Simple Procedures to Accelerate Analyses of Large-Scale Models for Sequential Tunnel Excavation

M. Noronha[1], Ch. Duenser[2], G. Beer[2]

Summary

Realistic simulations of tunnel excavation normally require large-scale models to represent the complex geometries. In these simulations, a major problem is that models undergo a large number of changes in order to represent the excavation of the ground and the application of the support and strengthening means. Verification of available computational codes shows that they do not have specific procedures to deal with this problem. In fact, they normally require a completely new analysis for every change, and the computational effort required for these simulations is proportional to the number of changes. Since this number tends to be high in practice, a complete simulation can take days even if high-speed computers are used. This work presents some simple procedures to speed up these simulations and which can be easily incorporated into existing software.

Introduction

Numerical simulations of sequential tunnel excavation require elaborate models and procedures to obtain accurate results. In these simulations, the Finite Element Method (FEM) and the Boundary Element Method (BEM) [1] have become well established tools which give results with good accuracy. For analyses of tunnelling problems the BEM is especially well suited because of the infinite or semi-infinite extent of the models domain. The required effort for mesh generation and computation is reduced by an order of magnitude because only the surface of the problem needs to be represented. This is a big advantage over methods like the FEM, where the volume of the problem has to be discretised and a truncation of the mesh is necessary for infinite domains. These are some of the reasons which make the BEM a preferable technique for excavation problems of tunnels or other underground constructions. However, a consideration of the sequential excavation and the data management which is involved in this procedure is not an easy task in all of the mentioned methods. Figure 1a shows how a tunnel excavation problem is discretised with the BEM, where the surface corresponding to the tunnel wall also represents the infinite region. A suitable number of finite regions represents the material to be excavated. Figure 1b depicts a particular case where an excavation of the tunnel in top heading and bench has been carried out.

[1] University of São Paulo, currently at Graz University of Technology (marcos.noronha@poli.usp.br)
[2] Institute for Structural Analysis, Graz University of Technology, Austria ({duenser, beer}@ifb.tu-graz.ac.at)

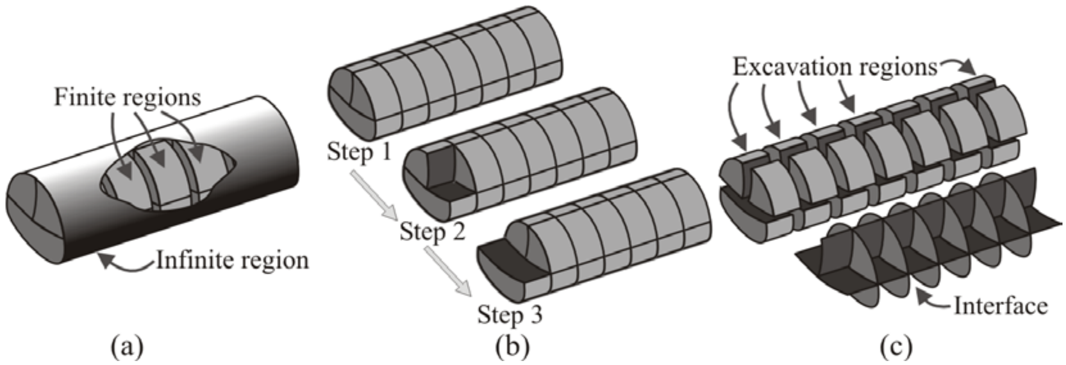

Figure 1: Tunnel excavation: (a) BEM discretisation; (b) sequential steps; (c) proposed geometric description.

This paper presents some simple procedures to speed up simulations of sequential excavation and which can be easily incorporated into existing software. Also, they are independent of the numerical formulation and can be applied to models based on FEM, BEM or coupled FEM/BEM. The proposed procedures are threefold. Firstly, they consider a geometric description that uses a suitable division of the model into regions, to represent the changing parts of the excavation, and a connecting interface that joins them together (Figure 1c). Secondly, instead of using a global matrix system, they use segmented storage for the system matrices. Finally, they accomplish the calculation in two steps. The first step consists in an initial evaluation of the coefficient matrices for all regions, which involves the most time-consuming operations. The second step consists in updating procedures of the system matrices when the model undergoes a change.

For the present application, multiple region strategies seems to be the natural choice. In the following sections the details of the Multi-Region BEM and of the key storage and calculation assets will be described.

Concept of the Multi-Region Boundary Element Method (MRBEM)

The MRBEM [1] [2] is based on the stiffness method. To explain the MRBEM, Figure 2 shows the cross section of a tunnel with a finite and an infinite region. In this model, nodes belonging to two or more regions are called interface nodes. Otherwise, they are called free nodes. In the present approach, a two-level calculation strategy has been implemented to solve the problem in terms of local (region) and global systems.

