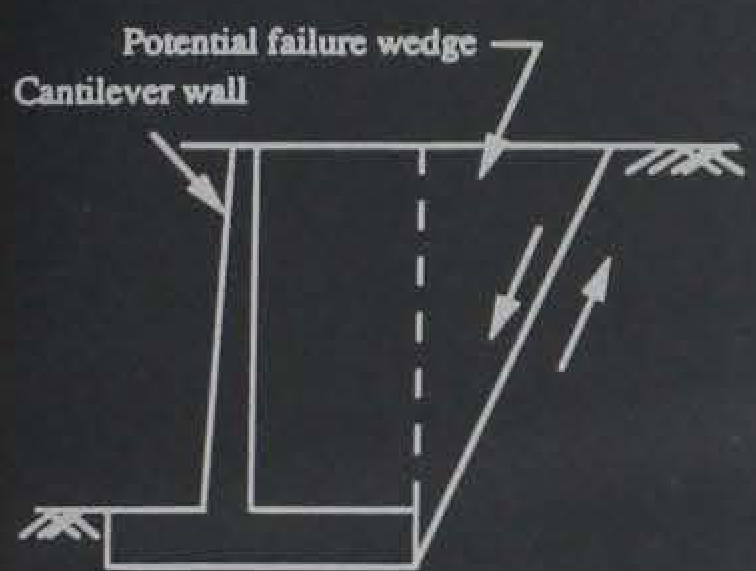
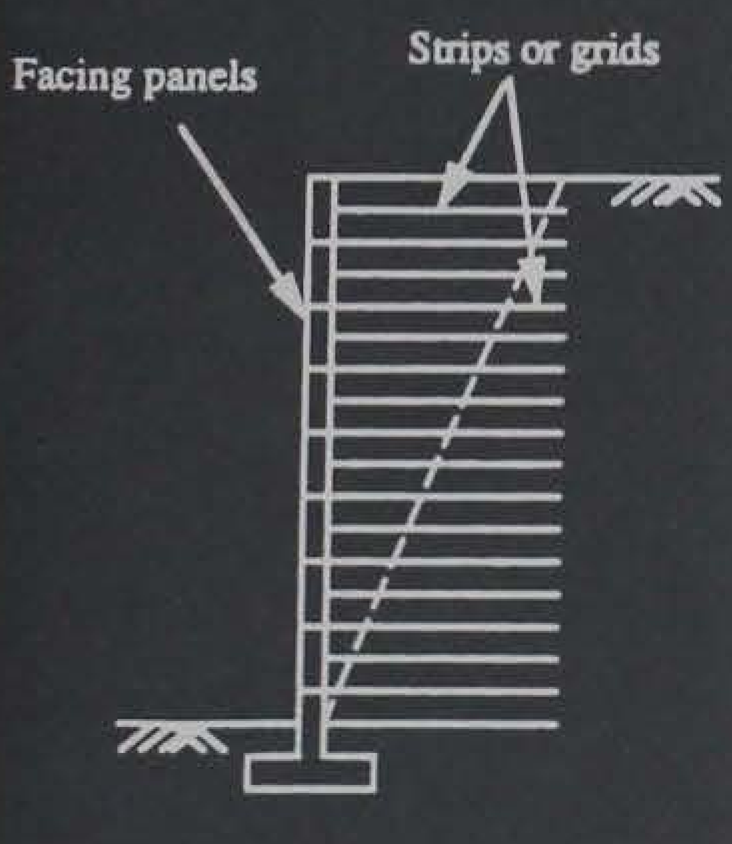
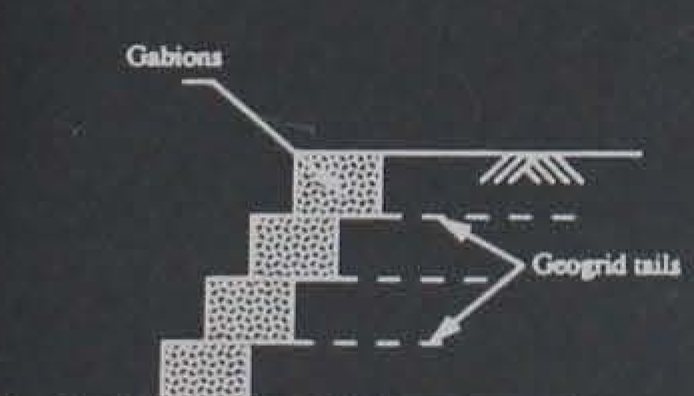
US Army Corps of Engineers




Cantilever




Reinforced soil






COMPUTER-AIDED STRUCTURAL ENGINEERING (CASE) PROJECT

CONTRACT REPORT ITL-92-2

# KNOWLEDGE-BASED EXPERT SYSTEM FOR SELECTION AND DESIGN OF RETAINING STRUCTURES

by

M. Arockiasamy, G. Sreenivasan, K. Shen

Department of Ocean Engineering
Florida Atlantic University
Boca Raton, Florida 33431



September 1992

Final Report

Approved For Public Release; Distribution Is Unlimited

Prepared for DEPARTMENT OF THE ARMY
US Army Corps of Engineers
Washington, DC 20314-1000

Monitored by Information Technology Laboratory
US Army Engineer Waterways Experiment Station
3909 Halls Ferry Road, Vicksburg, Mississippi 39180-6199

Under Contract No. DACW39-90-R-0041

26937644

TA7 W34c no.ITL-92-2 c.2

# REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188



**1. AGENCY USE ONLY** *(Leave blank)*

**2. REPORT DATE**
September 1992

**3. REPORT TYPE AND DATES COVERED**
Final report

**4. TITLE AND SUBTITLE**
Knowledge-Based Expert System for Selection and Design of Retaining Structures

**5. FUNDING NUMBERS**
C DACW39-90-R-0041

**6. AUTHOR(S)**
M. Arockiasamy, G. Sreenivasan, K. Shen

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Department of Ocean Engineering
Florida Atlantic University
Boca Raton, FL 33431

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
US Army Corps of Engineers, Washington, DC 20314-1000
USAE Waterways Experiment Station, Information Technology Laboratory, 3909 Halls Ferry Road, Vicksburg, MS 39180-6199

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
Contract Report
ITL-92-2

**11. SUPPLEMENTARY NOTES**
Available from National Technical Information Service, 5285 Port Royal Road Springfield, VA 22161.

**12a. DISTRIBUTION/AVAILABILITY STATEMENT**
Approved for public release; distribution is unlimited.

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** *(Maximum 200 words)*

This report describes the development of a microcomputer-based prototype expert system (ES), RETAININGEARTH, for the selection and design of earth retaining structures. The ES RETAININGEARTH is an interactive menu-driven system and consists of two modules - the selection module, SELECTWALL, and the design module. SELECTWALL is developed using the rule-based M.1 knowledge engineering shell and it makes a choice of the most appropriate retaining structure from a list of ten typical walls. A concise review of the design concepts related to the structures is presented together with the design flowcharts. The design module is then developed which consists of five independent design programs which perform detailed designs of the concrete gravity and cantilever walls, gabions, reinforced earth and sheetpile structures. The SELECTWALL and the design module are linked by an M.1 external code EXT through a control program CALL. All the design procedures are coded using the C programming language. The methodology of the development of the prototype ES 'RETAININGEARTH' is described together with illustrated

(Continued)

**14. SUBJECT TERMS**
Knowledge-based expert system
Retaining walls
Structures

**15. NUMBER OF PAGES**
368

**16. PRICE CODE**

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| UNCLASSIFIED | UNCLASSIFIED | | |

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

13. (Concluded).

consultations and the hard copy of the codes developed for the selection and design modules are given in Appendices A-G.

## PREFACE

This report presents a developmental prototype knowledge-based expert system (ES) RETAININGEARTH for the selection and design of earth retaining structures. This study will provide guidance for the selection of an appropriate retaining structure, based on a given set of input conditions and the subsequent detailed design of the selected structure. Funding for this study was provided by Headquarters, US Army Corps of Engineers, and the study was monitored by the Information Technology Laboratory (ITL), US Army Engineer Waterways Experiment Station (WES), Vicksburg, Mississippi. Partial graduate assistantships were provided by the Department of Ocean Engineering, Florida Atlantic University, Boca Raton, Florida.

The report was written by Dr. M. Arockiasamy as Principal Investigator with able assistance from graduate students. The computer code of the ES was developed by Giri Sreenivasan and Keling Shen who worked in this project as Graduate Research Assistants. Ms. Barbara Steinberg typed the report coordinating the text and tables layout.

The work was monitored at WES by Mr. Michael E. Pace, under the general supervision of Mr. H. Wayne Jones, Chief, Scientific and Engineering Applications Center, and Dr. N. Radhakrishnan, Director, ITL.

At the time of publication of this report, Director of WES was Dr. Robert W. Whalin. Commander and Deputy Director was COL Leonard G. Hassell, EN.

**ACKNOWLEDGEMENT**

The authors would like to thank Dr. N. Radhakrishnan, Chief, Information Technology Laboratory (ITL), US Army Engineer Waterways Experiment Station (WES), Vicksburg, Mississippi (the sponsor of this project), for his valuable guidance, constructive criticism and cooperation. The authors sincerely appreciate the constructive and critical comments of Mr. Michael Pace, Civil Engineer, ITL at WES which were very valuable in revising the draft final report. Grateful acknowledgments are due to several researchers and organizations whose references have been adapted, used or reproduced freely in the preparation of this report. They wish to express their appreciation to Dr. S.E. Dunn, Professor and Chairman, Department of Ocean Engineering and Dr. Craig S. Hartley, Dean of Engineering, Florida Atlantic University for their support, continued interest and encouragement.

Finally, Ms. Barbara Steinberg is acknowledged for her excellent and patient typing.

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

## LIST OF TABLES

**SUMMARY**

This report describes the development of a microcomputer based prototype expert system (ES), RETAININGEARTH, for the selection and design of earth retaining structures. The expert system RETAININGEARTH is an interactive menu-driven system and consists of two modules - the selection module, SELECTWALL and the design module. SELECTWALL is developed using the rule-based M.1 knowledge engineering shell and it makes a choice of the most appropriate retaining structure from a list of ten typical walls. A concise review of the design concepts related to the structures is presented together with the design flowcharts. The design module is then developed which consists of five independent design programs which performs detailed designs of the concrete gravity and cantilever walls, gabions, reinforced earth and sheetpile structures. The SELECTWALL and the design module are linked by an M.1 external code EXT through a control program CALL. All the design procedures are coded using the C programming language. The methodology of the development of the prototype ES 'RETAININGEARTH' is described together with illustrated consultations and the hard copy of the codes developed for the selection and design modules are given in Appendices A-G.

# CHAPTER 1

# INTRODUCTION

## 1.1 INTRODUCTION TO EXPERT SYSTEMS

Knowledge-based expert systems (KBES) are computer programs based on artificial intelligence (AI) techniques, designed to reach the level of performance of a human expert in a specialized problem solving domain. Expert Systems (ES) have a great potential for practical use in ill-structured problem solving domain, such as retaining structure selection, where explicit algorithms do not exist. Table 1.1 shows the characterizing features of traditional programs and ES.

