix [47, 51], or using an efficient block preconditioning strategy accounting for contact and non-conforming mesh tying [16]. Nevertheless, this improvement factor represents an essential ingredient for the increase of the overall efficiency of the integrated BIM-supported simulation strategy proposed in this paper.

5. Conclusions

In current engineering practice of tunnel design assessment, significant manual, time-consuming preparation and analysis as well as a significant amount of computing resources are required. To ensure a seamless workflow during the design and the analysis, and to enable efficient investigations of design alternatives and their numerical assessment, a BIM-based multi-level numerical simulation concept was proposed. It enables choosing different levels of detail in regard to the representation of the involved components. The model described in this paper is tightly coupled with a parallelised process-oriented simulation model (ekate) to enable large-scale simulations using standard hardware resources, which take all construction stages and relevant components, including the existing infrastructure in urban tunnelling, into account. In case of simulation-based prognoses required during construction, the framework enables near-instant predictions by means of simulation-based meta models used for forward analysis. The efficiency of the generation of the training set and the actual training of the meta models is increased by applying the BIM-based numerical modelling and training algorithms implemented within the framework SATBIM.

The proposed approach enables efficient numerical assessment of different design alternatives reducing both user and computational workload. The user workload is reduced by different aspects of the SATBIM framework, including:

- User-friendly design of the software, where the user assigns the design parameters in the visual environment of Revit/Dynamo, while the rest of the workflow is automated. The complete model is parametrised and dependencies between different parameters are already incorporated in the model, avoiding spatial conflicts. Therefore, the time needed to learn how to use software is minimised.
- The software structure is highly modular, and therefore flexible for adaptation or extensions. For example, an additional component or an additional LoD of an existing component can be added without affecting the existing parts of the software.
- By adopting BIM-based numerical modelling, the geometry and semantics describing the problem can be directly imported from the design tools. Therefore, the user does not need to create and maintain separate design and numerical models. Changes in the design models are directly transferred to numerical models.
- `SatBimModeller` enables the automatic generation of numerical models based on BIM, as well as the execution of simulations and storing the results of the analysis in a form suitable for further processing. The SATBIM approach significantly reduces the user workload for the generation of the model. This is particularly important

if different design alternatives have to be investigated, or a large number of models has to be generated for parametric studies.

In some cases, the simulation of large tunnel sections is required, for example, for the determination of the general tunnel routing of a project or in case of urban tunnelling, where the existing infrastructure needs to be considered. This consequently leads to large numerical models. The SATBIM framework offers three measures which can be applied to minimise the computational workload:

- Adopting the multi-level approach and enabling the selection of optimal LoDs of different components based on the objective of the analysis. This reduces the model size, and consequently the computational costs.
- Applying parallelisation strategies to perform large-scale simulations to decrease the computation time. The parallelisation strategy used in this paper obtains good speed-up and small memory requirements as compared to conventional parallel direct solvers, and exhibits strong scalability characteristic.
- Applying simulation-based meta models for forward analyses if real-time predictions are required. This provides a tool for instant prediction of tunnelling-induced effects within a state-of-the-art design environment, which can be used to support design, collaboration and decision making in the planning and construction phase.

The SATBIM framework offers a number of solutions for reducing the total user and computer workload for the assessment of tunnel design. However, there is room for extensions of this prototype concept. The efficiency of this framework could be further improved by integrating re-meshing tools directly into the BIM design environment, and implementing a direct interface between the design tool and the simulation software ekate, avoiding the need for GiD as a separate step.

The SATBIM toolkit will be made available as open source software together with tutorials, a complete manual, and a number of benchmark examples.

The project's Github repository (not yet public) can be found at https://github.com/satbim

Acknowledgements

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under the Marie Sklodowska-Curie grant agreement No 702874 "SATBIM — Simulations for multi-level Analysis of interactions in Tunnelling based on the Building Information Modelling technology". Financial support to the second author is provided by the German Research Foundation (DFG) in the framework of project C1 of the Collaborative Research Center SFB 837 "Interaction Modeling in Mechanized Tunnelling". This support is gratefully acknowledged.