At local level, the region assembly involves two steps. The first one is the solution of the system with fixed interface nodes and given boundary conditions at the free nodes, as shown in Figure 2a. Free nodes are assigned with the given boundary conditions, which in this case are the tractions at the free surface. On the other hand, the interface nodes are assigned with zero displacements.

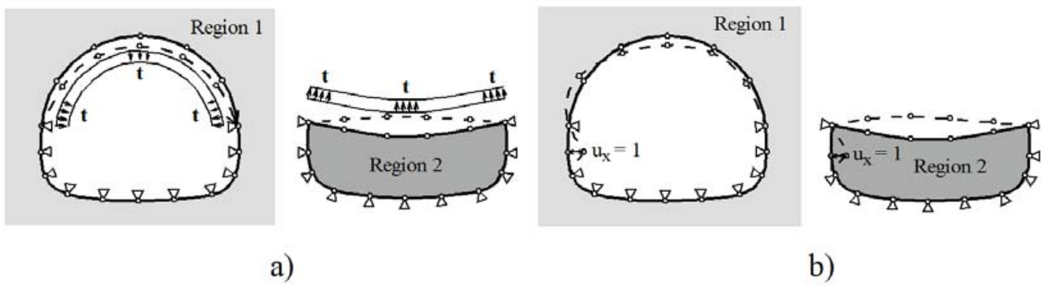

Figure 2: Multiple region problem: (a) fixed interface nodes; (b) unit displacements.

For each of the N regions of a model the following equation system can be written:

$$[\mathbf{B}]\begin{Bmatrix}\{\mathbf{t}\}_{c0}^{N}\\\{\mathbf{x}\}_{f0}^{N}\end{Bmatrix}=\{\mathbf{F}\}_{0}^{N}\ , \qquad (1)$$

where $[\mathbf{B}]$ is the assembled left hand side, $\{\mathbf{F}\}_0^N$ is the right hand side due to the boundary conditions, $\{\mathbf{t}\}_{c0}^N$ has the interface tractions and $\{\mathbf{x}\}_{f0}^N$ has the results at the free nodes. For the BEM, matrices $[\mathbf{B}]$ and $\{\mathbf{F}\}_o^N$ are formed out of **[ΔU]** or **[ΔT]** matrices [1].

The second step is the equation assembly for unit displacements at each of the *n* interface nodes in turn and zero tractions at the free nodes (Figure 2b), which gives

$$[\mathbf{B}]\begin{Bmatrix}\{\mathbf{t}\}_{cn}^{N}\\\{\mathbf{x}\}_{fn}^{N}\end{Bmatrix}=\{\mathbf{F}\}_{n}^{N} \qquad n=1,2.....n_c\ , \qquad (2)$$

where $\{\mathbf{t}\}_{cn}^N$ contains the interface tractions, $\{\mathbf{x}\}_{fn}^N$ the displacements at the free nodes and n_c is the number of interface degree of freedoms (DoFs).

Superposition of the solutions of both problems gives the interface tractions $\{\mathbf{t}\}_c^N$ and the free nodes results $\{\mathbf{x}\}_f^N$ in terms of displacements at the interface nodes $\{\mathbf{u}\}_c^N$ as:

$$\begin{Bmatrix}\{\mathbf{t}\}_{c}^{N}\\\{\mathbf{x}\}_{f}^{N}\end{Bmatrix}=\begin{Bmatrix}\{\mathbf{t}\}_{c0}^{N}\\\{\mathbf{x}\}_{f0}^{N}\end{Bmatrix}+\begin{bmatrix}\mathbf{K}^{N}\\\mathbf{A}^{N}\end{bmatrix}\{\mathbf{u}\}_{c}^{N}\ , \qquad (3)$$

where $\mathbf{K}^N$ is the stiffness matrix related to the interface nodes of region N and $\mathbf{A}^N$ is a transformation matrix related to the free nodes.

The next task consists in assembling all $\mathbf{K}^N$ matrices into the global system by application of compatibility and equilibrium conditions at the interface, which results in

$$[\mathbf{K}]\{\mathbf{u}\}_c=\{\mathbf{F}\}\ , \qquad (4)$$

where $[\mathbf{K}]$ is the global stiffness matrix and $\{\mathbf{F}\}$ is the assembled right hand side. The system is solved for the unknown $\{\mathbf{u}\}_c$ at the interface nodes of the model.

Segmented storage of the System Matrices

Together with the MRBEM, the segmented storage is another important aspect to speed up simulations of sequential excavation. The idea is to calculate most time-consuming operations just once and to update the results for each step of the analysis.