Table 1.1. Characteristics of traditional programs and expert systems (Maher, 1987)

| | Traditional Programs | Expert Systems |
|---|---|---|
| i) | Representation and use of data | Representation and use of knowledge |
| ii) | Knowledge and control integrated | Knowledge and control separated |
| iii) | Algorithmic (repetitive) process | Heuristic (inferential) process |
| iv) | Effective manipulation of large data bases | Effective manipulation of knowledge bases |
| v) | Programmer must ensure uniqueness and completeness | Knowledge engineer inevitably relaxes uniqueness and completeness restraint |
| vi) | Midrun explanation impossible | Midrun explanation desirable and achievable |
| vii) | Oriented toward numerical processing | Oriented toward symbolic processing |

## 1.2 ES APPLICATIONS TO RETAINING STRUCTURES

Retaining structures are frequently used to support earth and maintain a difference in elevation of the ground surface. With a variety of earth retaining systems which are in practice, and with the rapid influx of new techniques, and the many restrictions imposed on construction, it is becoming more difficult for the design engineer/owner to determine which of the available systems is the most suitable one for a given set of site conditions. Selection of an appropriate and technically feasible structure for a given set of criteria requires considerable judgement and expertise. ES technology provides a medium that formalizes heuristic and algorithmic techniques. By separating the knowledge from the control structure of the program, these judgements and expertise can be incorporated into the knowledge base of the ES, where addition and modification of rules are easily accomplished. The rapid development of new techniques in the area of retaining structures leads to knowledge improvements and modifications to the state-of-the-art which could be advantageously exploited in the development of the ES. Discussions on the state-of-the-art of ES applications in the design of structures are presented in Reference 1.1.

A wide choice of retaining wall types is available to a designer faced with an earth retention problem. The decision as to which type of wall is most appropriate for a particular situation depends on several factors, including wall height, soil conditions and the location of the wall. Although there is an abundance of literature available on design and construction details for each particular wall type, very little documentation exist on which wall type is better for a particular situation. Instead, designers have to rely upon their own experience and expertise. The ES specifically addresses the problem of choosing applicable wall types.

An existing developmental prototype called RETWALL was developed in Quintus Prolog, which addresses the problem of choosing an appropriate wall type for a given set of conditions [1.2]. RETWALL is based on an ES shell called BUILD. The knowledge in the system is represented as a series of production rules that exist in a separate module accessed by the BUILD shell. BUILD provides the control structure for the overall system. BUILD consists of a series of production rules that provide both goal driven and data driven control, in addition to an explanation facility. The system queries the user by stepping through the rules in a sequential manner. To obtain a second level of control within the knowledge base, advantage was taken of the sequential operation of the shell. By placing the rules in the knowledge base in an order that reflects the expert's preference of wall types, the system bases its recommendations on the first wall type that is considered acceptable. In its present state, RETWALL is able to make a choice among ten available wall types. Additionally the system will carry out the actual wall design for the blockwork wall and provides the user with a set of construction drawings.

Another developmental prototype ES [1.3] also presented a methodology for retaining wall selection and design in which the selection part was formulated in the form of production rules using OPS5 and the design part written in BASIC. This ES consists of the selection and preliminary design modules. The selection part is composed of two modules for wall selection and comparison. The selection module chooses possible types of walls and the comparison module selects the best wall if the selection module chooses more than one possible type of wall.

## 1.3 OBJECTIVES

The present study describes (i) the overall concept of the ES for the retaining structure selection, (ii) development of the higher level selection and design modules. The design modules present the detailed design of five different types of retaining structures.

## REFERENCES


1.1 M. Arockiasamy & S. Lee, State-of-the-Art on Expert Systems Applications to the Design, Construction and Maintenance of Structures, Final Draft Report, U.S. Army Engineer Waterways Expt. Station, Vicksburg, Mississippi, 1989, pp. 150.

1.2 PJ Hutchinson, An Expert System for the Selection of Earth Retaining Structures, M.S. Thesis, Dept. of Architectural Science, Univ. of Sydney, 1985, pp. 215.

1.3 SJ Fenves, M.L. Maher and D. Sriram, Expert Systems: C.E. Potential, Civil Engineering, ASCE, October 1984, pp. 45-47.3

1.4 ML Maher, Expert System Components, ASCE Expert Systems Committee Report on Expert Systems for Civil Engineering: Technology and Application, 1987.

# CHAPTER 2

## FORMULATION OF THE KNOWLEDGE BASE FOR THE EXPERT SYSTEM

### 2.1 KNOWLEDGE SOURCES

The real problems involved in building expert systems are those related to knowledge representation. The emphasis in the building of expert systems always seems to be on investigating technical issues and the implementation of the knowledge already elicitated. The information on the knowledge base of the expert system can be obtained from two sources - literature and domain specific knowledge from experts. Literature sources include technical journals, textbooks, manuals, public and commerical documents and reports. A second source of domain specific knowledge is from experts to aid in the development of the system by providing their experience, intuition, judgments, rules of thumb, etc.

In the domain of earth retaining structure selection the literature is fairly scant. However, a number of good references exist for the actual design of various types of earth retaining structures. The problems involved with the design of earth retaining structures are discussed in References 2.1, 2.2, and 2.3. However, these references give the reader only little guidance as to the steps and factors involved in the selection of one type of earth retaining structure over another.

The knowledge of human experts in the retaining structure construction industry, and public organizations has been reported in the recent conference on "Design and Performance of Earth Retaining Structures", organized by the

American Society of Civil Engineers [2.4]. A major portion of the knowledge base for the selection module was formulated using the above reference. Numerous fact and rules were formulated using information gathered from various geotechnical and structural engineering text books and journals.

## 2.2 KNOWLEDGE BASE ON EARTH RETAINING STRUCTURES

### 2.2.1. General

Selection of an earth retaining structure is influenced by many factors involving technical, practical, economical and political considerations. Earth retaining structures are widely used for support of cuts or fills in surface, marine and underground construction. Site conditions usually dictate the basic wall type and size. The typical structure types for the earth fill and cut situations are shown in Table 2.1.

### 2.2.2. Earth Retaining Structures

Depending on the basic mechanisms of support, retaining walls can be broadly classified under externally and internally stabilized systems. Fig. 2.1 shows the classification scheme for earth retention systems. An externally stabilized system uses an external structural wall, against which stabilizing forces are mobilized whereas an internally stabilized system involves reinforcements installed within and extending beyond the potential failure wedge [2.5].

Table 2.1 Fill/Cut Structure Categorization [2.5]

| Fill Retention | Cut Retention |
|---|---|
| Mechanically Stabilized | Permanent Ground Anchor (tiebacks) |
| Crib | Soil Nailing |
| Metal Bin | Reticulated Micro-Pile |
| Gabion | Slurry |
| Gravity | Cylinder Pile |
| Cantilever | Lagged Soldier Pile |
| Counterfort | Sheet Pile |
| Deadman Anchor | |

- EXTERNALLY STABILIZED SYSTEMS
  - In-Situ Walls
    - * timber
    - * precast concrete
    - * sheet piles
    - * soldier piles
    - * cast in-situ
      - - slurry wall
      - - secant pile
      - - tangent pile
    - * bored-in-place
    - * soil-cement
    - Braced
      - * cross-lot
      - * rakers
    - Tied-back
      - * augered
      - * belled
      - * pressure injected
  - Gravity Walls
    - * masonry
    - * concrete
    - * cantilever
    - * counterfort
    - * gabion
    - * crib
    - * bin
- INTERNALLY STABILIZED SYSTEMS
  - Reinforced Soils
    - * metallic, polymeric and organic reinforcing strips and grids
    - * anchored earth
  - In-Situ Reinforcement
    - * soil nailing
    - * reticulated micro piles
    - * soil doweling
- HYBRID SYSTEMS --- SPECIAL MATERIALS
  - * tailed gabions
  - * tailed masonry
  - * polymer impregnated soil
  - * low density fills
    - - low density cement
    - - expanded polystyrene

Figure 2.1 Classification Scheme for Earth Retaining Systems [2.5]

Virtually all traditional walls may be regarded as externally stabilized systems. These systems can further be classified into gravity walls and in-situ walls. The gravity walls, e.g. cantilever, counterfort, gabions, bins, cribs, etc. support the soil through weight and stiffness to resist sliding, overturning, and excessive shear and moments. The in-situ walls consist of timber, precast concrete, sheet piles, soldier piles, cast in-situ (slurry wall, secant pile, tangent pile), soil-cement, tied-back and braced structures.

Internally stabilized walls consist of reinforced soils with horizontally layered elements, such as metallic strips or polymer grids, and soil nailing, in which metallic bars or dowels are installed during the in-situ construction. The reinforced soil systems consist of metallic, polymeric and organic reinforcing strips and grid types of reinforcements and anchored earth structures. The in-situ reinforcement systems consist of soil nailing, reticulated micro piles and soil doweling. A fundamentally new concept has been introduced in the area of internally stabilized systems in which the shear transfer is used to mobilize the tensile capacity of closely spaced reinforcing elements. This has eliminated the need for a structural wall in retaining structures and substituted instead a composite system of reinforcing elements and soil as the primary structural entity. Typical examples of externally and internally stabilized earth retaining structures are shown in Fig. 2.2.

Hybrid systems have emerged as a result of the combination of both internally and externally supported systems. Examples include the

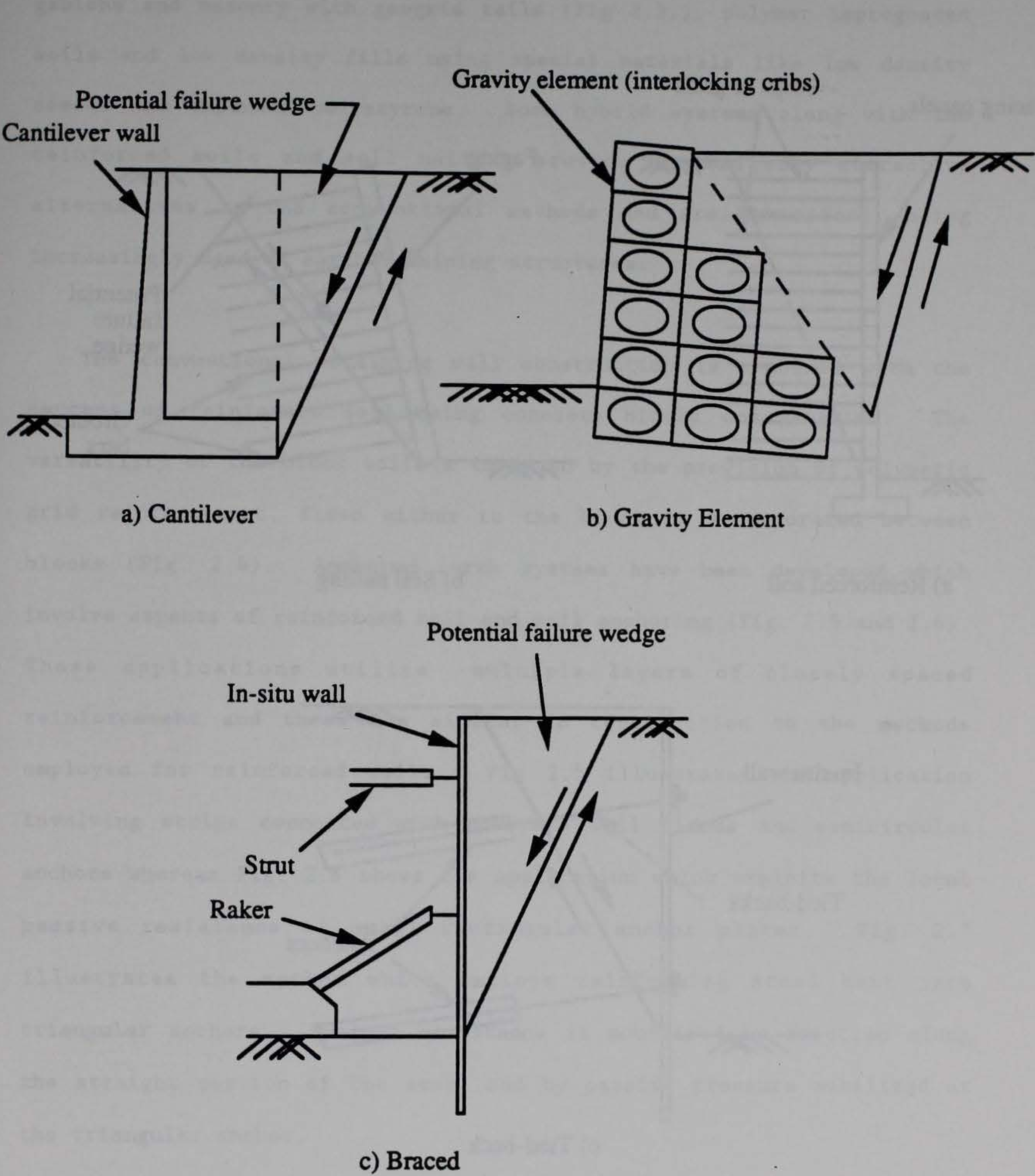



Figure 2.2a Typical Earth Retaining Systems (Externally Stabilized) [2.6]

