Representing Settlement for Soft Ground Tunneling

by

Takahiro Aoyagi

B.E. in Civil Engineering
Nagasaki University, Japan
(1983)

M.E. in Civil Engineering
Kyushu University, Japan
(1985)

Submitted to the Department of Civil and Environmental Engineering in
Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE
in Civil and Environmental Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1995

© 1995 Takahiro Aoyagi
All Rights Reserved

The author hereby grants to MIT permission to reproduce and to distribute publicly
paper and electronic copies of this thesis document in whole or in part.

Signature of Author ______________________________
Department of Civil and Environmental Engineering
May 12, 1995

Certified by ______________________________
Dr. Herbert H. Einstein
Department of Civil and Environmental Engineering
Thesis Supervisor

Accepted by ______________________________
Dr. Joseph M. Sussman
Chairman, Departmental Committee on Graduate Studies

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

JUN 27 1995

LIBRARIES

Barker Eng

Representing Settlement for Soft Ground Tunneling

by

Takahiro Aoyagi

Submitted to the Department of Civil and Environmental Engineering on May 12, 1995 in partial fulfillment of the requirements for the Degree of Master of Science in Civil and Environmental Engineering

Abstract

The basic purpose of Decision Aids for Tunneling (DAT) is to deal with uncertain conditions in planning a tunnel project and to give effective information on construction cost and duration. Since the DAT have been developed for tunneling in hard rock, the program can not consider ground movement in soft ground tunneling. In this thesis, a program which allows one to predict ground movements in soft ground tunneling by using the DAT, was developed.

The factors influencing ground movement are associated with the geological conditions, tunnel dimensions, and construction methods. Among the many ground movements, this thesis focuses on the settlement profile. The empirical equation proposed by O'Reilly and New (1982) is used as a predictive model to relate the settlement profile to factors of ground movement. This settlement model is developed by associating the parameters of the equation, the volume of lost ground and the width of the settlement trough, with the ground conditions and construction procedures. The settlement model has been coded in the C programming language and incorporated in the existing DAT program which was developed at MIT.

It is possible to illustrate a settlement profile in a transverse section and in a longitudinal section on the graphic interface. Such predicted settlement profiles provide significant information in the selection of the construction methods and procedures affecting the cost and duration of tunnel projects.

Thesis Supervisor: Dr. Herbert H. Einstein
Title: Professor of Civil and Environmental Engineering

Acknowledgments

I would like to express my special thanks to all the people and friends who have encouraged me not only in fulfilling this research but also in getting through my life at MIT.

Above all, I would like to thank Professor Herbert Einstein for his valuable comments and suggestions despite his busy schedule. I highly appreciate his approachable nature which allowed frequent and friendly discussions to tale place any time of the day. I am also grateful to Mr. Vijaya Halabe, a Ph.D. student, who guided me through the SIMSUPER simulations and was always available for trouble shooting. I could not achieve this research without his continuous supports. Thanks to Mr. Albert Essiam among Geotech friends, I could have a sort of life in this nerd kingdom with him during the tough and not-so-tough days of grad life.

Not only financial support but also generous encouragement from KAJIMA Corporation has helped me complete my study at MIT, Furthermore, I would like to express appreciations the people at KAJIMA who encouraged and recommended me, in particular, Mr. Kenji Sakamoto, Mr. Susumu Ukai, and Mr. Tasuku Kubota. The alumni at MIT, Minoru Doi and Yasuaki Shimizu, gave me invaluable and practical information to survive at MIT.

Finally, I would like to thank my dedicated wife, Chikage, who inspired to study abroad and supported throughout our stay at MIT and our son, Iori, who always cheered and revitalized me to study hard. I was confused by my two year-old son's naive question “Do you really have to go to the class, Dad?” even when I go out somewhere.

May 1995,
Cambridge,
Massachusetts

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

One of the goals of tunnel designs is to plan a construction method by which the required work can be carried out in a certain amount of time at minimum cost. A decision strategy and cost estimating method are, thus, required in order to allow one to evaluate a project during the planning or early design phase. The controlling factors influencing the cost and duration of an underground project, such as geology, geometry, construction, labor productivity and equipment availability, vary for each job. Therefore, few general conclusions can be drawn regarding the "true" cost of a project.

Tunneling is affected by a variety of uncertainties. For example, the geological conditions at the tunnel level are largely unknown before construction; but even during construction the parameters that affect excavation and support are known only to a limited extent.

In the past decade, a wide variety of groups have been interested in developing an applicable "decision support tool" to predict construction cost and duration by considering the above-mentioned factors. Although a variety of modeling tools have been applied to analyze several tunnel projects, they have limitations in that these models need further processing of the construction data and the special knowledge of an engineer. In order to overcome these limitations, the Decision Aids for Tunneling (DAT) have been created at MIT to deal with the tunnel project with specified geologic conditions, tunnel dimensions, resource allocation, and construction methods. The DAT have been applied so far to various tunnel projects. However, the application of the DAT has been limited to rock tunneling.

Soft ground tunneling, unlike hard rock tunneling, usually produces ground movement. None of the existing modules of the DAT, so far, has considered the control of the ground movements despite the recently increasing number of tunnels in urban areas, which are mostly located in soft ground. In soft ground tunneling, forecasting the ground conditions and selecting construction procedures that will permit control of ground movements are critical. These procedures also affect the duration and the cost of the whole tunnel project. Therefore, there is a need for a program that can reflect the effects of ground movement.

As mentioned above, soft ground tunneling is associated with ground movements such as surface settlement and face stability. This study focuses on the surface settlement. First, empirical equations that have been developed for the settlement profile are examined. Among various empirical equations, the generalized equation of O'Reilly and New (1982) is selected as a predictive equation to relate the settlement profile to the controlling factors of ground movement. In this equation, there are two parameters defining a transverse profile: the volume of settlement trough and width of the settlement trough. Second, in order to incorporate the settlement model into the DAT, these parameters are associated with the ground conditions and tunnel configurations. The settlement model has been coded in the C programming language and is run as a tunnel activity in the DAT.

The DAT is composed of two main programs (GEOLOGY and SIMSUPER) and a user interface (NETWORK). The settlement model is run in one of the main programs, the construction and resource simulation module called SIMSUPER. SIMSUPER can consider the uncertainties in ground conditions and construction procedures through a probabilistic analysis. The computational results are then illustrated by using the graphic interface.

This thesis is organized as follows: Chapter 2 presents the literature review on the empirical methods regarding settlement in soft ground tunneling. Chapter 3 describes the chosen empirical equations in detail. Chapter 4 discusses the incorporation of the settlement model into the existing DAT. Chapter 5 is the User's manual and Chapter 6 presents applications of the settlement model. Conclusions and recommendations for future studies are given in Chapter 7. Additionally, the program source code of the developed model is listed in Appendix I.

Chapter 2
Literature Review

2.1 General

The basic aim of settlement prediction methods in soft ground tunneling is to produce an accurate assessment of ground settlement induced by the tunnel advance as well as to appraise the associated effects on surface structures and ground conditions. In general, most of the studies developing such predictive methods have relied on a large number of case studies and evaluated observations. One major objective of predictive methods is to offer a reasonable estimate of the settlement. The principal methods for predicting settlement can be grouped as follows: empirically derived relationships, numerical models and theoretically developed models.

Empirically derived relationships are in the form of formulae which have been established from observed surface settlement behavior: Peck and Schmidt (1969) assumed a particular geometric form of the settlement profile, specifically that the shape of the settlement trough above a tunnel is reasonably represented by an error function curve (normal distribution curve). This concept is well established and accepted as the basic form of the settlement profile by many researchers (Cording (1972), Attewell (1978), and O'Reilly (1991) et al.). A more generalized form of the error function curve, the three dimensional form which considers the direction of tunnel advance, was derived by Attewell and Woodman (1982). Since Peck's research, many researchers have concentrated on evaluating the volume of ground loss due to tunneling and the shape of the surface trough in different soil types.

Second, with the advent of powerful computing tools, numerical methods have prevailed in recent years. The application of numerical methods to the problem of ground settlement induced by tunneling is appropriate. Numerical methods are applied not only to the ground settlement prediction but also to the entire tunnel design procedures, including simulation of the excavation sequence and placing of the lining, soil - tunnel lining interaction, effects of nearby tunnels, seepage, and consolidation. One of the more refined numerical methods is the Finite Element Method (FEM). Clough and Leca (1989) reviewed recent work using the FEM as a means to analyze soft ground tunnels. They pointed out that soil tunneling problems have proved difficult for FEM modeling because they are complex, often involving many parameters that are poorly estimated or indefinite if one does not properly model both the soil and the construction procedure. There are also many cases in which the available information about the soil properties is scarce and does not justify the use of a complex constitutive model and a sophisticated numerical method. However, the flexibility of FEM models can be exploited in performing back analyses of ground movements, and can assist in understanding the ground movements at particular sites.

Finally, theoretical models exist which are based on the fundamental equations of the elastic and continuum theories. Sagaseta (1987) presented closed form solutions for obtaining the strain field in initially isotropic, homogeneous and incompressible soil due to near-surface ground loss. He showed that the calculated movements agree with the experimental observations and compare favorably with commonly used numerical methods. Although the simplified theoretical model can predict the general tendency of ground movement, it has yet to reach the stage where it can describe more complicated soil behavior such as high shear strain and consolidation.

This chapter concentrates on empirically derived relationships of ground settlement, since this is the method applied to the model used in the research presented in my thesis.

2.2 Empirical Methods

2.2.1 Research by Peck and Schmidt (1969, 1974)

Peck (1969) and Schmidt (1974) assumed that the permanent settlement profile can be described in terms of a normal distribution function curve (error function curve). They showed that this approach adequately models the shape of the settlement trough caused by tunneling in soft ground on the basis of a statistical evaluation of field observations.

(1) Basic Equations

The equation used by Peck and Schmidt to specify settlement profile is

$$S = S_{max} \exp(-x^2/2i^2), \tag{2.1}$$

where S is the vertical settlement of a point which is at a distance x from the vertical plane containing the tunnel axis (see Figure 2.1), S_{max} is the settlement of the point directly above the tunnel, and i is a parameter which defines the width of the settlement trough.

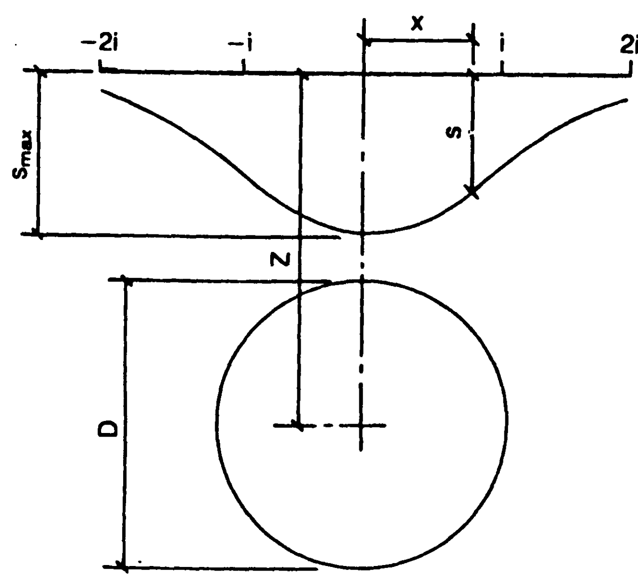


Figure 2.1 Surface settlement represented by the error curve (from Peck (1969))

If settlement occurs with no change in the volume of the soil, then the volume of the soil (V_s) between the settlement trough and the original ground surface is obtained by the integration of Equation (2.1), that is,

$$V_s = \sqrt{(2\pi)}\ i\ S_{max}, \tag{2.2}$$

where V_s is the volume of the settlement trough per unit length of tunnel, i is a parameter which defines the width of the settlement trough, and S_{max} is the settlement of the point directly above the tunnel.

Peck also produced a dimensionless plot of the observed width of the settlement profile where different types of soil are compared to the depth of the tunnel axis.(see Figure 2.2).

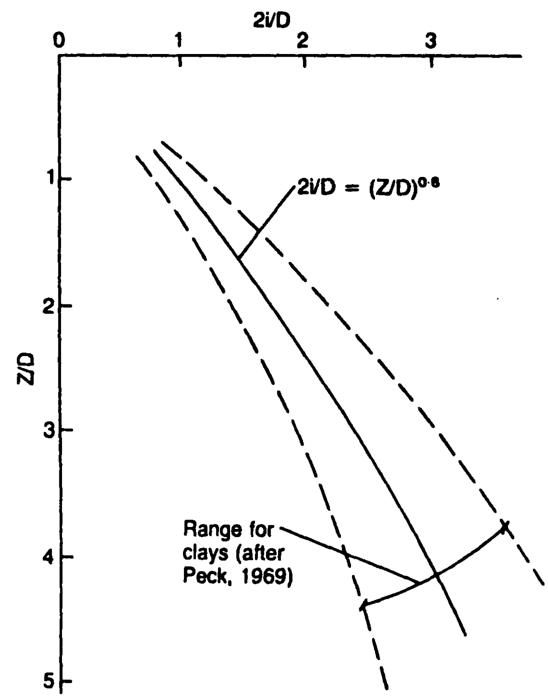


Figure 2.2 Relation between trough width and tunnel depth (from Peck (1969))

The relationship that can fit Peck's plot (broken lines in Figure 2.2) was given by Schmidt as follows:

$$(2i/D) = (Z/D)^{0.8\text{-}1.0}, \tag{2.3}$$

where D is the diameter of the tunnel, i is a parameter which defines the width of the settlement trough, and Z is the depth of the tunnel (see Figure 2.1).

(2) Procedures for Computation

These relationships and observations can be used to estimate the settlement above a real tunnel as follows:

(1) One estimates the volume of settlement trough (V_s) on the basis of experience with similar tunneling techniques in similar soils.

(2) The value of i is obtained by using Peck's chart (Figure 2.2) or Eq. (2.3) or a similar relationship.

(3) Once both the volume of the settlement trough (V_s) and the trough width parameter (i) are calculated, Eq.(2.2) is used to find S_{max}. Then Eq.(2.1) can predict the surface settlement at any point (x).

2.2.2 Research of Cording and Hansmire (1972, 1975, 1989)

Cording and Hansmire (1972, 1975, 1989) at the University of Illinois are Peck's successors. They concentrated on estimating the volume of ground loss that causes surface settlement. They have stated that the difference between the volume lost into the tunnel and the volume of the surface settlement trough is largely due to compression of the soil at the side of the tunnel and the volume increase of granular materials over the crown.

They modified the trough width relation by using a vertical angle β, which is the angle between the vertical line and the line drawn from springline to the edge of surface trough (See Figure 2.4).

(1) Basic Equations

As described in 2.2.1, the shape of the settlement trough at the ground surface resembles the shape of a normal or error distribution curve. Cording and Hansmire (1972, 1975, 1989) used the properties of the normal distribution curve (by Peck and Schmidt (1969)) as an expedient method for describing the trough widths of tunnels:

$$S = S_{max} \exp(-x^2/2i^2), \tag{2.4}$$

where S is the vertical settlement of a point which is at a distance x from the vertical plane containing the tunnel axis, S_{max} is the settlement of the point directly above the tunnel, and i is a parameter which defines the width of the settlement trough.

The correlation of (i) with tunnel radius, depth, and soil type is shown in Figure 2.3.

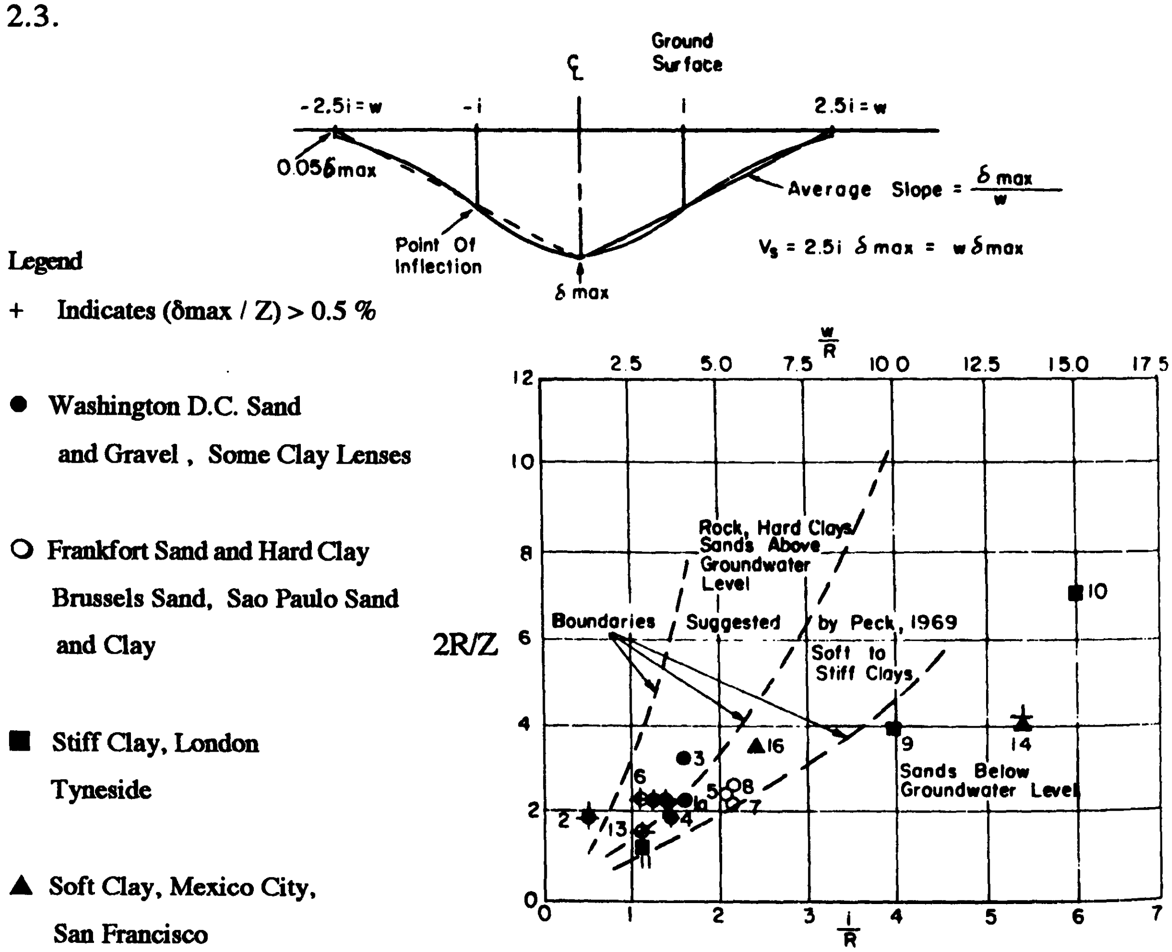


Figure 2.3 Width of settlement trough (modified by Peck (1969))

(2) Definition of Lost Ground

As shown in Figure 2.4 (a), the volume of the settlement trough can be simply defined by the width (w) and maximum displacement (δ_{max}). Thus the volume of the settlement trough can be computed as a triangle with the base (2w) and a height (δmax) as follows:

$$Vs = \frac{1}{2} 2w \cdot \delta_{max} = w \cdot \delta_{max} , \qquad (2.5)$$

where V_s is the volume of the settlement trough per unit length of tunnel, w is the half width of the base of the triangular trough, and δ_{max} is the maximum displacement of the trough. When superposed on a normal distribution curve, the width (w) is equal to 2.5 i. Moreover, Cording and Hansmire (1975) stated that the relation between trough width and depth can be expressed as a vertical angle (β) drawn from the springline of the tunnel to the defined width (w) of the settlement trough at the surface (see Figure 2.4 (a) and (b)). Figure 2.4 (b) relates β to different ground types.

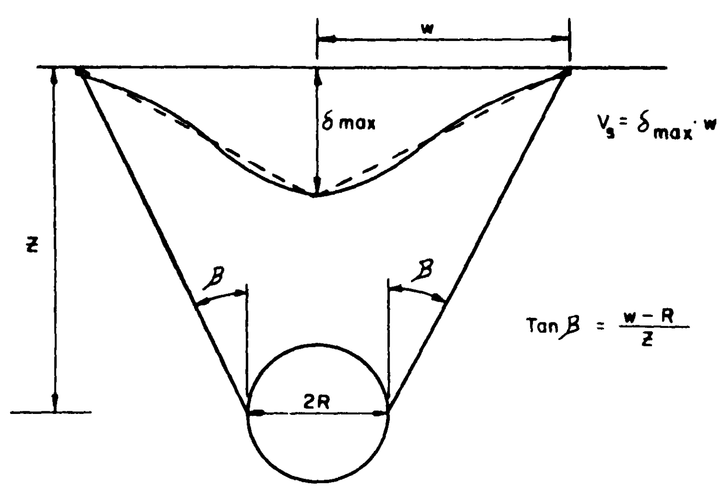


Figure 2.4 (a) Relation of β to trough width (from Cording and Hansmire (1975))

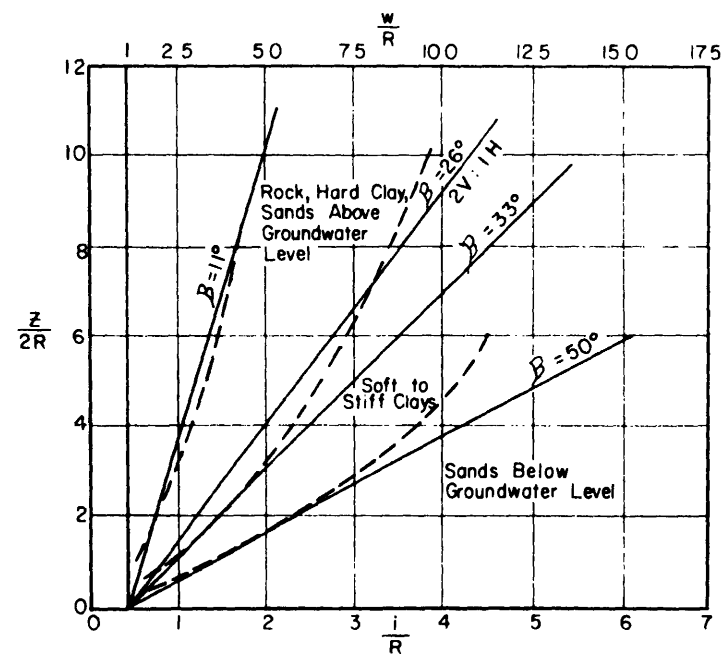


Figure 2.4 (b) Relationship between trough width and tunnel depth (from Cording and Hansmire (1975))

In addition, Cording and Hansmire (1975) stated that some settlement data, in particular granular soil, might not fit the normal distribution curve. The settlement at the edge of the trough did not continue to increase in proportion to the settlement at the center of the trough, after settlement at the center became large. Rather, as δmax increased, further settlement was concentrated just above the tunnel where the zone of high shear strain exists. After that, the settlement trough no longer fits the normal probability curve, and the calculated values of i decrease gradually. When applying the normal distribution curve in predicting the surface settlement, one should know the limitations, especially for cohesionless soil in which localized yield zones rapidly propagate from the tunnel sidewalls to the surface.

2.2.3 Research of Atkinson and Potts (1975, 1977)

Atkinson and Potts (1977) illustrated how the distribution of displacement throughout the soil around a tunnel depends on the nature of the soil and the depth of the tunnel. They demonstrated, in particular, how deformation occurring at the periphery of the tunnel migrates through the soil and appears as surface settlement. The magnitude and the shape of the trough of surface settlement can be related empirically to the settlement of the tunnel crown, the depth of burial, and the characteristics of both sand and clay. Through model tests, they proposed a relation between trough width parameter(i), overburden depth from surface to the tunnel crown (C), and tunnel diameter (D). Additionally, they derived the relationship between maximum surface settlement (S_{max}) and the settlement of the crown (S_C).

(1) Basic Equation

The shape of the surface settlement can again be represented with an error function curve of the form introduced by Peck and Schmidt (1969) (see Figure 2.5):

$$S = S_{max} \exp(-x^2/2i^2), \qquad (2.6)$$

where S is the vertical settlement of a point which is a distance x from the vertical plane containing the tunnel axis, S_{max} is the settlement of the point directly above the tunnel, and i is a parameter which defines the width of the settlement trough.

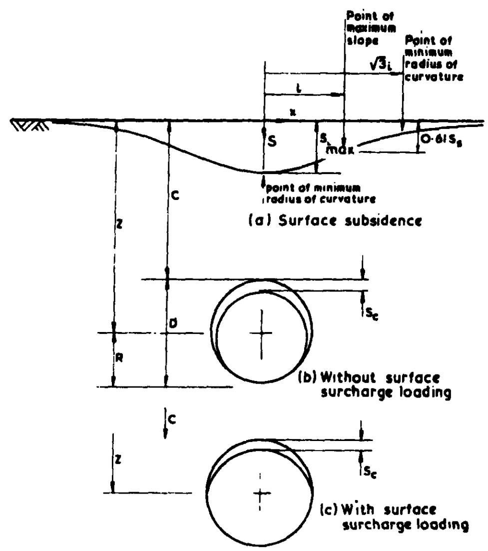


Figure 2.5 Geometry of tunnel and surface deformations observed for circular tunnel (from Atkinson and Potts (1977))

(2) Estimation of Trough Width

In order to examine the relationship between the trough width parameter (i) and the depth of the tunnel, model tests were conducted with and without surface surcharge.

For settlements above tunnels in medium sand without surface surcharge, i is evaluated as follows:

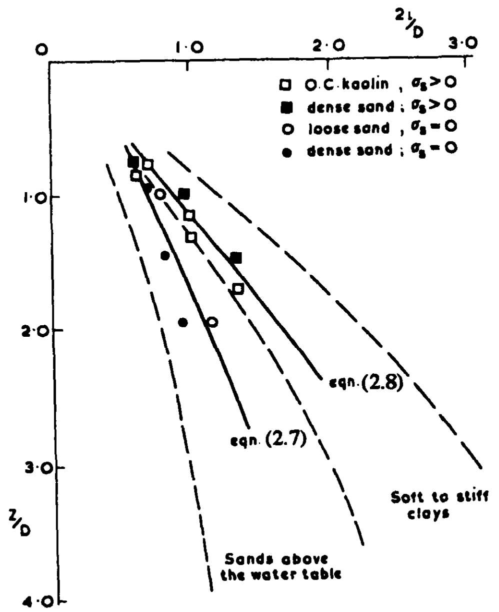
$$i = 0.25\,(C + D), \tag{2.7}$$

where i is a parameter which defines the width of the settlement profile, C is the depth of cover to the tunnel crown shown in Figure 2.5, and D is the tunnel diameter.

For settlements above tunnels in dense sand and in overconsolidated Kaolinite with surcharge,

$$i = 0.25\,(1.5\,C + D), \tag{2.8}$$

where i is a parameter which defines the width of the settlement trough, C is the depth of cover to the tunnel crown shown in Figure 2.5, and D is the tunnel diameter.



σ_s = Surface surcharge

Figure 2.6 Variation of maximum surface settlement profile with depth for model tunnels (from Atkinson (1977))

(3) Volume of Ground Loss

Assuming the surface settlement trough may be approximated by an error function curve, Atkinson and Potts (1977) defined the volume of the settlement trough (V_s) for per unit length of tunnel as

$$V_S = \sqrt{2\pi}\, i S_{max}, \tag{2.9}$$

where V_s is the volume of the settlement trough per unit length of tunnel, i is a parameter which defines the width of the settlement trough, and S_{max} is the settlement of the point directly above the tunnel.

If it is then assumed that the tunnel deforms as indicated in Figure 2.5 and that the magnitude of the crown settlement is relatively small compared to the tunnel diameter, the volume of ground lost in the tunnel per unit length is

$$V_T = \frac{\pi}{2} D S_c, \tag{2.10}$$

where V_T is the ground loss during excavation, D is the tunnel diameter, and S_c is the settlement of the tunnel crown.

Thus, the ratio of the volume of ground lost at the surface and in the tunnel is

$$\frac{V_S}{V_T} = 2\sqrt{\frac{2}{\pi}}\left(\frac{i}{2a}\right)\left(\frac{S_{max}}{S_c}\right), \tag{2.11}$$

where V_s is the volume of the settlement trough per unit length of tunnel, V_T is the ground loss during excavation, a is the half-width of the opening, i is a parameter which defines the width of the settlement trough, S_C is the vertical settlement at the crown of the tunnel, and S_{max} is the settlement of the surface point directly above the tunnel.

(4) Relationship between Surface Settlement and Crown Settlement

Atkinson and Potts (1977) assumed that the magnitude of S_{max}/S_C for a tunnel depends on the depth of burial, the presence of a surface surcharge, and any compression and dilation in the soil around the tunnel. Figure 2.7 shows the variation of maximum surface settlement with crown settlement for tunnels in both sand and clay. In this Figure, the relationship between S_{max} and S_C can be represented by a linear expression. Figure 2.8 also shows the variation S_{max}/S_C with the depth of tunnels in both sand and clay: (a) Dense sand; (b) Sands; (c) Overconsolidated Kaolinite.

Smax = Maximum surface settlement

Sc = Crown settlement

C = Overburden depth

D = Tunnel diameter

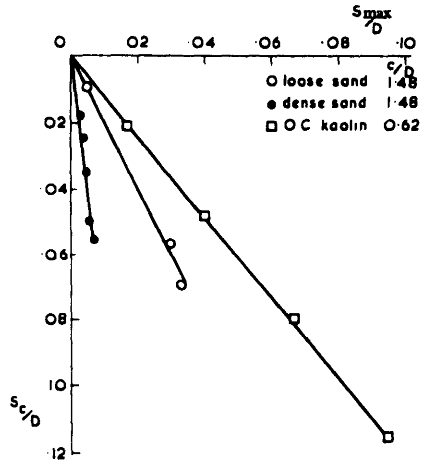


Figure 2.7 Variation of maximum surface settlement with crown settlement for tunnels in sand and clay (from Atkinson (1977))

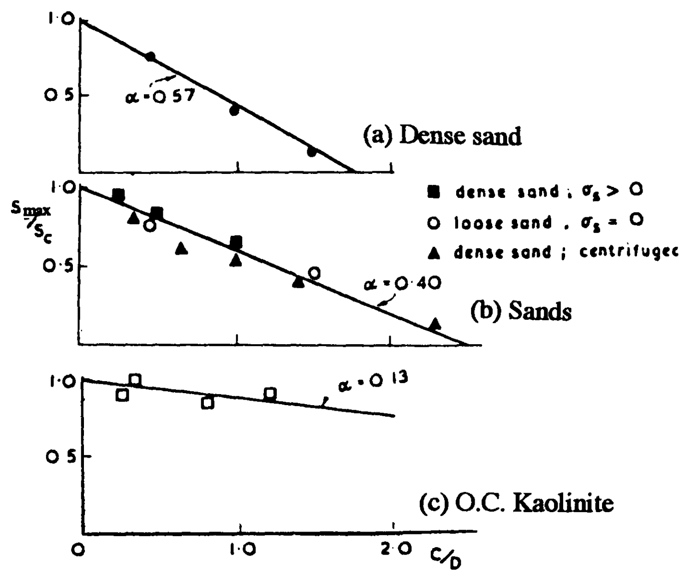


Figure 2.8 Variation of S_{max}/S_C with depth of burial for model tunnels in sand and clay (from Atkinson (1977))

The relationship between maximum surface settlement and crown settlement can be described by the following equation;

$$\frac{S_c}{S_{max}} = 1.0 - \alpha\left(\frac{C}{D}\right), \tag{2.12}$$

where S_{max} is the maximum settlement that occurs above the tunnel axis, S_C is the crown settlement, C is the overburden depth from surface to tunnel crown, D is the diameter of opening, and α is the slope of the graph plotted S_{max}/S_C versus C/D (see Figure 2.8).

Table 2.1 also defines the values of α used in Eq. (2.12).

Table 2.1 Value α for different soil types

Soil	Value of α	Relative rates of volume strain in laboratory tests
Dense sand at low stresses	0.57	Large dilation
Loose sand and dense sand at large stresses	0.40	Small or moderate dilation
Overconsolidated Kaolinite	0.13	Very small or zero dilation, possibly small compression

2.2.4 Research of Attewell (1978, 1982)

Attewell (1978, 1982) was involved in the fundamental research on Peck's basic formulae. He applied Peck's error function, not only to cohesive soil but also to cohesionless soil (granular soil), by using information extracted from a case history. Furthermore, Attewell and Woodman (1982) derived a three dimensional form of the error function curve.