References

[1] AIA (2013). “Aia contract document g202-2013, building information modeling protocol form..” *Report no.*, American Institute of Architects, Washington DC, USA.

[2] Alsahly, A., Stascheit, J., and Meschke, G. (2016). “Advanced finite element modeling of excavation and advancement processes in mechanized tunneling.” *Advances in Engineering Software*, 100, 198 – 214.

[3] AUTODESK (2017). *Autodesk Revit*, <http://www.autodesk.co.uk/products/revit-family/>. http://www.autodesk.co.uk/products/revit-family/.

[4] Balay, S., Abhyankar, S., Adams, M., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Eijkhout, V., Gropp, W., Kaushik, D., Knepley, M., McInnes, L., Rupp, K., Smith, B., Zampini, S., Zhang, H., and Zhang, H. (2016). “PETSc users manual.” *Report no.*, Argonne National Laboratory.

[5] Balay, S., Gropp, W. D., McInnes, L. C., and Smith, B. F. (1997). *Efficient Management of Parallelism in Object-Oriented Numerical Software Libraries*. Birkhäuser Boston, Boston, MA, 163–202.

[6] Benzi, M. (2002). “Preconditioning techniques for large linear systems: A survey.” *Journal of Computational Physics*, 182(2), 418 – 477.

[7] Benzi, M., Golub, G. H., and Liesen, J. (2005). “Numerical solution of saddle point problems.” *Acta Numerica*, 14, 1–137.

[8] BIMforum (2013). “Level of development specification guide., <https://bimforum.org/wp-content/uploads/2017/11/LOD-Spec-2017-Guide_2017-11-06-1.pdf>. Las.

[9] Borrmann, A., Flurl, M., Jubierre, J. R., Mundani, R.-P., and Rank, E. (2014). “Synchronous collaborative tunnel design based on consistency-preserving multi-scale models.” *Advanced Engineering Informatics*, 28(4), 499 – 517.

[10] Borrmann, A., Kolbe, T., Donaubauer, A., Steuer, H., Jubierre, J., and Flurl, M. (2015). “Multi-scale geometric-semantic modeling of shield tunnels for gis and bim applications.” *Computer-Aided Civil and Infrastructure Engineering*, 30(4), 263–281.

[11] Bui, H. G. and Meschke, G. (2017). “A parallelization strategy for hydro-mechanically coupled mechanized tunnelling simulation.” *Computer and Geotechnics* submitted.

[12] Chen, X., Jie, Y., and Yu, Y. (2013). “Gpu-accelerated iterative solutions for finite element analysis of soil–structure interaction problems.” *Computational Geosciences*, 17(4), 723–738.

[13] Dadvand, P., Rossi, R., and Oñate, E. (2010). “An object-oriented environment for developing finite element codes for multi-disciplinary applications.” *Archives of Computational Methods in Engineering*, 17, 253–297.

[14] Do, N., Dias, D., Oreste, P., and Djeran-Maigre, I. (2014). “Three-dimensional numerical simulation for mechanized tunnelling in soft ground: the influence of the joint pattern.” *Acta Geotechnica*, 9(4), 673–694.

[15] Falgout, R. and Yang, U. (2002). “hypre: a library of high performance preconditioners.” *Computational Science – ICCS 2002. Lecture Notes in Computer Science*, 632 – 641.

[16] Ferronato, M., Janna, C., and Gambolati, G. (2008). “Mixed constraint preconditioning in computational contact mechanics.” *Computer Methods in Applied Mechanics and Engineering*, 197(45), 3922 – 3931.