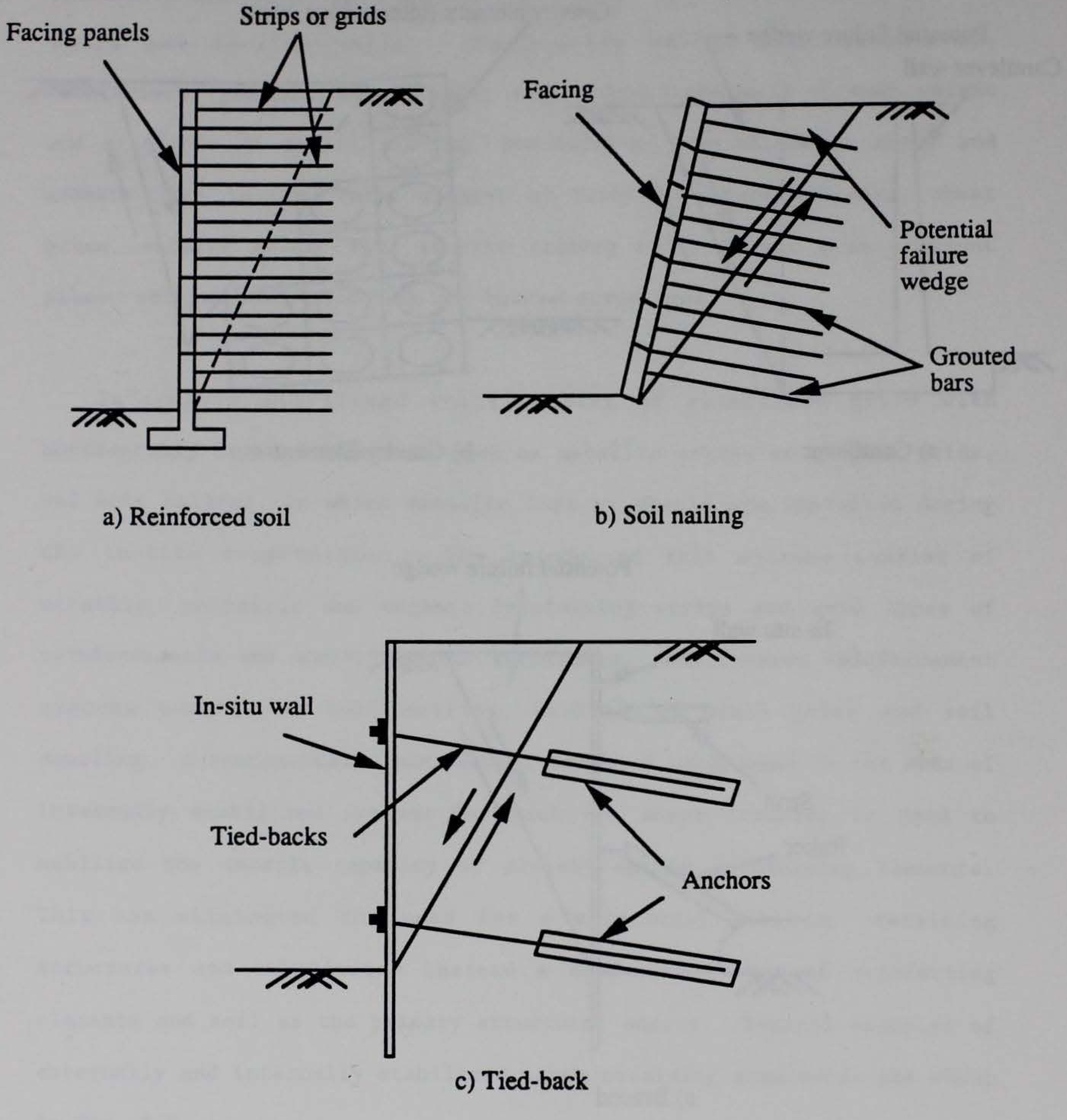



Figure 2.2b Typical Earth Retaining Structures (Internally Stabilized) [2.6]

gabions and masonry with geogrid tails (Fig 2.3.), polymer impregnated soils and low density fills using special materials like low density cement and expanded polystyrene. Such hybrid systems along with the reinforced soils and soil nailing provide economically attractive alternatives to the conventional methods and are, therefore, being increasingly used as earth retaining structures.

The conventional retaining wall construction is combined with the concept of reinforced soil using concrete blocks and geogrids. The versatility of the block wall is improved by the provision of polymeric grid reinforcement, fixed either to the block or incorporated between blocks (Fig. 2.4). Anchored earth systems have been developed which involve aspects of reinforced soil and soil anchoring (Fig. 2.5 and 2.6). These applications utilize multiple layers of closely spaced reinforcement and these are similar in construction to the methods employed for reinforced soils. Fig 2.5 illustrates the application involving strips connected with concrete wall blocks and semicircular anchors whereas Fig. 2.6 shows the application which exploits the local passive resistance of small rectangular anchor plates. Fig. 2.7 illustrates the method which employs reinforcing steel bent into triangular anchors. Pullout resistance is mobilized by friction along the straight portion of the steel and by passive pressure mobilized at the triangular anchor.

The development of earth retention systems can be viewed as an evolutionary process in which methods for supporting soil have involved progressively more alterations and insertions of reinforcing elements.

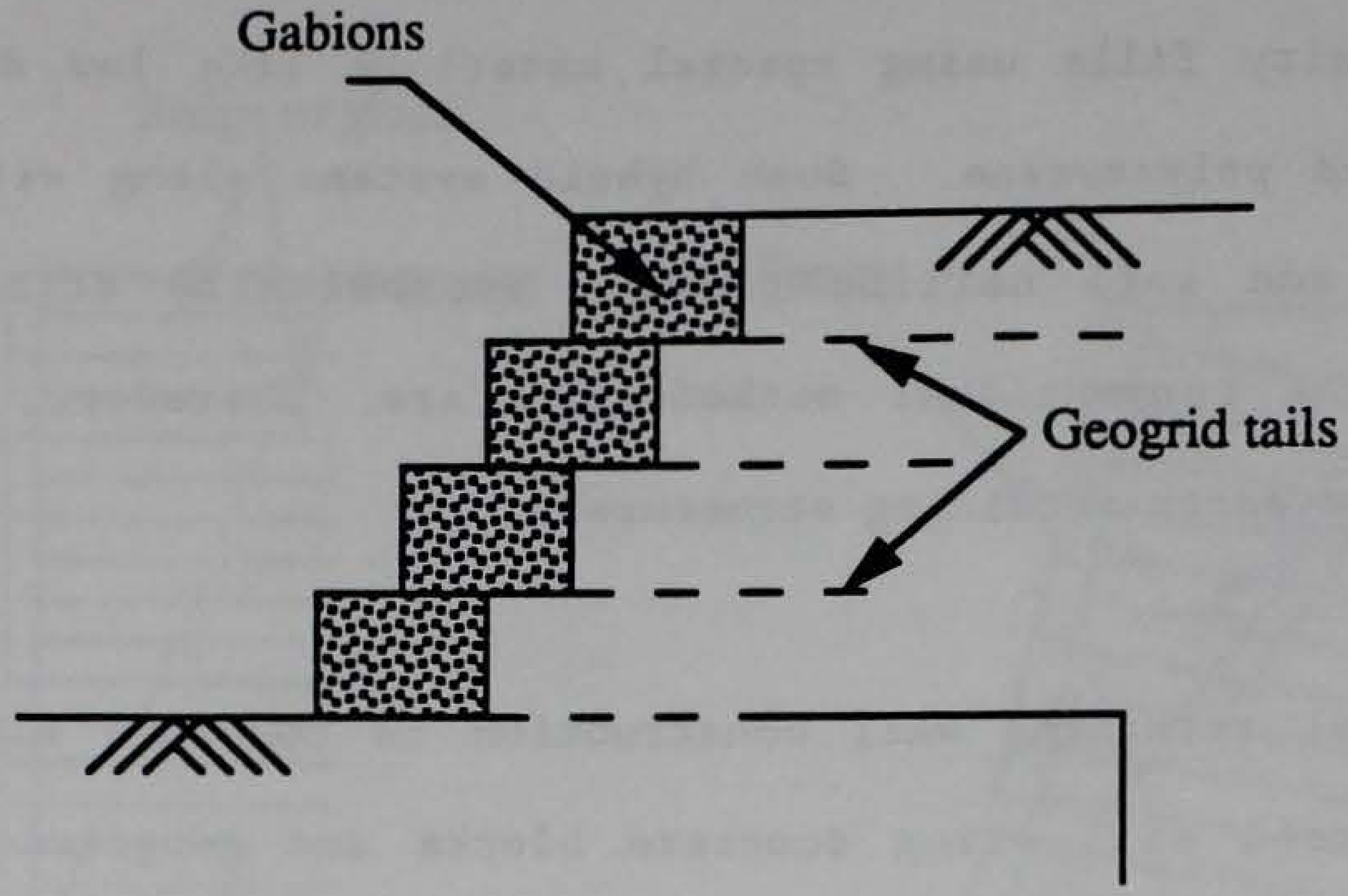



Figure 2.3 Gabions with Geogrid Tails. [2.6]