(1) Estimation of Trough Width

Attewell derived the trough width parameter by using the following equation (see Figure 2.9) along with the relationship between the depth and tunnel opening size.

$$i = aK_a[z/2a]^n , \qquad (2.13)$$

where i is the transverse horizontal distance between the points of maximum settlement and of inflection, z is the depth from surface to tunnel center, and a is the half-width of the opening. K_a (ordinate intercept) and n (slope) can be determined empirically by plotting log(i/a) against log(z/2a).

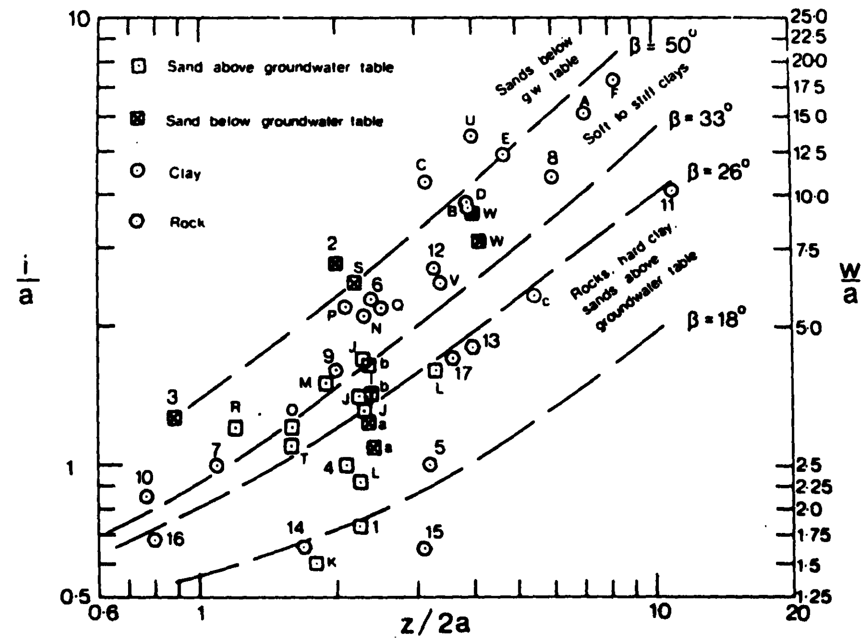


Figure 2.9 Settlement trough width as a function depth and diameter

(from Attewell (1978))

(2) Deriving Three Dimensional Equations for Settlement

Attewell and Woodman (1982) derived a three dimensional equation for settlement, lateral displacement, and strain that might be used for computation at any point at a center distance from the tunnel face.

Eq. (2.14) expresses the three dimensional equation for settlement. In this expression, the vertical displacement (w) in the Z axis is given as follows (see Figure 2.10);

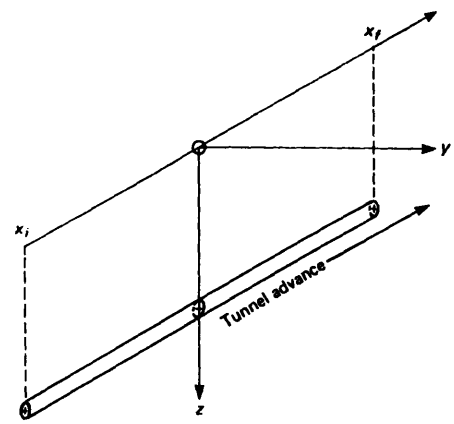


Figure 2.10 Tunnel coordinate system (from Attewell and Woodman (1982))

Figure 2.10 shows the coordinate system to be adopted: x - parallel to the tunnel center line; y - transverse to the tunnel center line; z - vertical axis through tunnel center.

$$w = \frac{V}{2\pi i}\int_{x_i}^{x_f} \exp\left[\frac{-(x-x_0)^2+y^2}{2i^2}\right]dx_0 \, , \qquad (2.14)$$

$$= \frac{V}{\sqrt{2\pi i^2}}\exp\left[\frac{-y^2}{2i^2}\right]\left[G\left(\frac{X-X_i}{i}\right)-G\left(\frac{X-X_f}{i}\right)\right],$$

where

$$G(\alpha) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\alpha}\exp\left(\frac{-\beta^2}{2}\right)d\beta \, ,$$

$$\alpha = \frac{X-X_{i,f}}{i} \, , \qquad (2.15)$$

where V is the volume(m^3) of settlement trough per unit face advance(m), i is a parameter which defines the width of the settlement trough, subscript (0) denotes the line of y=0, subscript (i) is used to denote 'initial location of tunnel,' and subscript (f) means 'final location of tunnel.' G(α) is a cumulative normal distribution function.

β is a variable to express the density function of normal distribution function. Therefore, by calculating the term α in Eq.(2.15), tables provide values of G ( ) (see Table 2.2).

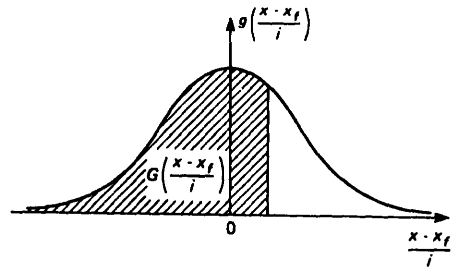


Table 2.2 Numerical integration of the normal probability curve

Table of $G\left(\frac{x-x_f}{i}\right)$

$(x-x_f)/i$	0	1	2	3	4	5	6	7	8	9
0.0	.500	.504	.508	.512	.516	.520	.524	.528	.532	.536
0.1	.540	.544	.548	.552	.556	.560	.564	.567	.571	.575
0.2	.579	.583	.587	.591	.595	.599	.603	.606	.610	.614
0.3	.618	.622	.626	.629	.633	.637	.641	.644	.648	.652
0.4	.655	.659	.663	.666	.670	.674	.677	.681	.684	.688
0.5	.691	.695	.698	.702	.705	.709	.712	.716	.719	.722
0.6	.726	.729	.732	.736	.739	.742	.745	.749	.752	.755
0.7	.758	.761	.764	.767	.770	.773	.776	.779	.782	.785
0.8	.788	.791	.794	.797	.800	.802	.805	.808	.811	.813
0.9	.816	.819	.821	.824	.826	.829	.831	.834	.836	.839
1.0	.841	.844	.846	.848	.851	.853	.855	.858	.860	.862
1.1	.864	.867	.869	.871	.873	.875	.877	.879	.881	.883
1.2	.885	.887	.889	.891	.893	.894	.896	.898	.900	.901
1.3	.903	.905	.907	.908	.910	.911	.913	.915	.916	.918
1.4	.919	.921	.922	.924	.925	.926	.928	.929	.931	.932
1.5	.933	.934	.936	.937	.938	.939	.941	.942	.943	.944
1.6	.945	.946	.947	.948	.949	.951	.952	.953	.954	.954
1.7	.955	.956	.957	.958	.959	.960	.961	.962	.962	.963
1.8	.964	.965	.966	.966	.967	.968	.969	.969	.970	.971
1.9	.971	.972	.973	.973	.974	.974	.975	.976	.976	.977
2.0	.977	.978	.978	.979	.979	.980	.980	.981	.981	.982
2.1	.982	.983	.983	.983	.984	.984	.985	.985	.985	.986
2.2	.986	.986	.987	.987	.987	.988	.988	.988	.989	.989
2.3	.989	.990	.990	.990	.990	.991	.991	.991	.991	.992
2.4	.992	.992	.992	.992	.993	.993	.993	.993	.993	.994
2.5	.994	.994	.994	.994	.994	.995	.995	.995	.995	.995
2.6	.995	.995	.996	.996	.996	.996	.996	.996	.996	.996
2.7	.997	.997	.997	.997	.997	.997	.997	.997	.997	.997
2.8	.997	.998	.998	.998	.998	.998	.998	.998	.998	.998
2.9	.998	.998	.998	.998	.998	.998	.998	.999	.999	.999
3.0	.999	.999	.999	.999	.999	.999	.999	.999	.999	.999

2.2.5 Research of O'Reilly and New (1982, 1988, 1991)

O'Reilly and New (1982) presented more generalized empirical forms of the settlement profile based on the Peck's research. They developed the general equations for both cohesive and cohesionless soil using effective width (i) and soil constant (K) on the basis of the three dimensional equation proposed by Attewell (1978).

(1) Basic Equations

The shape of the surface settlement may again be represented by an error function curve of the form proposed by Peck and Schmidt (1969) (see Figure 2.11).

$$S = S_{max} \exp(-x^2/2i^2), \tag{2.16}$$

where S is the vertical settlement of a point which is at a distance x from the vertical plane containing the tunnel axis, S_{max} is the settlement of the point directly above the tunnel, and i is a parameter which defines the width of the settlement trough.

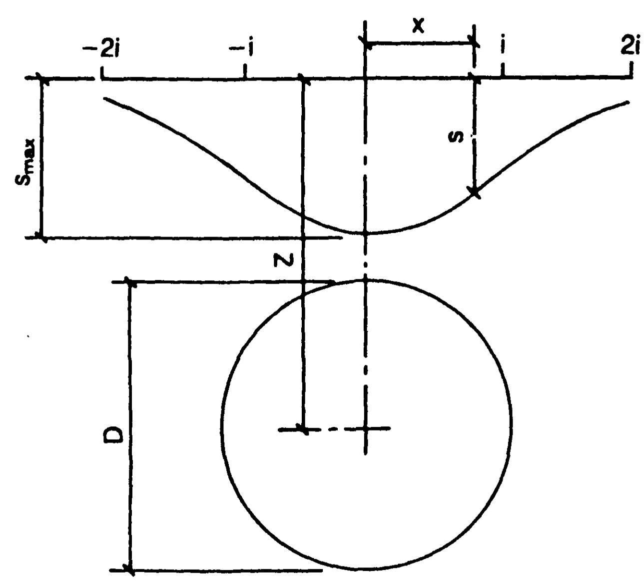


Figure 2.11 Surface settlement by the error function curve (from Peck (1969))

(2) Estimation of Trough Width Parameter (i) and Soil Constant (K)

O'Reilly and New (1982) assumed the radial flow of soil displacement (see Figure 2.12) in defining the trough width parameter (i). The adoption of radial flow means that the width of the zone of deformed ground decreases linearly with depth below the ground surface. Therefore, the trough width parameter is simply derived as Eq. (2.17).

$$i = KZ, \tag{2.17}$$

where i is a parameter which defines the width of the settlement trough, K can be defined from the slope by plotting i against Z by using field data, and Z is the depth of tunnel center.

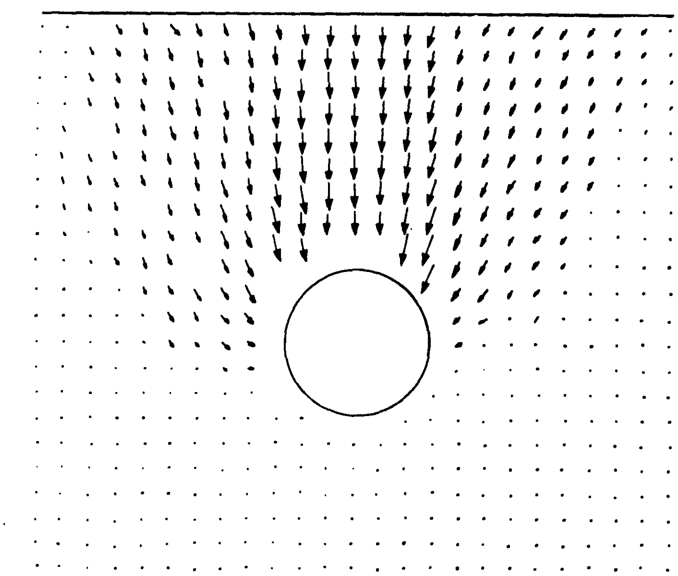


Figure 2.12 Soil displacement around model tunnel in clay (from Mair (1979))

Furthermore, O'Reilly and New (1982) proposed the relationship between horizontal and vertical components of displacement. Similarly, Glossop's (1977) stochastic analysis of surface movement around tunnels gives results which are identical to the equation below, as do Martos' (1958) results for horizontal surface displacements above tabular openings. The general equation can be expressed as follows:

$$H_{(y,z)} = \frac{y}{z} S_{(y,z)}, \tag{2.18}$$

where H(y, z), and S(y, z) are, respectively, the horizontal and vertical components at a transverse distance(y) and at a vertical distance(z) from the tunnel axis.

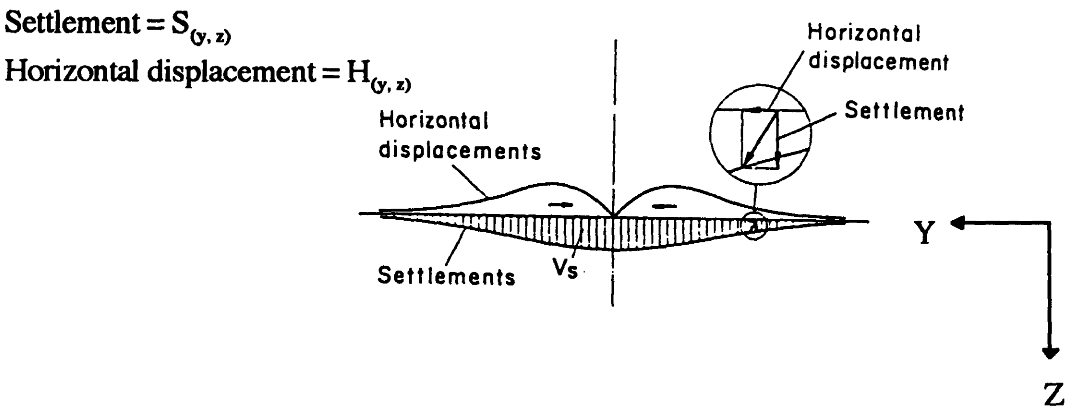


Figure 2.13 Patterns of horizontal displacement (from Sagaseta (1989))

The width of the settlement trough is defined using the distance from the tunnel center to the point of the inflection (y= i). Multiple linear regression analyses were performed on field data (21 cohesive soils, 16 cohesionless soils) to build the relationship between the trough parameter and depth for both cohesive and cohesionless soils. Figure 2.14 shows the trough width parameter plotted against tunnel axis depth for both ground types. Finally, Eq. (2.19) defines the trough width parameter for both soil types.

$$i = 0.43Z + 1.1 \text{ (m)} \quad \text{for cohesive soil,}$$

$$i = 0.28Z - 0.12 \text{ (m)} \quad \text{for cohesionless soil,} \tag{2.19}$$

where i is the trough width parameter, and Z is the depth to the tunnel axis.

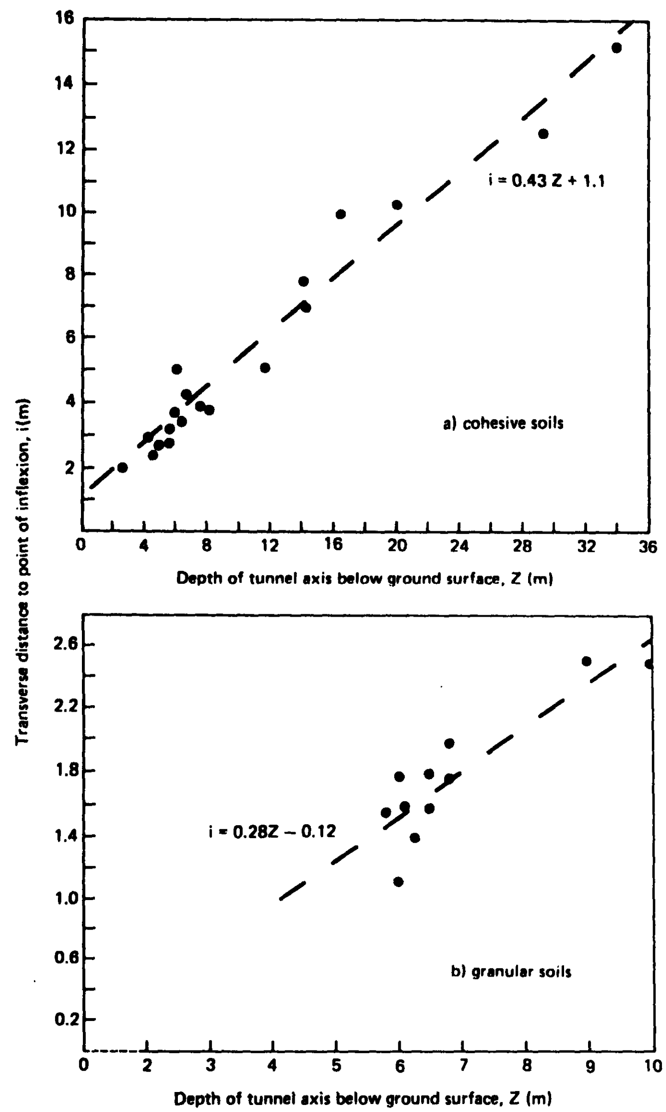


Figure 2.14 Relation of trough width parameter to tunnel depth (from O'Reilly (1982))

In the case of two strata, Eq. (2.19) can be combined as follows:

$$i = 0.43Z_a + 0.28Z_b + 1.1 \text{ (m)} \quad \text{for a tunnel in clay overlain by sand,}$$

$$i = 0.28Z_a + 0.43Z_b - 0.12 \text{ (m)} \quad \text{for a tunnel in sand overlain by clay,} \qquad (2.20)$$

where i is a parameter which defines the width of the settlement trough, Z_a is the depth of the tunnel axis beneath the interface, and Z_b is the thickness of the surface layer.

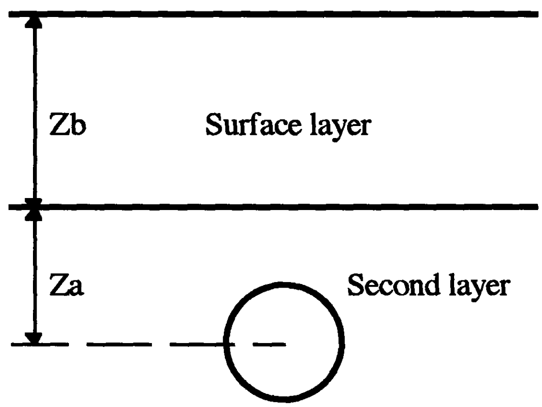


Figure 2.15 Schematic of layered strata

Moreover, in the case of multilayered strata (N layers), trough width(i) is given as

$$i_N = K_1Z_1+K_2Z_2+K_3Z_3+...............+K_NZ_N. \tag{2.21}$$

Field data analysis indicates K varies from about 0.4 for stiff clays to about 0.7 for soft and silty clays. For cohesionless materials above the water table, K ranges from 0.2 to 0.3.

(3) Generalized Equations

(i) Single Tunnel

The two dimensional equations for single tunnel are,

for vertical settlement, (see Figure 2.16)

$$S_{(y,z)} = S_{(\max,y,z)} \exp(-y^2/2i^2) = \frac{V_s}{\sqrt{(2\pi)}KZ} \exp(-y^2/2(KZ)^2),$$

and for horizontal displacement,

$$H_{(y,z)} = \frac{y}{Z} S_{(\max,y,z)} \exp(-y^2/2i^2) = \frac{V_s y}{\sqrt{(2\pi)}KZ^2} \exp(-y^2/2(KZ)^2), \tag{2.22}$$

where $S_{(y,z)}$ and $H_{(y,z)}$ are the vertical and horizontal components of displacement at a transverse distance y and a vertical distance z, i is a parameter which defines settlement trough width, V_s is the volume of the settlement trough per unit length of tunnel, and K can be determined as the slope in the graph plotting i against Z.

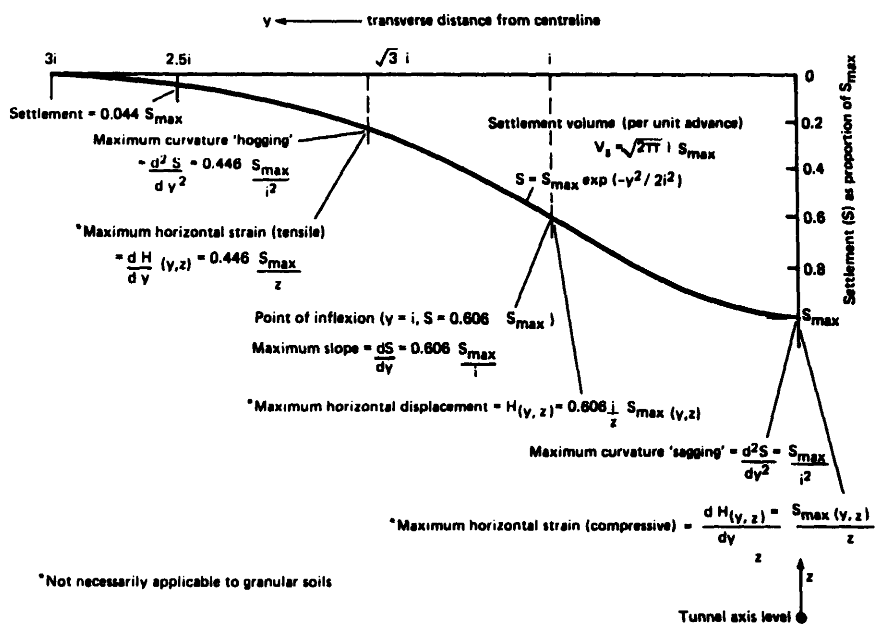
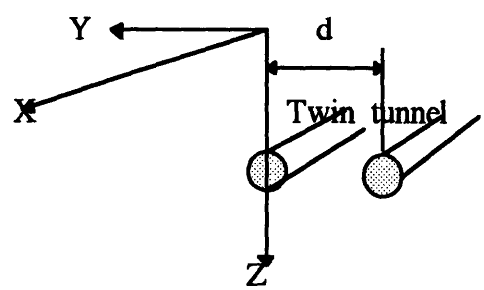


Figure 2.16 Settlement semi-profile with error function form (from O'Reilly (1988))

(ii) Twin Tunnels

The two dimensional equations for twin tunnels are:



for vertical settlement (see Figure 2.17),

$$S_{(y,z)} = \frac{V_s}{\sqrt{(2\pi)}KZ}\left[\exp(-y^2/2(KZ)^2) + \exp(-(y-d)^2/2(KZ)^2)\right],$$

and for horizontal settlement,

$$H_{(y,z)} = \frac{V_s}{\sqrt{(2\pi)}KZ^2}\left[y\exp(-y^2/2(KZ)^2) + (y-d)\exp(-(y-d)^2/2(KZ)^2)\right] \tag{2.23}$$

where $S_{(y,\ z)}$ and $H_{(y,\ z)}$ are the vertical and horizontal components of displacement, respectively, at a transverse distance y and a vertical distance z from the tunnel axis, i is a parameter which defines settlement trough width, V_s is the volume of the settlement trough per unit length of tunnel , and K can be determined as the slope in the graph plotting i against Z, and d is the axial separation of the tunnels.

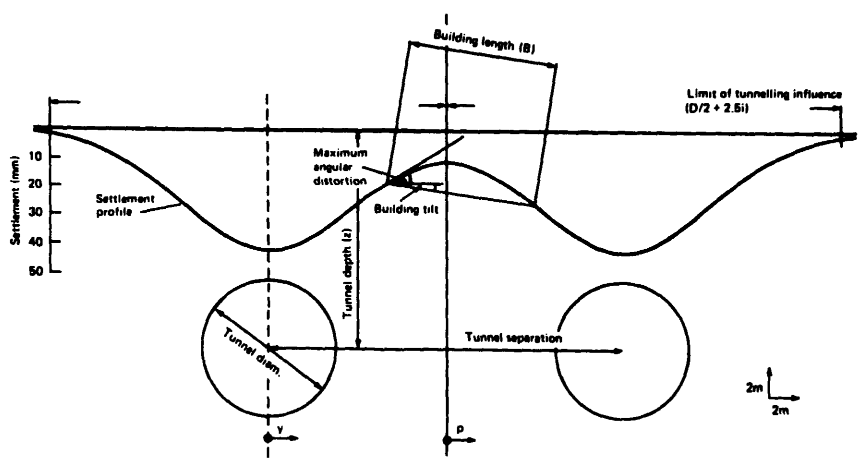


Figure 2.17 Surface settlement profile for twin tunnels (from O'Reilly (1988))

The vertical and horizontal displacements for any point with coordinates X, Y, Z follow the assumption of a normal probability form for the transverse profile, while the longitudinal profile (x direction) should take a cumulative probability form; this has been reasonably validated by examining field study reports (Attewell and Woodman, 1982).

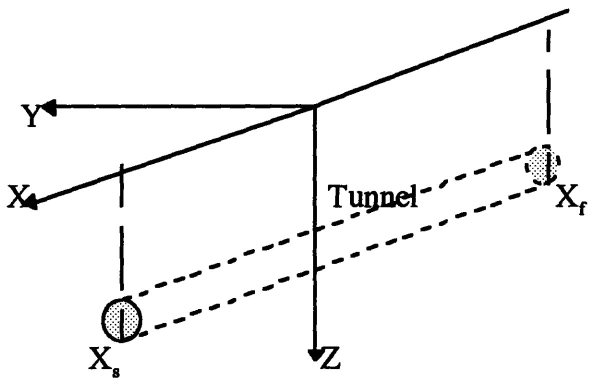


Figure 2.18 Coordinate system in three dimensions

The vertical and horizontal displacements for any points are given by

$$S_{(y,z)} = \frac{V_s}{\sqrt{2\pi}KZ} \exp(-y^2/2(KZ)^2)\left[F\left(\frac{X-X_s}{KZ}\right) - F\left(\frac{X-X_f}{KZ}\right)\right],$$

$$H_{(y,z)} = \frac{y}{Z} S_{(y,z)}, \qquad (2.24)$$

$$H_{(x,z)} = \frac{V_s}{2\pi Z}\left[\exp(-(X-X_s)^2 - y^2)/2(KZ)^2 - \exp(-(X-X_f)^2 - y^2)/2(KZ)^2\right],$$

where $S_{(y,\ z)}$ and $H_{(y,\ z)}$ are the vertical and horizontal components of displacement, respectively, at a transverse distance y and a vertical distance z from the tunnel axis, i is a parameter which defines settlement trough width, V_s is the volume of the settlement trough per unit length of tunnel , K is an empirical constant which depends on ground conditions, X_s and X_f are respectively the starting and final locations of the tunnel face. The function F( ) represents a cumulative distribution function of a standardized normal random variable, that is,

$$F_{(a)} = \int_{-\infty}^{a} \frac{1}{\sqrt{(2\pi)}} e^{-\frac{1}{2}t^2}\, dt,$$

$$a = \frac{X - X_{s,f}}{KZ} \tag{2.25}$$

In this equation, t is a variable expressing the density function of the normal distribution function. Therefore, by calculating the term a, tables (see table 2.2) yield values of F ( ). In particular, F(0) gives 0.5, and F(1) provides 1.0.

Chapter 3
Definition of the Ground Settlement

3.1 General

A satisfactory tunnel should be designed in such a way that its construction will cause as little damage as possible to overlying or adjacent structures and services. With soft ground tunneling, settlement is often a problem in built-up areas, where significant structures can be put at risk. To minimize overall project costs and the risk of damage or accidents, the engineer who designs a tunnel must be able to predict the extent and amount of settlement that is likely to arise from tunneling in a variety of conditions. Although various prediction methods ranging from simplified equations to complex analytical formulae have been presented, the chosen method should be simple enough to allow one to easily determine the next appropriate steps. Because of this, there is a need for generalized empirical equations.

Given reliable forecasts of ground deformations, one would be in a position to choose between a number of options that, depending on the particular location, might include (1) relocation of the tunnel far away from sensitive structures or services, (2) an alternative tunnel in better ground, (3) adoption of the appropriate method for ground control on a more direct route, and (4) the underpinning of existing buildings and the relocation of water and gas lines. Such considerations, in addition to the growing emphasis on environmental problems, have led to a considerable amount of research regarding settlements and ground deformations caused by tunneling in soft ground.

In this chapter, the mechanism of soft ground tunneling and the evaluation of empirically derived equations are discussed.

3.2 Definition of Soft Ground Tunneling

Due to the relatively low strength and high deformability of soils, tunneling through soft ground is very difficult. These adverse mechanical characteristics have a direct influence on the excavation method; the stability of the roof, the face, and walls of the tunnel; the effect of the tunnel construction on its environment; and the design of the tunnel lining. These problems must be investigated during the early design stage.

Considering the geotechnical aspects in tunnels, excavation leads to the redistribution of the pore pressures, which could be negative or positive depending on the stress distribution around the tunnel opening. With low permeability cohesive soils, the unconfined compression strength ($q_U = 2S_U$) is one of the adequate measures of the shear strength of the soils(S_U), as pore pressures will change slowly. As long as the soil around the tunnels maintains its shear strength, the ground at the face as well as the tunnel periphery can remain stable. The overload factor is a useful index for assessing tunnel stability. It is the ratio of the overburden stress at the tunnel crown to the inherent shear strength (S_U);

$$N = \frac{\sigma_v}{S_u} \tag{3.1}$$

where N is the overload factor, σ_v is overburden stress (unit weight multiplied by depth), and S_U is the undrained shear strength at the tunnel crown.

From many field data, Broms and Bennermark (1967) stated that values of N below 6 indicate that the tunnel opening can remain stable.

For coarse cohesionless soils, the permeability of the soil increases and the pore pressure tends to reduce to the atmospheric condition. The strength of these soils is governed only by frictional properties, and with the removal of stress on the side of the tunnel opening, there can be relatively rapid reduction in ground strength and an increase in deformation.

Wong and Kaiser (1987) showed different types of yield zone propagation and stress redistribution for cohesive soils and cohesionless soils (see Figure 3.1).

Mode I = cohesionless soils

Mode II = cohesive soils

S_s = Surface settlement

S_c = Crown settlement

R = Radius of continuous yield zone

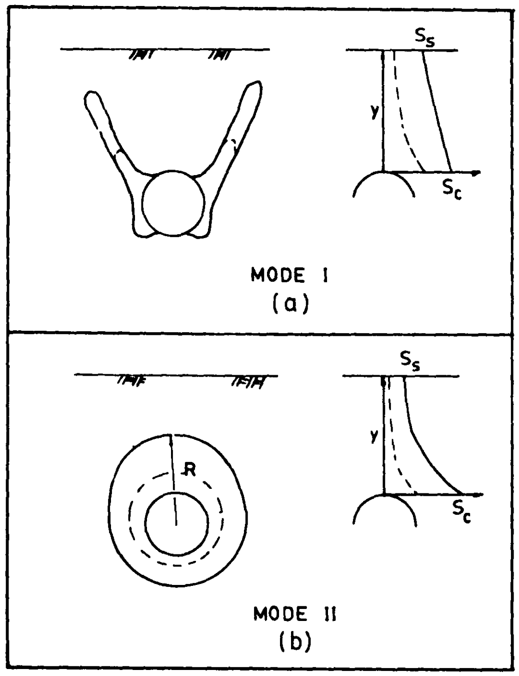


Figure 3.1 Schematic of subsurface settlement profiles

(from Wong and Kaiser (1987))

For a shallow tunnel in cohesionless soils, a localized yield zone starts to form at the tunnel circumference, and the yield zone propagates towards the ground surface from both sidewalls. The soil above the roof still does not yield and moves downward as a rigid block. In contrast, for a shallow tunnel in cohesive soils, a continuous yield zone surrounds the tunnel opening and no localized yield takes place. The yield zone occurring around the opening expands gradually. As a result, the magnitude of settlement in cohesionless soils is larger than that of cohesive soils, and the settlement trough width for cohesionless soils tends to be smaller than that in cohesive soils.

3.3 Evaluation of the Equations for the Ground Settlement

As described in Chapter 2, most of the empirical settlement profiles can be represented by an error function curve presented by Peck (1969).

Eq. (3.2) shows the vertical and horizontal components of O'Reilly and New's settlement profile (1982). Both the vertical and horizontal displacement can be expressed by employing the same parameters.