In the model herein proposed for excavation problems, the first task consists in the calculation and storage of matrices **[ΔU]** and **[ΔT]** for each model region. These matrices do not change during the whole analysis. The next task consists in performing **[B]** matrix assembly and inversion for the boundary conditions established for the first analysis step. The inverse $\mathbf{[B]}^{-1}$ matrix is then stored in memory and from this it is possible to obtain matrices $\mathbf{[K]}^{N}$, $\mathbf{[A]}^{N}$ and **[K]** in a straightforward manner. The storage of $\mathbf{[B]}^{-1}$ matrix is also helpful for the non-linear calculations. For regions where the boundary conditions are changing, a new calculation of the $\mathbf{[B]}^{-1}$ matrix is necessary. The last section of this paper presents a low-cost procedure to accomplish this calculation.

The segmented storage of matrices **[ΔU]**, **[ΔT]**, and $\mathbf{[B]}^{-1}$ provides an adequate basis for the calculation of the global system. In the proposed implementation, the model interface carries out the managing tasks of matrix assembly at both local and global level.

Model changes

In simulations of tunnel excavation the model undergoes several changes, which correspond to the application of load/displacement values at the free nodes or to the removal/inclusion of a region. Figure 3 illustrates the case of a load application for the simulation of tunnelling with compressed air .

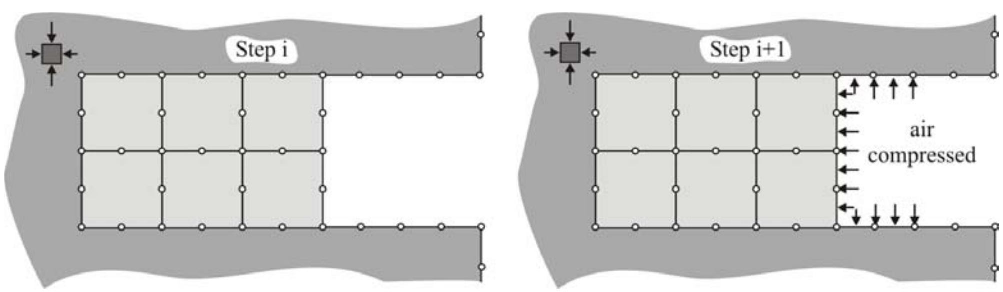

Figure 3: Changes due to a load application at the model free nodes.

The calculation using the MRBEM considers displacement boundary conditions at the nodes of a region connected to the model interface. Therefore, removal of some regions requires two major changes into the model. First, it is necessary a modification of the connecting interface, removing or deactivating its newly exposed parts. As a consequence, the corresponding parts of the remaining regions shift their boundary condition type from displacement to traction. Figure 4 illustrates this situation.

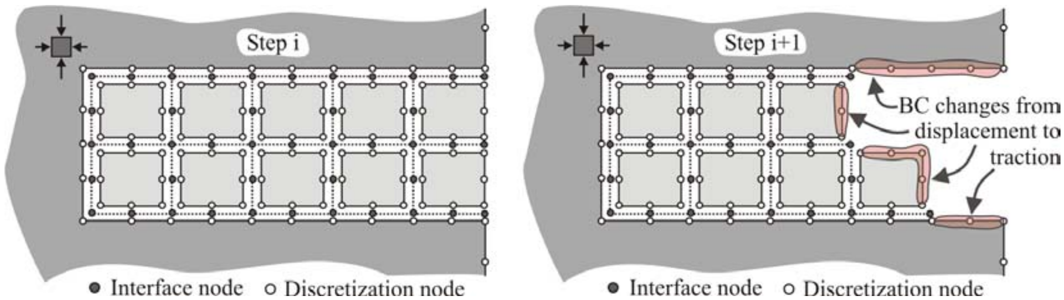

Figure 4: Changes due to the removal of some model regions.

Updating the System Matrices

According to the modifications introduced into a region during a simulation, two different updating processes have to be considered. The first and simplest case occurs when only the values of the boundary conditions change. Due to the segmented storage, this is a straightforward task with no need of matrix inversion processes. It only requires simple substitutions of the new values in the equations of the MRBEM.

The second case handles the changes in boundary condition type. In analysis with BEMs, the **[B]** matrix typical changes involve some column permutations with **[ΔU]** or **[ΔT]**. For the new inversion of the $n \times n$ matrix **[B]**, the computational effort is of order n^3. For this case, the Sherman-Morrison-Woodbury (SMW) method requires a much lower effort. The changes between two simulation steps can be represented in the form

$$[\mathbf{B}]_{i+1} = [\mathbf{B}]_i + [\mathbf{\Delta S}] \cdot [\mathbf{C}]^T, \tag{5}$$

where **[ΔS]** is a $n \times c$ matrix formed by some columns out of **[ΔU]** or **[ΔT]**, c is the number of exchanged columns and **[C]** is a $n \times c$ mapping matrix for the new columns.