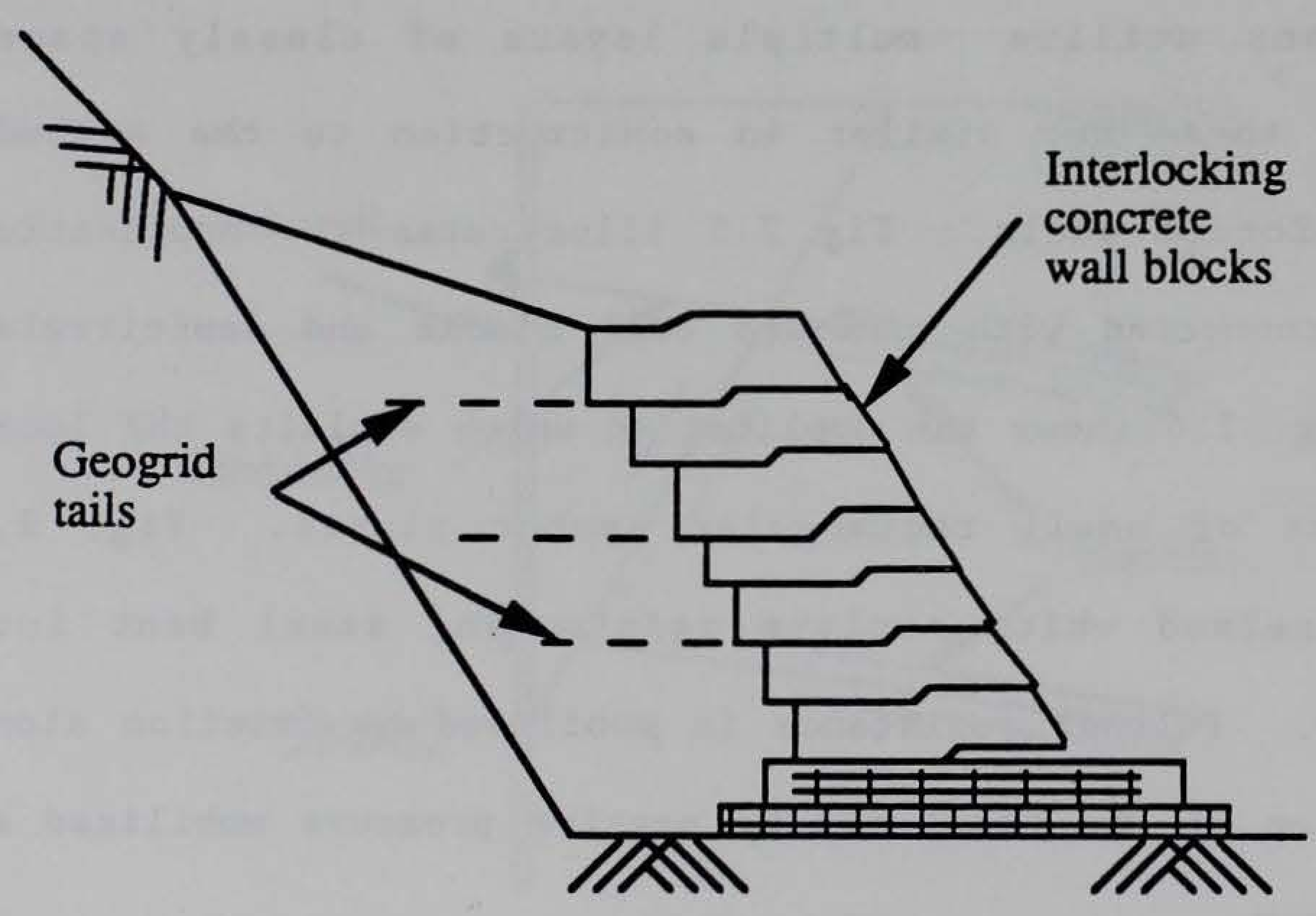



Figure 2.4 Concrete Block Wall with Geogrid Tails [2.6]

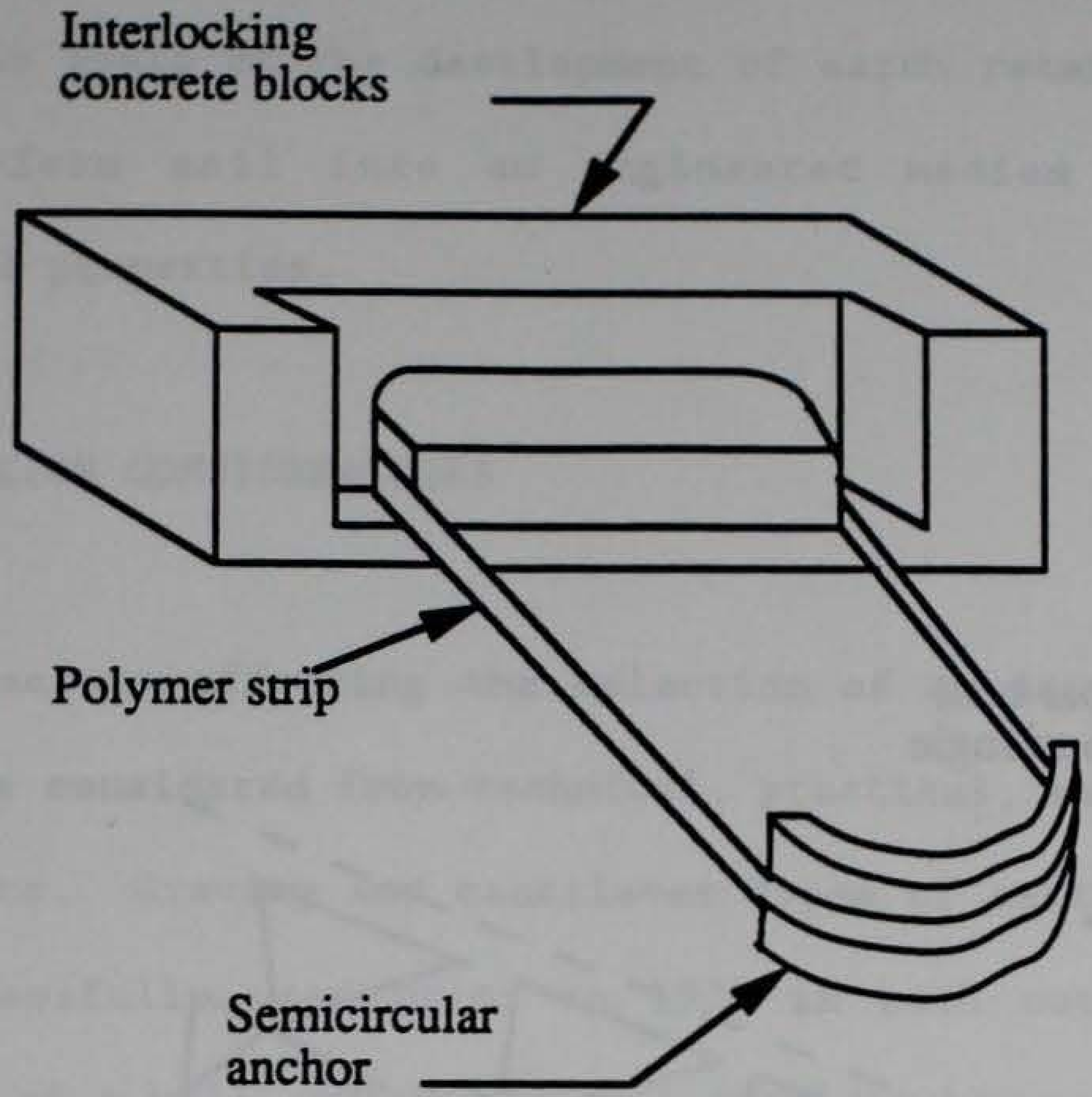



Figure 2.5 Wall System with Concrete Blocks, Polymer Strips, and Anchors. [2.6]

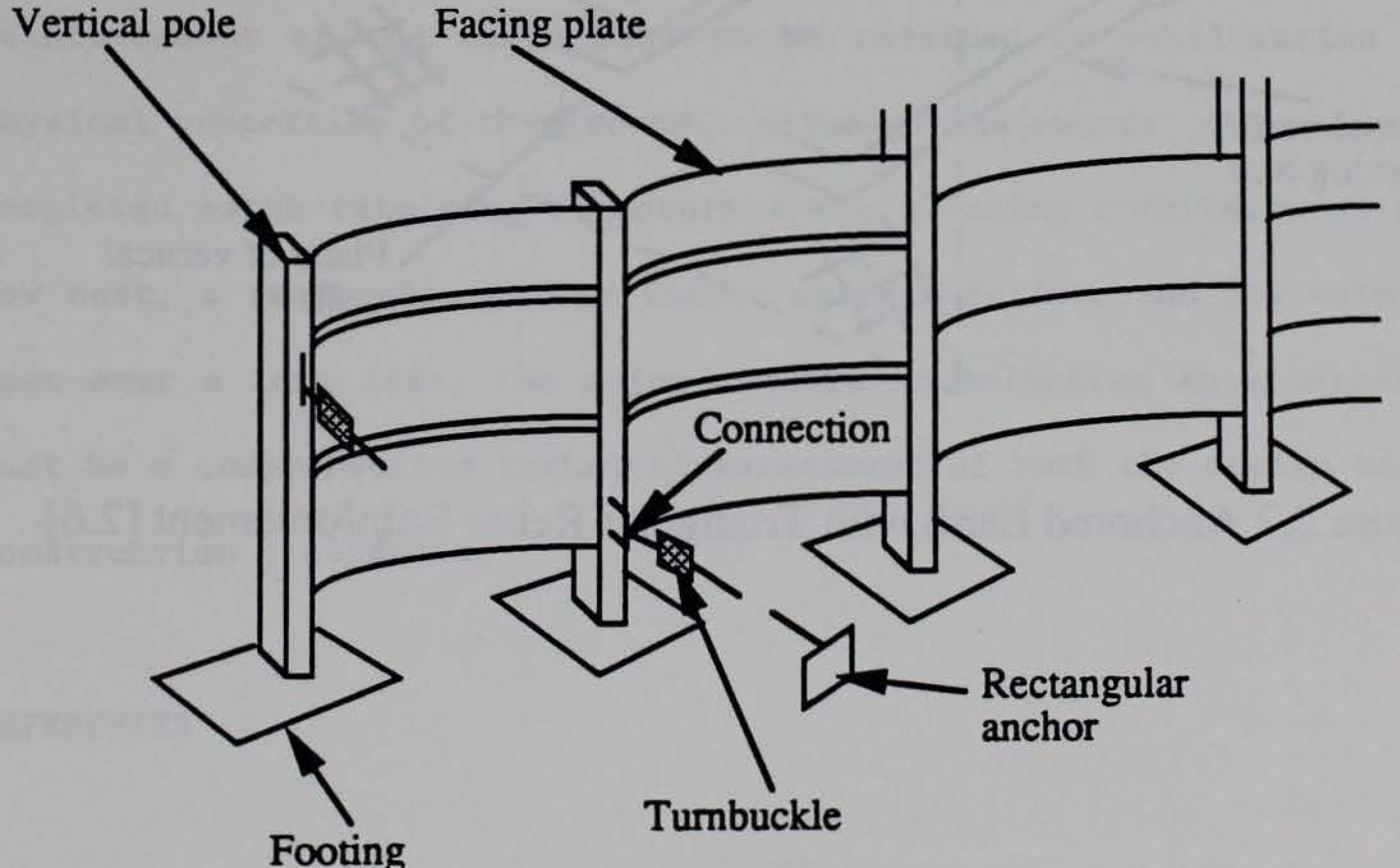



Figure 2.6 Wall System with Facing Plates and Rectangular Anchors [2.6]