For vertical displacement,

$$S_{(y,z)} = S_{(\max,y,z)} \exp(-y^2/2i^2) = \frac{V_s}{\sqrt{(2\pi)}KZ} \exp(-y^2/2(KZ)^2),$$

and for horizontal displacement,

$$H_{(y,z)} = \frac{y}{Z} S_{(\max,y,z)} \exp(-y^2/2i^2) = \frac{V_s y}{\sqrt{(2\pi)}KZ^2} \exp(-y^2/2(KZ)^2), \qquad (3.2)$$

where $S_{(y,z)}$ and $H_{(y,z)}$ are the vertical and horizontal components of displacement at a transverse distance y and a vertical distance z, i is a parameter which defines settlement trough width, V_s is the volume of the settlement trough per unit length of tunnel, and K can be determined as the slope in the graph plotting i against Z (see Figure 2.14).

When computing the maximum surface settlement, which often becomes the significant parameter regarding structural damages, Eq. (3.3) is obtained by substituting y = 0 in Eq.(3.2).

$$S_{\max} = \frac{V_s}{\sqrt{2\pi}KZ}, \qquad (3.3)$$

where S_{max} is the maximum settlement in the tunnel axis, i is a parameter which defines the width of the settlement trough, and V_s is the volume of the settlement trough per unit length of tunnel.

In Eq.(3.3), the values of the trough width parameter i (= KZ), and the volume of the settlement trough V_s, are the two critical constants that define the maximum surface settlement. Therefore, there is a need to provide the appropriate values for i and V_s so that the settlement profile can be uniquely defined.

O'Reilly and New (1991) clearly defined one of the primary constants, trough width parameter i, in the form of a linear equation (see Eq.(3.7)). The other critical constant V_S, was expressed in the form of a percentage of the volume of excavation based on the field data. For this reason, O'Reilly and New's settlement profile is adopted as a basic equation for developing a model to predict settlement.

3.3.1 Settlement Trough Width

The transverse distance from the tunnel center line to the point of inflection (y = i) is used to describe the width of the settlement trough and should be related to both the depth from the ground surface and, to a lesser extent, the diameter of the tunnel. O'Reilly and New (1982) performed multiple linear regression analyses on field data. In their analyses, they found no significant correlation between the trough width parameter i, and the tunnel diameter D, although the expected strong correlation of i and tunnel depth Z, was found. This finding is also indicated by Glossop (1988), who carried out an analysis based on stochastic and numerical modeling techniques.

As explained in 2.2.5, a two-variable regression analysis was carried out. It provided the following relationships:

$$i = 0.43Z + 1.1 \text{ (m) for cohesive soil,}$$

$$i = 0.28Z - 0.12 \text{ (m) for cohesionless soil,} \tag{3.4}$$

where i is the trough width parameter, and Z is the depth of the tunnel center.

Figure 2.14 shows the trough width parameter plotted against the tunnel axis depth for both cohesive and cohesionless ground. Data for cohesionless soils are more scattered and reflect the unpredictable consequences of tunneling in such ground. The data suggests that a linear relationship between i and Z can appropriate for both ground conditions.

Similarly, in the case of two strata, the equations for each soil in Eq. (3.4) can be combined as follows;

$$i = 0.43Z_a + 0.28Z_b + 1.1 \text{ (m) for a tunnel in clay overlain by sand,}$$

$$i = 0.28Z_a + 0.43Z_b - 0.12 \text{ (m) for a tunnel in sand overlain by clay,} \tag{3.5}$$

where i is a parameter which defines the width of the settlement trough, Z_a is the depth of the tunnel axis beneath the interface, and Z_b is the thickness of the surface layer.

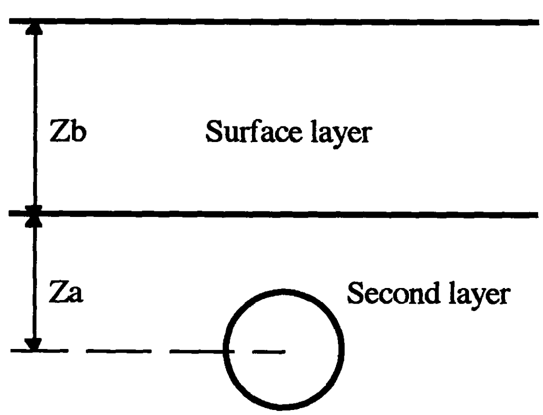


Figure 3.2 Schematic of layered strata

Moreover, in the case of multilayered strata (N layers), trough width(i) is given as

$$i_N = K_1Z_1+K_2Z_2+K_3Z_3+\ldots\ldots\ldots\ldots\ldots+K_NZ_N. \tag{3.6}$$

where i_N is the trough width parameter for multilayered strata, K_i is the soil constant in determining the trough width parameter of each layer, and Z_i is the thickness of each layer.

The linear regression lines may, for most practical purposes, be simplified to the form

$$i = KZ, \tag{3.7}$$

where i is a parameter which defines the settlement trough width, Z is the tunnel depth to the tunnel center, and K is the soil constant which is determined from the slope of the plot i as a function of Z.

O'Reilly and New (1982) suggested the value K for both cohesive and cohesionless soils based on field data. Table 3.1 provides the value K in cohesive soils for different ground conditions and for different ground support methods as follows;

Table 3.1 Values K for cohesive soils (from O'Reilly and New(1982))

Ground conditions	Ground support method in tunnels	Trough width parameter, constant, K
Stiff fissured clay	Shield or none	0.4 - 0.5
Glacial deposits	Shield in free air	0.5 - 0.6
Glacial deposits	Shield in compressed air	0.5 - 0.6
Recent silty clay deposits (C_u = 10 - 40 KN/m^2)	Shield in free air	0.6 - 0.7
Recent silty clay deposits (C_u = 10 - 40 KN/m^2)	Shield in compressed air	0.6 - 0.7

Although it is difficult to provide a reliable K value for cohesionless soils because of insufficient field data, O'Reilly and New (1982) indicate that, for cohesionless materials above the water-table, K ranges between 0.2 and 0.3.

As an average value, K = 0.5 for cohesive and K = 0.25 for cohesionless soils are adopted.

3.3.2 Volume of Lost Ground

As already discussed, both the ground conditions and the construction method determine the ground losses induced by tunneling. Consequently, defining the volume of the settlement trough at the surface in relation to ground conditions, as well as construction method, is complex.

One effective method for determining the relationship is to use the field data obtained from the various sites. The volume of the settlement trough at the surface should be related to the tunnel size or the tunnel volume excavated. To normalize the volume of lost ground with respect to tunnel size, the volume of the settlement trough at the surface V_s, is expressed as a percentage of the excavated tunnel volume V_{exc}. Given the diameter of the tunnel, the excavated tunnel volume is calculated by a simple mathematical expression ($V_{exc} = \pi D^2/4$).

Table 3.2 shows the relation between the volume of the settlement trough at the surface and the excavated tunnel volume for cohesive soils. Estimates of ground loss in cohesionless soils are difficult to predict with certainty because poor tunneling techniques can result in large and almost immediate ground settlements. Table 3.3, compiled from the literature review regarding ground loss, provides the value of the ratio of ground loss for cohesionless soils.

Table 3.2 Ground loss for cohesive soils (from O'Reilly and New (1982))

Ground conditions	Ground support method in tunnels	Ground loss V_s/V_{exc} (%)	Remarks
Stiff fissured clay	Shield or none	0.5 - 3.0	considerable data available; loses normally 1 - 2 %
Glacial deposits	Shield in free air	2.0 - 2.5	
Glacial deposits	Shield in compressed air	1.0 - 1.5	compressed air used to control ground movements
Recent silty clay deposits (C_u = 10 - 40 KN/m^2)	Shield in free air	30.0 - 45.0	failure or near failure conditions
Recent silty clay deposits (C_u = 10 - 40 KN/m^2)	Shield in compressed air	5.0 - 20.0	some partial face value included

Table 3.3 Ground loss for cohesionless soils

Ground conditions	Ground loss V_s/V_{exc} (%)	Remarks
Dense sand	0.5 - 1.0	In the case of dilating soils, 1% of the tunnel volume excavated
Medium sand	1.0 - 2.5	1% to 2% for well-constructed tunnels in cohesionless soils
Loose sand	3.0 - 5.0	loosely-compacted soil gives upper limit of 5 %

3.3.3 Prediction of Ground Displacements

Considering the uncertainty affecting tunnel designs, computations for design purposes should include probabilistic estimates of ground displacement. To provide a useful starting point in any assessment, estimates of the best and worst cases should be performed to bracket the extent and amount of ground deformation. It is also important to realize that such a predictive model can only give a general indication of the form and magnitude of the potential settlement. In practice, unexpected ground conditions on site, construction difficulties, poor tunneling techniques, or a combination of all three, could lead to significantly different ground displacements.

In general, settlement starts to appear before the passing of the tunnel face and the maximum settlement takes place after the passing of the tunnel face. A preliminary analysis is performed with a two-dimensional model, and although this may be satisfactory for the prediction of conditions subsequent to tunnel construction, other significant ground deformations of a three-dimensional character may occur during the passing of the tunnel face. However, the settlement model given in this research focuses on the maximum settlement in a transverse section, which occurs long after the passing of the tunnel face, rather than the progress of settlement in a longitudinal section with the tunnel advancing.

Considerable monitoring of ground and building settlement is now routinely carried out in most tunneling projects in urban areas. Where the extent and magnitude of the predicted settlement are important, consideration should be given to the construction program so that alternative methods are determined as early as possible. The framework given here makes it possible to review a tunnel project taking the problematic aspects of ground settlement into consideration. Finally, this framework should be employed to minimize the overall project cost and the risk of damage.

Chapter 4

Incorporation into the DAT

4.1 General

The model for ground settlement has been incorporated into the DAT (Decision Aids for Tunneling) which were developed as "decision making tools" to address uncertain conditions involved in tunnel construction. The existing DAT have features to evaluate overall project cost, duration, and resources distribution as a function of specified geologic conditions, tunnel dimensions, and construction methods.

The DAT are basically composed of two main program modules (GEOLOGY, SIMSUPER) in addition to the user interface (NETWORK) (Halabe (1995)). GEOLOGY produces probabilistic geologic/geotechnical profiles. The profiles, which reflect the probabilities of geologic conditions occurring at a particular tunnel location, are obtained by considering the uncertainty of given geologic data. SIMSUPER simulates the construction process through the profiles. The construction process involves relating geologic conditions (ground classes) to construction classes. Construction classes define tunnel cross sections, initial and final support, as well as excavation methods which are best suited for particular ground classes. Most importantly, construction is modeled by a number of activities which, in turn, are described by equations relating ground dependent or ground independent parameters to activity, duration, and cost. Parameters can be determined probabilistically in the form of different types of distributions.

The settlement model is run as a part of SIMSUPER. In SIMSUPER, the settlement model is established as a tunnel activity and the settlement equations are set up in a subroutine in the tunnel activity files. The data on ground conditions and tunnel configurations are assigned as variable files to the settlement subroutine. Both deterministic and probabilistic analyses can be implemented in the program by giving the ground condition data of either a specific value or a distribution of values.

4.2 Creation of Subroutines

4.2.1 Settlement Equation

As described in Chapter 3, the empirical equations by O'Reilly and New (1982) are used for creating the settlement model. These equations produce transverse settlement/horizontal displacement profiles from parameters on tunnel configurations, ground conditions, and construction quality. In these equations (see Eq.3.2), there are three parameters; two of the three parameters are the volume of settlement trough (V_S) and soil constant for trough width (K) which are affected by both ground conditions and construction characteristics. The third parameter is the depth of the tunnel (Z) which is part of the data on tunnel configuration. Since the volume of the settlement trough (V_S) is defined by the ratio of the volume of settlement trough (V_S) to the excavated tunnel volume (V_{exc}), tunnel diameter (D) is required to calculate the excavated tunnel volume. These calculations of the volume are performed per unit length of tunnel. In order to consider both cases of a single tunnel and twin tunnels, the distance between tunnels (d) is also defined. Therefore, the required data to calculate the settlement model are as follows:

Table 4.1 Required data for the settlement equation

Factor dependency	Variables	Description
Dependent of ground conditions and construction quality	K	Soil Constant to determine trough width (i = KZ (refer to Eq.(3.7))
	Vs/Vexc (%)	The ratio of the volume of the settlement trough to the excavated tunnel volume
Independent of ground conditions and construction quality	Z (m)	Depth from surface to the tunnel center
	D (m)	Diameter of the tunnel
	d (m)	The distance between two tunnels

4.2.2 Data Description for Settlement Subroutine

Since SIMSUPER simulates the construction process through the ground class, two ground dependent parameters, V_S/V_{exc} and K (as shown Table 4.1) must be categorized by associating them with ground classes (conditions) and construction classes (qualities). Table 3.1 lists the Values of K for cohesive soil based on field data (O'Reilly and New (1982)). Table 3.2 and Table 3.3 list the relations between the volume of the settlement trough (V_S) at the surface and the excavated tunnel volume (V_{exc}) for both cohesive and cohesionless soils.

On the basis of O'Reilly and New's research, both K values and the ratio of V_s/V_{exc} are associated with two major soil types (CLAY and SAND) and three construction qualities. Here construction quality is roughly grouped into three classes: poor, average, and good. A value of the ratio of V_s/V_{exc} is associated with each construction quality. In addition, each major soil type is subdivided into three subcategories. Hence, the K values and the ratio of V_s/V_{exc} are defined by six soil types and three construction qualities, that is, a total of 18 conditions as shown in Table 4.2.

Although some field data by O'Reilly and New indicate large values of V_s/V_{exc} , an upper limit of 5% may be used for the purpose of estimation, recognizing that this will usually be a very conservative figure (Attewell, 1978).

Table 4.2 Summary of value K and the ratio of Vs/Vexc

Ground condition Soil type	Soil property	Trough width parameter Constant, K	Construction Quality	Vs / Vexc (%)
CLAY	Stiff clay	0.4 - 0.5	Good	0.50
		0.4 - 0.5	Average	1.75
		0.4 - 0.5	Poor	3.00
	Glacial deposits (N. C. Clay)*	0.5 - 0.6	Good	1.00
		0.5 - 0.6	Average	1.75
		0.5 - 0.6	Poor	2.50
	Silty Clay (Cu=10-40KN/m2)	0.6 - 0.7	Good	3.00
		0.6 - 0.7	Average	4.00
		0.6 - 0.7	Poor	5.00
SAND	Dense	0.2 - 0.3	Good	0.50
		0.2 - 0.3	Average	0.75
		0.2 - 0.3	Poor	1.00
	Medium	0.2 - 0.3	Good	1.00
		0.2 - 0.3	Average	1.75
		0.2 - 0.3	Poor	2.50
	Loose	0.2 - 0.3	Good	3.00
		0.2 - 0.3	Average	4.00
		0.2 - 0.3	Poor	5.00

(N. C. Clay)* = Normal Consolidated Clay

4.3 Incorporation into the DAT

4.3.1 Description of the DAT

As described in 4.1, the DAT are composed of two major program modules (GEOLOGY, SIMSUPER), and the user interface (NETWORK). In general, the user performs the following procedures to run the programs:

1) The user creates all the necessary input data by using NETWORK 2) The program GEOLOGY runs based on the file input through NETWORK 3) By using the output of GEOLOGY, SIMSUPER runs construction and resource simulations.

GEOLOGY can perform either deterministic or probabilistic analyses depending on the uncertainty of the given data by applying a Markov process approach, and it can create a ground class profile as an output. Using the output file from GEOLOGY, SIMSUPER can perform the construction process and resource distribution simulations.

In the computation of settlement, Table 4.2 is used instead of running GEOLOGY. Therefore, the user dose not have to run GEOLOGY to create the ground class profiles for SIMSUPER. With regard to data input, the user can select the user interface (NETWORK) or input data directly into the files. NETWORK guides the user step by step through the input process and facilitates the organization of the complex data.

4.3.2 Description of Settlement Model

The settlement model runs as a part of SIMSUPER. Figure 4.1 shows a schematic of settlement computations. As shown in Figure 4.1, the controlling factors influencing settlement are ground conditions, construction quality, and tunnel configuration. These factors are associated with the parameters in the settlement equations. Both parameters, the ratio of the settlement trough to the excavated tunnel volume (V_S/V_{exc}) and the soil constant for the trough width parameter (K), are related to ground conditions and construction quality. Additional tunnel configuration parameters, which are tunnel depth (Z), tunnel diameter (D), and the distance between two tunnels (d) are taken directly from the tunnel geometry. After these parameters are determined, the calculation of settlement is performed in the settlement subroutines. The computational results produce the maximum surface settlement and horizontal displacement in a transverse section. Moreover, by choosing each maximum value in a transverse section, the longitudinal settlement profile can be obtained.

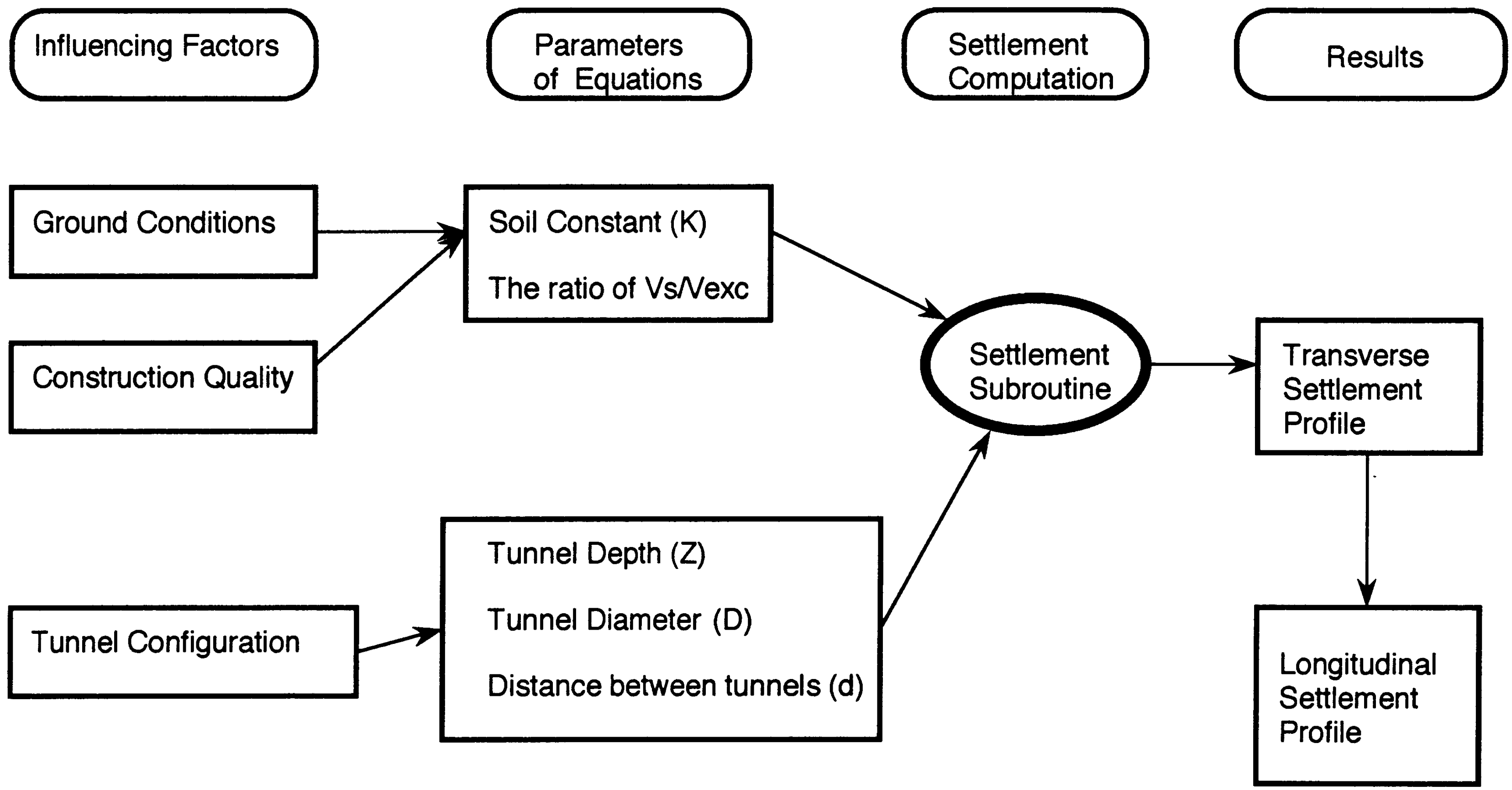
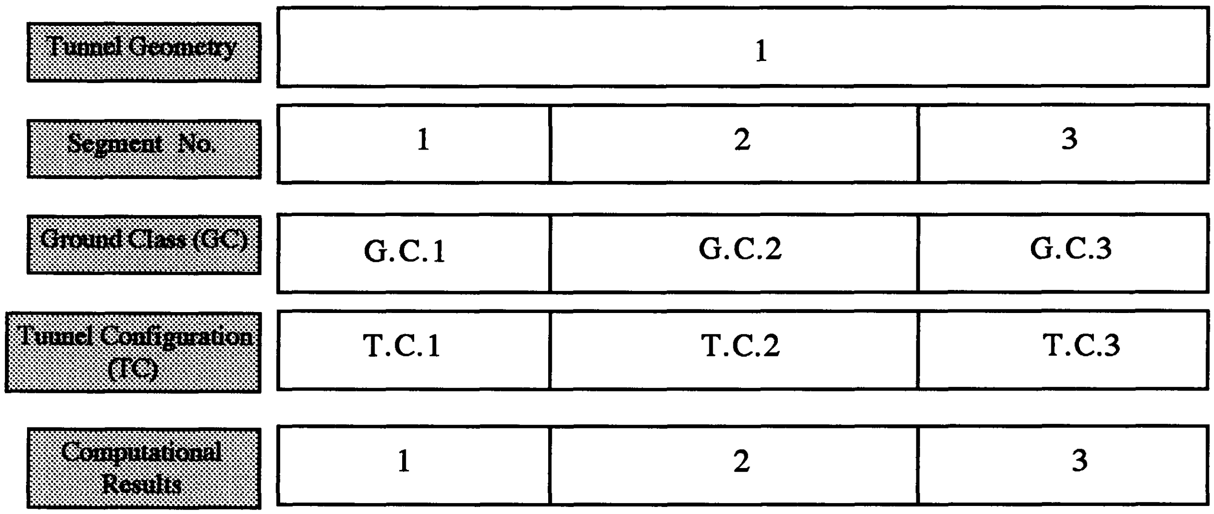


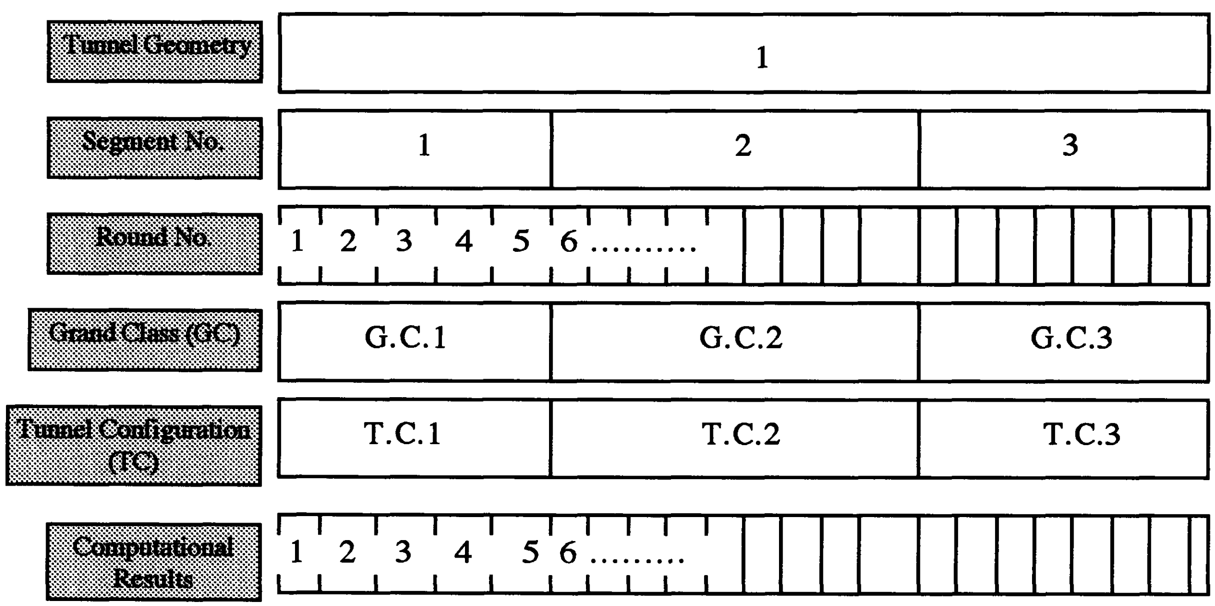
Figure 4.1 Schematic flow of the settlement calculation

Considering the uncertainty of ground conditions and construction procedures through probabilistic analyses can be automatically implemented in SIMSUPER depending on the input data: If the user defines one specified value as a variable, the computation is performed as a deterministic analysis. On the other hand, if the user inputs three values which correspond to the pessimistic, average, and optimistic values, the calculation is performed probabilistically based on a triangular probability distribution function.

Figure 4.2 illustrates the data allocation for both deterministic and probabilistic analyses. In general, the data regarding ground conditions, tunnel configuration, and construction quality are considered by segment which is a unit of tunnel length. A segment is defined by having unique information on the ground class and construction classes. For each calculation, the simulator (SIMSUPER) selects the settlement activity for each ground class and construction class. The computation is performed segment by segment using corresponding input data (see Figure 4.2(a)). If the user provides a distribution of data, the settlement simulation automatically enters into the probabilistic analysis and then proceeds round by round in each segment being assigned a value from the parameter distribution (see Figure 4.2(b)). The procedure is then repeated by simulating the next segment.



(a) Deterministic analysis



(b) Probabilistic Analysis

Figure 4.2 Concept of data allocation in the Settlement model

From this, the settlement model derives the settlement/horizontal displacement profiles in a transverse section as well as along the tunnel length.

Figure 4.3 shows a schematic flow chart of SIMSUPER. In SIMSUPER, the settlement model is established as one of the tunnel activities and the settlement equations are set up as a subroutine in the tunnel activity files (see Figure 4.3). Similarly, the equations regarding cost, time, and resource are set up in the activity files. Furthermore, these activity files are associated with the construction procedures and construction methods.

Since SIMSUPER was originally created to simulate construction processes and resource distribution, the user must provide information on the connections between all tunnel activities. For the computation of settlement, however, the user does not have to define the connections for all tunnel activities because the settlement activity is the only tunnel activity referred to in the settlement model.

Figure 4.4 is an overview of the relationships among all the input files in SIMSUPER. As shown in this figure, the required data files to perform SIMSUPER are roughly composed of five parts: general, time, cost, resource, and tunnel data, since SIMSUPER was originally made to perform time and cost simulations by considering construction process and resource management. For the settlement simulation, although data files regarding time, cost, and resource are assigned to operate SIMSUPER, these data are ignored as dummy data and do not affect the computation results.

For detailed descriptions of the data files, the user should refer to the Programmer's Manual and User's Manual (Vijaya Halabe(1995)).

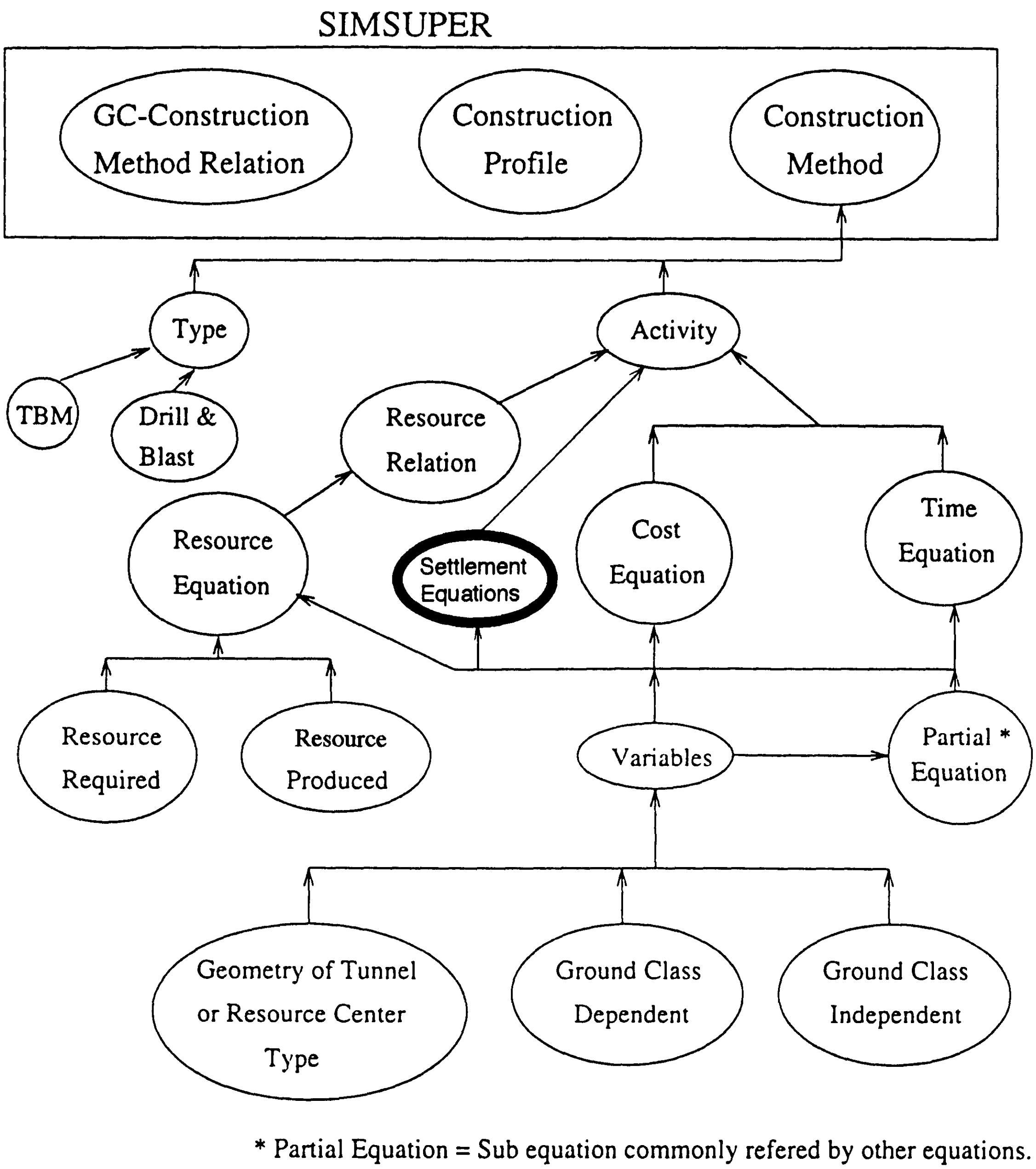


Figure 4.3 Schematic of SIMSUPER (from Halabe (1995))

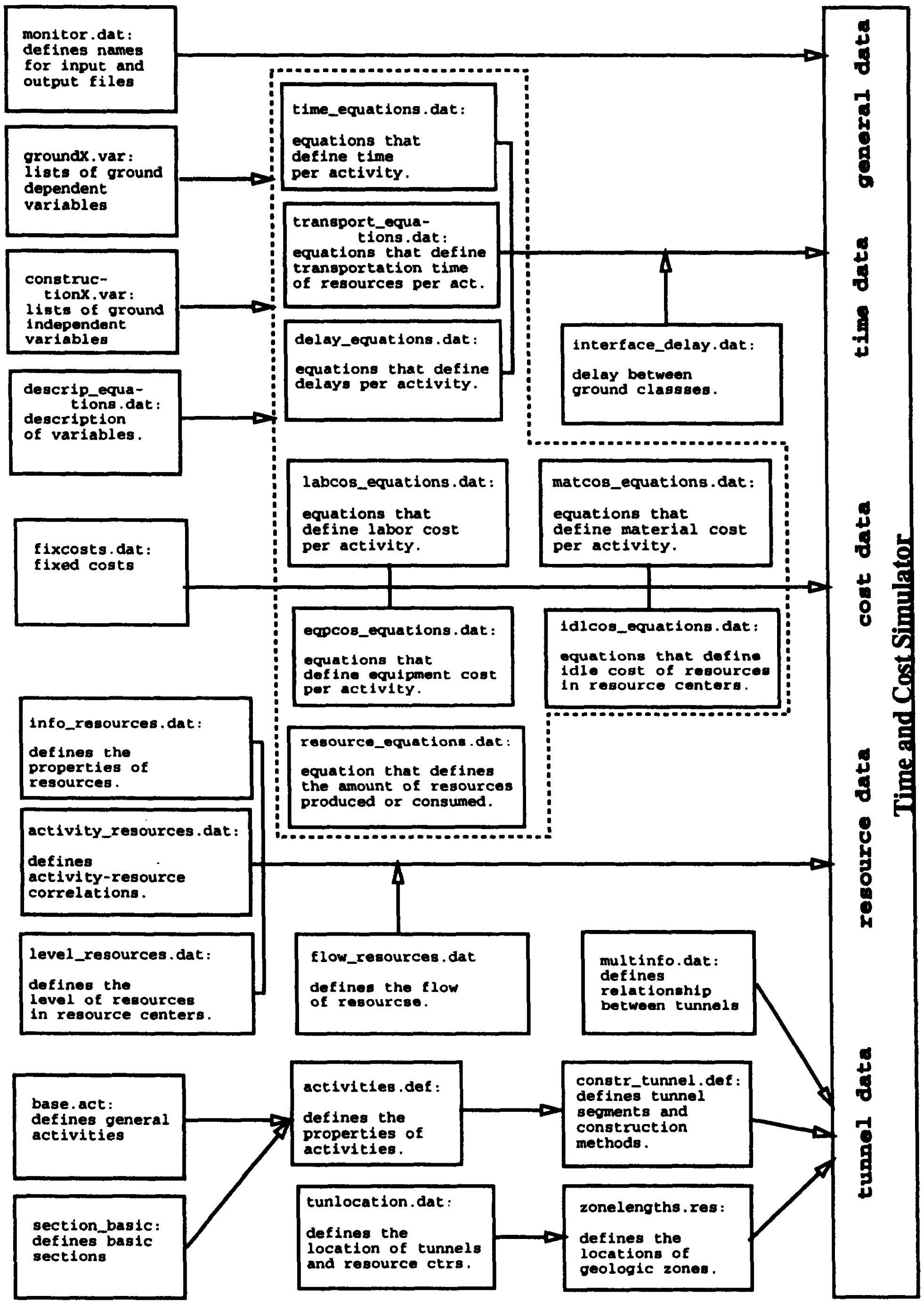


Figure 4.4 Relationship of input file in SIMSUPER (from Halabe (1995))

Chapter 5

User's Manual

5.1 General

This chapter describes all the information necessary to run the settlement program in the DAT. The information given here includes how to install the programs, create input data, start the programs, and obtain the computational results. The input data can be handled through the special program, NETWORK, which is the interface module facilitating access to SIMSUPER (Construction and Resource simulation program).