[17] Founta, V., J.Ninic, Whittle, A. J., Meschke, G., and Stascheit, J. (2013). “Numerical simulation of ground movements due to epb tunnelling in clay.” *EURO:TUN 2013, 3rd International Conference on Computational Methods in Tunneling and Subsurface Engineering*, Ruhr University Bochum, 97–108 (April).

[18] Hegemann, F., Manickam, P., Lehner, K., Koch, C., and König, M. (2013). “Hybrid ground data model for interacting simulations in mechanized tunneling.” *Journal of Computing in Civil Engineering*, 27(6), 708–718.

[19] Karypis, G. and Kumar, V. (1998). “A fast and high quality multilevel scheme for partitioning irregular graphs.” *SIAM J. Sci. Comput.*, 20(1), 359–392.

[20] Kasper, T. and Meschke, G. (2004). “A 3D finite element model for TBM tunneling in soft ground.” *International Journal for Numerical and Analytical Methods in Geomechanics*, 28, 1441–1460.

[21] Kennedy, J. and Eberhart, R. C. (1995). “Particle swarm optimization.” *Proceedings of the IEEE International Conference on Neural Networks*, I. Press, ed., Piscataway, NJ, USA, 1942 – 1948 (November-December).

[22] Koch, C., Vonthron, A., and König, M. (2017). "A tunnel information modelling framework to support management, simulations and visualisations in mechanised tunnelling projects." *Automation in Construction*, 83, 78 – 90.

[23] Kolymbas, D. (1998). *Geotechnik - Tunnelbau und Tunnelmechanik*. Springer.

[24] Komiya, K. (2009). "Fe modelling of excavation and operation of a shield tunnelling machine. fe-modellierung des ausbruchs und betriebs einer schildvortriebmaschine." *Geomechanik und Tunnelbau*, 2(2), 199–208.

[25] Koopmann, J. (2008). "What is your definition of database workload?." *Database Journal*.

[26] Lambrughi, A., Rodríguez, L. M., and Castellanza, R. (2012). "Development and validation of a 3d numerical model for tbm–epb mechanised excavations." *Computers and Geotechnics*, 40, 97 – 113.

[27] Lee, P.-C., Wang, Y., Lo, T.-P., and Long, D. (2018). "An integrated system framework of building information modelling and geographical information system for utility tunnel maintenance management." *Tunnelling and Underground Space Technology*, 79, 263 – 273.

[28] Mayer, P.-M., Frodl, S., and Hegemann, F. (2016). "Bim as a process in tunnelling / bim als prozess im tunnelbau." *Geomechanics and Tunnelling*, 9(6), 684–695.

[29] Melendo, A., Coll, A., Pasenau, M., Escolano, E., and Monros, A. (2016). "www.gidhome.com, <http://www.gidhome.com/>.

[30] Meschke, G. (1996). "Consideration of aging of shotcrete in the context of a 3D viscoplastic material model." *International Journal for Numerical Methods in Engineering*, 39, 3123–3143.

[31] Miro, S., Hartmann, D., and Schanz, T. (2014). "Global sensitivity analysis for subsoil parameter estimation in mechanized tunneling." *Computers and Geotechnics*, 56, 80–88.

[32] Montgomery, D., Peck, E., and Vining, G. (2012). *Introduction to Linear Regression Analysis*. John Wiley & Sons, 5th edition edition.

[33] Morris, M. (1991). "Factorial sampling plans for preliminary computational experiments." *Technometrics*, 33, 161 – 174.

[34] N. Yabuki, T. Aruga, H. F. (2013). "Development and application of a product model for shield tunnels." *Proceedings of the 30th International Symposium on Automation and Robotics in Construction*, Montreal.

[35] Nagel, F. and Meschke, G. (2010). "An elasto-plastic three phase model for partially saturated soil for the finite element simulation of compressed air support in tunnelling." *International Journal for Numerical and Analytical Methods in Geomechanics*, 34, 605–625 doi:10.1002/nag.828.