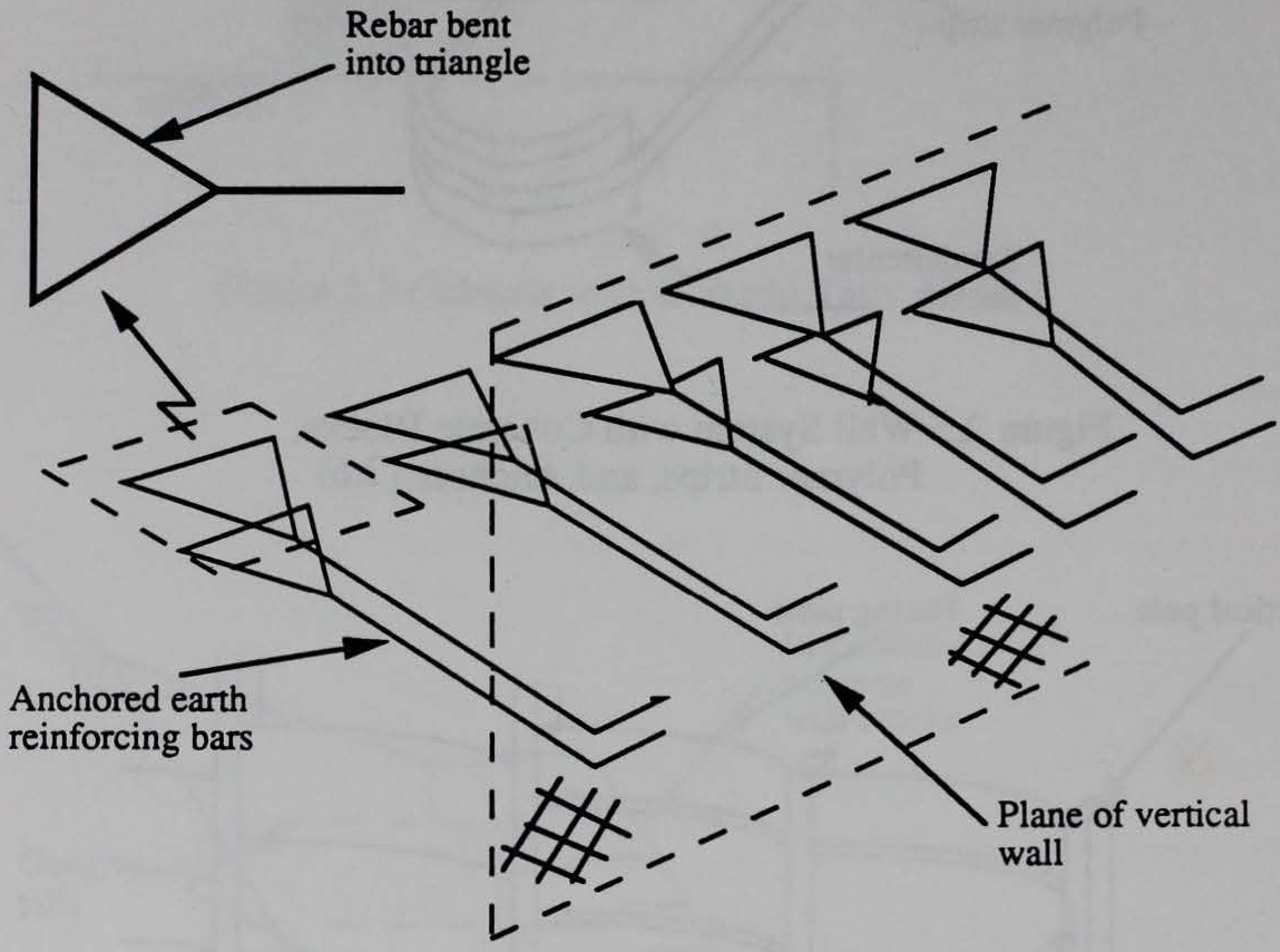



Figure 2.7 Anchored Earth with Triangular Rebar Reinforcement [2.6].

One of the goals of the development of earth retention systems has been to transform soil into an engineered medium because of enhanced mechanical properties.

## 2.3 SELECTION CONSIDERATIONS

The factors affecting the selection of an earth retaining structure have to be considered from technical, practical, economical and political view points. Gravity and cantilever types of earth retaining walls have been successfully used prior to 1970 in both cut and fill situations. Selection of a wall system today is considerably more complex than in 1970 since an ever growing number of innovative alternates are now available to the designer in preference to the conventional gravity and cantilever walls. These alternate systems generally rely on either reinforcement of the earth mass to be retained or mobilization of the physical properties of the ground. Although the owners' objectives for a completed earth retaining structure are a pleasing architectural facade, low cost, a reasonable safety factor against failure and low maintenance cost over a long life, the major concern in selecting an alternate wall must be a comprehensive technical assessment of both the design basis and construction procedures.

## REFERENCES

2.1 JE Bowles, Foundation Analysis and Design, 4th Ed., McGraw Hill Book Co., 1988, 530-642.

2.2 M Carter, Geotechnical Engineering Handbook, Pentech Press, Plymouth, 1983.

2.3 K Terzaghgi & RB Peck, Soil Mechanics in Engineering Practice, Wiley, New York, 1967.

2.4 Design and Performance of Earth Retaining Structures, GSP 25, Ed PC Lambe & LA Hansen, ASCE, New York, 1990, pp. 904.

2.5 RS Cheney, Selection of Retaining Structures: the Owner's Perspective, GSP 25, Proc. Conference on Design and Performance of Earth Retaining Structures, Ed PC Lambe & LA Hansen, ASCE, New York, 1990, 52-65.

2.6 TD O'Rourke & CJFP Jones, Overview of Retention Systems: 1970 - 1990, GSP 25, Ed PC Lambe & LA Hansen, ASCE, New York, 1990, 22-51.

2.7 JL Walkinshaw, Handout on Retaining Wall Alternates, FHWA - HST-09, San Francisco, CA 94105, April 1989.

# CHAPTER 3

# KNOWLEDGE REPRESENTATION

## 3.1 INTRODUCTION

Knowledge-based expert systems (KBES) are identified by their method of representing and processing domain-specific problem-solving knowledge. A general block diagram of an expert system (ES) is illustrated in Fig. 3.1. The essential components of an ES are the knowledge base, the inference engine, the context and the user interface. Each of these components is described in Section 3.3.

The representation and management of knowledge form the core in artificial intelligence (AI) since a comprehensive collection of experiential/heuristic facts is the key to a high-performance intelligent system. Thus the purpose of knowledge representation is to organize required information in a form such that the ES can readily access it for making decisions, planning, analyzing scenes, recognizing objects and situations, drawing conclusions and other cognitive functions.

## 3.2. Types of Knowledge Representation

The main types of knowledge representation are logic-based, rule-based and network-based representations and schemes.

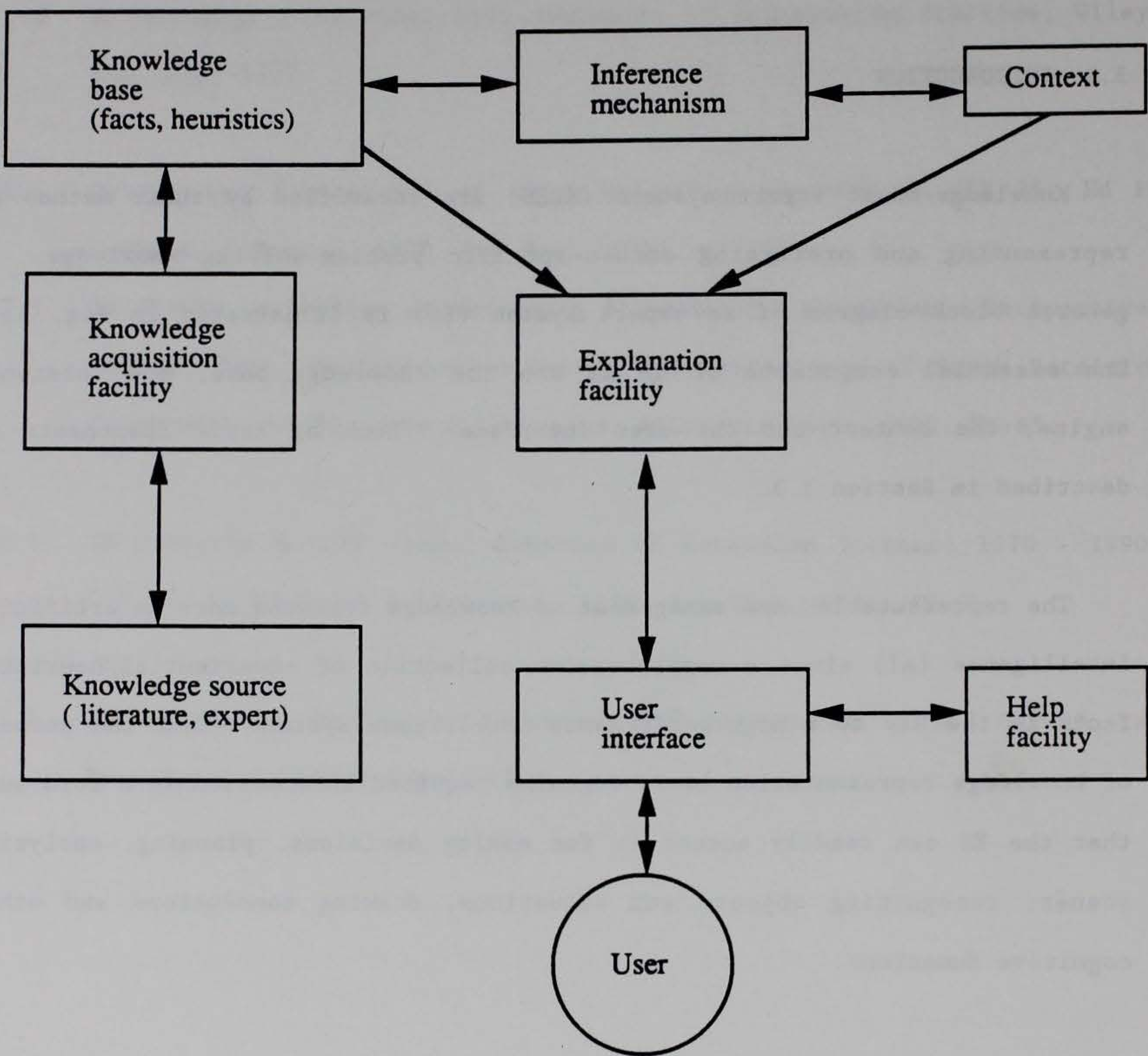



Figure 3.1 Block Diagram of an Expert System

### 3.2.1 Logic-based representations

A logic-based scheme is one in which knowledge is represented as assertions in logic, usually first-order predicate logic or a variant thereof [3.1]. This form of representation is generally coupled with an inference procedure based on theorem proofs. Logic-based language allows quantified statements and other well-defined formulae as assertions. PROLOG is an example of logic-based representation language. Overall, the flexibility and precision of mathematical logic make the logic-based representation both a useful tool and a standard of comparison for alternative knowledge representation schemes [3.2]. The rigor of logic has an advantage since it is clear what is known and how that knowledge will be used.

Fig. 3.2. illustrates the general form of a typical logical process in which the inputs to the logical process are called **premises** and the output (conclusions) as **inferences**. Both **deductive** and **inductive** types of reasoning are used in logic to make inferences from premises. When a specific inference is obtained using general premises, the process is called deductive reasoning or deduction. A number of established facts or premises is used in inductive reasoning in order to draw some general conclusion. The conclusion is never final or absolute unless all possible facts are included in the premises.