5.2 Installing the Programs

To run the settlement model in SIMSUPER, the following related data files are installed: SIMSUPER, GEOLOGY, and NETWORK. These procedures are also described in detail in the User's Manual (Halabe (1995)).

5.2.1 Installing the Data Files of SIMSUPER

The following commands are necessary to install the data files of SIMSUPER.

Step	Prompt	Commands	Explanation
1.	athena%	add simsuper	
			Adds the locker simsuper to a workstation so that all the files in the locker become accessible when the

correct pathname of the file is given. The path of locker then will be '/mit/simsuper/'.

2. athena% /mit/simsuper/SIM_COPY

1) Creates the directory ~/simsuper at the top level.

2) Creates three sub directories

~/simsuper/sim.dir.

~/simsuper/geology.dir.

~/simsuper/network.dir.

3) Copies data files necessary for the SIMSUPER program from the locker into the subdirectory ~/simsuper/sim.dir

After installing the SIMSUPER data files as described above, the user should then install NETWORK data files.

5.2.2 Installing the Data Files of NETWORK

The following commands are necessary to install the data files of NETWORK.

Step	Prompt	Commands	Explanation
1.	athena%	/mit/simsuper/NETWORK_COPY	

1) Copies data files necessary for the NETWORK program from the locker into the subdirectory

~/simsuper/network.dir

5.2.3 Installing the Data Files of GEOLOGY

The following commands are necessary to install the data files of GEOLOGY.

Step	Prompt	Commands	Explanation
1.	athena%	/mit/simsuper/GEO_COPY	
		1) Copies data files necessary for the GEOLOGY program from the locker into the subdirectory ~/simsuper/geology.dir	

These procedures create three directories under simsuper and copy data files necessary to run the settlement model as a part of SIMSUPER.

5.3 Starting the Settlement Program

The following commands should be typed in a workstation at MIT to start the settlement program.

athena%	cd /mit/simsuper/sim.dir
	(change directories to SIMSUPER)
athena%	attach X11r5 (for allocating the window system)
athena%	/mit/simsuper/sim.dir/SIMSUPER (starts the program)

These commands will start the settlement program in SIMSUPER. After the program starts and reads the related files, the message below asks the user if the graphic interface is necessary for output data.

Graphic mode (y/n) ?

If the user needs to obtain the computational results by using the graphic interface, type y. The graphic window shown in Figure 5.1 will appear on the screen.

Otherwise, the computational results will be automatically dumped into the assigned output files.

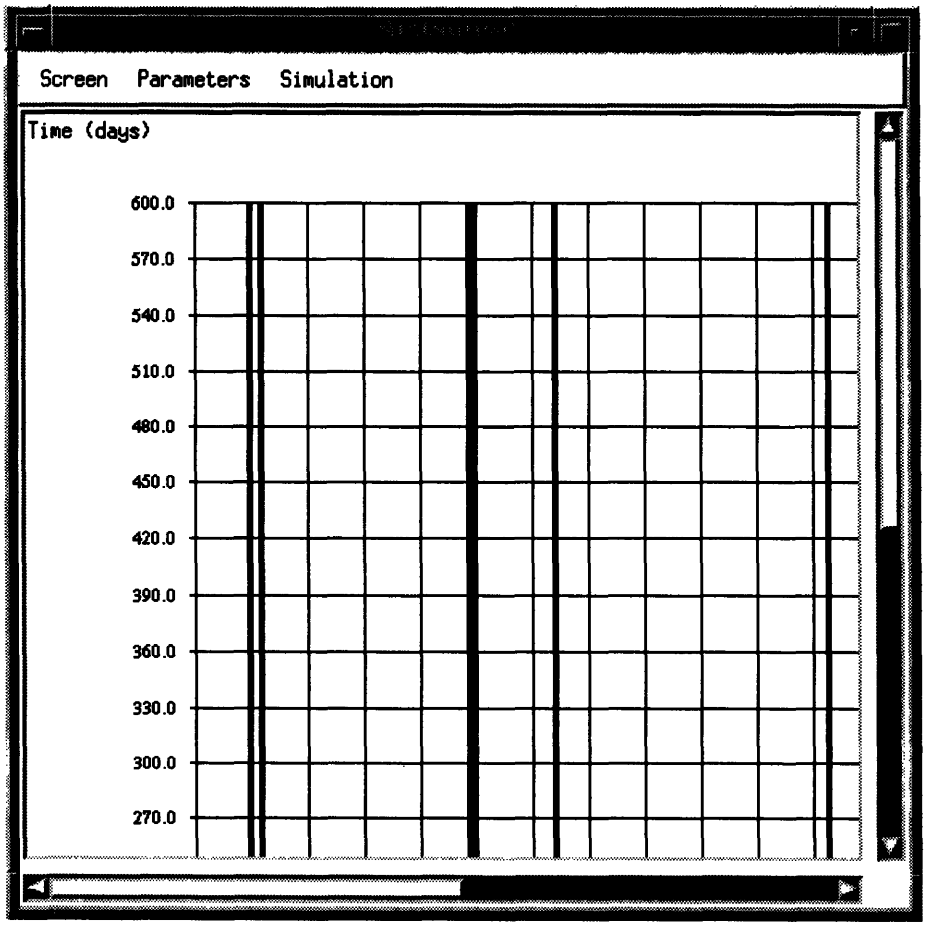


Figure 5.1 Main window for graphic interface

5.4 Input Data Files

Since the simulator files hold information used to perform time and cost simulations, the simulator includes the activities that comprise the construction methods used to build the tunnels, the cost and the time needed to perform the activities, and the geology of the tunnels. As shown in Figure 4.4, these input files are composed of general input files, variable files, cost and time files, activity and section files, tunnel defining files, and output files for SIMSUPER.

Although SIMSUPER contains data files regarding time, cost, and resources, the information is not needed for the settlement simulation. Therefore, the data files regarding cost, time, and resource are defined as dummy data.

In the following, only the data files which the user must modify to run the settlement model are explained. Since the rest of the data files are copied from the original files when the programs are installed, the user does not have to revise these data files. For detailed descriptions of the input data files, the user should refer to both the Programmer's Manual and User's Manual (Halabe(1995)).

5.4.1 General Input File

The monitor.dat file is the main input file. It defines the names of input and output data files and holds general information about the tunnels. These files in Figure 5.3 are required to operate SIMSUPER even though all of the files are not needed for the settlement simulation. The structure of a monitor.dat file appears in Figure 5.2 and a sample file appears in Figure 5.3.

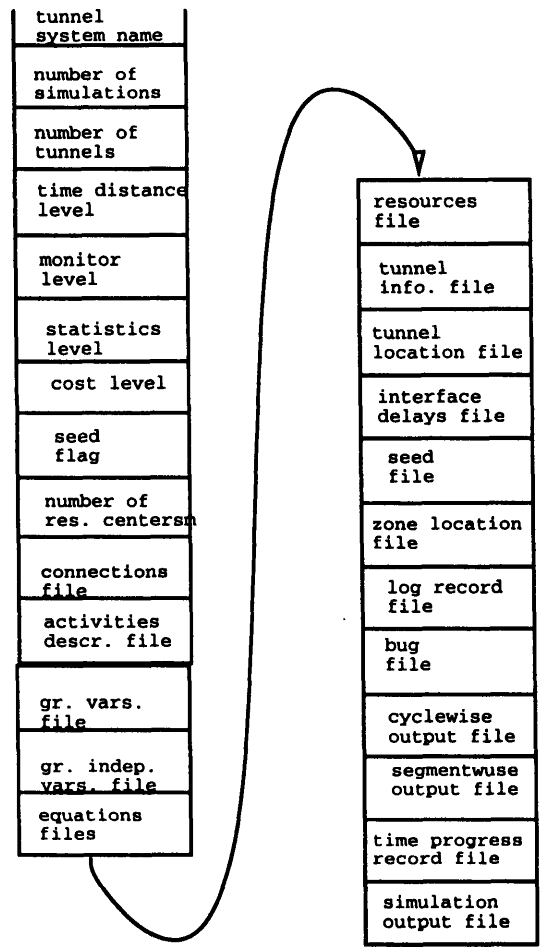


Figure 5.2 Structure of monitor.dat (from Halabe (1995))

```
Settlement Model Sample Program    ; Name of the simulation
1                                  ; #simulations
1                                  ; #tunnels
0                                  ; time-distance level
1                                  ; monitor level (1, 2, or 3)
0                                  ; statistics level (0 or 1)
0                                  ; cost level (0 or 1)
1                                  ; seed flag
0                                  ; number of resource centers
constr_tunnel.def                  ; activity connections input file
activities.def                     ; activity description input file
ground                             ; ground dependent variable input files
construction                       ; ground independent variable files
equations.dat                      ; equation input file
resources.dat                      ; resources input file
multinfo.dat                       ; tunnel description file
../network.dir/tunlocation_new.dat ; tunnel location file
interface_delay.dat                ; interface delay file
../geology.dir/seed                ; seed file
../geology.dir/zonelengths.res     ; zone location data file
log_file                           ; real time output file
test.b                             ; bug output file
/tmp/sim1.rep                      ; roundwise output file
/tmp/sim1.stat                     ; segmentwise output file
/tmp/settle                        ; settlement output file
tun.out                            ; tunnel information output file
```

Figure 5.3 A sample data of monitor.dat

Here, the level of output files shown in Figure 5.3 is described briefly.

(1) time-distance level; The number entered here specifies whether or not the time distance output files will record output in a file named log_file. If the number is zero, no time-distance output will be recorded. The user specifies the number (how many tunnel-distance output wants to record) and the tunnel geometry number (which tunnel he wants to record).

(2) monitor level; This specifies the frequency at which output is presented to the 'time-output file.' If 1 is entered, output appears once per simulation. If 2 is entered, output appears once per round. If 3 is entered, output appears once per activity.

(3) statistics level; The number entered here specifies whether or not the roundwise and segmentwise output files will record output in a file named sim1.rep and sim1.stat

respectively. If 0 is entered, output will not be recorded. If 1 is entered, output will be recorded.

(4) Cost level; The number entered here specifies whether or not the cost output will be recorded. The number can be either 0 or 1 as explained in the statistics level.

5.4.2 Variable Input Files

The variable files representing ground conditions and construction quality contain the ground dependent and ground independent variables used in the settlement model. These files are composed of groundX.var and constructionX.var. The "X" denotes the tunnel geometry type. For example, when the user defines only one geometry, the user must set up ground1.var and construction1.var.

Both groundX.var file and constructionX.var files can be defined in terms of seven types of tunnel geometry.

(1) The groundX.var files

The groundX.var files describe all ground dependent variables for each ground class in a single geometry. The groundX.var file is organized into groups, each of which contains all the ground dependent variables for one ground class. The structure of a groundX.var is shown in Figure 5.4.

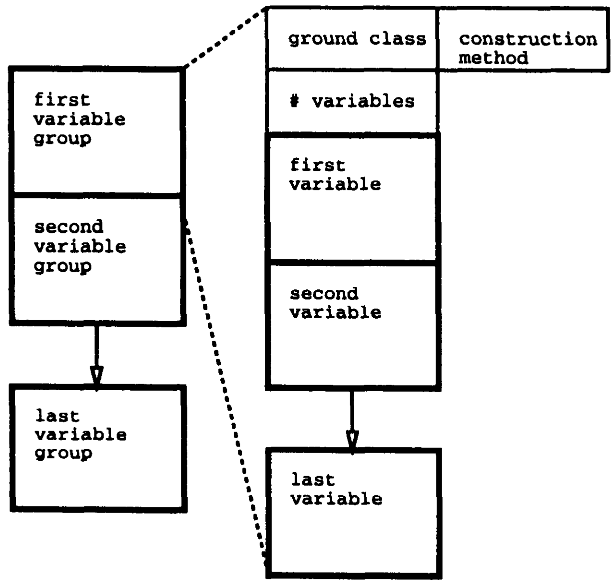
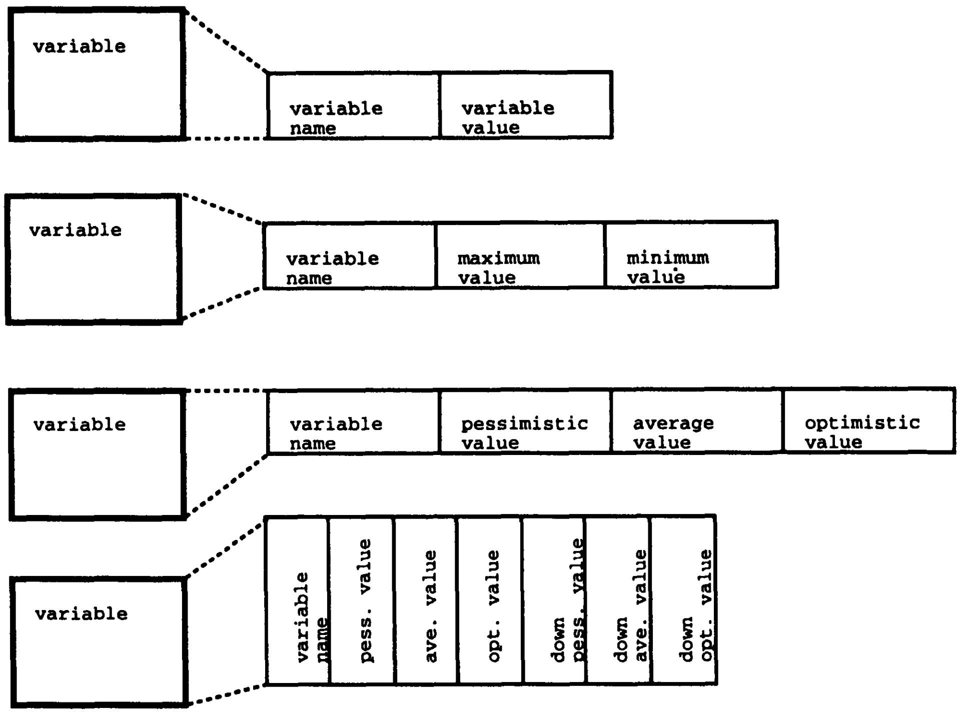


Figure 5.4 groundX.var structure (from Halabe (1995))

The typical structures are shown in Figure 5.5. There are a few types of variable input that determine whether deterministic or probabilistic analysis will be used. The first type of variable input in Figure 5.5 has only one value. That input means the computation is performed as a deterministic analysis by specifying a value. The second type of variable input has two values which are maximum and minimum values and the value for a particular cycle of the simulation is computed from the uniform distribution. The third type of variable input has three values which are pessimistic, average, and optimistic values and the value for a particular cycle of the simulation is computed from the triangular distribution. The groundX.var file is shown for the settlement model in Figure 5.6.



;; If one specifies a single value, then the variable will be of the first type. If one specifies three values, then the variable will be of the third type. If one specifies six values, then the variable will be of the fourth type.

Figure 5.5 Typical variable structures (from Halabe (1995))

```
1       1                                    ;Ground Class/Construction Class
2                                            ;# variables
k_constant        0.40                       ;variables
ratio_volume      0.50                       ;variables
1       2                                    ;Ground C./Construction C.
2                                            ;
k_constant        0.50   0.55   0.60         ;
ratio_volume      1.00   1.75   2.50         ;
1       3                                    ;Ground C./Construction C.
2                                            ;
k_constant        0.60   0.70                ;
ratio_volume      3.00   5.00                ;
1       4                                    ;Ground C./Construction C.
2                                            ;
k_constant        0.20   0.25   0.30         ;
ratio_volume      0.50   0.75   1.00         ;
1       5                                    ;Ground C./Construction C.
2                                            ;
k_constant        0.20                       ;
ratio_volume      1.00                       ;
```

Figure 5.6 groundX.var file

(2) The constructionX.var files

The constructionX.var files hold the variables and values for ground independent variables. Like the groundX.var files, there is a constructionX.var file for each geometry. The structure of a constructionX.var file appears in Figure 5.7.

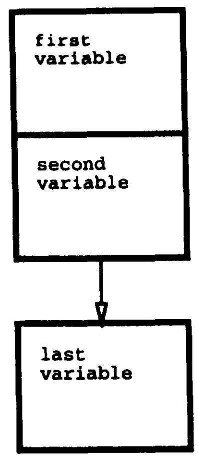


Figure 5.7 The constructionX.var file structure (from Halabe (1995))

The user defines variable name and variable value. Similar to the groundX.var file, the user can determine deterministic and probabilistic analyses by using a few types of variable input (see Figure 5.5). A constructionX.var file shows in Figure 5.8.

```
diameter              10              ;variables
height                50              ;
width                100              ;
dist                  50              ;
round_length           5              ;
```

Figure 5.8 constructionX.var file

5.4.3 Equations Files

Five types of equations exist as follows:

1) Partial equations.

2) Length of the cycle equations.

3) Transportation and time equations.

4) Resource amount equations for labor, material and equipment.

5) Resource cost equations for labor, material, equipment and idle cost including block and fixed costs.

Each of the above equations is defined for a simulation cycle. As shown in Figure 5.9, these equations have two parts. The left hand side of an equation is the value or activity name to be computed and the right hand side is the expression and variables to be used for the computation. The equations are defined in terms of the variables listed in groundX.var and constructionX.var. In the equations dat. files, the user may also define a function (subroutine) instead of an equation.

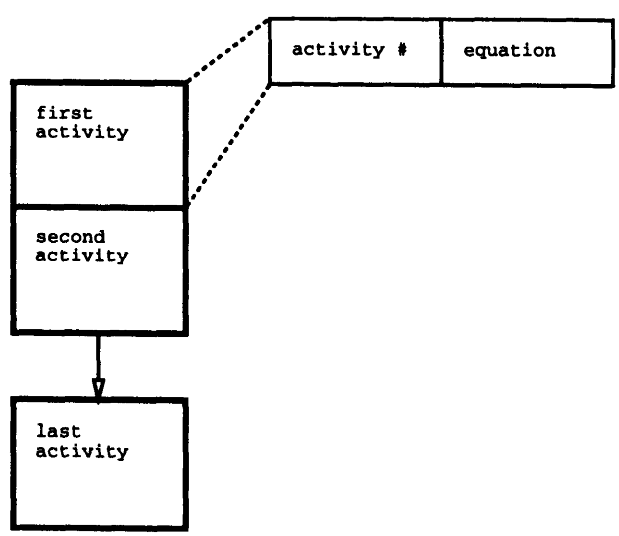


Figure 5.9 equations dat. File structure (from Halabe (1995))

Since SIMSUPER refers to all equations related to activities in running the simulation, the user should define the settlement activity in one of the files regarding cost and time. The rest of the files should be defined as zero so that these data files have no connection with the settlement calculation. Figure 5.10 shows how the settlement activity is assigned to time_equations dat. Figure 5.11 shows a sample data to be defined for the rest of data files.

```
1  Settlement =                                 ;activity number, name
settlement(dist,k_constant,ratio_volume,    ;function (arguments)
           height,diameter,width);
0;                                              ;end
```

Figure 5.10 Time_equations.dat for settlement model

```
1  Settlement = 0;            ;activity number, name, equation
0;                            ;end
```

Figure 5.11 A sample data for the rest of equation data.file

5.4.4 Activities and Section Files

(1) The base.act file

The base.act file defines and assigns numbers to each activity in the tunnel. The structure and a sample file appear in Figure 5.12 and Figure 5.13.

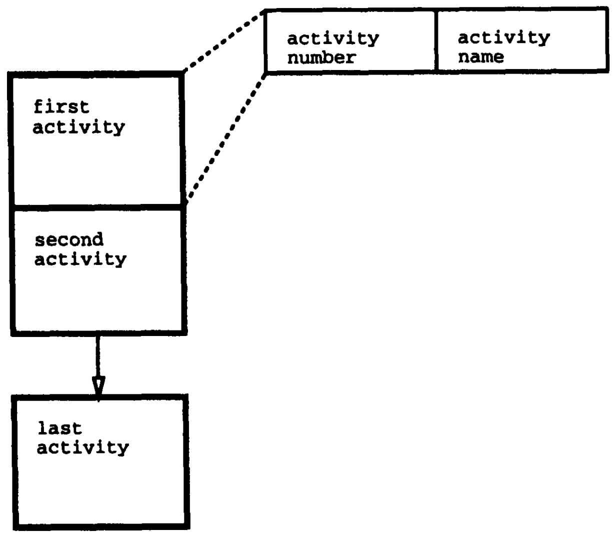


Figure 5.12 base act. File Structure (from Halabe (1995))

1 Probe

2 Drill

3 Load

4 Blast

5 Muck

6 Mapping

7 Geotechnical_observation

12 Steel_sets

8 RockBolts

9 Shotcrete

10 Refill_invert

11 Delays

Figure 5.13 A sample data of base act. file (from Halabe 1995))

(2) The activities.def file

The activities.def file defines tunnel segment activities for the program. This file contains information concerning all activities that appear in the construction method used for tunnels. The structure of this file appears in Figure 5.14.

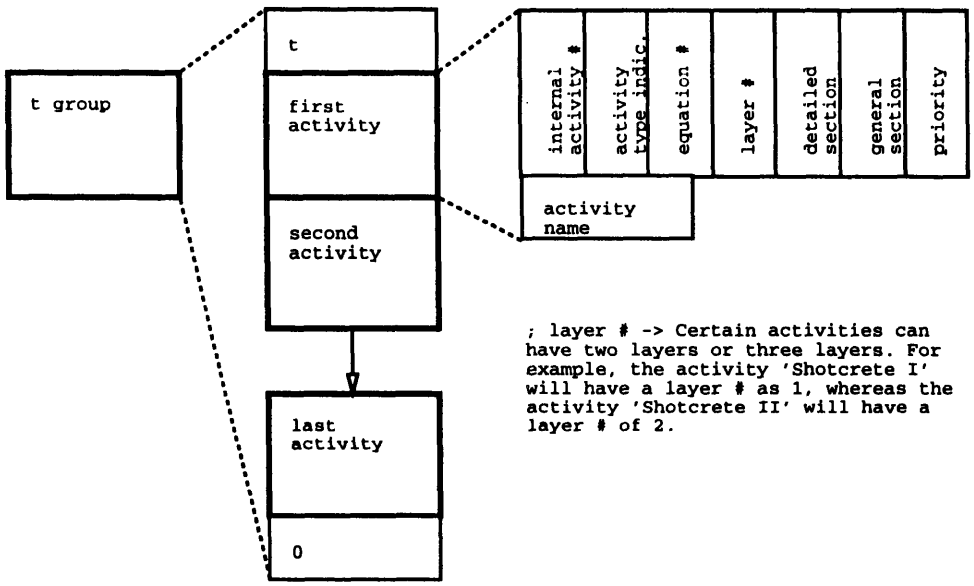


Figure 5.14 activities.def file structure (from Halabe (1995))

The equation number refers to the equation in the base.act file for each activity. The detailed section column indicates the specific subdivisions of the tunnel face for which the activity applies. Figure 5.15 shows a sample activities.def file for the settlement file. For the settlement activity, the only data the user must specify is the equation number used by the settlement activity.

```
 1  1  1  0  1  0  0          ;refer to file structure (in  Fig.5.14)
Settlement N                  ;activity name
 0                            ;end
```

Figure 5.15 A sample data of activities.def file

5.4.5 Tunnel Input Files

The two tunnel input files, constr_tunnel.def and multiinfo.dat, define the tunnels for the simulation. The constr_tunnel.def file defines information of segments and the tunnel activities used for each segment. The multiinfo.dat file defines tunnel types and the relationship between tunnels. The following section describes constr_tunnel.def file.

(1) The constr_tunnel.def file

This file contains information on segments for tunnels and the connections between activities. The connections determine the order of the activities; one specifies which activities precede and follow others. However, in the settlement model, since there is only one activity, the settlement activity, the user does not have to create the data regarding connections. The general structure of the file appears in Figure 5.16.

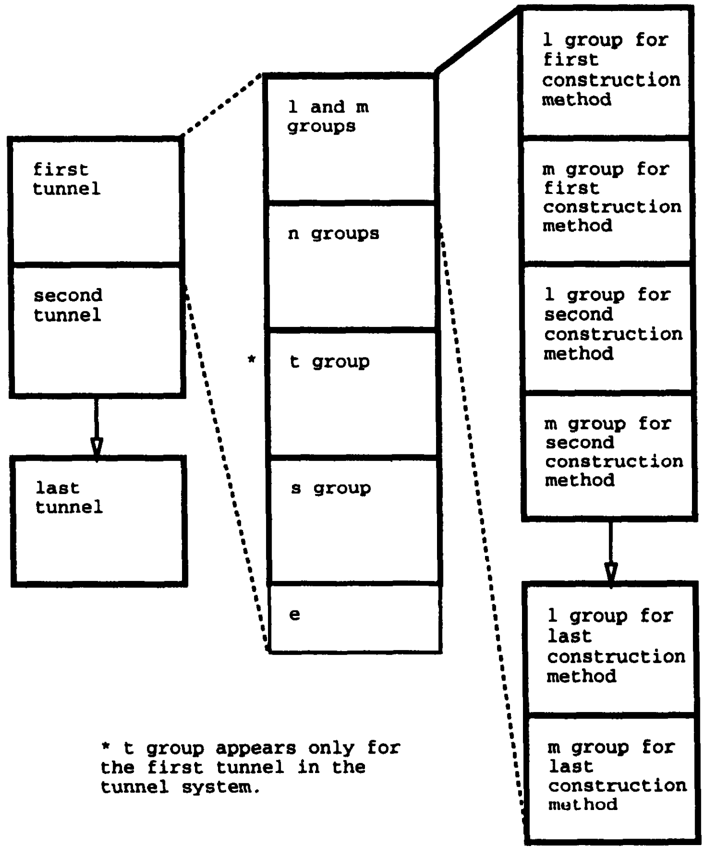


Figure 5.16 constr_tunnel.def file Structure (from Halabe (1995))

The data for each tunnel can be divided into several groups. The ℓ group lists the activities in a construction method. The m group defines the connection for the activities in the ℓ groups. The s group defines the segment of a tunnel. The e flag indicates the end of a tunnel's data. The following four subsections detail each of the above groups.

(i) The ℓ group

The ℓ group is a list of activities that are used in one construction method. The structure of the group appears in Figure 5.17.

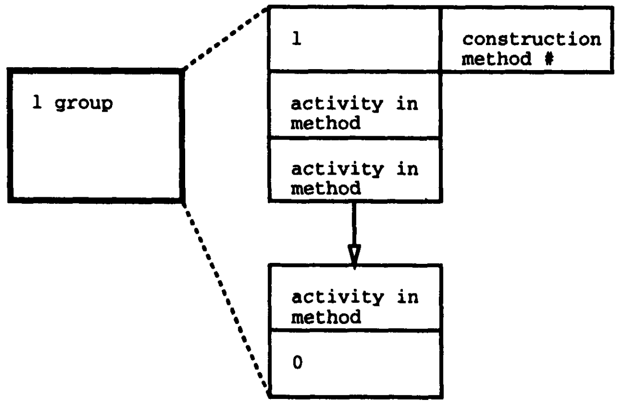


Figure 5.17 ℓ group structure (from Halabe (1995))

(ii) The m group

The m group defines the connection between activities for a construction method. The structure for m group appears in Figure 5.18.

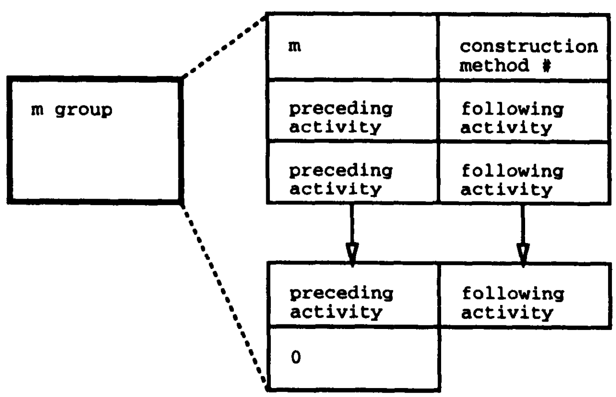


Figure 5.18 m group structure (from Halabe (1995))

Figure 5.20 shows sample data of ℓ and m groups for the settlement model.

(iii) The s group

The s group contains information that defines the segments of the tunnel. It is structured according to Figure 5. 19.