The SMW method offers a low-cost alternative to calculate the inverse matrix $([\mathbf{B}]+ [\mathbf{\Delta S}].\, [\mathbf{C}]^{T})^{-1}$, which can be found by the formula

$$\left([\mathbf{B}]+[\mathbf{\Delta S}]\cdot[\mathbf{C}]^T\right)^{-1} = [\mathbf{B}]^{-1} - [\mathbf{B}]^{-1}\cdot[\mathbf{\Delta S}]\cdot\left([\mathbf{I}]+[\mathbf{C}]^T\cdot[\mathbf{B}]^{-1}\cdot[\mathbf{\Delta S}]\right)^{-1}\cdot[\mathbf{C}]^T\cdot[\mathbf{B}]^{-1}\,. \tag{6}$$

Since $[\mathbf{B}]^{-1}$ is in storage, the most time-consuming procedure of the SMW method is the inversion of a $c \times c$ matrix. For the simulations of sequential excavation models, usually $c < n$. Therefore, the computational effort required by the SWM method is much smaller ($c^3 << n^3$) than the application of the commonly used matrix inverse algorithms.

Figure 5 shows an example of the matrix $[\mathbf{B}]^{-1}$ updating process for a region with 372 DoFs. Due to the removal of an adjacent region, 52 DoFs undergo changes in their boundary condition type. While the usual inversion algorithms require a computational effort of order 372^3, the SMW method reduces the inversion process to order 52^3. In this case, the SMW algorithm is about 360 times faster than the conventional algorithms.

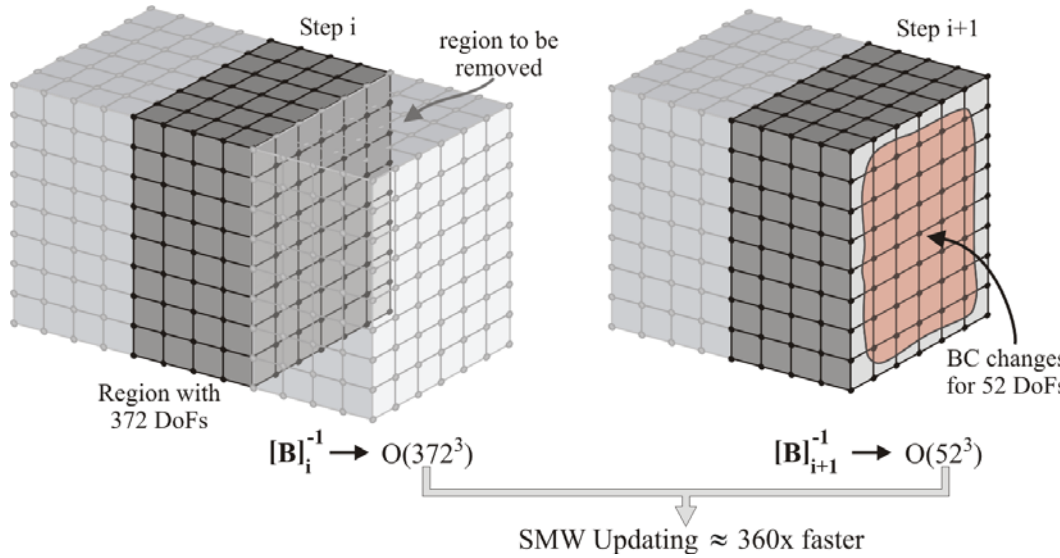

Figure 5: Application of the SMW method to a large-scale model.

Conclusion

The proposed procedures result in a greatly reduced computational effort for sequential excavation calculations, since most time-consuming operations are performed just once and the updating procedures are much faster than the commonly used strategies. Also, they can be easily incorporated into existing software, especially for object-oriented codes using distributed or parallel computation [4].

Acknowledgement

The work reported here is supported by the Austrian national science fund (FWF).

References

1 Beer G. (2001): *Programming the boundary element method*, Wiley.

2 Duenser Ch. (2001): *Simulation of sequential tunnel excavation with the Boundary Element Method*, PhD Thesis, Technical University of Graz.

3 Press, W., Teukolsky, S., Vetterling, W, Flannery, B. (2002): *Numerical Recipes in C++*, Cambridge University Press.

4 Beer, G., Duenser, Ch., Noronha, M. (2003): "Recent advances in the numerical simulation of tunnel excavation", ISRM 2003, South Africa.