**Computational logic** is used to convert the deductive or inductive reasoning process into a form suitable for manipulation by a computer. The two basic forms of computational logic are:

i) **propositional logic** and ii) **predicate logic**. In propositional logic symbols such as letters of the alphabet are used to represent various propositions, premises or conclusions. More complex premises are formed

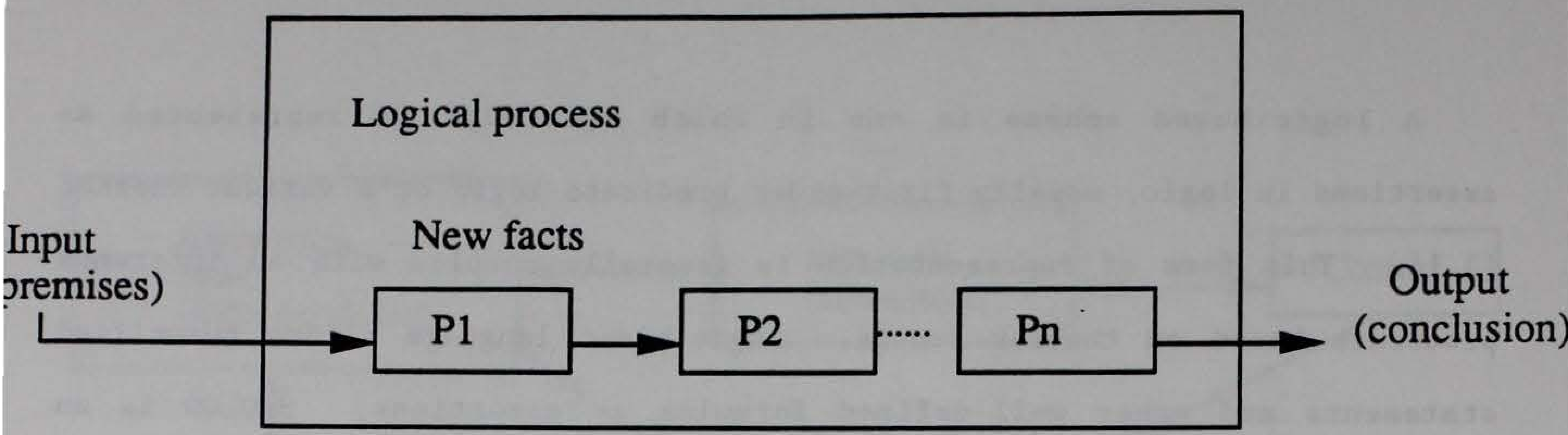



Figure 3.2 Reasoning by Logic

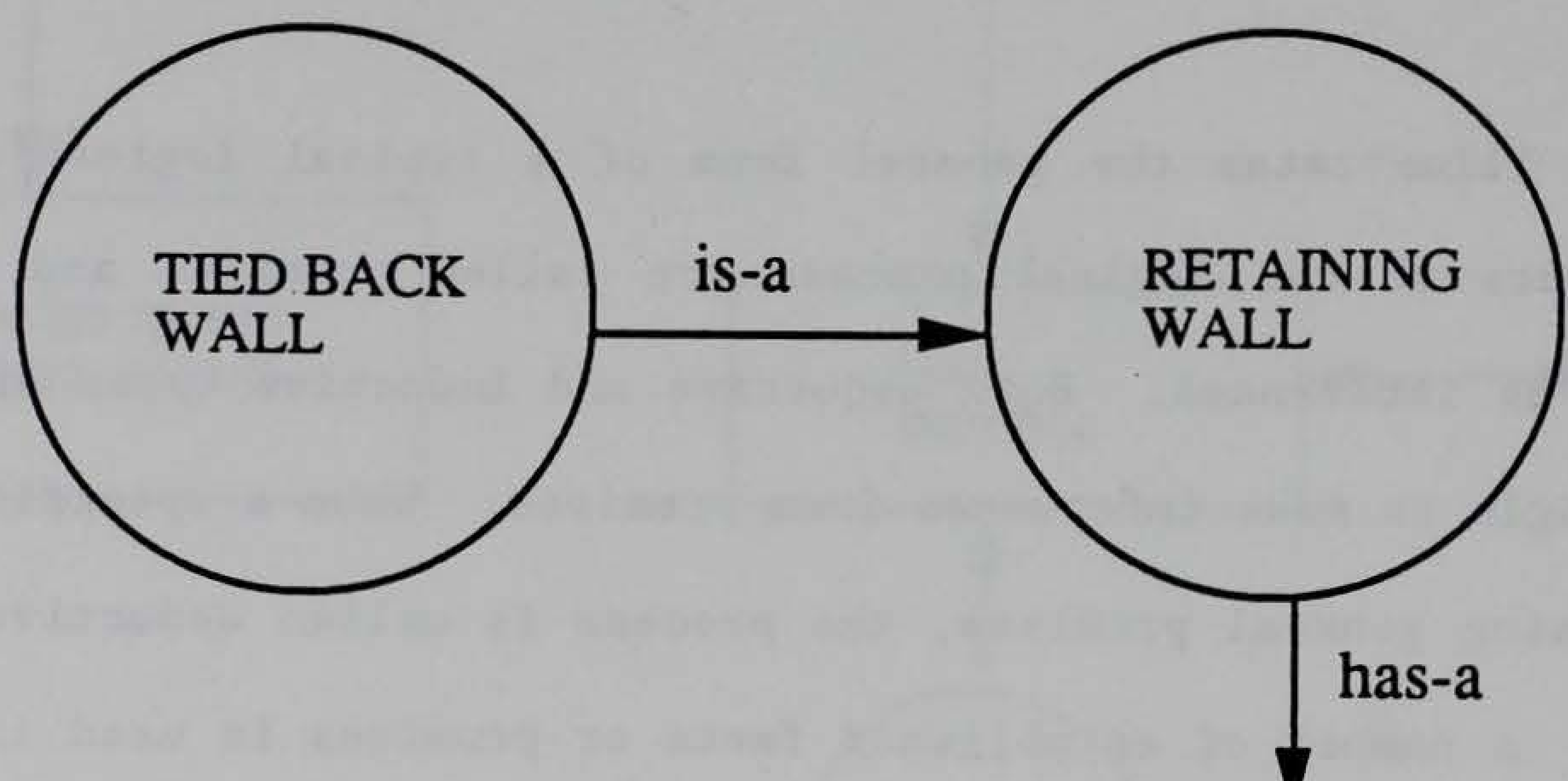



Figure 3.3 Example of Representing Knowledge in Semantic Network

combining two or more propositions using logical connectives. Predicate logic is a more sophisticated form of logic which has enhanced ability to represent knowledge in finer detail. It permits one to break a statement down into component parts namely an object, a characteristic of the object, or some assertion about the object. The use of variables and functions of variables in a symbolic logic statement leads to a more powerful knowledge representation scheme which is more directly applicable to practical problems to be solved on a computer.

A valid combination of objects and constructor (quantifiers, connectivities, functions, and predicates) forms Well-Formed Formula (WFF) in logic. Two basic quantifiers are used in logic: $\forall$ (for all) and $\exists$ (there exists). Connectivities are $\wedge$ (and), $\vee$ (or), $\neg$ (not), $\rightarrow$ (implies), and $\equiv$ (equivalent). A function returns a value by defined function and its arguments. The function PLUS(2,4), for example, returns the sum of 2 and 4 if the function PLUS is defined as sum of first and second argument. The predicate gives a value of true or false: Examples of predicates are < (less than), = (equal to), >= (greater than or equal to).

To illustrate the use of logic to represent knowledge, the following example is considered:

Tied-back wall is a retaining wall.

Every retaining wall has soil pressure.

The above facts are expressed in logic in the following form:

retaining-wall (Tied-back-wall)

$\forall$ X retaining-wall (X) soil-pressure (X)

By deductive reasoning, a new fact can be inferred as follows:

retaining-wall (Tied-back-wall) → soil-pressure (Tied-back-wall). That is, tied-back-wall has soil pressure.

### 3.2.2. Rule-based representation

Most rule-based systems (RBS) can be classified as production systems. The core idea of these tools is that the domain knowledge is represented in the form of modular rules known as production rules. The first part of the rule, called the **antecedent**, expresses a situation or premise while the second part, called the **consequent**, states a particular action or conclusion that applies if the situation or premise is true. The most common forms of production rule are of these formats:

| | | |
|---|---|---|
| ANTECEDENT | ——— | CONSEQUENT |
| SITUATION | ——— | ACTION |
| PREMISE | ——— | CONCLUSION |

The first or left-hand part of the rule is a statement with the prefix IF. The second or right-hand part of the rule is a statement with the prefix THEN. The action, consequence or conclusion stated in the THEN part is valid if the IF part of the rule is true or meets certain conditions. Production rules are by far the most popular and widespread means of converting human knowledge into a format suitable for symbolic representation in a computer.

The first widely used ES shell was created by stripping the medical knowledge ban from MYCIN and called EMYCIN (for Essential MYCIN or Empty MYCIN) which was used to construct diagnosis systems. DENDRAL analyzes mass spectrographic, nuclear magnetic resonance, and other chemical experimental data nuclear magnetic resonance, and other checmical experimental data to infer the plausible structures of an unknown compound. CADUCEUS consists of an extremely large semantic network of relationships between diseases and symptoms in internal medicine. These systems were constructed using rule-based representations. In general , RBS have the advantages of homogeneity and simplicity which permit self-inspection (explanation, meta-rules, consistency and self-knowledge) and help with rule acquisition. The main drawback is that the ability to express general relations between pieces of knowledge is severely restricted.

IF height of retaining wall exceeds 15 ft. and
location of wall to be built is not building and
foundation is good or poor and
reinforced earth is aesthetically acceptable and
enough excavation space exists and
there is no future excavation behind the wall
THEN reinforced earth is possible selection.

A set of production rules forms a production system to define some domain knowledge accurately, and this results in the solution of the subproblem by inference which is the clue to the final solution. For example, a set of production rules may be of the form:

(a b c) ----> (d e)

(d f) ----> (g)

(g h i j) ----> (k)

These rules imply that if a, b, and c are true, d and e are fired. By using d which is obtained from the previous rule and f, new consequent g is generated.

3.2.3. Network-based Representation

Semantic networks are basically graphical representations of knowledge that show hierarchical relationships between objects. It is made up of a number of circles or nodes which represent objects and descriptive information about those objects. Objects can be any physical item and the nodes can also be concepts, events, or actions. The nodes in a semantic network are also interconnected by links or arcs. These arcs show the relationships between the various objects and descriptive factors. Some of the most common arcs are of the is-a or has-a type. Is-a is used to show class relationship; i.e., an object belongs to a larger class or category of objects. Has-a links are used to identify characteristics or attributes of the object nodes.

The semantic network is a very flexible method of representing knowledge. Almost any kind of object, attribute or concept can be defined and the relationships created with links. Although the semantic network is graphic in nature the various objects and their relationships are stated in verbal terms and are programmed into the computer using one of several different kinds of languages, viz. C, PROLOG, LISP, COBOL, PL1 and PASCAL. The computer uses various search and pattern matching techniques to look through the network structure to identify the desired objects and determine their relationship as posed by the question.

A simple semantic network is shown in Fig. 3.3. The ability of higher level nodes to pass down properties to lower level nodes is called inheritance. By establishing inheritance hierarchy in the network, a third statement can be inferred. In this example, one can infer that "tied-back wall has soil pressure". Semantic network has an advantage that huge amounts of space are saved in the complex domain since information can be stored in one central location without repeating the information about similar nodes at each note.