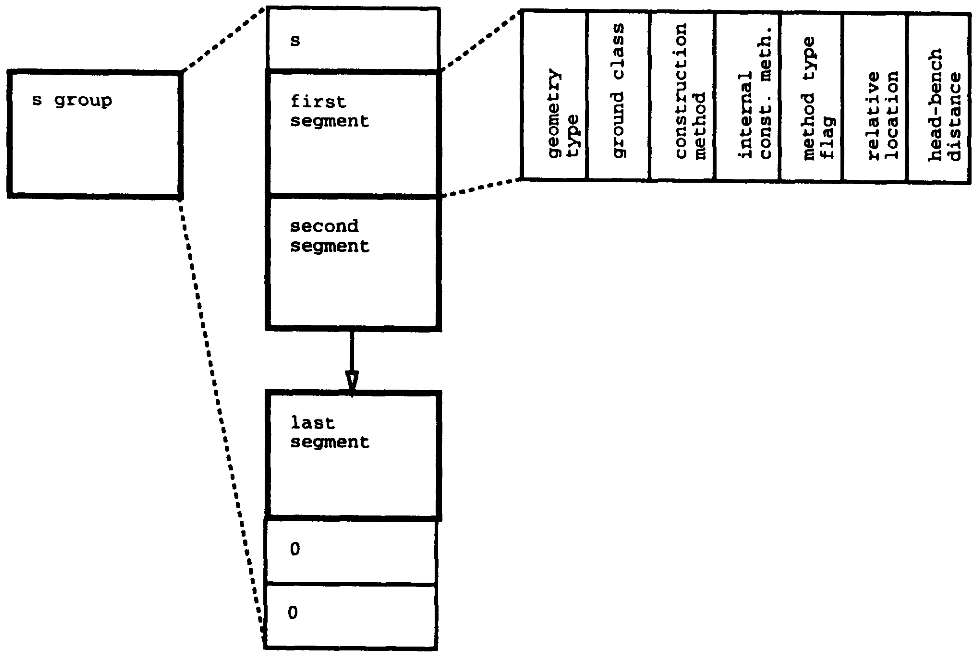


Figure 5.19 s group structure (from Halabe (1995))

The third column, the construction method, indicates the number of the construction method as it is defined in the tunlocation.dat file and the groundX.var files. The relative location column indicates length measured from the beginning of the tunnel. The internal construction method is the number of the construction activity as defined in the ℓ and m groups of this file. A sample s group appears in Figure 5.20.

```
l 1                                  ;l group construction method
  1                                  ;activity number
  0                                  ;end
m  1                                 ;m group construction method
  0                                  ;end (no preceding and following activity)
s                                    ;s group
  1  1  3  1  0    20  0             ;refer to s group structure (in Fig. 5.19)
  1  1  2  1  0   400  0             ;
  1  1  4  1  0   900  0             ;
  1  1  5  1  0  1200  0             ;
  1  1  3  1  0  1400  0             ;
  1  1  1  1  0  1800  0             ;
  1  1  6  1  0  2200  0             ;
  1  1  5  1  0  2500  0             ;
  1  1  2  1  0  3000  0             ;
  0                                  ;end
  0                                  ;end
e                                    ;e flag for end
```

Figure 5.20 A sample data of ℓ, m, and s group

(iv) The e group

The e flag appears after the s group and marks the end of the data for a tunnel. The l group of the next tunnel should follow the e flag.

5.4.6 GEOLOGY output files

(1) Tunlocation.dat file

Tunlocation.dat file is the link between the geology output and the SIMSUPER input file. SIMSUPER relates ground classes to the relative location in the tunnel. Each tunnel is defined in tunlocation.dat file as a portion of one area. This file also contains data that assign a construction method to each ground class. The construction method number corresponds to the method number defined in the groundX.var file. The structure of the file appears in Figure 5.21. A sample file appears in Figure 5.22.

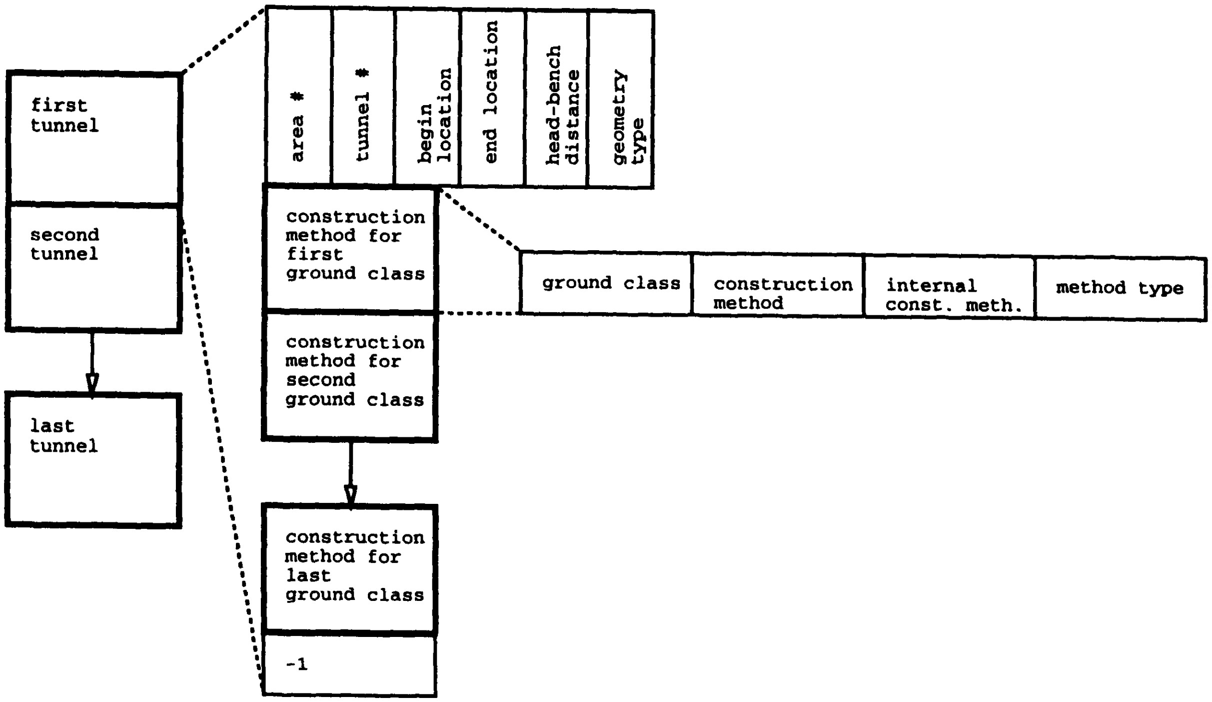


Figure 5.21 tunlocation.dat file structure (from Halabe (1995))

```
Area    Tunnel No.   Begin Point      End Point    HBDist    GeomType
                      x  y  z          x     y  z
1          1          0  0  0        10000   0  0     0          1
Ground Class    Method#    Internal Mehod#    Method Type
1                 1            1               0
1                 2            1               0
1                 3            1               0
1                 4            1               0
1                 5            1               0
-1
```

Figure 5.22 A sample data of tunlocation.dat file (from Halabe (1995))

5.5 Outputs

5.5.1 File Outputs

The outputs of the settlement simulation contain the results of maximum settlement and horizontal displacement in the longitudinal section as well as the settlement and horizontal displacement in the transverse section. There are two output files: Settle_seg.out and Settle.X_X.out. Settle_seg.out file shows the summary results of the segments such as Maximum, Minimum, and Average settlement/horizontal displacement. Settle.X_X.out file describes the transverse profiles of settlement/horizontal displacement in a segment.

(1) Settle_seg.out file

The Settle_seg.out file is the segmentwise output file. The file provides the computation results of each segment of each tunnel geometry. For each segment of each tunnel, the file includes the following information (see Figure 5.23). Figure 5.24 shows parts of Settle_seg.out file.

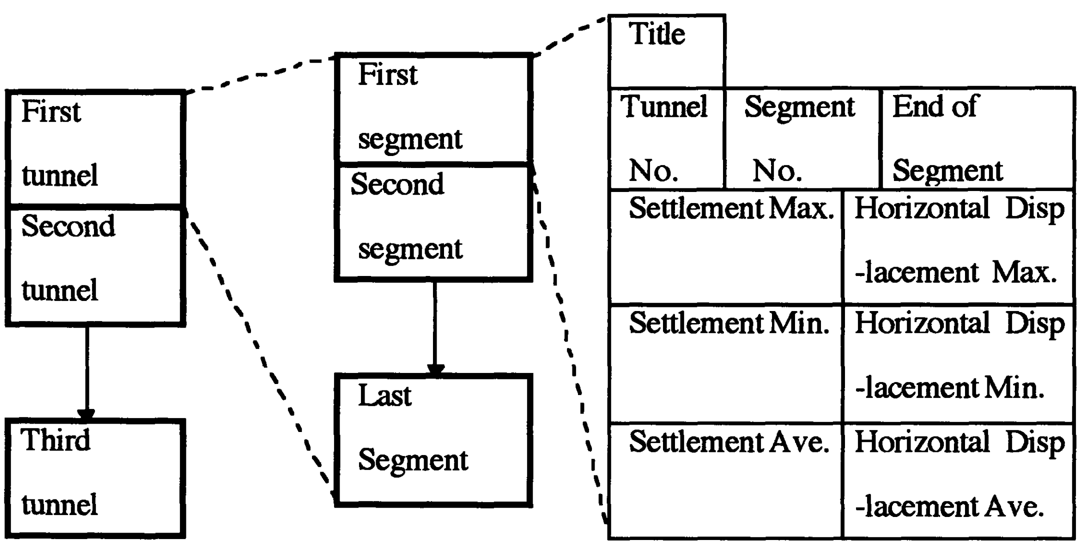


Figure 5.23 Settle_seg.out file structure

```
COMPUTATIONAL RESULTS for SEGMENTS
 Tunnel No.=  1  Segment No=  1  End Location (m)=  20.00
seg_settlement_vmax (m)=    0.060   seg_settlement_hmax (m)=    0.020
seg_settlement_vmin (m)=    0.047   seg_settlement_hmin (m)=    0.015
seg_settlement_vave (m)=    0.052   seg_settlement_have (m)=    0.017

COMPUTATIONAL RESULTS for SEGMENTS
 Tunnel No.=  1  Segment No=  2  End Location (m)= 400.00
seg_settlement_vmax (m)=    0.035   seg_settlement_hmax (m)=    0.010
seg_settlement_vmin (m)=    0.015   seg_settlement_hmin (m)=    0.004
seg_settlement_vave (m)=    0.026   seg_settlement_have (m)=    0.007

COMPUTATIONAL RESULTS for SEGMENTS
 Tunnel No.=  1  Segment No=  3  End Location (m)= 900.00
seg_settlement_vmax (m)=    0.028   seg_settlement_hmax (m)=    0.004
seg_settlement_vmin (m)=    0.012   seg_settlement_hmin (m)=    0.002
seg_settlement_vave (m)=    0.019   seg_settlement_have (m)=    0.003

COMPUTATIONAL RESULTS for SEGMENTS
 Tunnel No.=  1  Segment No=  4  End Location (m)= 1200.00
seg_settlement_vmax (m)=    0.065   seg_settlement_hmax (m)=    0.009
seg_settlement_vmin (m)=    0.031   seg_settlement_hmin (m)=    0.005
seg_settlement_vave (m)=    0.046   seg_settlement_have (m)=    0.007

COMPUTATIONAL RESULTS for SEGMENTS
 Tunnel No.=  1  Segment No=  5  End Location (m)= 1400.00
seg_settlement_vmax (m)=    0.066   seg_settlement_hmax (m)=    0.022
seg_settlement_vmin (m)=    0.045   seg_settlement_hmin (m)=    0.015
seg_settlement_vave (m)=    0.057   seg_settlement_have (m)=    0.018

COMPUTATIONAL RESULTS for SEGMENTS
 Tunnel No.=  1  Segment No=  6  End Location (m)= 1800.00
seg_settlement_vmax (m)=    0.046   seg_settlement_hmax (m)=    0.012
seg_settlement_vmin (m)=    0.012   seg_settlement_hmin (m)=    0.003
seg_settlement_vave (m)=    0.027   seg_settlement_have (m)=    0.007
```

Figure 5.24 parts of Settle_seg.out

(2) Settle.X_X.out

The Settle.X_X.out file is the roundwise output file in a segment. The file provides the transverse profiles of settlement/horizontal displacement. The first X in this file name denotes the tunnel geometry number and the second X denotes the segment number. The structure of the file appears in Figure 5.25. A sample result is shown in Figure 5.26.

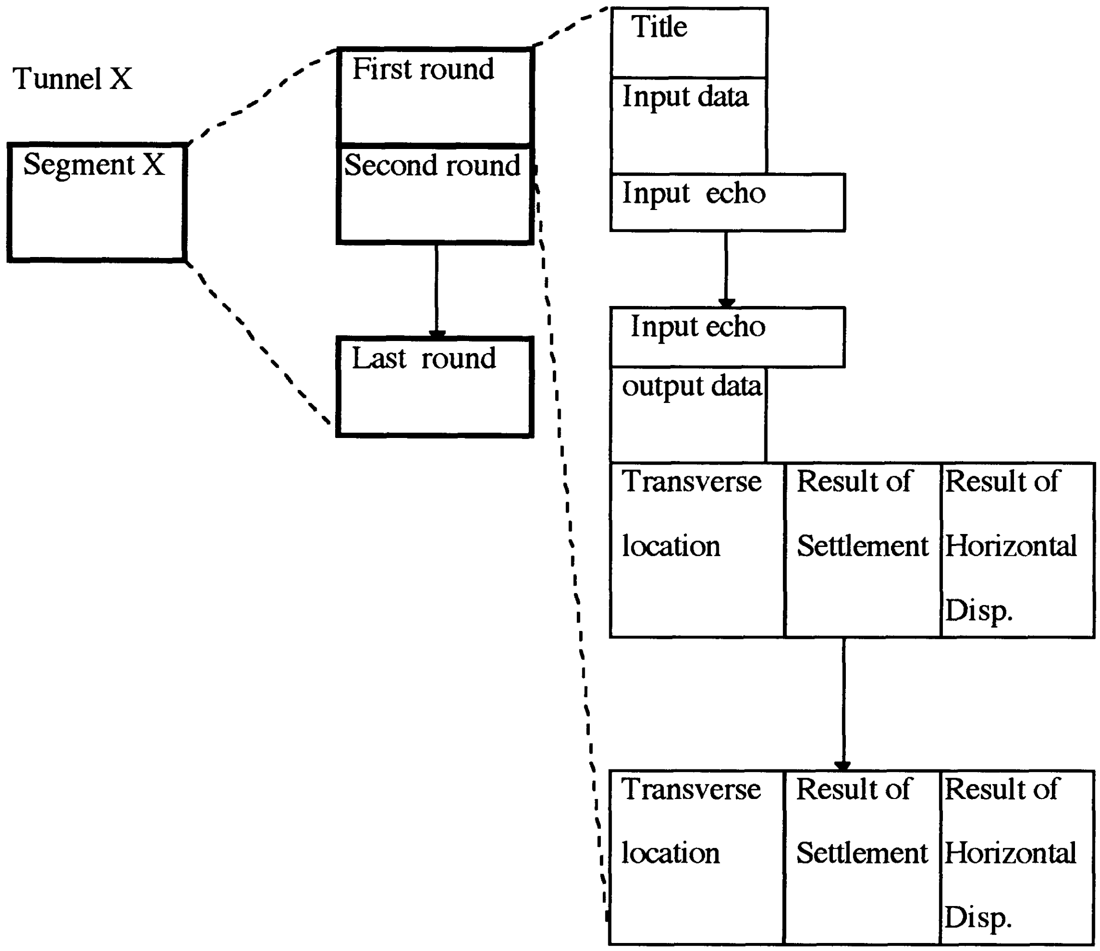


Figure 5.25 Settle.X_X.out file structure

```
CALCULATION of TRANSVERSE PROFILE     ;Transverse Profile (Probabilistic)

INPUT_DATA                            ;Input data echo
 k_constant =  0.69                   ;Soil constant K
 Ratio_volume(%) =  4.33              ;Ratio of Vs/Vexc
 Depth(m) = 50.00                     ;Tunnel depth
 Diameter(m) = 10.00                  ;Tunnel diameter
 Distance(m) = 50.00                  ;Distance between tunnels (Twin)
 Range for Calculation(m) = 100.0     ;Transverse range for calculation
 Location(m) =    5.00                ;End location of 1 round

OUTPUT                                                    ;Output
Trans Loc.(m)   Settlement(m)   H.Displacement(m)         ;Title
  -100.00         0.00057         -0.00115                ;results
   -99.00         0.00062         -0.00124                ;
   -98.00         0.00068         -0.00133                ;
   -97.00         0.00074         -0.00143                ;
   -96.00         0.00080         -0.00154                ;
   -95.00         0.00087         -0.00165                ;
   -94.00         0.00094         -0.00177                ;
   -93.00         0.00102         -0.00190                ;
```

Figure 5.26 Parts of Settle.X_X.out file

5.5.2 Graphic Outputs

In order to obtain the computational results in the form of plots, the user must use the graphic interface as described in 5.3. These plots are displayed in color on the screen of the workstation. The following is the procedure to obtain the graphic output.

After the main window in Figure 5.1 appears, the user clicks **Screen** from the top menu bar in the main window to define the parameters of the plots. The **Screen** pane shows up (see Figure 5.27). In that pane, the user selects **Parameter** and the sub-pane appears for defining either the dynamic or static option (see Figure 5.27). Dynamic option means items contained in the pane (Figure 5.27) can be freely modified at any time during the simulation. To start the settlement calculation, the user clicks **Dynamic** option.

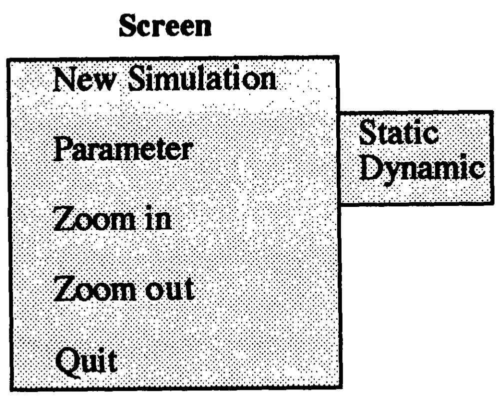


Figure 5.27 Contents of screen pane

Soon after clicking **Dynamic**, the next pane to define the scale of the plots appears as follows (see Figure 5.28):

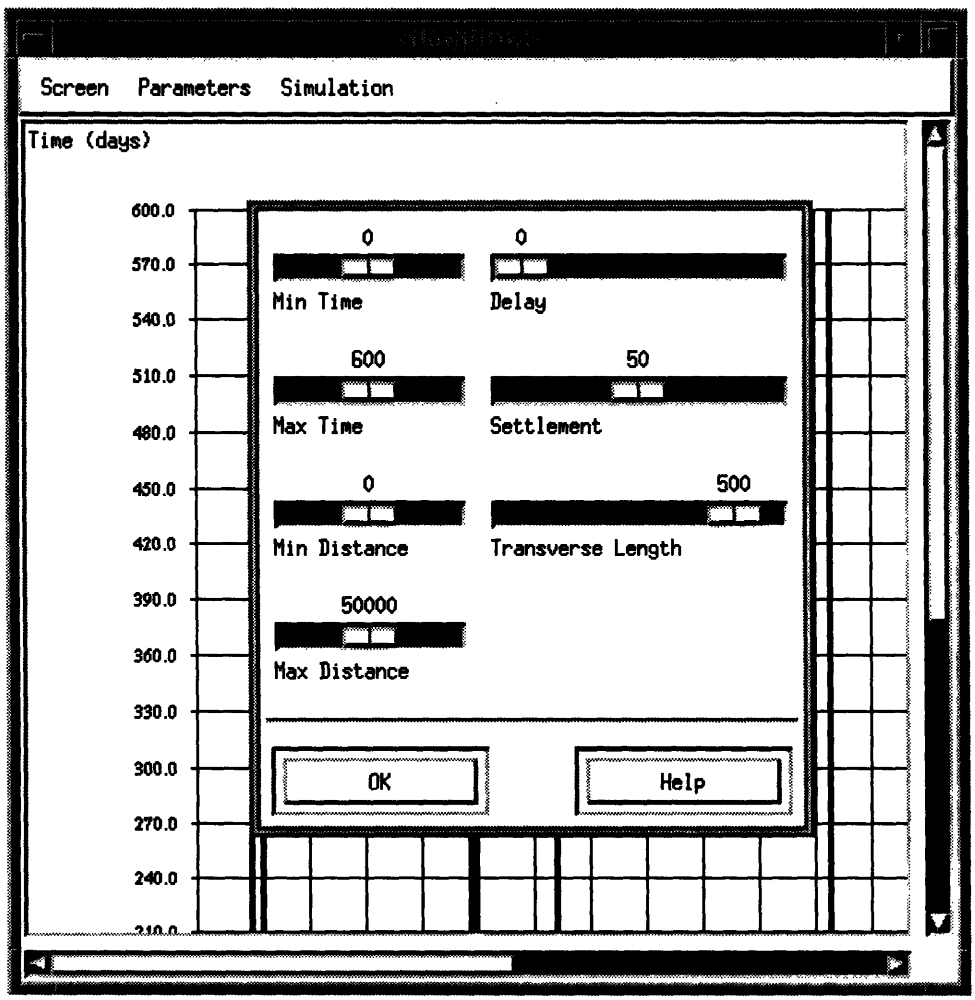


Figure 5.28 pane for defining scale of plot

In this pane, there are three buttons which the user must move to change the range with each calculation; Max Distance, Settlement, and Transverse length. Max Distance denotes the maximum length of the tunnel profile to be displayed in the plot. Settlement represents the expected maximum settlement or horizontal displacement on the plot. Transverse length expresses the length in the transverse profile plot. These values can be modified by either dragging the mouse cursor or pressing the up/down arrow key on the keyboard. After establishing the scale of the plot, the **OK** button is clicked to complete the procedure.

For the next procedure, the user returns to the top menu bar in the main window and clicks the **Parameters** button (see Figure 5.29). This pane shows the items which the user expects to obtain from the computation. In this pane, **Settle** (settlement) is selected to obtain the result of the settlement model. Soon after selecting the **Settle**, both the longitudinal and transverse profile plots appear next to the main window (see Figure 5.30).

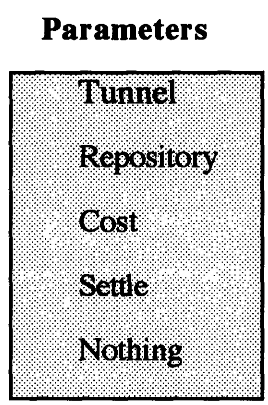


Figure 5.29 Contents of Parameter pane

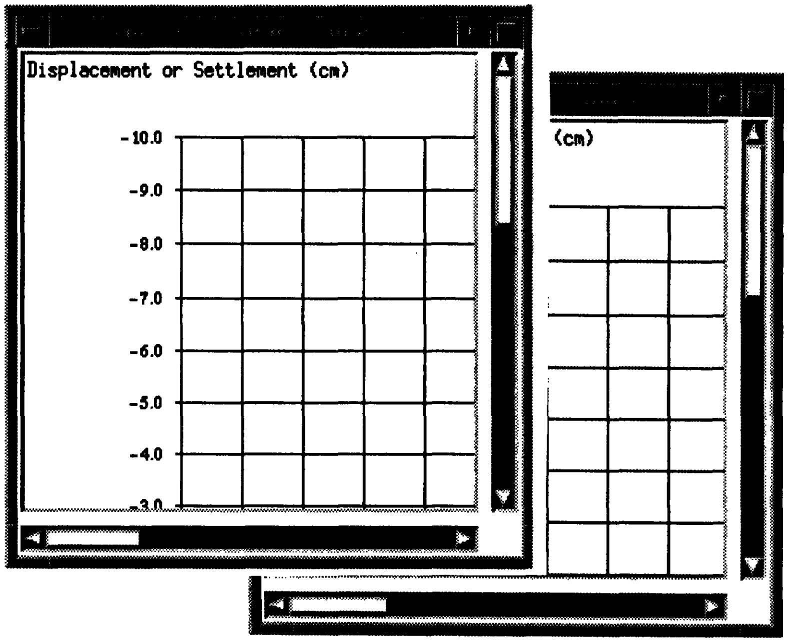


Figure 5.30 window of the settlement plot

The computation will start by clicking **Start** in **Simulation** from the top menu bar in the main window.

Simulation

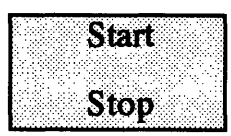


Figure 5.31 Contents of Simulation pane

Finally, the computational results will appear on the settlement plot. Once the calculation starts, the user should expand the longitudinal profile plot by dragging the corner of the pane to display the entire plot (see Figure 5.32).

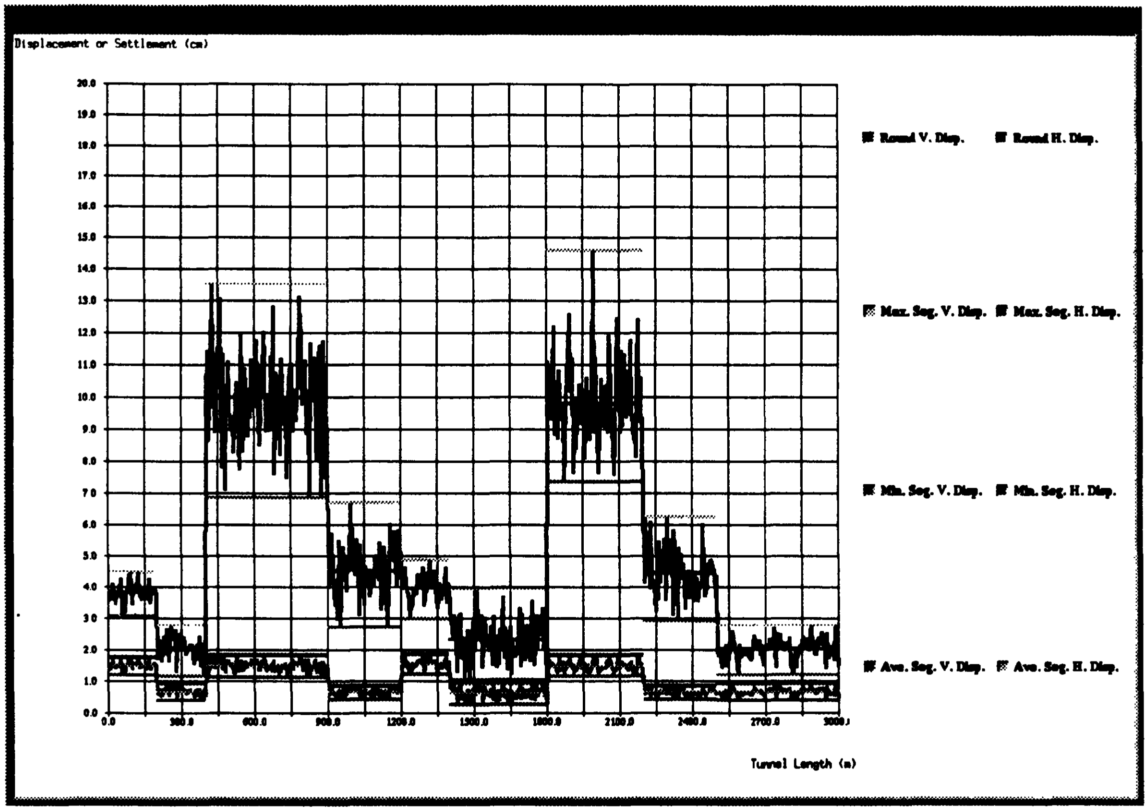


Figure 5.32 A sample plot of a longitudinal profile

Figure 5.32 shows maximum surface settlement and horizontal displacement versus tunnel length. Additionally, the maximum, average, and minimum values in a segment are indicated by horizontal lines. If the user chooses to erase an item of these results, the toggle button in front of each item should be clicked.

Furthermore, when considering a transverse profile at some locations in the tunnel, the user is required to click on the corresponding location on the longitudinal profile screen. The transverse profile corresponding to the point is automatically selected and appears on the transverse profile plot (see Figure 5.33). This transverse plot is capable of displaying both settlement and horizontal displacement at up to five locations. If the user chooses to erase one of these results, the toggle button in front of particular result should be clicked.

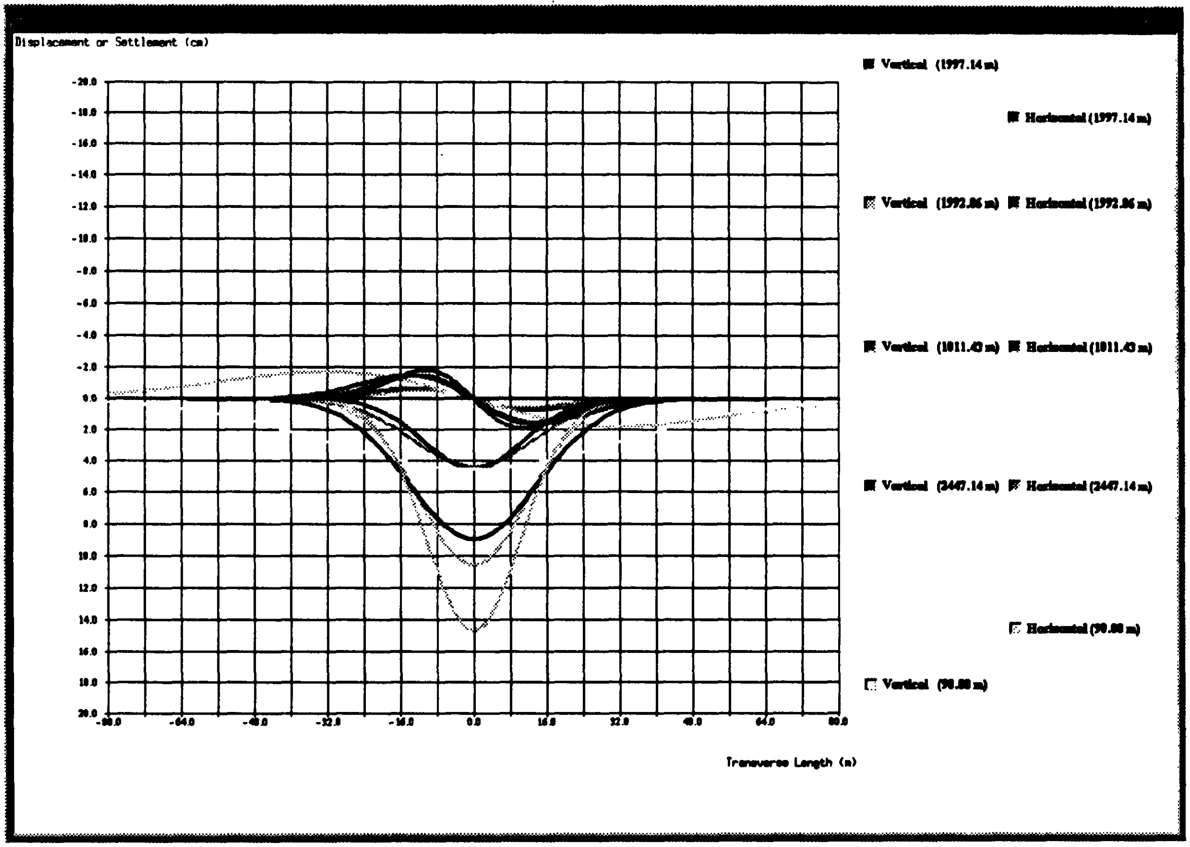


Figure 5.33 A sample plot of a transverse profile

Chapter 6
Case Studies

6.1 General

6.1.1 General

The intent of this chapter is to discuss the possibility to realistically simulate ground settlement by means of the settlement model. In the process of tunnel planning, preliminary settlement computations are performed to predict the potential extent and amount of ground displacement. Such predicted settlement profiles can then be used to determine the next appropriate procedures in a tunnel project.

As case studies, the calculations for two geometries, a single tunnel and twin tunnels with the same diameter, are chosen. Before starting an estimation of settlement, the user should select either the deterministic or the probabilistic analysis depending on the uncertainty of the given geologic and construction information.

In this chapter, the following cases shown in Table 6.1 are considered.

Table 6.1 Case studies

Case	Tunnel type	Analysis type	The distance between two tunnels (m)	Tunnel diameter (m)	Range of tunnel depth (m)
1	Single tunnel	Deterministic	_	5	10 - 30
2	Single tunnel	Probabilistic	_	5	10 - 30
3	Twin tunnels	Probabilistic	50	10	50
4	Twin tunnels	Probabilistic	20	10	50

6.1.2 Procedures for Computation

In general, the user performs the following procedures to compute the amount of settlement. 1) Before starting the calculation, the information regarding geologic/geographic conditions, tunnel layout, and construction procedures is collected. 2) After considering geologic conditions and tunnel configurations, ground conditions and tunnel configurations are set up along a planned tunnel as shown in Figure 6.1. 3) The user defines the input data by reflecting the uncertainty of ground conditions. The input data on ground conditions are given by referring to Table 4.2 (Values for K and the ratio of V_s/V_{exc}) while the data for the tunnel configurations are taken directly from the geometry. Provided that the construction quality of the planned tunnel is still uncertain, the user should employ the probabilistic analysis using a distribution of data.

6.2 Single Tunnel

6.2.1 Case Study I (Deterministic Analysis)

(1) Layout of Tunnel

As Case study I, the following tunnel layout is considered.