[36] Ninić, J., Koch, C., and Stascheit, J. (2017). "An integrated platform for design and numerical analysis of shield tunnelling processes on different levels of detail." *Advances in Engineering Software*.

[37] Ninić, J. and Meschke, G. (2015). "Model update and real-time steering of tunnel boring machines using simulation-based meta models." *Tunnelling and Underground Space Technology*, 45, 138 – 152 Online.

[38] Ochmański, M., Modoni, G., and Bzówka, J. (2018). "Automated numerical modelling for the control of epb technology." *Tunnelling and Underground Space Technology*, 75, 117 – 128.

[39] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). "Scikit-learn: Machine learning in Python." *Journal of Machine Learning Research*, 12, 2825–2830.

[40] Rahm, T., Scheffer, M., Thewes, M., König, M., and Duhme, R. (2016). "Evaluation of disturbances in mechanized tunneling using process simulation." *Computer-Aided Civil and Infrastructure Engineering*, 31(3), 176–192.

[41] Rüde, U. (2017). "Lean and mean: Finite elements solvers beyond a trillion degrees of freedom." *Domain Decomposition Conference*, number 10.13140/RG.2.2.13824.23049, Longyearbyen, Norway (Februar).

[42] Saad, Y. (2000). *Iterative Methods for Sparse Linear Systems* (3).

[43] Schillinger, D. and Ruess, M. (2015). "The Finite Cell Method: A Review in the Context of Higher-Order Structural Analysis of CAD and Image-Based Geometric Models." *Archives of Computational Methods in Engineering*, 22(3), 391–455 00000.

[44] Schindler, S. and Mark, P. (2013). "Evaluation of building stiffness in the risk-assessment of structures affected by settlements.." *Proc. 3rd Int. Conf. on Comp. Meth. in Tunneling and Subsurface Engineering - EURO:TUN 2013, RUB*, Bochum, Germany, 477–486.

[45] Stascheit, J., Ninić, J., Meschke, G., Hegemann, F., and Maidl, U. (2018). "Building information modelling in mechanised shield tunnelling

– a practitioner's outlook to the near future." *Geomechanics and Tunnelling*, 11(1), 34–49.

[46] Swoboda, G. and Abu-Krisha, A. (1999). "Three-dimensional numerical modelling for tbm tunnelling in consolidated clay." *Tunnelling and Underground Space Technology*, 14(3), 327 – 333.

[47] Toh, K.-C., Phoon, K.-K., and Chan, S.-H. (2004). "Block preconditioners for symmetric indefinite linear systems." *International Journal for Numerical Methods in Engineering*, 60(8), 1361–1381.

[48] Toll, D., Zhu, H., Osman, A., Coombs, W., Li, X., and (eds.), M. R. (2014). "Information technology in geo-engineering." *2nd International Conference on Information Technology in Geo-Engineering*.

[49] Vapnik, V. (2000). *The nature of statistical learning theory*. Springer-Verlag.

[50] Vesic, A. B. (1963). "Beams on elastic subgrade and the winkler's hypothesis." *Proceedings of 5th International Conference of Soil Mechanics*, 845 – 850.

[51] White, J. A. and Borja, R. I. (2011). "Block-preconditioned newton–krylov solvers for fully coupled flow and geomechanics." *Computational Geosciences*, 1–13.

[52] Yu, H. (1998). "CASM: a unified state parameter model for clay and sand." *International Journal for Numerical and Analytical Methods in Geomechanics*, 48, 773–778.

[53] Zienkiewicz, O. (2004). "The birth of the finite element method and of computational mechanics." *International Journal for Numerical Methods in Engineering*, 60, 3–10.

[54] Zienkiewicz, O. and Taylor, R. (2000). *The Finite Element Method*, Vol. 1. Butterworth Heinemann, Oxford, Fifth edition.

[55] Zobl, F. and Marschallinger, R. (2008). "Subsurface geo building information modelling geobim." *GEO Informatics*, 11.