### 3.2.4. Schemas

Stereo-typed knowledge which is based on previous exposure to typical objects and situations as well as experience with conventional unvarying processes is represented with **schemas**. A schema is a method of organizing, presenting and using stereo-typed knowledge for computer reasoning. Frames and scripts are the two basic types of schemas.

#### 3.2.4.1. Frames and scripts

A **frame** is a relatively large block of knowledge about a particular object, event, location, situation, or other element. The frame describes that object in great detail. The detail is given in the form of slots which describe the various attributes and characteristics of the object or situation. It is a connection of nodes and relations organized in such a way that the higher level nodes represent general concepts and the lower nodes are more specific to illustrate those concepts. In a frame system, the concepts include a number of attributes (e.g., location, purpose, height, length) called the slot and the values of those attributes (road, permanent, 32 ft, 230 ft.) A set of procedures is attached to each slot to make a reasoning process by communication between higher and lower level information. The procedure uses the values in the higher level frame to fill the slot in the lower level frame. A new frame is selected and the filling process repeated until the frame has satisfactory results. Frames are usually used to represent stereotyped knowledge. Fig. 3.4 shows an example of knowledge representation by frames. In this figure, "gravity type" frame has seven slots to be filled.

A **script** is a knowledge representation scheme similar to a frame but instead of describing an object, the script describes a sequence of events. The script, like the frame, portrays a stereotyped situation. Unlike the frame it is usually presented in a particular context. A sequence of events is described by the script using a series of slots containing information about the people, objects, and actions that are involved in the events. The elements of a typical script include entry conditions, props, roles, tracks and scenes. The entry conditions describe situations which must be satisfied before events in the script can occur or be valid. **Props** refer to objects which are used in the sequence of events that occur. **Roles** refer to the people involved in the script. The result is conditions which exist after the events in the script have occurred. **Track** refers to variations that might occur in a particular script. And finally, **scenes** describe the actual sequence of events that occur. A script is useful in predicting what will happen in a certain situation. Knowledge is stored in the computer in

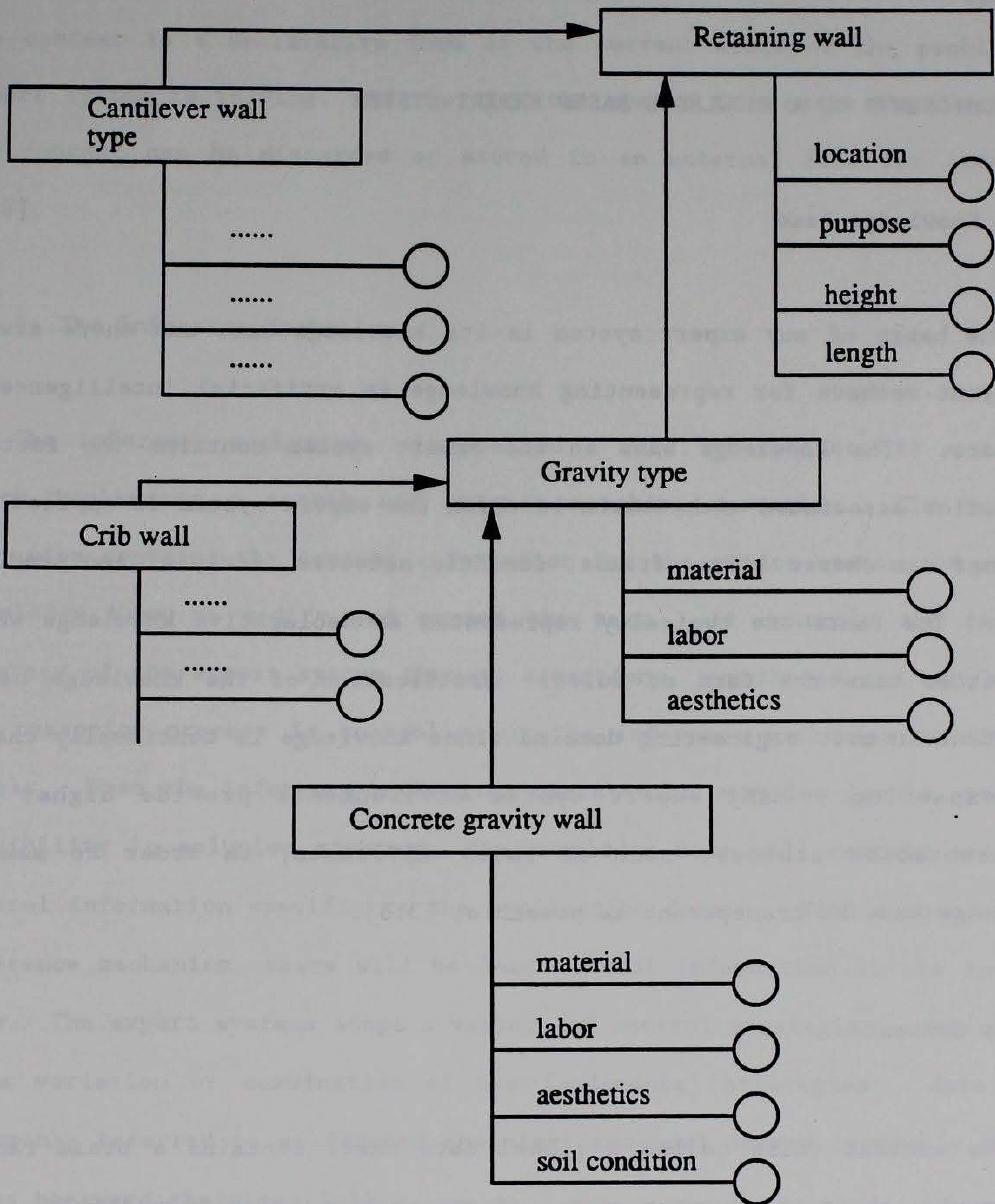



Figure 3.4 Knowledge Representation by Frame

symbolic form using LISP, or any other language which enables the use of the script.

## 3.3 COMPONENTS OF A KNOWLEDGE-BASED EXPERT SYSTEM

### 3.3.1 Knowledge Base

The heart of any expert system is its knowledge base and there are many different methods for representing knowledge in artificial intelligence (AI) software. The knowledge base in the expert system contains the facts and heuristics associated with domain in which the expert system is applied. The designer can choose lists, frames, semantic networks, scripts, and production rules. The facts are typically represented as declarative knowledge whereas heuristics take the form of rules. Modification of the knowledge base is important in most engineering domains since knowledge is continually changing and expanding. Many expert system environments provide higher level representation schemes, such as rules or frames, in order to make the knowledge base as transparent as possible. [3.3].

### 3.3.2. Context

The context (also called a global data base) contains a broad range of information about the current status of the problem being solved. The context is the component of the expert system which initially contains the information that defines the parameters of the problem. Prior to execution of the KBES the context is empty. As the KBES reasons about the given problem the context expands to include the information generated by the expert system to solve it.

At the completion of the problem-solving process, the context contains all the intermediate results of the problem-solving process as well as the solution. The context is a declarative form of the current state of the problems the expert system is solving. At the time the system terminates the contents of the context can be discarded or stored in an external file for later use. [3.3].

### 3.3.3 The Inference Mechanism

The inference mechanism, also called a rule interpreter, is a software which implements a search and pattern matching operation. The inference mechanism contains the control information for the expert system and uses the knowledge base to modify and expand the text. It controls the reasoning strategy of the expert system through assertions, hypothesis and conclusions. The reasoning process is controlled by the inference mechanism at different levels. When the inference mechanisms operates at very low levels, providing flexibility in solution strategy, the knowledge base will contain additional control information specific to the application domain. With a more specific inference mechanism, there will be less control information in the knowledge base. The expert systems adopt a variety of control strategies, each of which is a variation or combination of two fundamental strategies: **data driven** (commonly referred to as **forward chaining**) and **goal driven** (commonly referred to as **backward chaining** [3.3].

#### 3.3.3.1. Forward chaining

Forward chaining is one of the two basic approaches used to search for an answer beginning with some initial information and working forward with an attempt to match that information with a rule. The rule interpreter attempts to match a fact or statement in the data base to the situation stated in the lefthand or IF part of the rule. Once a fact in the database has been matched to the IF part of the rule, the rule is fired and the action or consequent stated, could produce new knowledge which is stored in the knowledge base. This new fact can then be used to search out the next appropriate rule and this searching and matching process continues until a final conclusion rule is fired.

3.3.3.2. Backward chaining

The second basic approach to search for an answer is backward chaining wherein the rule interpreter starts with a fact (hypothesis) in the database. The rule interpreter then begins examining the right hand or THEN parts of rules looking for a match. The inference engine searches for evidence to support the original hypothesis. If a match is found, the database is updated, recording the conditions stated in the rule for supporting the matched conclusion. The chaining process continues with the system continuously attempting to match the right side of the rule against the current system status. The corresponding IF sides of the rules matched are used to generate new intermediate hypothesis or goal states which are recorded in the database. This backward chaining continues until the hypothesis is proved.

3.3.3.3. Forward chaining vs. backward chaining

The choice of a control strategy with either forward or backward chaining is determined by the design of the system and the problem being solved. The forward chaining or data driven approach may be too slow in large systems with many rules since it will generate many sequences of rules. In such cases a backward chaining or goal driven approach may be more advantageous. On the other hand, depending on the rules themselves, the backward chaining process could get a fixation on a particular hypothesis and continue to explore it in spite of the data unavailability to support it. Some expert systems incorporate both backward and forward chaining which usually speeds up the process and ensures a solution.

### 3.3.4 User Interface

The user interface is a software in the expert system and it lets the user communicate with the system. The user interface asks questions or presents menu choices for entering initial information in the database. It provides a means of communicating the answer or solution once it has been found. Intermediate communications during the problem solving process are provided by the user interface. The user interface contains "canned" questions, statements or menu sequences. Information supplied by the user must usually be formatted in a special way or entered in a unique restricted syntax.

### 3.3.5 Explanation and Knowledge Acquisition Facilities

The explanation facility in an expert system provides answers to questions about the reasoning process used to develop a solution. A good explanation facility can explain to the user both why a certain fact is requested and how a certain conclusion was reached. The knowledge acquisition facility in an expert system is the component that facilitates the structuring and development of the knowledge base. This facility acts as an editor, and the expert should be able to add to or modify the knowledge base as the expert system reveals gaps in the knowledge base. The knowledge acquisition facility understands the inference mechanism being used and can actively aid the expert in defining the knowledge base.

### 3.3.6 Certainty Factors

Expert systems have the ability to deal with incomplete or uncertain information. At times, the user does not have the answer to a question put forth by an expert system in gathering initial inputs. There are many situations where the information is known with less than 100% confidence in its truthfulness. Under these circumstances, certainty factors are used with both the premise (IF) and conclusion (THEN) portions of a rule.

## 3.4 KNOWLEDGE REPRESENTATION IN M.1

M.1 is a powerful PC-based knowledge system software tool, implemented in the C programming language, capable of developing and using knowledge systems up to 2500 rules. Knowledge systems built with M.1 are designed around a knowledge base of facts and rules relating to a particular task or application and an inference engine that performs the reasoning process to solve specific problems in that application area. The M.1 knowledge base consists of facts, rules and expressions which are formulated by the knowledge engineer in M.1 syntax using a standard text editor.

The M.1 inference engine is activated when a consultation is initiated, usually with the **go** command which clears any previous conclusions and begins the consultation. M.1 then seeks the values for expressions that appear in the **goal** or **initialdata** specifications within the knowledge. In the ES RETAININGEARTH the initial data is given as

initialdata = [begin the consultation, end the consultation].