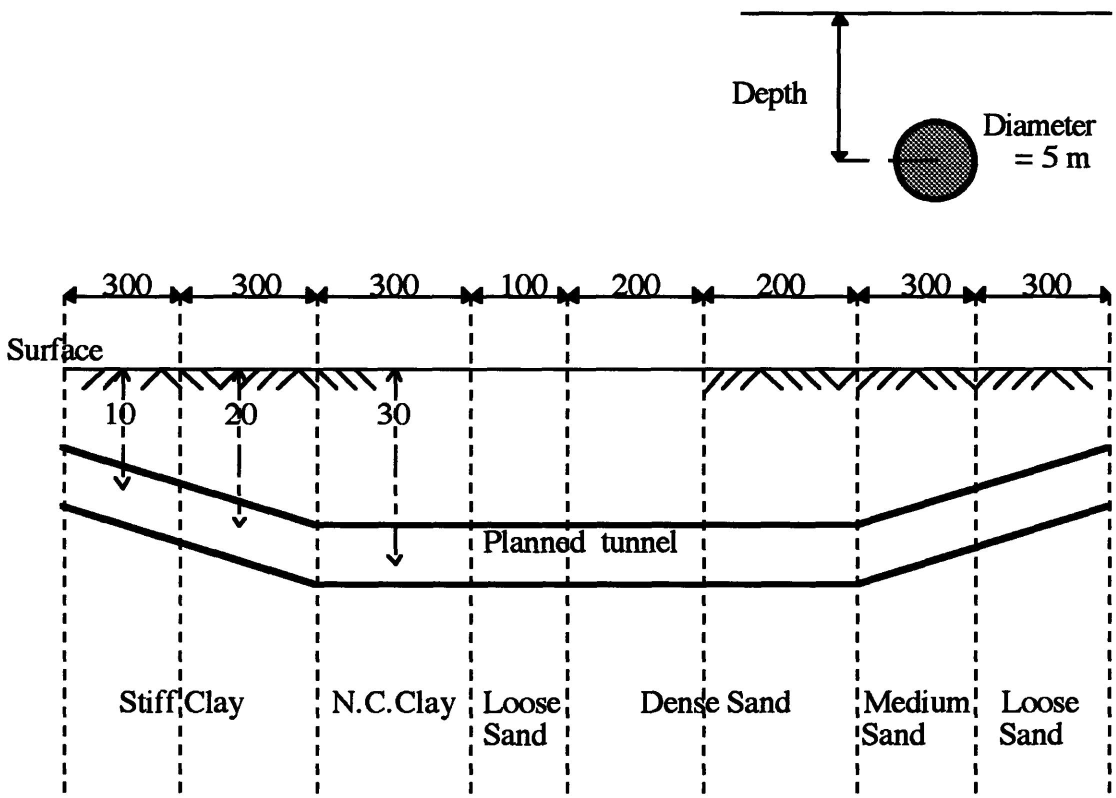


Figure 6.1 Layout of planned tunnel (Case Study I) (unit: m)

(2) Input Data

Table 6.2 summarizes the ground conditions, tunnel configuration, and construction quality of the planned tunnel based on Figure 6.1. Table 6.3 shows the input data for the settlement model based on Table 6.2. Since the input for K and the ratio of V_S/V_{exc} are specified as single values, SIMSUPER automatically performs the deterministic analysis.

Table 6.2 Ground Conditions and Tunnel Configurations (Case Study I)

Segment No.		1		2		3		4		5		6		7		8	
Tunnel Length (m)	0		300		600		900		1000		1200		1400		1700		2000
Tunnel Diameter (m)		5		5		5		5		5		5		5		5	
Tunnel Depth (m)		10		20		30		30		30		30		20		10	
Segment Length (m)		300		300		300		100		200		200		300		300	
Soil Type		Stiff Clay		Stiff Clay		N. C. Clay		Loose Sand		Dense Sand		Dense Sand		Medium sand		Loose sand	
Construction Quality		Good		Poor		Average		Average		Average		Poor		Poor		Good	

Table 6.3 Input Data for the Settlement Model (Case Study I)

Segment No.		1		2		3		4		5		6		7		8	
Tunnel Length (m)	0		300		600		900		1000		1200		1400		1700		2000
Soil Type		Stiff Clay		Stiff Clay		N. C. Clay		Loose Sand		Dense Sand		Dense Sand		Medium sand		Loose Sand	
Construction Quality		Good		Poor		Average		Average		Average		Poor		Poor		Good	
Parameter K		0.45		0.45		0.55		0.25		0.25		0.25		0.25		0.25	
Ratio of Vs/Vexc(%)		0.50		3.00		1.75		4.00		0.75		1.00		2.50		3.00	

(3) Computational Results

Figure 6.2 shows the longitudinal profile of maximum surface settlement and horizontal displacement along the planned tunnel. In this figure, the horizontal axis indicates the tunnel length and the vertical axis displays the amount of settlement and horizontal displacement. Since the input data are assigned by segment, the computational results are also provided by segment. In this analysis, as the analysis is deterministic, the maximum, minimum, and average segment values are all the same.

Figure 6.3 also represents the transverse profiles of surface settlement and horizontal displacement in all segments. In this figure, the horizontal axis indicates the transverse distance and the vertical axis displays the amount of settlement and horizontal displacement. This figure can display information on up to five locations. Each transverse profile is distinguished by the clicked location, which is shown in the parentheses after each toggle button, as well as by the colored lines.

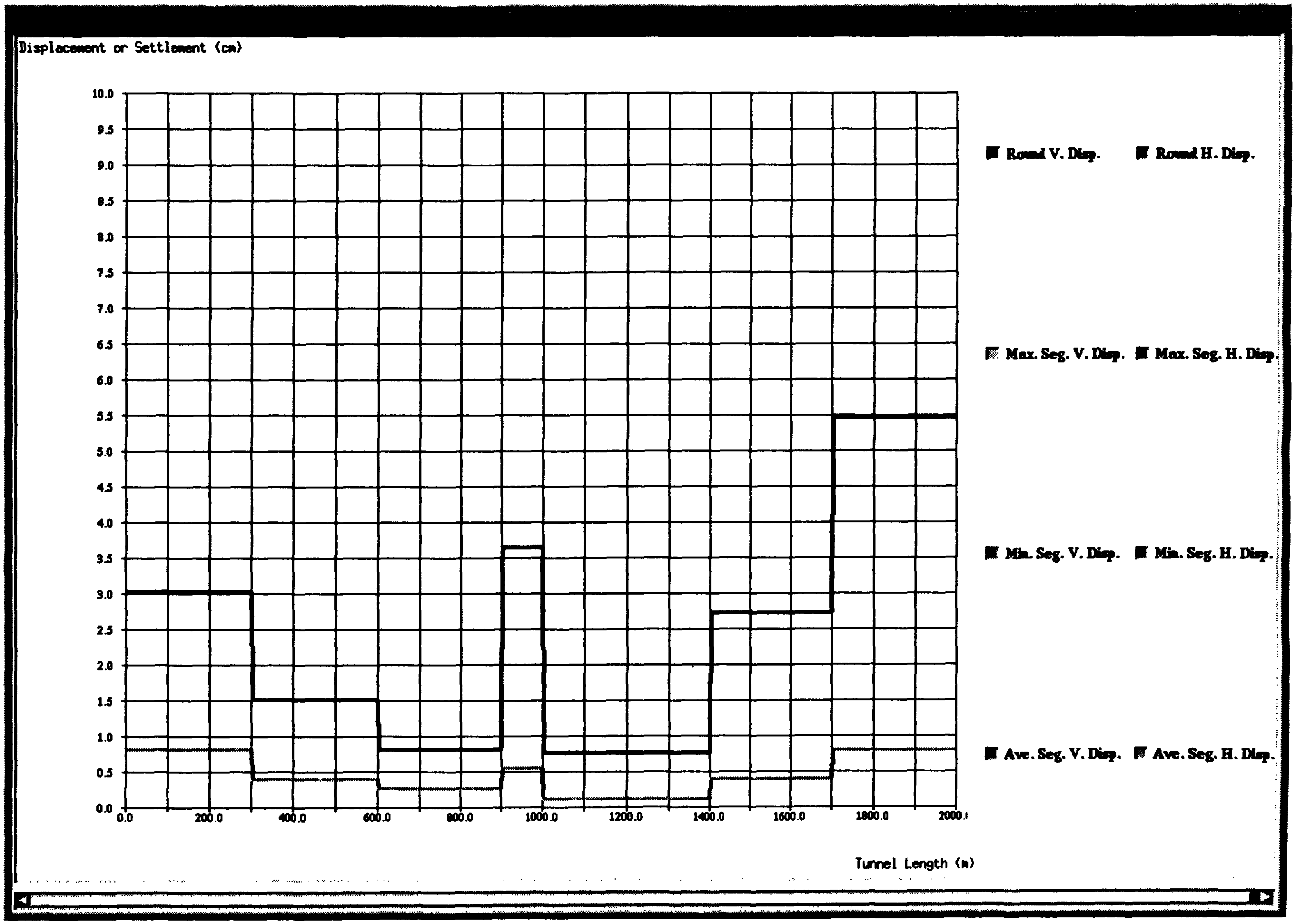


Figure 6.2 Longitudinal Profile (Case Study I)

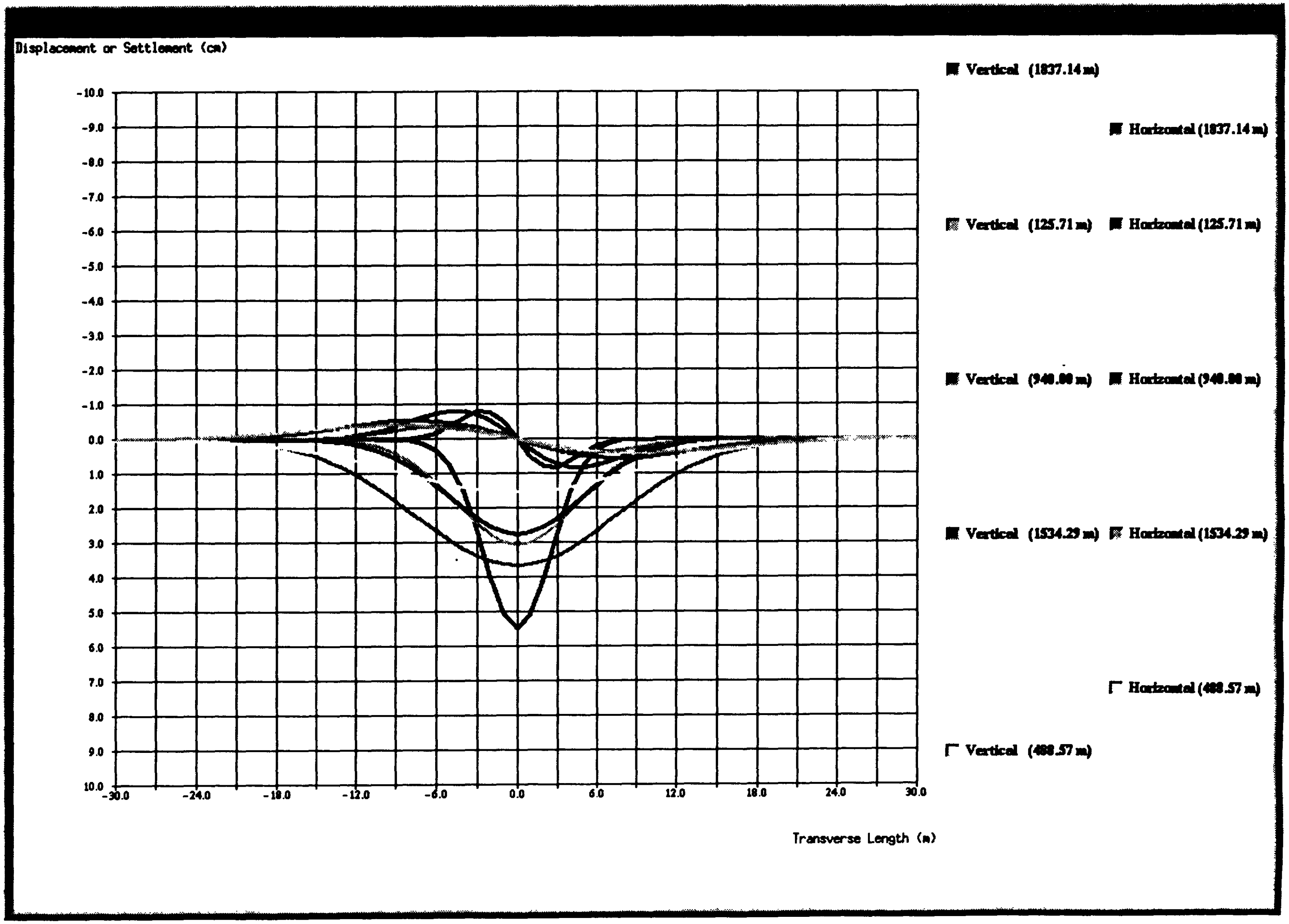


Figure 6.3 Transverse Profile (Case Study I)

6.2.2 Case Study II (Probabilistic Analysis)

(1) Layout of Tunnel

As Case study II, the same tunnel layout as in Case study I is considered.

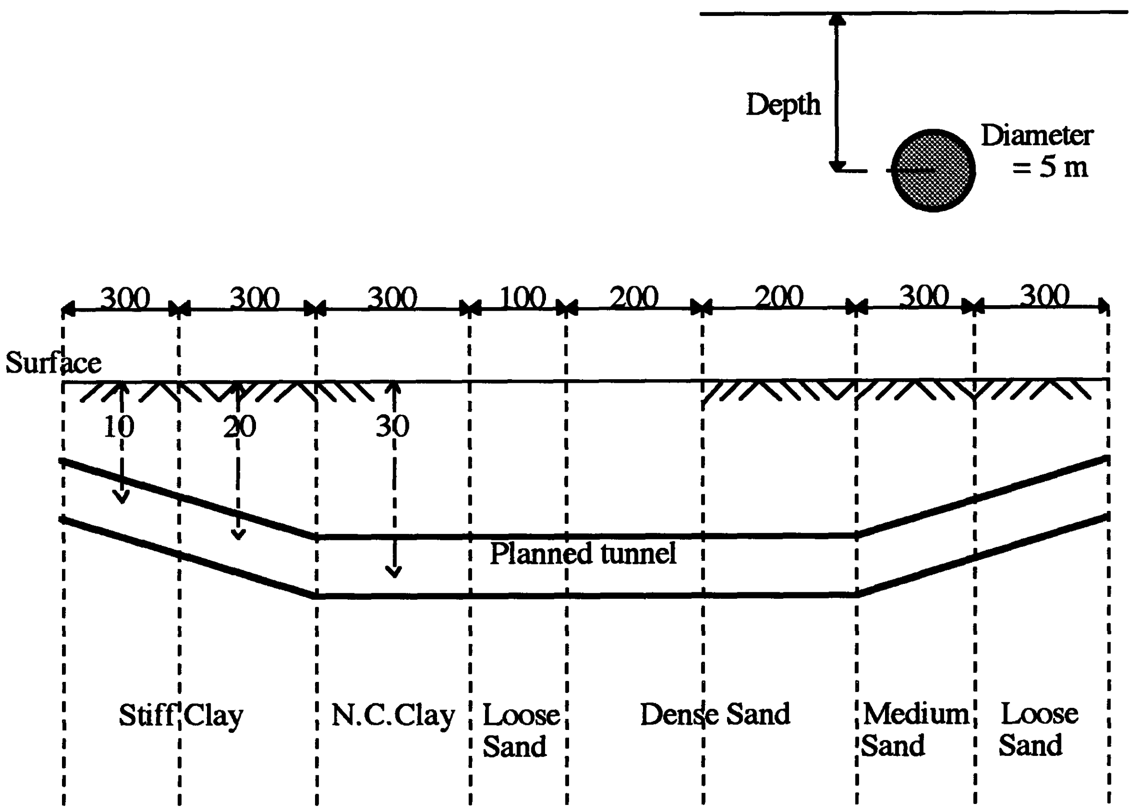


Figure 6.4 Layout of planned tunnel (Case Study II) (unit: m)

(2) Input Data

In Case II, the ground conditions and tunnel configurations are the same as in Case I. Assuming that the construction procedures (quality) are uncertain, the ratio of V_S/V_{exc} can not be specified with certainty. Hence, both parameters, K and V_S/V_{exc}, are provided as a range of data. Table 6.4 summarizes the ground conditions and tunnel configuration in the planned tunnel based on Figure 6.4. Table 6.5 describes the input data for the settlement model based on Table 6.4. Since K and

V_S/V_{exc} are defined by a range of data, SIMSUPER automatically performs the probabilistic analysis. In order to input the triangular distribution for the probabilistic analysis, the maximum, mean, and minimum values of K and V_S/V_{exc} are used (see Table 6.5).

Table 6.4 Ground Conditions and Tunnel Configurations (Case Study II)

Segment No.		1		2		3		4		5		6		7		8	
Tunnel Length (m)	0		300		600		900		1000		1200		1400		1700		2000
Tunnel Diameter (m)		5		5		5		5		5		5		5		5	
Tunnel Depth (m)		10		20		30		30		30		30		20		10	
Segment Length (m)		300		300		300		100		200		200		300		300	
Soil Type		Stiff Clay		Stiff Clay		N. C. Clay		Loose Sand		Dense Sand		Dense Sand		Medium sand		Loose sand	

Table 6.5 Input Data for the Settlement Model (Case Study II)

Segment No.		1		2		3		4		5		6		7		8	
Tunnel Length (m)	0		300		600		900		1000		1200		1400		1700		2000
Soil Type		Stiff Clay		Stiff Clay		N. C. Clay		Loose Sand		Dense Sand		Dense Sand		Medium sand		Loose Sand	
Parameter K		0.4-0.5		0.4-0.5		0.5-0.6		0.2-0.3		0.2-0.3		0.2-0.3		0.2-0.3		0.2-0.3	
Ratio of Vs/Vexc(%)		0.5-3.0		0.5-3.0		1.0-2.50		3.0-5.0		0.5-1.0		0.5-1.0		1.0-1.75		3.0-5.0	

(3) Computational Results

Figure 6.5 shows the longitudinal profile of maximum surface settlement and horizontal displacement along the planned tunnel. In this figure, the horizontal axis indicates the tunnel length and the vertical axis displays the amount of settlement and horizontal displacement. Since the input data is in form of a range of data, the computational results fluctuate per the round length based on the given random values. The maximum, average, and minimum values in each segment are indicated by the horizontal lines. Compared with Case Study I, the probabilistic analysis produces a wide range of potential settlement, especially at shallow depth.

Figure 6.6 shows the transverse profiles of surface settlement and horizontal displacement in all segments. In this figure, the horizontal axis indicates the transverse distance and the vertical axis displays the amount of settlement and horizontal displacement. Each transverse profile is distinguished by the clicked location which is shown in the parentheses after each toggle button as well as by the colored lines.

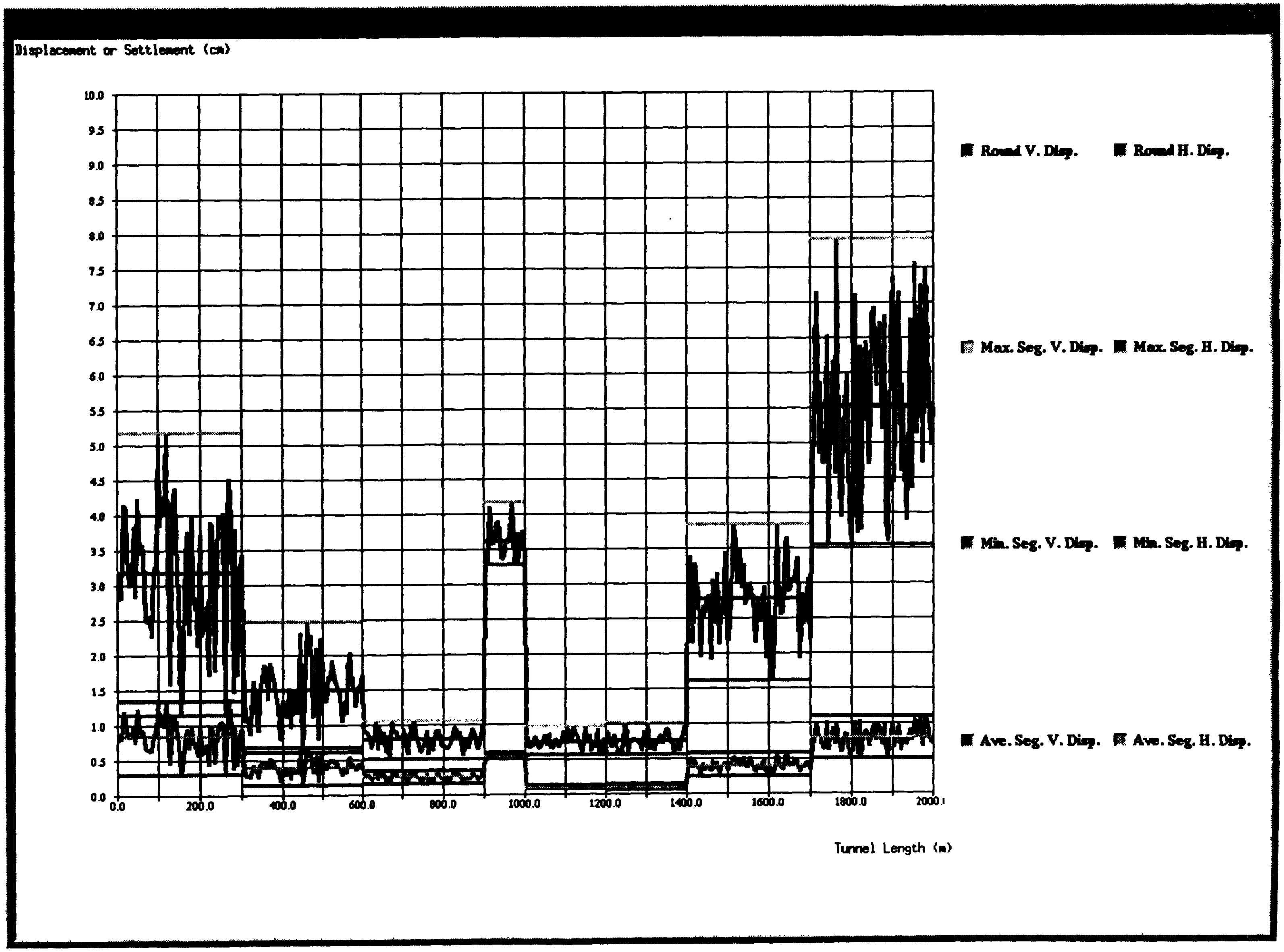


Figure 6.5 Longitudinal Profile (Case Study II)

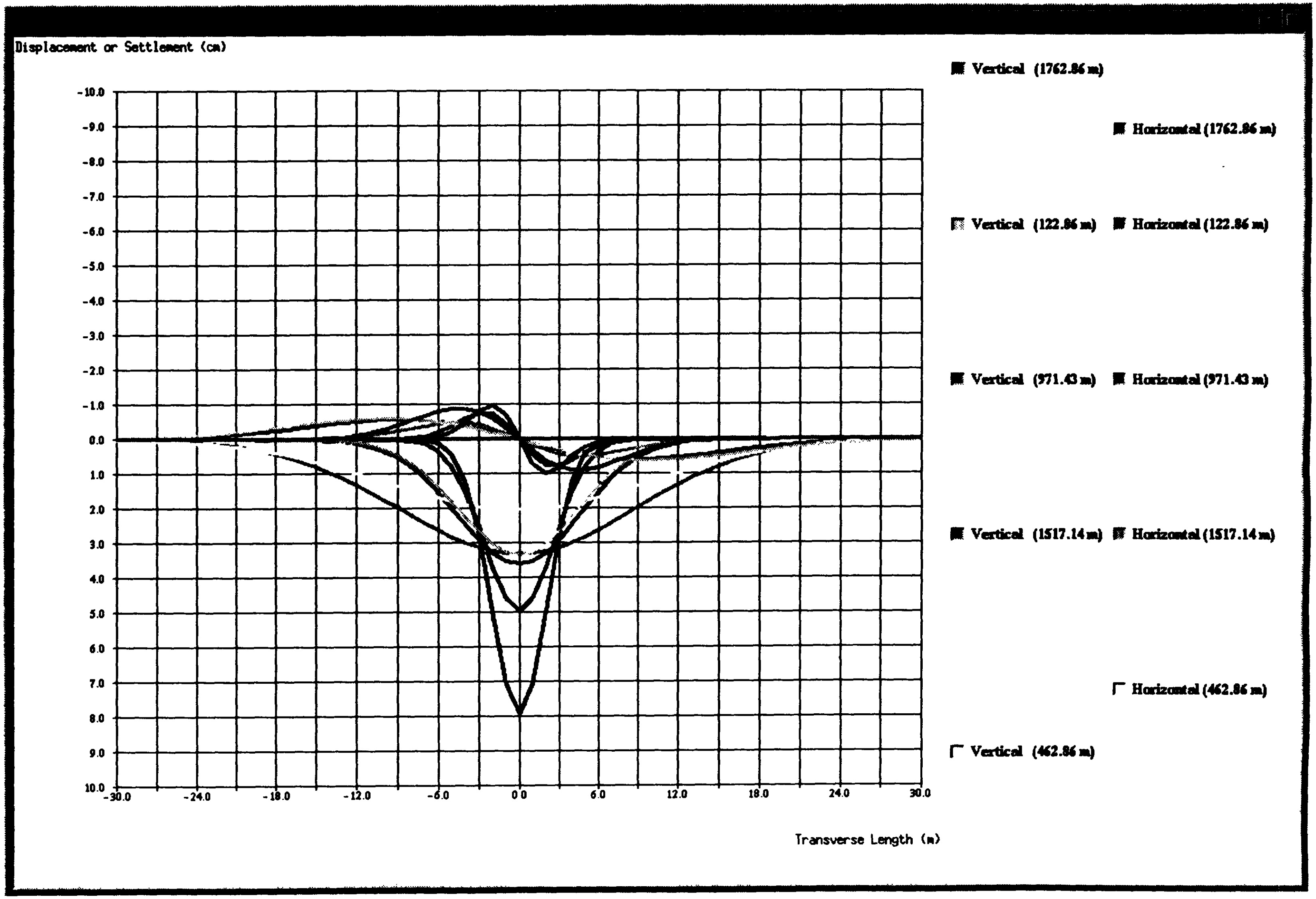


Figure 6.6 Transverse Profile (Case Study II)

6.3 Twin tunnel

6.3.1 Case Study III (Probabilistic analysis)

(1) Layout of Tunnel

As case study III, the following tunnel layout is considered.

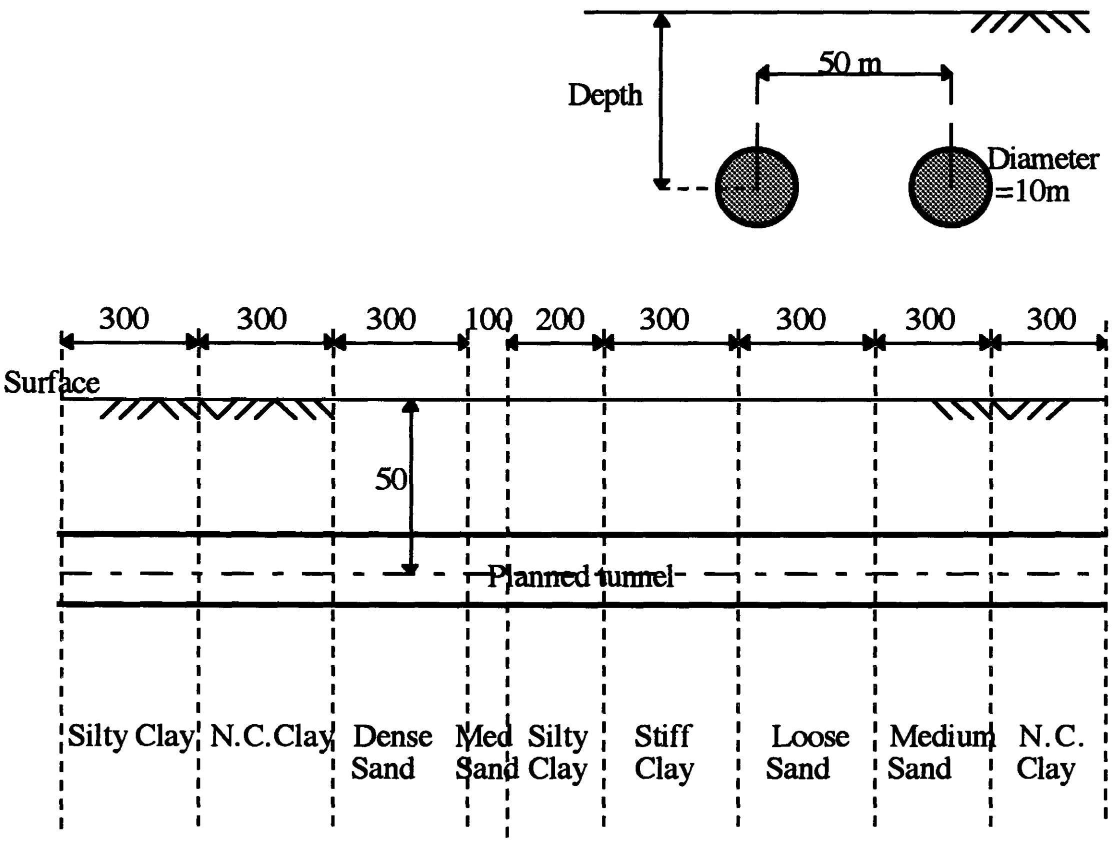


Figure 6.7 Layout of planned tunnel (Case Study III) (unit: m)

(2) Input Data

Assuming that the construction procedures (quality) are uncertain, the ratio of V_S/V_{exc} can not be specified with certainty. Hence, both parameters, K and V_S/V_{exc} are provided as a range of data. Table 6.6 summarizes the ground condition and tunnel configuration in the planned tunnel based on Figure 6.7. Table 6.7 shows the input data for the settlement model based on Table 6.6. Since the input data, K and

V_S/V_{exc} are defined by a range of data, SIMSUPER automatically performs the probabilistic analysis. In order to input the triangular distribution for the probabilistic analysis, the maximum, mean, and minimum values of K and V_S/V_{exc} are used (see Table 6.7).

Table 6.6 Ground Conditions and Tunnel Configurations (Case Study III)

Segment No.		1		2		3		4		5		6		7		8		9	
Tunnel Length (m)	0		200		400		900		1200		1400		1800		2200		2500		3000
Tunnel Diameter (m)		10		10		10		10		10		10		10		10		10	
Tunnel Depth (m)		50		50		50		50		50		50		50		50		50	
Tunnel Distance (m)		50		50		50		50		50		50		50		50		50	
Segment Length (m)		300		300		300		100		200		300		300		300		300	
Soil Type		Silty Clay		N.C. Clay		Dense Sand		Medium Sand		Silty Clay		Stiff Clay		Loose Sand		Medium Sand		N.C.Clay	

Table 6.7 Input Data for the Settlement Model (Case Study III)

Segment No.		1		2		3		4		5		6		7		8		9	
Tunnel Length (m)	0		200		400		900		1200		1400		1800		2200		2500		3000
Soil Type		Silty Clay		N.C. Clay		Dense Sand		Medium Sand		Silty Clay		Stiff Clay		Loose Sand		Medium Sand		N.C.Clay	
Parameter K		0.60-0.70		0.40-0.50		0.20-0.30		0.20-0.30		0.60-0.70		0.40-0.50		0.20-0.30		0.20-0.30		0.50-0.60	
Ratio of Vs/Vexc(%)		3.0-5.0		1.0-2.50		0.50-1.0		1.0-2.50		3.0-5.0		0.50-3.0		3.0-5.0		1.0-2.50		1.0-2.50	

(3) Computation Results

Figure 6.8 shows the longitudinal profile of maximum surface settlement and horizontal displacement along the planned tunnel. In this figure, the horizontal axis indicates the tunnel length and the vertical axis displays the amount of settlement and horizontal displacement. Since the input data is in form of a range of data, the computational results fluctuate per round length based on the given random values. The maximum, average, and minimum values in each segment are indicated by the horizontal lines.

Figure 6.9 also shows the transverse profiles of surface settlement and horizontal displacement in all segments. In this figure, the horizontal axis indicates the transverse distance and the vertical axis displays the amount of settlement and horizontal displacement. Each transverse profile is distinguished by the clicked location which is shown in the parentheses after each toggle button as well as by the colored lines. Since this computation is performed for twin tunnels, the transverse profiles display two depressions.

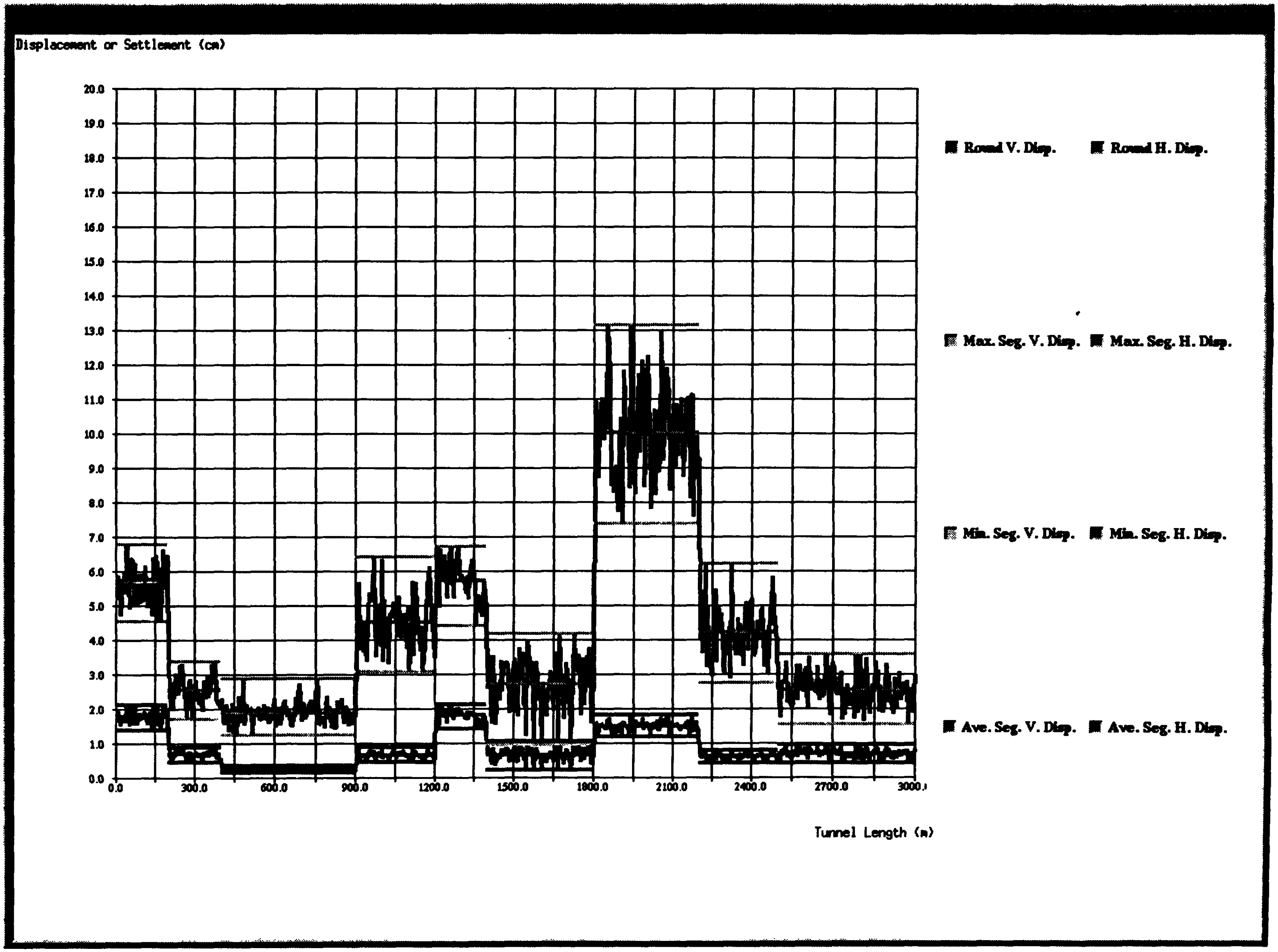


Figure 6.8 Longitudinal Profile (Case Study III)

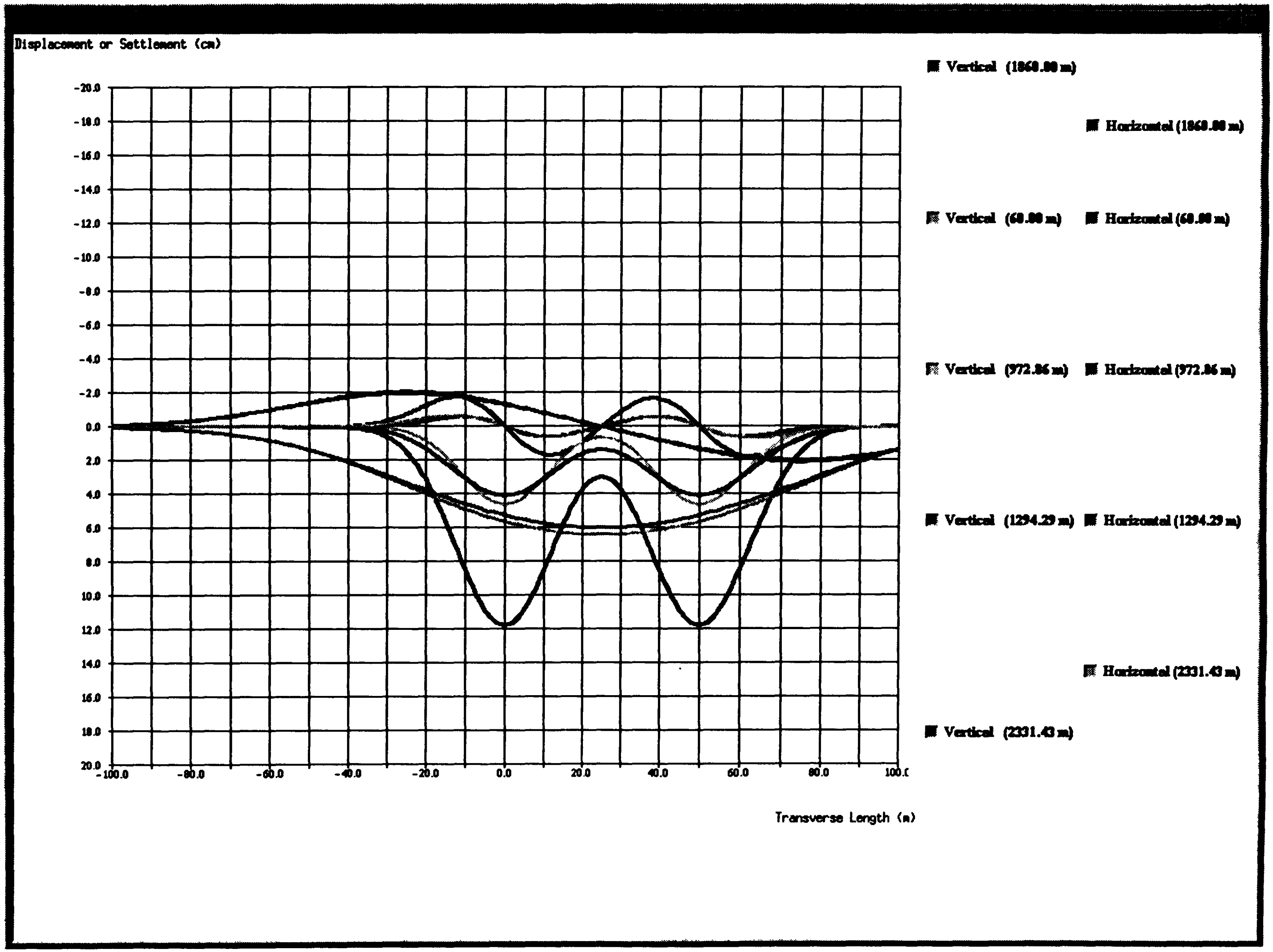


Figure 6.9 Transverse Profile (Case Study III)

6.3.2 Case Study IV (Probabilistic Analysis)

(1) Layout of Tunnel

As Case study IV, the following tunnel layout is considered.

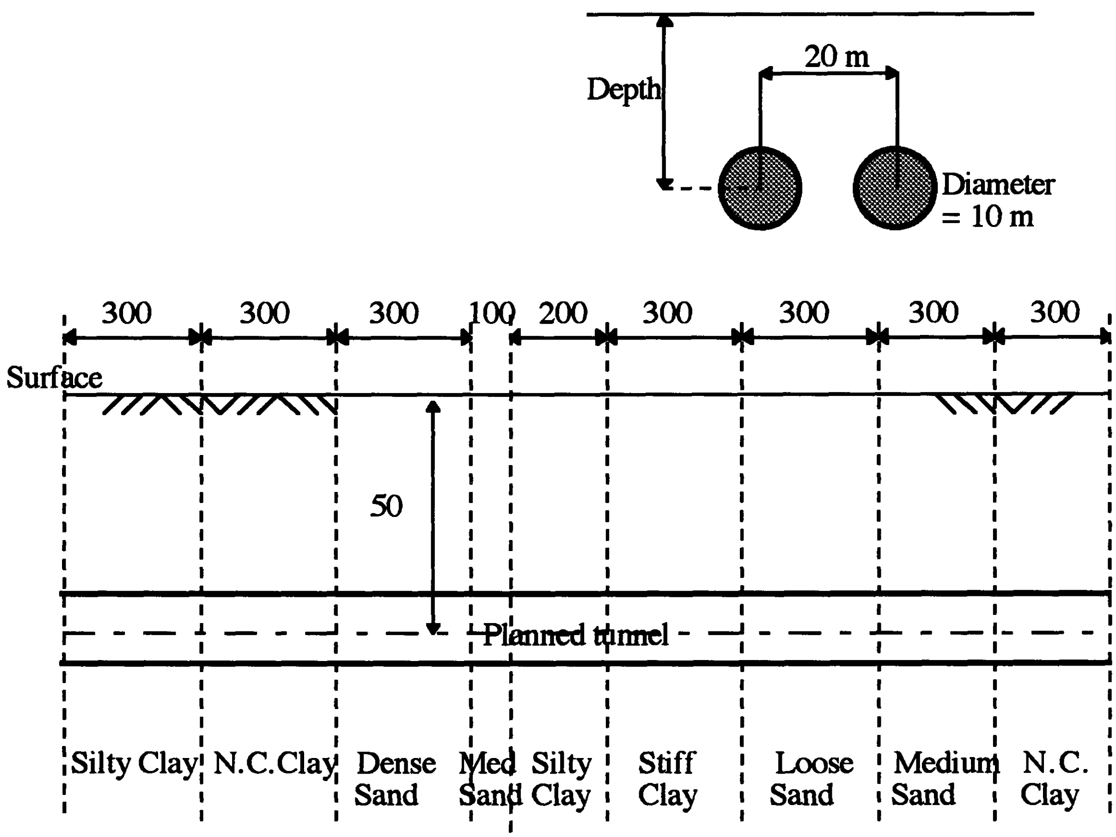


Figure 6.10 Layout of planned tunnel (Case Study IV) (unit: m)

(2) Input Data

The input data in Case study IV are the same as in Case study III except for the distance between two tunnels. In order to consider the effect of distance, the distance between two tunnels is input as 20m instead of 50m as in Case III. The rest of the data is the same as in Case III. Assuming that the construction procedures (quality) are uncertain, the ratio of V_S/V_{exc} can not be specified with certainty. Hence, both parameters, K and V_S/V_{exc}, are provided as a range of data. Table 6.8 summarizes the ground condition and tunnel configuration in the planned tunnel based on Figure 6.10. Table 6.9 shows the input data for the settlement model based on Table 6.8. Since K and V_S/V_{exc} are defined by a range of data, SIMSUPER automatically performs the probabilistic analysis. In order to input the triangular distribution for the probabilistic analysis, the maximum, mean, and minimum values of K and V_S/V_{exc} are used (see Table 6.9).

Table 6.8 Ground Conditions and Tunnel Configurations (Case Study IV)

Segment No.		1		2		3		4		5		6		7		8		9	
Tunnel Length (m)	0		200		400		900		1200		1400		1800		2200		2500		3000
Tunnel Diameter (m)		10		10		10		10		10		10		10		10		10	
Tunnel Depth (m)		50		50		50		50		50		50		50		50		50	
Tunnel Distance (m)		20		20		20		20		20		20		20		20		20	
Segment Length (m)		300		300		300		100		200		300		300		300		300	
Soil Type		Silty Clay		N.C. Clay		Dense Sand		Medium Sand		Silty Clay		Stiff Clay		Loose Sand		Medium Sand		N.C.Clay	

Table 6.9 Input Data for the Settlement Model (Case Study IV)

Segment No.		1		2		3		4		5		6		7		8		9	
Tunnel Length (m)	0		200		400		900		1200		1400		1800		2200		2500		3000
Soil Type		Silty Clay		N.C. Clay		Dense Sand		Medium Sand		Silty Clay		Stiff Clay		Loose Sand		Medium Sand		N.C.Clay	
Parameter K		0.60-0.70		0.40-0.50		0.20-0.30		0.20-0.30		0.60-0.70		0.40-0.50		0.20-0.30		0.20-0.30		0.50-0.60	
Ratio of Vs/Vexc(%)		3.0-5.0		1.0-2.50		0.50-1.0		1.0-2.50		3.0-5.0		0.50-3.0		3.0-5.0		1.0-2.50		1.0-2.50	

(3) Computational Results

Figure 6.10 shows the longitudinal profile of maximum surface settlement and horizontal displacement along the planned tunnel. In this figure, the horizontal axis indicates the tunnel length and the vertical axis displays the amount of settlement and horizontal displacement. Since the input data is in form of a range of data, the computational results fluctuate per the round length based on the given random values. The maximum, average, and minimum values in each segment are indicated by the horizontal lines.

Figure 6.11 also shows the transverse profiles of surface settlement and horizontal displacement in all segments. In this figure, the horizontal axis indicates the transverse distance and the vertical axis displays the amount of settlement and horizontal displacement. Each transverse profile is distinguished by the clicked location which is shown in the parentheses after each toggle button as well as by the colored lines. Compared with Case study III, the transverse profiles look similar to those of single tunnel and the amount of maximum settlement tends to be larger than that of Case III because two tunnels are constructed to the close distance.

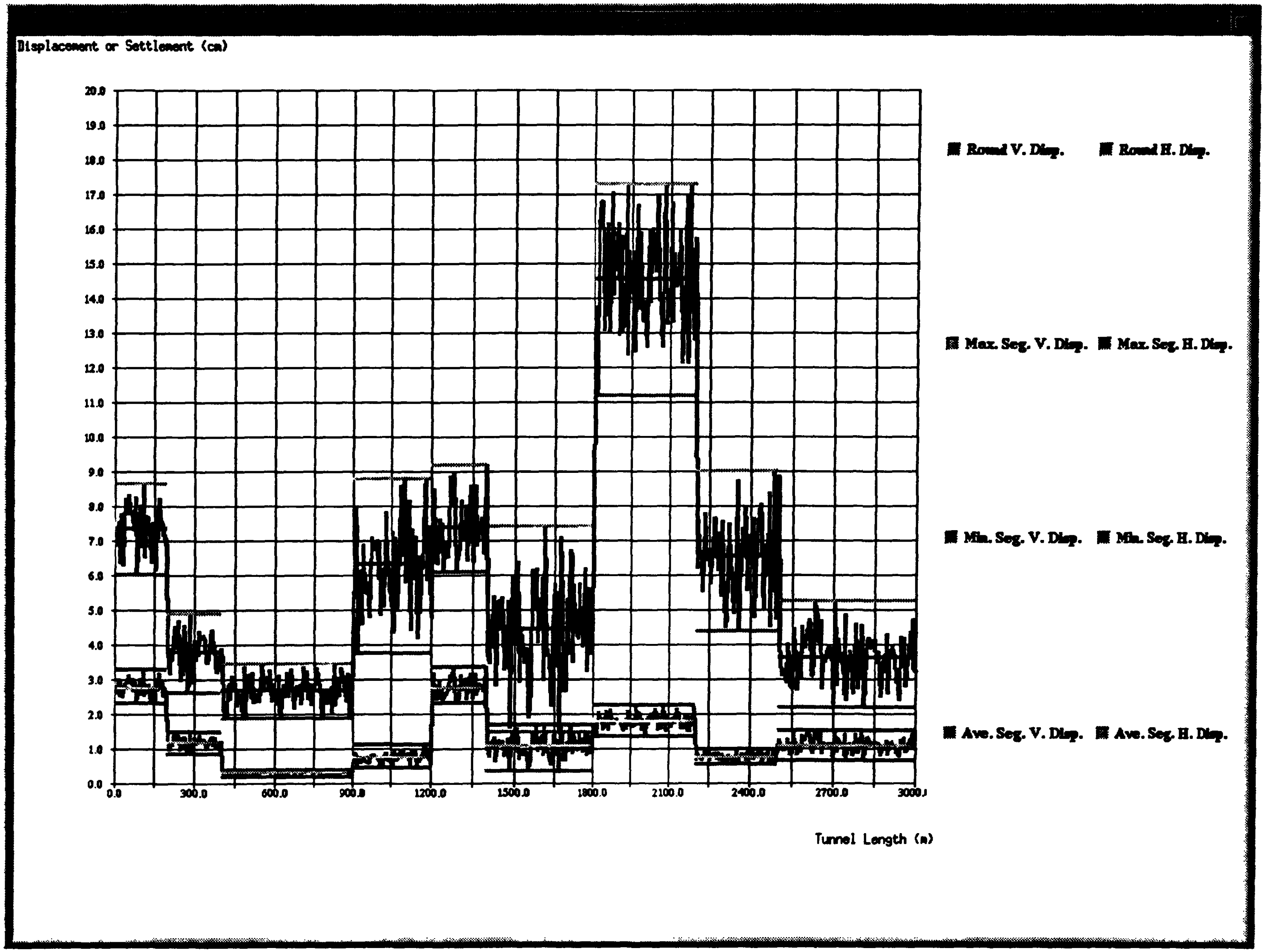


Figure 6.11 Longitudinal Profile (Case Study IV)

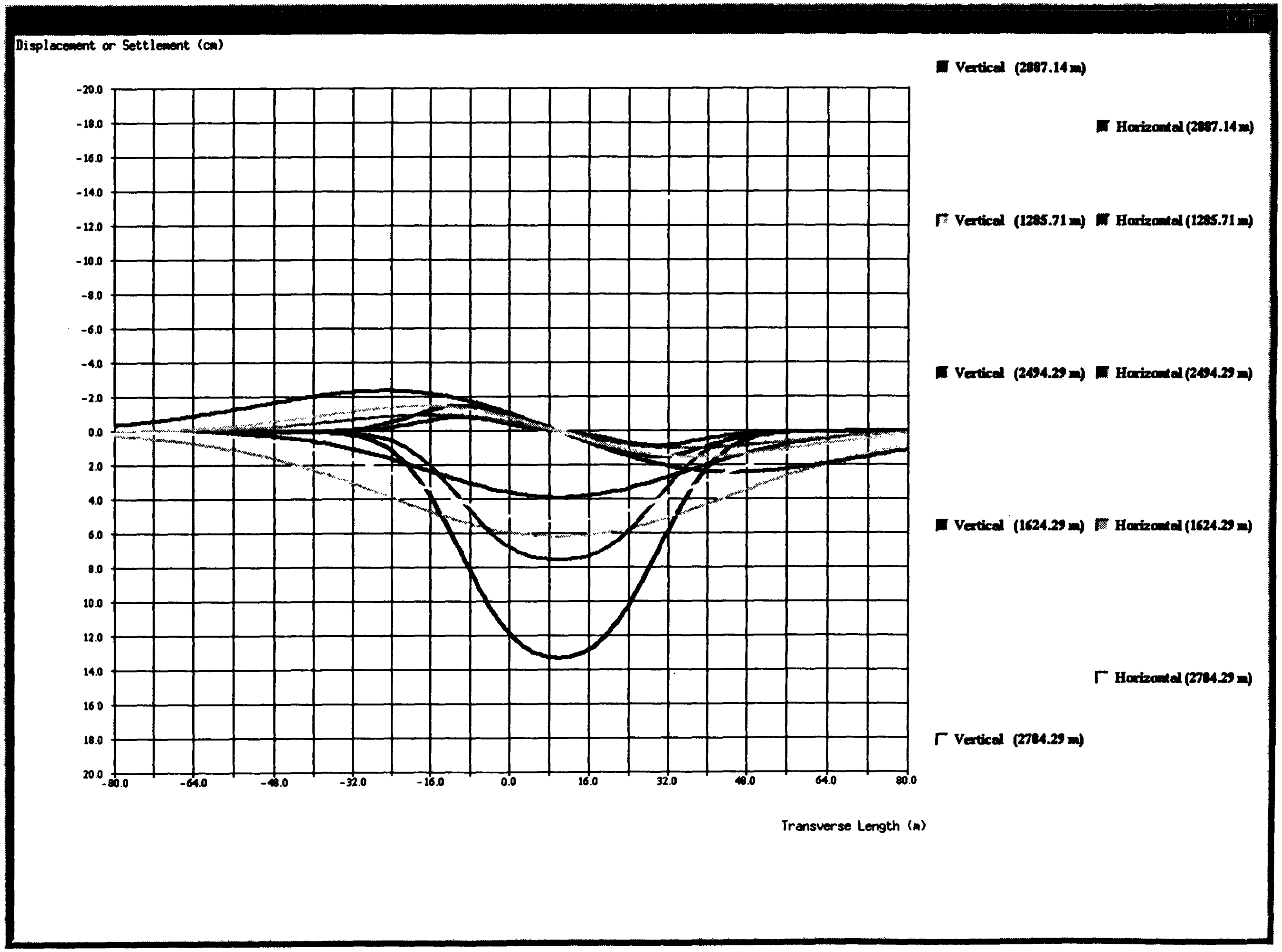


Figure 6.12 Transverse Profile (Case Study IV)

Chapter 7
Conclusions and Recommendations

7.1 Conclusions

In order to evaluate the influence of ground movements in soft ground tunneling, a settlement model was incorporated in the DAT (Decision Aids for Tunneling). After reviewing predictive methods for obtaining settlement profiles, the empirical equations by O'Reilly and New (1982) were used as the settlement model.

The computer program described in this thesis has the following principal features:

(1) When a tunnel is planned at a certain location, the program can be employed as a predictive method to evaluate potential settlement along the planned tunnel. Once the user defines the basic input data, the program can be easily used to perform parametric studies.

(2) If the given information regarding ground conditions and tunnel configurations is uncertain, the user can input a range of data and the program can perform a probabilistic analysis. As a result, the user can obtain possible ranges of settlement and horizontal displacements. These results will provide the necessary information to determine the next appropriate steps in tunnel design and construction.

(3) The user can observe the computational results through the graphic output which provides the longitudinal profiles along a planned tunnel. Furthermore, by clicking a specified point on the longitudinal profile, the plotter automatically gives a transverse settlement/horizontal displacement profile. Examining the transverse profile is an effective way to evaluate the influence of settlement on an existing structure when a tunnel is constructed.

7.2 Recommendations for Future Development

Future revisions of the settlement program will make it practically more applicable. Currently, this program is capable of evaluating only the ground conditions for a single homogeneous layer. In general, ground is composed of multilayered strata, and also at the early design stage, various types of tunnel geometries need to be considered. Therefore, there is a necessity to develop a program which can evaluate the multilayered ground strata and various types of tunnel geometries. Right now, the settlement equations are defined by two parameters, the trough width parameter and the volume of settlement trough related to the ground conditions and tunnel configuration. Hence, the next developments should concentrate on associating the parameters with multilayered ground conditions and different tunnel geometries.

Bibliography

Atkinson, J. and Potts, M. "Collapse of shallow unlined tunnels in dense sand." Tunnel and Tunneling, May 1975.

Atkinson, J. and Potts, M. "Ground movements near shallow model tunnels in sand." Proceedings of large ground movement and settlement, July 1977.

Atkinson, H. and Potts, D. "Settlement above shallow tunnels in soft ground." Journal of Geotechnical Engineering, ASCE, 103 (GT4), 1977: 307-325.

Attewell, B. "Ground deformation caused by tunneling in silty alluvial clay", Ground Engineering 11, 1978.

Attewell, P. and Woodman, P. "Predicting the dynamics of ground settlement and it's derivatives caused by tunneling in soil." Ground Engineering 15, 1982: 13-22.

Broms, B. and Bennermark, H. "Stability of clays at vertical openings." Journal of Geotechnical Engineering ASCE, 93(SM1), 1967: 71-74.

Clough, W. and Leca, E. "With focus on use of finite element methods for soft ground tunneling." Review paper in Tunnels et Micro. Paris 1989: 531-573.

Cording, E and Hansmire, W. "Controlling ground movement during tunneling." The Art and Science of Geotechnical Engineering, Chapter 25, 1989.

Cording, E and Hansmire, W. "Displacements around ground tunnels." Proceedings of Fifth Pan-American conference on Soil Mechanics and Foundation Engineering. General Report: Session IV, 1975: 571-633.

Cording, E and Hansmire, W. "Performance of a soft ground tunnel of the Washington Metro." Proceedings of Rapid Excavation Tunneling Conference, Chicago Vol. 1, 1972.

Halabe, V. "Resource modeling for DAT." Ph.D. Thesis, Massachusetts Institute of Technology, 1995.

Glossop, H. "Soil deformation caused by soft ground tunneling." Ph.D. Thesis, University of Durham, 1977.

Mair, R. J. "Centrifugal modeling of tunnel of tunnel construction in soft clay." Ph.D. Thesis, Cambridge University, 1979.

Martos, F. "Concerning an approximate equation of subsidence trough and its time factors." In international strata control congress, Leipzig, 1958: 191-205.

O'Reilly, P. and New, M. "Settlement above tunnels in the United Kingdom — their magnitude and prediction." Proceedings of Tunneling Symposium 1982:173-181.

O'Reilly, M. and New, B. "Tunneling induced ground movements; predicting their magnitude and effects." Proceedings of Fourth International Conference of ground movement and structures, Cardiff, 1991.

O'Reilly, M. and New, B. "Evaluating and predicting ground settlements caused by tunneling in London clay." Proceedings of Tunneling Symposium 1988.

Peck, B. "Deep excavations and tunneling in soft ground." Proceedings of 7th International Conference on Soil Mechanics and Foundation Engineering. Mexico city, 1969: 225-290.

Sagaseta, C. "Analysis of undrained soil deformation due to ground loss." Geotechnique 37, No. 3, 1987.

Salazar, F. "Stochastic and Economic Evaluation of Adaptability in Tunnel Design and Construction.", Ph.D. Thesis, Massachusetts Institute of Technology, 1983.

Schmidt, B. "Settlements and ground movements associated with tunneling in soil.", Ph.D. Thesis, University of Illinois, 1969.

Wong, R. and Kaiser, P. "Prediction of ground movements above shallow tunnels." Proceedings of the international symposium on Prediction and Performance in Geotechnical Engineering, Calgary, 1987.

Appendix I

Program sources of the settlement model

(1) Settle.h

(2) Settle.c

(3) Action_list.h

(4) Action_list.c

settle.h

```
#ifndef _SETTLE_
#define _SETTLE_

#ifdef __cplusplus
extern "C" {
#endif
void init_settle(const char *filename);
char *get_settle_file();
float settlement_single_tunnel (float distance, float k_constant, float ratio_volume,
                                float height, float diameter,
                                float width, int isegment,
                                int itunnel, int iround,  float location);
float settlement_twin_tunnel (float distance, float k_constant,
                              float ratio_volume, float height, float diameter,
                              float width, int isegment, int itunnel,
                              int iround,  float location);

void compute_seg_value( int itunnel,int isegment,int iround, float location,
                        float settlement_vmax, float settlement_hmax);

void settlement_tunnel (int itunnel, int isegment, int iround, float location,
                        float distance, float k_constant, float ratio_volume,
                        float height, float diameter, float width);

#ifdef __cplusplus
}
#endif

#endif
```

```c
#ifdef SETTLE
/**************************************************************
 **         Subroutine of settlement profile
 **************************************************************/

#include "defs.h"
#include "macro.h"
#include "utils.h"
#include "define.h"
#include "settle.h"
#include "graphic.h"
#include "global.h"
#include "math.h"
#include "tunnel.h"
#include "tun_loc.h"
#include "tun_info.h"

#define R2PI  2.50662827        /* definition of root 2 pai */
#define DIM       500           /* definition of array dimension */

#define ROUND_VERTICAL 0
#define ROUND_HORIZONTAL 1
#define MAX_SEG_VERTICAL 2
#define MAX_SEG_HORIZONTAL 3
#define MIN_SEG_VERTICAL 4
#define MIN_SEG_HORIZONTAL 5
#define MEAN_SEG_VERTICAL 6
#define MEAN_SEG_HORIZONTAL 7
#define EXPF(x)   ((float) exp((double) (x)))

FILE *fpw=NULL, *fpwl, *fopen(const char *filename, const char *type);
char settle_file[80];

/*  DATA DESCRIPTION */
/*  distance is the distance between both tunnels */
/* k_constant is soil constant determining trough width */
/* ratio_volume is ration of settlement volume to excavation volume */
/* height is height from surface to tunnel center */
/* diameter is diameter of tunnel */
/* width is width for settlement calculation */


/* Initialization */

void init_settle(const char *filename)
{
  sprintf(settle_file, filename);
}

char *get_settle_file()
{
  return settle_file;
}

/***********************************************************/
/********* function of settlement        tunnel) **********/
/***********************************************************/

void settlement_tunnel(int itunnel, int isegment, int iround, float location,
                   float distance, float k_constant, float ratio_volume,
                   float height, float diameter, float width)
{
  static boolean called_before = 0;
  static int last_seg[MAX_TUN];
  char stresult[80];

  if(repo_id(itunnel))
    return;
  if(!called_before) {
    sprintf(stresult,"%s_seg.out", settle_file);
    fpwl=fopen(stresult, "w");
    called_before = 1;
    for (int i=0;i<NumTun();i++)
      last_seg[i] = ABSENT;
  }

  if(last_seg[itunnel] NEQ isegment) {
    sprintf(stresult, "%s.%d_%d.out", settle_file, tunnel_id(itunnel),
            isegment+1);
    fpw=fopen(stresult, "w");
    last_seg[itunnel] = isegment;
  }


  if (distance  EQ  0)
    settlement_single_tunnel (distance, k_constant, ratio_volume, height, diameter,
                             width, iround, isegment, itunnel, location);
  else
    settlement_twin_tunnel (distance, k_constant, ratio_volume, height,
                           diameter, width, iround, isegment, itunnel,
                           location);
}


void compute_seg_value( int itunnel,int isegment,int iround, float location,float settlem
ent_vmax, float settlement_hmax)

{

  static float seg_settlement_vmax[MAX_TUN];
  static float seg_settlement_hmax[MAX_TUN];
  static float seg_settlement_vmin[MAX_TUN];
  static float seg_settlement_hmin[MAX_TUN];
  static float seg_settlement_hsum[MAX_TUN];
  static float seg_settlement_vsum[MAX_TUN],seg_length[MAX_TUN];
  static int last_seg[MAX_TUN];
  float round_length=length_advance(itunnel);


  if  (iround  EQ 1)
    {
      draw_settle(itunnel, ROUND_VERTICAL,
                  current_position(itunnel),settlement_vmax, FALSE);
      draw_settle(itunnel, ROUND_HORIZONTAL,
                  current_position(itunnel),settlement_hmax, FALSE);
      last_seg[itunnel]=isegment;
      seg_settlement_vmax[itunnel]=settlement_vmax;
      seg_settlement_hmax[itunnel]=settlement_hmax;
      seg_settlement_vmin[itunnel]=settlement_vmax;
      seg_settlement_hmin[itunnel]=settlement_hmax;
      seg_settlement_vsum[itunnel]=settlement_vmax*round_length;
      seg_settlement_hsum[itunnel]=settlement_hmax*round_length;
      seg_length[itunnel]      = round_length;
      draw_settle(itunnel, ROUND_VERTICAL,
                  current_position(itunnel),
                  settlement_vmax, FALSE);
      draw_settle(itunnel, ROUND_HORIZONTAL,
```

```
                current_position(itunnel),
                settlement_hmax,FALSE);
    }
  else if   (last_seg[itunnel] EQ isegment)
    {
      seg_settlement_vmax[itunnel]= MAX(seg_settlement_vmax[itunnel],settlement_vmax);
      seg_settlement_hmax[itunnel]= MAX(seg_settlement_hmax[itunnel],settlement_hmax);
      seg_settlement_vmin[itunnel]=MIN(seg_settlement_vmin[itunnel],settlement_vmax);
      seg_settlement_hmin[itunnel]=MIN(seg_settlement_hmin[itunnel],settlement_hmax);
      seg_settlement_vsum[itunnel]+=settlement_vmax*round_length;
      seg_settlement_hsum[itunnel]+=settlement_hmax*round_length;
      seg_length[itunnel]    += round_length;
    }


  if(last_seg[itunnel] NEQ isegment || iround EQ ABSENT)
    {
      draw_settle(itunnel,MAX_SEG_VERTICAL,
                current_position(itunnel)-seg_length[itunnel],
                seg_settlement_vmax[itunnel],FALSE);
      draw_settle(itunnel,MAX_SEG_HORIZONTAL,
                current_position(itunnel)-seg_length[itunnel],
                seg_settlement_hmax[itunnel],FALSE);
      draw_settle(itunnel,MIN_SEG_VERTICAL,
                current_position(itunnel)-seg_length[itunnel],
                seg_settlement_vmin[itunnel],FALSE);
      draw_settle(itunnel,MIN_SEG_HORIZONTAL,
                current_position(itunnel)-seg_length[itunnel],
                seg_settlement_hmin[itunnel],FALSE);
      draw_settle(itunnel,MEAN_SEG_VERTICAL,
                current_position(itunnel)-seg_length[itunnel],
                seg_settlement_vsum[itunnel]/seg_length[itunnel],FALSE);
      draw_settle(itunnel,MEAN_SEG_HORIZONTAL,
                current_position(itunnel)-seg_length[itunnel],
                seg_settlement_hsum[itunnel]/seg_length[itunnel],FALSE);

      draw_settle(itunnel,MAX_SEG_VERTICAL,current_position(itunnel),
                seg_settlement_vmax[itunnel],TRUE);
      draw_settle(itunnel,MAX_SEG_HORIZONTAL,current_position(itunnel),
                seg_settlement_hmax[itunnel],TRUE);
      draw_settle(itunnel,MIN_SEG_VERTICAL,current_position(itunnel),
                seg_settlement_vmin[itunnel],TRUE);
      draw_settle(itunnel,MIN_SEG_HORIZONTAL,current_position(itunnel),
                seg_settlement_hmin[itunnel],TRUE);
      draw_settle(itunnel,MEAN_SEG_VERTICAL,current_position(itunnel),
                seg_settlement_vsum[itunnel]/seg_length[itunnel],TRUE);
      draw_settle(itunnel,MEAN_SEG_HORIZONTAL,current_position(itunnel),
                seg_settlement_hsum[itunnel]/seg_length[itunnel],TRUE);

      fprintf (fpwl,  "COMPUTATIONAL RESULTS for SEGMENTS\n Tunnel No.=%3d  Segment No=%
3d  End Location (m)= %6.2f\nseg_settlement_vmax (m)= %8.3f   seg_settlement_hmax (m)= %
8.3f\nseg_settlement_vmin (m)= %8.3f   seg_settlement_hmin (m)= %8.3f\nseg_settlement_va
ve (m)= %8.3f   seg_settlement_have (m)= %8.3f\n\n",
              tunnel_id(itunnel), last_seg[itunnel]+1,current_position(itunnel),
              seg_settlement_vmax[itunnel],seg_settlement_hmax[itunnel],
              seg_settlement_vmin[itunnel],seg_settlement_hmin[itunnel],
              seg_settlement_vsum[itunnel]/seg_length[itunnel],
              seg_settlement_hsum[itunnel]/seg_length[itunnel]);

      if(iround EQ ABSENT)
        return;
      last_seg[itunnel]=isegment;
      seg_settlement_vmax[itunnel]=settlement_vmax;
      seg_settlement_hmax[itunnel]=settlement_hmax;
      seg_settlement_vmin[itunnel]=settlement_vmax;
      seg_settlement_hmin[itunnel]=settlement_hmax;
      seg_settlement_vsum[itunnel]=settlement_vmax*round_length;
      seg_settlement_hsum[itunnel]=settlement_hmax*round_length;
      seg_length[itunnel]       = round_length;

    }

  draw_settle(itunnel, ROUND_VERTICAL,
            current_position(itunnel)+length_advance(itunnel),
            settlement_vmax, TRUE);
  draw_settle(itunnel, ROUND_HORIZONTAL,
            current_position(itunnel)+length_advance(itunnel),
            settlement_hmax,TRUE);

}


/****************************************************************
  Subroutine of settlement profile (Twin tuunel)
 ****************************************************************/


/*  DATA DESCRIPTION */
/*  distance is the distance between both tunnels */
/*  k_constant is soil constant determining trough width */
/*  ratio_volume is ration of settlement volume to excavation volume */
/*  height is height from surface to tunnel center */
/*  diameter is diameter of tunnel */
/*  width is width for settlement calculation */



#ifdef TEST

main()

{

  char stresult[80];
  float distance, k_constant, ratio_volume, height, diameter, width, location;
  int iround, isegment, itunnel;
  int called_before = 0;

  while(TRUE) {
    printf("INPUT_DATA\n distance, k_constant, ratio_volume,  height, diameter,
 width, itunnel, isegment, iround, location\n");

    scanf("%f%f%f%f%f%f%d%d%d%f", &distance, &k_constant, &ratio_volume, &height,
          &diameter, &width, &itunnel, &isegment, &iround, &location);

    if (k_constant  EQ  0) {
      compute_seg_value(itunnel, isegment, 0, location, 0., 0.);
      break;
    }

    sprintf(stresult, "%s.%d_%d_%d.out", settle_file, itunnel, isegment,
            iround);

    fpw=fopen(stresult, "w");

    if(!called_before) {
      sprintf(stresult, "%s_seg.out", settle_file);
```

```
    fpw1=fopen(stresult, "w");
    called_before = 1;

    settlement_twin_tunnel(distance, k_constant, ratio_volume, height, diameter,
                           width, iround, isegment, itunnel, location);

    fclose(fpw);

   }
   fclose(fpw1);
  }
}
#endif

  /**********************************************************/
  /********* function of settlement(twin tunnel) **********/
  /**********************************************************/

#ifdef __STDC__
float settlement_twin_tunnel (float distance, float k_constant,
                              float ratio_volume, float height,
                              float diameter, float width, int iround,
                              int isegment, int itunnel, float location)
#else
float settlement_twin_tunnel (distance, k_constant, ratio_volume, height,
                              diameter, width, iround, isegment,
                              itunnel,  location)

  float distance, k_constant, ratio_volume, height, diameter, width, location;
  int iround, isegment, itunnel;
#endif

{
  int y, j;
  float settlement_vmax=0.0, settlement_hmax=0.0, x_vmax, x_hmax;
  float i, x, volume_exc, volume_set;
  float  range[DIM], settlement_v[DIM],settlement_h[DIM];

  i=k_constant*height;  /* define trough width parameter */
  volume_exc=PI*diameter*diameter/4.0;
  /* calculation of excavation Area */

  volume_set=ratio_volume/100.0*volume_exc;
  /* calculation of trough volume  by Vexc */


  for (j=0; j<DIM; j++)
    {
      settlement_v[j]=0;          /* Initialization */
      settlement_h[j]=0;          /* Initialization */
    }

  y=1;                            /* Initialization */

  fprintf (fpw, "CALCULATION of SURFACE DISPLACEMENT\n\nINPUT_DATA\n k_constant = %5.2f\
n Ratio_volume(%%) = %5.2f\n Height(m) = %5.2f\n Diameter(m) = %5.2f\n Distance(m) = %5.
2f\n Width(m) = %5.2f\n Location(m) = %7.2f\n\n",k_constant,
          ratio_volume, height, diameter, distance, width, location);

  fprintf (fpw, "\nOUTPUT\n Trnsv.Loc.(m)   Settlement(m)   H.Displacement(m)\n");

  for (x= -width; x<=width+distance; x++,y++)
    {
      settlement_v[y]=volume_set*(EXPF(-(x*x)/(2*i*i))+EXPF(-((x-distance)*(x-distance))/
(2*i*i)))/(R2PI*i);

      /*calculation of vertical displacement */

      settlement_h[y]=volume_set*(x*EXPF(-(x*x)/(2*i*i))+(x-distance)*EXPF(-((x-distance)
*(x-distance))/(2*i*i)))/(R2PI*i*height);

      /* calculation of horizontal displacement */


      /**************************************/
      /****** find out maximum value ********/
      /**************************************/

      if(settlement_v[y]>= settlement_vmax)
        {settlement_vmax=settlement_v[y];x_vmax=x;}


      if(settlement_h[y]>= settlement_hmax)
        {settlement_hmax=settlement_h[y];x_hmax=x;}


      range[y]=x;
      /* define range or calculation */


      fprintf (fpw, "%9.2f %15.5f %15.5f\n", range[y], settlement_v[y], settlement_h[y]);



    }

  fprintf(fpw,  "MAXIMUM VALUE\nX_V(m) = %5.2f  Vmax (m)= %8.3f\nX_H(m) = %5.2f  Hmax (m)
= %8.3f\n",  x_vmax,   settlement_vmax,   x_hmax,   settlement_hmax);

  compute_seg_value(itunnel, isegment, iround, location, settlement_vmax, settlement_hmax
);

}


/************************************************************
     Subroutine of settlement profile
************************************************************/
#ifdef TEST

main()

{
  char stresult[80];
  float k_constant, ratio_volume, height, diameter, width, location;
  int iround, isegment, itunnel;
  int called_before = 0;

  while(TRUE) {
    printf("INPUT_DATA\n k_constant, ratio_volume, height, diameter, width, itunnel, iseg
ment, iround, location\n");

    scanf("%f%f%f%f%f%d%d%d%f", &k_constant, &ratio_volume, &height, &diameter, &width, &
itunnel, &isegment, &iround, &location);

    if (k_constant  EQ  0) {
      compute_seg_value(itunnel, isegment, 0, location, 0., 0 ),
```

```c
    break;
  }
  sprintfs(stresult, "settlement.%d_%d_%d.out", itunnel, isegment, iround);

  fpw=fopen(stresult, "w");

  if(!called_before) {
    sprintf(stresult, "%s_seg.out", settle_file);
    fpwl=fopen(stresult, "w");
    called_before = 1;
  }

  settlement_single_tunnel(distance, k_constant, ratio_volume, height, diameter, width
, iround, isegment, itunnel, location);
       fclose(fpw);
  }
  fclose(fpwl);
}
#endif

/*********************************************************/
/********** function of settlement(single tunnel)***********/
/*********************************************************/
#ifdef __STDC__
float settlement_single_tunnel (float distance, float k_constant, float ratio_volume,
                               float height, float diameter, float width,  int iround,

                               int isegment, int itunnel, float location)
#else
float settlement_single_tunnel (distance, k_constant, ratio_volume, height, diameter,wid
th, iround,  isegment,  itunnel,  location)
     float distance, k_constant, ratio_volume, height, diameter, width, location;
     int iround, isegment, itunnel;
#endif

{

  int y, j;
  float settlement_vmax=0.0,  settlement_hmax=0.0,  x_vmax,  x_hmax;
  float i, x, volume_exc, volume_set, range[DIM], settlement_v[DIM],
  settlement_h[DIM];


  i=k_constant*height;    /* define trough width parameter */
  volume_exc=PI*diameter*diameter/4.0;
  /* calculation of excavation Area */

  volume_set=ratio_volume/100.0*volume_exc;
  /* calculation of trough volume  by Vexc */


  for (j=0; j<DIM; j++)
    {
      settlement_v[j]=0;          /* Initialization */
      settlement_h[j]=0;          /* Initialization */
    }

  y=1;                        /* Initialization */

  fprintf (fpw,"CALCULATICN of SURFACE DISPLACEMENT\n\nINPUT_DATA\n k_constant = %5.2f\n
 Ratio_volume(%%) = %5.2f\n Height(m) = %5.2f\n Diameter(m) = %5.2f\n Distance(m) = %5.2
f\n Width(m) = %5.2f\n Location(m) = %7.2f\n\n", k_constant,
          ratio_volume, height, diameter, distance,width, location);

  fprintf (fpw, "\nOUTPUT\n Trnsv.Loc.(m)   Settlement(m)   H Displacement(m)\n");

  for (x= -width; x<=width; x++,y++)
    {

      settlement_v[y]=volume_set*EXPF(-(x*x)/(2*i*i))/(R2PI*i),

      /*calculation of vertical displacement */

      settlement_h[y]=x*settlement_v[y]/height;

      /* calculation of horizontal displacement */


      /**************************************/
      /****** find out maximum value  ********/
      /**************************************/

      if(settlement_v[y]>= settlement_vmax)
        {settlement_vmax=settlement_v[y];x_vmax=x;}


      if(settlement_h[y]>= settlement_hmax)
        {settlement_hmax=settlement_h[y];x_hmax=x;}


      range[y]=x;   /* define range for calculation */


      fprintf (fpw, "%9.2f %15.5f %15.5f\n", range[y], settlement_v[y],  settlement_h[y])
;

    }

  fprintf (fpw,  "MAXIMUM VALUE\nX_V(m) = %5.2f  Vmax (m)= %8 4f\nX_H(m) = %5.2f  Hmax (m
)=  %8.4f\n",
           x_vmax,   settlement_vmax,  x_hmax,   settlement_hmax);


  compute_seg_value(itunnel, isegment, iround, location, settlement_vmax, settlement hmax
);


}

#endif
```

action_list.h

```c
#ifndef _FUNC_LIST_
#define _FUNC_LIST_

#include "equations.h"

struct parameter_t
{
  struct expression_t *param;
  struct parameter_t  *next;
};

#ifdef __cplusplus
extern "C" {
#endif

void                  init_list_actions();

action_t              give_function(char *func_name,int *nb_params);

action_t              give_table(char *table_name);
struct expression_t*  size_of_table_decalage(char *table_name);

#ifdef __cplusplus
}
#endif

#endif
```

```c
#include "action_list.h"
#include "util.dir/defs.h"
#include "util.dir/macro.h"
#include "simple_proba.h"
#include "prob_util.h"
#include "tun_loc.h"
#include "tun_info.h"
#include "evaluator.h"
#include "delay.h"
#include "timeway_out.h"
#include "up_down.h"
#include "tun_activity.h"
#include "tunnel_out.h"
#include "ress_info.h"
#include "global.h"
#include "scheduler.h"

#define MAX_TABLE 20
#define MAX_FUNC  20

struct function_list
{
  char *name;
  int  nb_params;
  action_t action;
};

static struct function_list list_func[MAX_FUNC];
static nb_functions;

// The value pt corresponds to the last argument entered by the user.
// To retrieve earlier argument use pt=pt->next;

#define ACTION      (pt->param->action)
#define LEFT_PARAM  pt->param->left_exp
#define RIGHT_PARAM pt->param->right_exp

static float func_constant(struct expression_t *expl, struct expression_t *exp2)
{
  struct parameter_t *pt = (struct parameter_t*) exp2;

#ifdef EQ_DEBUG
  int *nb = (int*) expl;
  assert(*nb==1);
  printf("constant retourne %f\n",
         ACTION(LEFT_PARAM,RIGHT_PARAM));
#endif

  return ACTION(LEFT_PARAM,RIGHT_PARAM);
}

static float func_uniform(struct expression_t *expl, struct expression_t *exp2)
{
  struct parameter_t *pt = (struct parameter_t*) exp2;

  float min,max;

  max = ACTION(LEFT_PARAM,RIGHT_PARAM);
  pt  = pt->next;
  min = ACTION(LEFT_PARAM,RIGHT_PARAM);

#ifdef EQ_DEBUG

  int *nb = (int*) expl;    /* nb parameters (not useful here) */
  assert(*nb==2);

  float val = uniform(min,max);
  printf("uniform(%f,%f) return %f\n",min,max,val);
  return val;
#else
  return uniform(min,max);
#endif
}

static float func_triangular(struct expression_t *expl, struct expression_t *exp2)
{
  struct parameter_t *pt = (struct parameter_t*) exp2;

  float min,ave,max;

  min = ACTION(LEFT_PARAM,RIGHT_PARAM);
  pt  = pt->next;
  ave = ACTION(LEFT_PARAM,RIGHT_PARAM);
  pt  = pt->next;
  max = ACTION(LEFT_PARAM,RIGHT_PARAM);

#ifdef EQ_DEBUG
  int *nb = (int*) expl;
  assert(*nb==3);
  float val = triangular(min,ave,max);
  printf("triangular(%f,%f,%f) return %f\n",min,ave,max,val);
  return val;
#else
  return triangular(min,ave,max);
#endif
}

static float func_up(struct expression_t *expl, struct expression_t *exp2)
{
  struct parameter_t *pt = (struct parameter_t*) exp2;

#ifdef EQ_DEBUG
  int *nb = (int*) expl;
  float val;

  assert(*nb==1);

  if(is_going_up(current_evaluated_tunnel()))
    val = ACTION(LEFT_PARAM,RIGHT_PARAM);
  else
  {
    val = 0;
  }
  printf("up returns %f\n",val);
  return val;
#else
  if(is_going_up(current_evaluated_tunnel()))
    return ACTION(LEFT_PARAM,RIGHT_PARAM);
  else
    return 0;
#endif
}

static float func_down(struct expression_t *expl, struct expression_t *exp2)
{
  struct parameter_t *pt = (struct parameter_t*) exp2;

#ifdef EQ_DEBUG
```

```c
  int *nb = (int*) expl;
  assert(*nb==1);
  float val;
  if(is_going_up(current_evaluated_tunnel()))
    val = 0;
  else
    val = ACTION(LEFT_PARAM,RIGHT_PARAM);
  printf("down retourn %f\n",val);
  return val;
#else
  if(is_going_up(current_evaluated_tunnel()))
    return 0;
  else
    return ACTION(LEFT_PARAM,RIGHT_PARAM);
#endif
}

static float func_round_apl(struct expression_t *expl, struct expression_t *exp2)
{
  struct parameter_t *pt = (struct parameter_t*) exp2;

  int val;

  val = current_round(current_evaluated_tunnel()) MOD
    (int) ACTION(LEFT_PARAM,RIGHT_PARAM);

#ifdef EQ_DEBUG

  int *nb = (int*) expl;
  assert(*nb==1);

  printf("RndApl<x> return %f\n",(float) (val EQ 0));
#endif

  return (float) (val EQ 0);
}

static float func_interface_delay(struct expression_t *expl, struct expression_t *exp2)
{
  int tunnel = current_evaluated_tunnel();

  if(is_tun_terminated(tunnel))
    return 0.0;

  if(current_ground_class(tunnel) != previous_ground_class(tunnel) &&
     current_constr_method(tunnel) != previous_constr_method(tunnel)) {

#ifdef EQ_DEBUG
    int *nb = (int*) expl;
    assert(*nb==0);

    printf("delay return %f comme interface entre %d et %d (tunnel %d)\n",
           give_interface_delay(give_geom_type(tunnel,current_segment(tunnel)),
                                previous_ground_class(tunnel),
                                current_ground_class(tunnel)),
           previous_ground_class(tunnel),
           current_ground_class(tunnel),
           tunnel);
#endif

    float idelay =
      give_interface_delay(give_geom_type(tunnel,current_segment(tunnel)),
                           previous_ground_class(tunnel),
                           current_ground_class(tunnel));
    if(idelay > 0.0)
      update_interface_delay(tunnel, idelay);
    return idelay;
  }
  else
    return 0.0;
}

static float func_distance(struct expression_t *expl, struct expression_t *exp2)
{
  int tunnel = current_evaluated_tunnel();

  if(is_tun_terminated(tunnel))
    return 0.0;

  int nrepo,repo[MAX_REPOSITORIES];
  float distance;

  tun_reposit(tunnel,&distance,&nrepo,repo);

#ifdef EQ_DEBUG
    int *nb = (int*) expl;
    assert(*nb==0);

    printf("distance return %f for tunnel %d\n", distance, tunnel);
#endif
  if(!nrepo)
    return 0.0;
  assert(distance>=0);
  return distance;

}

static float func_layer(struct expression_t *expl, struct expression_t *exp2)
{
  struct parameter_t *pt = (struct parameter_t*) exp2;

  float layer;

  layer = (int) ACTION(LEFT_PARAM,RIGHT_PARAM);

#ifdef EQ_DEBUG

  int *nb = (int*) expl;
  float val;

  assert(*nb==2);

  if(layer EQ current_layer(current_evaluated_tunnel()))
  {
    pt = pt->next;
    val = ACTION(LEFT_PARAM,RIGHT_PARAM);
  }
  else
    val = 0;

  printf("Layer returns %f\n",val);

  return val;
#else
  if(layer EQ current_layer(current_evaluated_tunnel()))
  {
    pt = pt->next;
    return ACTION(LEFT_PARAM,RIGHT_PARAM);
```

```
  }
  else
    return 0;
#endif
}

static float func_pick(struct expression_t *expl, struct expression_t *exp2)
{
  static int   lst_rnd[MAX_TUN][MAX_EQUATION],called_before=0;
  static float cum_val[MAX_TUN][MAX_EQUATION];
  struct parameter_t *pt = (struct parameter_t*) exp2;
  int    ipick=0,tunnel,i,j;
  float  val;

  if(!called_before) {
    for(j=0;j<MAX_EQUATION;j++)
      for(i=0;i<NumTun();i++) {
        lst_rnd[i][j] = 0;  /* last round number */
        cum_val[i][j] = -2.;  /* cumulative value = sum of % - UNIF_0_1 */
      }
    called_before = 1;
  }

  val = ACTION(LEFT_PARAM,RIGHT_PARAM);
  assert(IN(val,0,1));
  pt  = pt->next;
  ipick = (int) ACTION(LEFT_PARAM,RIGHT_PARAM);
  assert(IN(ipick,1,MAX_EQUATION));

  /* check if the last round number is same as the current round number     */
  /* If it is select random number between 0 and 1 and upate the cumulative */
  // value. If the value is greater than zero then the state is 1 else it is 0.
  tunnel = current_evaluated_tunnel();
  if(current_round(tunnel) NEQ lst_rnd[tunnel][ipick]) {
/*    assert(cum_val[tunnel][ipick] EQ -2.);*/ /* A state was selected previously*/
    lst_rnd[tunnel][ipick] = current_round(tunnel);
    cum_val[tunnel][ipick] = val-UNIF_0_1;
  }
  else if(cum_val[tunnel][ipick] NEQ -2.)
    cum_val[tunnel][ipick] += val;

  if(cum_val[tunnel][ipick] >= 0.) {
    cum_val[tunnel][ipick] = -2.;
    return 1.;
  }
  else
    return 0.;
}

static float func_act_time_delay(struct expression_t *expl,
                              struct expression_t *exp2)
{
  static float *clock_int[MAX_TUN],called_before=0;
  static float *clock_del[MAX_TUN];
  struct parameter_t *pt = (struct parameter_t*) exp2,*ptl;
  int    tunnel,activity;
  float  val=0,vall;

  if(!called_before) {
    for(tunnel=0;tunnel<NumTun();tunnel++) {
      CREATE(clock_int[tunnel], float, MAX_ACTIVITIES);
      bzero(clock_int[tunnel],MAX_ACTIVITIES*sizeof(float));
      CREATE(clock_del[tunnel], float, MAX_ACTIVITIES);
      bzero(clock_del[tunnel],MAX_ACTIVITIES*sizeof(float));
    }
    called_before = 1;
  }

  tunnel   = current_evaluated_tunnel();
  activity = current_evaluated_activity();

  ptl = pt;
  if(!clock_del[tunnel][activity]) {
    pt = pt->next;
    clock_del[tunnel][activity] = ACTION(LEFT_PARAM,RIGHT_PARAM);
    assert(IN(clock_del[tunnel][activity],0,HUGE));
   }
  clock_int[tunnel][activity] += time_duration(tunnel),
  if(clock_del[tunnel][activity] <= clock_int[tunnel][activity])
    while (clock_del[tunnel][activity] <= clock_int[tunnel][activity]) {
      pt  = ptl;
      vall = ACTION(LEFT_PARAM,RIGHT_PARAM);
      assert(IN(vall,0,HUGE));
      val += vall;
      pt = pt->next;
      vall = ACTION(LEFT_PARAM,RIGHT_PARAM);
      assert(IN(vall,0,HUGE));
      clock_del[tunnel][activity] += vall;
    }

  return val;
}


static float func_time_delay(struct expression_t *expl,
                             struct expression_t *exp2)
{
  static float *clock_glbdel[MAX_TUN],called_before=0,
  struct parameter_t *pt = (struct parameter_t*) exp2,*ptl;
  int    tunnel,activity;
  float  val=0,vall;

  if(!called_before) {
    for(tunnel=0;tunnel<NumTun();tunnel++) {
      CREATE(clock_glbdel[tunnel], float, MAX_ACTIVITIES);
      bzero(clock_glbdel[tunnel],MAX_ACTIVITIES*sizeof(float));
    }
    called_before = 1;
  }


  tunnel   = current_evaluated_tunnel();
  activity = current_evaluated_activity();

  ptl = pt;
  if(!clock_glbdel[tunnel][activity]) {
    pt = pt->next;
    clock_glbdel[tunnel][activity] = ACTION(LEFT_PARAM,RIGHT_PARAM);
    assert(IN(clock_glbdel[tunnel][activity],0,HUGE));
   }

  if(clock_glbdel[tunnel][activity] <= now()+time_duration(tunnel))
    while (clock_glbdel[tunnel][activity] <= now()+time_duration(tunnel)) {
      pt  = ptl;
      vall = ACTION(LEFT_PARAM,RIGHT_PARAM);
      assert(IN(vall,0,HUGE));
      val += vall;
```

```
    pt = pt->next;
    val1 = ACTION(LEFT_PARAM,RIGHT_PARAM);
    assert(IN(val1,0,HUGE));
    clock_glbdel[tunnel][activity] += val1;
  }

  return val;
}


static float func_length_delay(struct expression_t *exp1,
                               struct expression_t *exp2)
{
  static float *length_del[MAX_TUN],called_before=0;
  struct parameter_t *pt = (struct parameter_t*) exp2,*pt1;
  int    tunnel,activity;
  float  val=0,val1;

  if(!called_before) {
    for(tunnel=0;tunnel<NumTun();tunnel++) {
      CREATE(length_del[tunnel], float, MAX_ACTIVITIES);
      bzero(length_del[tunnel],MAX_ACTIVITIES*sizeof(float));
    }
    called_before = 1;
  }


  tunnel   = current_evaluated_tunnel();
  activity = current_evaluated_activity();

  pt1 = pt;
  if(!length_del[tunnel][activity]) {
    pt = pt->next;
    length_del[tunnel][activity] = ACTION(LEFT_PARAM,RIGHT_PARAM);
    assert(IN(length_del[tunnel][activity],0,HUGE));
  }
  if(length_del[tunnel][activity] <= current_length(tunnel))
    while (length_del[tunnel][activity] <= current_length(tunnel)) {
      pt = pt1;
      val1 = ACTION(LEFT_PARAM,RIGHT_PARAM);
      assert(IN(val1,0,HUGE));
      val += val1;
      pt = pt->next;
      val1 = ACTION(LEFT_PARAM,RIGHT_PARAM);
      assert(IN(val1,0,HUGE));
      length_del[tunnel][activity] += val1;
    }

  return val;
}

#ifdef SETTLE
#include "graphic.h"
#include "settle.h"
static float func_settlement(struct expression_t *exp1,
                             struct expression_t *exp2)
{
  struct parameter_t *pt = (struct parameter_t*) exp2;


  float s_width = ACTION(LEFT_PARAM,RIGHT_PARAM);
  pt = pt->next;
  float s_diameter = ACTION(LEFT_PARAM,RIGHT_PARAM);
  pt = pt->next;
  float s_height = ACTION(LEFT_PARAM,RIGHT_PARAM);
  pt = pt->next;
  float s_ratio_volume = ACTION(LEFT_PARAM,RIGHT_PARAM);
  pt = pt->next;
  float s_k_constant = ACTION(LEFT_PARAM,RIGHT_PARAM);
  pt = pt->next;
  float s_distance = ACTION(LEFT_PARAM,RIGHT_PARAM);

  int tunnel = current_evaluated_tunnel();
  settlement_tunnel(tunnel,current_segment(tunnel),current_round(tunnel),
                    current_position(tunnel)+length_advance(tunnel),
                    s_distance,s_k_constant,s_ratio_volume,s_height,
                    s_diameter,s_width);
#ifdef EQ_DEBUG
  printf("\nDist=%f,K=%f,R=%f,H=%f,D=%f,W=%f", s_distance, s_k_constant,
         s_ratio_volume,s_height,s_diameter,s_width);
#endif

  return 0.;
}
#endif

action_t give_function(char *name, int *nb_params)
{
  int i;

  for (i=0;i<nb_functions;i++)
    if(!strcmp(list_func[i].name,name)) /* strcasecmp useless (tolower in equation c) */
    {
      *nb_params = list_func[i].nb_params;
      return list_func[i].action;
    }
  return NULL;
}

/*********************************************************/
/*                    External tables                    */
/*********************************************************/

struct table_list
{
  char      *name;
  action_t  action;
  int       nb_dimensions;
};

static struct table_list list_table[MAX_TABLE];
static nb_tables;

static float table_round_modifier(struct expression_t *exp1, struct expression_t *)
{
#ifdef EQ_DEBUG
  printf("Rndlthmod[Thisrn] return %f\n",1.0);
#endif

  return 1.0;
  /*
    return round_location((int) (exp1->action)(exp1->left_exp,exp1->right_exp)),
  */
}

static float table_round_location(struct expression_t *exp1, struct expression_t *)
{
#ifdef EQ_DEBUG
```

```
  printf("RndLoc retourne %f\n",1.0);
#endif

  return 1.0;
  /*
    return round_location((int) (expl->action)(expl->left_exp,expl->right_exp));
  */
}


action_t give_table(char *name)
{
  int i;

  for (i=0;i<nb_tables;i++)
    if(!strcmp(list_table[i].name,name)) return list_table[i].action;

  return NULL;
}

struct expression_t* size_of_table_decalage(char *name)
{
  int *p;

  CREATE(p,int,1);
  *p = 1;

  return (struct expression_t*) p;
}

/*********************************************************/


#define PUT_FUNC(v1,v2,v3) (list_func[nb_functions].name        = strdup(v1);\
                              list_func[nb_functions].action      = (v2);\
                                list_func[nb_functions++].nb_params = (v3);)

#define PUT_TABLE(v1,v2,v3) (list_table[nb_tables].name           = strdup(v1);\
                               list_table[nb_tables].action         = (v2);\
                                 list_table[nb_tables++].nb_dimensions = (v3);)

void init_list_actions()
{
  nb_functions = nb_tables = 0;
  PUT_FUNC("constant",func_constant,1);
  PUT_FUNC("uniform",func_uniform,2);
  PUT_FUNC("triangular",func_triangular,3);
  PUT_FUNC("down",func_down,1);
  PUT_FUNC("up",func_up,1);
  PUT_FUNC("rndapl", func_round_apl,1);
  PUT_FUNC("interface_delay",func_interface_delay,0);
  PUT_FUNC("layer",func_layer,2);
  PUT_FUNC("pick",func_pick,2);
  PUT_FUNC("act_time_delay",func_act_time_delay,2);
  PUT_FUNC("time_delay",func_time_delay,2);
  PUT_FUNC("length_delay",func_length_delay,2);
  PUT_FUNC("distance",func_distance,0);
#ifdef SETTLE
  PUT_FUNC("settlement",func_settlement,6);
#endif
  PUT_TABLE("rndlthmod",table_round_modifier,1);
  PUT_TABLE("rndloc",table_round_location,1);
}
```