The above statement could also be represented as

goal = end the consultation or

goal = select-wall.

The following example can be used to illustrate the M.1 syntax:

multivalued(location).

question(location) =

'Which of the following list of sites best describes the location of the proposed retaining structure?'

legalvals(location) =

[excavation, roadway, sidehill, abutment, forest-area, building-related-excavation, mountainous-terrain, waterfront-area, railway, building, marine].

automaticmenu(location).

enumeratedanswers (location).

explanation (location) = ('The location where the structure is to be built is a very important consideration in the selection of retaining walls, since the parameter location, when coupled with one or more parameters e.g., site geometry and height, yields a specific type of wall to be selected for that particular application. You may choose the site which is closest to one of the above locations.')

The term 'multivalued (location)' means that the variable 'location' is multivalued in nature. The system asks the user to select the site which best describes the location of the proposed retaining structure. The legal values or answers that the system accepts are given as an **enumerated** list which is **automatically** displayed on the screen along with the question. This is made possible by the command **legalvals** (location), **automaticmenu** (location) and **enumeratedanswers** (location). The user may respond to this question by entering either a number corresponding to his selection or by entering sufficient letters to unambiguously distinguish the response or type 'why' to get an explanation to the question being asked.

A typical example of a rule formulated in M.1 is shown below:

```
rule-14:  if site = fill and
    not (location = excavation) or
    not (location = building-related-excavation) and
    height = H and (H > 15 and H < =50) and
    (length = L and L> 0) and
    rway = no and
    soile = no and
    time = medium or
    time = long and
    soila = yes and
    foundation = good or
    foundation = poor and
    vertload = yes or
    vertload = no and
    noise = no
    then possible-wall = reinforced-earth-wall.
```

For the **possible-wall = reinforced-earth-wall** to be activated or fired, all the conditions stated in the rule have to be satisfied. M.1 examines each condition in the rule and checks to see if there is a question in the knowledge base which relates to this condition. If so, this question is displayed to the user and the corresponding response is stored in the cache (or temporary memory location). Likewise, all the conditions are checked for and the facts are noted in the cache. If all the conditions are favorably answered, then the rule is fired or invoked. In this case the **possible-wall = reinforced-earth-wall** is displayed to the user and M.1 then proceeds to the next stage of the selection. As stated before, the inference engine continues this process of searching the rules, during which it displays the intermediate results to the user, until the goal stated in the initial data is achieved. The process of the selection of retaining structures using M.1 is discussed in detail in Chapter 4.

**REFERENCES**


3.1 BG Buchanan & RO Duda, Principles of Rule-Based Expert Systems, Technical Report HPP-82-14, Stanford Univ., Aug. 1982.

3.2 RO Duda & EH Shortliffe, Expert Systems Research, Science 220, April 1983, 261-268.

3.3 ML Maher, Expert System Components, ASCE Expert Systems Committee Report on Expert Systems for Civil Engineering: Technology and Application, 1987.

3.4 H. Adeli, Expert Systems in Construction and Structural Engineering, Chapman and Hall, New York, 1988.

3.5 F Hayes-Roth, DA Waterman and DB Lenat, Building Expert Systems, Addison-Wesley, Reading, MA, 1983.

3.6 WJ Rasdorf and SJ Fenves, Design Specifications Representation and Analysis, Proceedings of the Second Conference on Computing in Civil

Engineering, American Society of Civil Engineers, Baltimore, MD, June, 1980, pp 102-111.

3.7 WJ Rasdorf and LM Parks, Expert Systems and Engineering Design Knowledge, Proceedings of the Ninth Conference on Electronic Computation, American Society of Civil Engineers, Birmingham, AL, February, 1986.

# CHAPTER 4

# SELECTION OF RETAINING STRUCTURES

## 4.1 INTRODUCTION

The rapid influx of new techniques for retaining structure construction and the many restrictions associated with it have made it difficult to determine which of the available systems is the most suitable for a given situation. To determine the most appropriate structure, a thorough evaluation of the many factors involving the design, construction, use and maintenance of each system is required. Some of the most important factors affecting the selection, development methodology, and its implementation in M.1. are described in detail in the following sections.

## 4.2 RETAINING STRUCTURE SELECTION FACTORS

Selection of a retaining structure can be done from the point of view of the owner or the design engineer. The selection by an owner of a particular retaining structure type is motivated by various factors, such as, aesthetics, cost, risk and durability [4.1]. The important factors that affect the selection by a design engineer are the soil properties at site, construction considerations (availability of materials and labor, site accessibility, equipment availability, etc), groundwater, right-of-way, aesthetics, environmental concerns, durability and maintenance, cost, politics and tradition [4.2]. A brief description of some of the important factors affecting selection is given in the following sections:

### 4.2.1 Ground

The selection of the retaining structure is very much influenced by the earth, the structure is designed to retain as well as the earth on which it rests. As discussed in Section 2.3, the ground or site geometry dictates the wall types that are best suited for either the fill or cut situations. Table 2.1 shows the typical structure types for fill or cut situations. The soil properties also play a very important part in structure selection, especially when the retained earth itself has a major load-carrying function. In the case of walls, such as reinforced earth structures, the reinforcing elements have to develop sufficient pull-out force which is resisted by the friction along the soil-reinforcement interface and also by the passive resistance along the transverse members of the reinforcement, if any (e.g., grid reinforcement). The pull-out force of the reinforcing element is dependent on soil properties such as internal angle of friction.

Strain compatibility is another important factor which is considered in the structure selection. For structures which utilize reinforcing elements (e.g., reinforced earth, tieback, soil nailing, etc.), the strains required to mobilize the full strength of the reinforcing elements are much smaller than those needed to mobilize the full strength of the soil. Thus these systems are not suitable for retaining soils with low residual strength. The possibility of saturation and creep in clayey soils are also factors that deter the selection of systems using reinforcing elements. Gravity structures, on the other hand, are less influenced by the soil properties. When the soils have large vertical and horizontal deformations, a flexible gravity system such as the gabion structure may be selected instead of a more rigid system that resists large deformations.

### 4.2.2. Groundwater

The water table level in a region is an important consideration in retaining structure selection. Generally the groundwater table behind the earth retaining structure is lowered to reduce the hydrostatic pressure acting on the structure, and also to prevent saturation of the soil which increases displacements significantly leading to instability. Corrosion in the presence of groundwater is also an important parameter. The negative impact of groundwater can be reduced by using a free-drainage system in reinforced earth and gabion structures. In some cases it is desirable to keep the water table high to prevent settlement of adjacent structures or to protect the existing untreated timber pile foundations from fungus decay resulting from the exposure to oxygen. In such cases (e.g., slurry wall, tangent piles, etc.) a relatively rigid watertight structure capable of withstanding the full hydrostatic pressure is used.

### 4.2.3 Construction Considerations

The construction considerations for retaining structure selection include a construction schedule, availability of material, site accessibility, and equipment and labor availability.

Conventional cast-in-place concrete structures generally take more time to construct and hence these structures are not considered when there is a constraint on project time. Earth reinforced walls are chosen under these conditions, since no form work is required. Furthermore they are more easily constructed using the locally available backfill material. Another advantage of earth walls is that the construction can be accomplished regardless of the climatic conditions.

Site inaccessibility to heavy construction equipment in high mountainous terrain/thick forest areas is a deciding factor in the choice of an appropriate retaining system. At the same time, material availability is also an important parameter. As an example, it can be cited that a gabion structure would be an ideal choice, when there is an abundance of suitable rock backfill. When suitable aggregate has to be hauled a long distance to the project site, concrete retaining structures are usually avoided.

#### 4.2.4 Right-of-way

When there is insufficient space behind the structure face, reinforced compacted earth systems are not preferred since these require relatively large space behind the structure face (0.7 times the wall height) compared to that needed for the construction of conventional concrete structures. When there is a very little space, a top-down staged excavation and support system such as soil nailing may be chosen as a suitable alternative. The feasibility of such structures which involves reinforcing elements as part of the structure, is influenced by the availability of utilities, buried structures in the vicinity, and good soil.

#### 4.2.5 Aesthetics

A pleasing architectural facade is usually mandated by environment impact statements on sensitive projects. Different types, shapes and color facings are used in the construction of earth walls. The types of facings range from built on-site continuous facings (shotcrete, welded wire mesh, cast-in place concrete, etc.) to prefabricated concrete or steel panels. The aesthetic factor is extremely important when building a retaining structure in parks, forests and natural habitats. Under such circumstances walls such as the Evergreen walls, which consist of precast concrete units with open spaces at the faces into which shrubs, vines, etc may be planted, are the most attractive choices of structures. These shrubs within a few years with adequate water supply aid the walls to merge with the environment.

4.2.6 Environmental Considerations

The selection of an earth retaining system is influenced by its potential environmental impact during and after construction. If the construction site is in the vicinity of a hospital, library or a laboratory which houses sound sensitive equipment, then earth systems which use pile driving or heavy construction machinery may not be considered. Other important parameters would include the excavation and disposal of contaminated material and the discharge of large quantities of water at the project site. When the problem of noise reduction is the primary concern, then the gravity-type gabion structures/the Evergreen walls are the most suitable types. The open nature of the former and the presence of foilage covering in the latter are effective in absorbing the noise, making these structures acoustically superior to hard and smooth concrete structures.

4.2.7 Durability and Maintenance

The effects of corrosion and weathering have to be taken into account while determining the selection of the appropriate retaining structure.

Concrete retaining structures generally are more durable against weathering and corrosion effects than earth reinforced systems which utilize considerable reinforcing elements. Earth reinforced systems are not preferable if the corrosion in the site is a critical issue. Gabion walls have similar durability problems since the wire metal cages are susceptible to corrosion. When geo-synthetics are used as reinforcement, their long term creep behavior and resistance to deterioration due to chemical attack and exposure to ultraviolet light are major concerns. Exposure of geotextile to ultraviolet radiation may significantly affect the structural strength of the system by way of reduction of tensile strength and elongation to failure of the geosynthetic structural elements. The durability of concrete structures is influenced by the quality of the aggregate and water used in the mix and the casting procedures adopted in the construction of the system. The durability factor is a very important consideration when the retaining structure is to be built in highly corrosive soils (with high acidic content) or non-conventional environments with waves, chemicals or marine borers.

### 4.2.8. Cost

Retaining wall costs are generally expressed per square foot of wall face. The major elements that contribute to the cost are wall materials, erection and select backfill for fill retention systems. The cost of wall materials and erection are relatively predictable for a particular wall type although backfill costs can vary dramatically depending on the quality of the backfill required and its local availability. In general, the construction cost of earth walls in fill areas is less than 60% that of conventional cast-in-place concrete walls. The non-conventional walls become more economical when the wall is more than 10 ft. in height [4.2]. Table 4.1 presents the average costs in the North West United States.

Table 4.1 Average Wall Costs in Northwest U.S.

| Wall Type | Average Cost in Dollars per S.F. Wall Face |
|---|---|
| Reinforced Earth | $18 - $22 |

| Wall Type | Average Cost in Dollars per S.F. Wall Face |
|---|---|
| VSL Retained Earth | $18 - $22 |
| Wire Wall | $13 - $20 |
| Geotextile Walls w/o Permanent Facing | $ 6 - $12 |
| w/3" (76mm) Shotcrete Face | $15 - $25 |
| Tensar Walls w/o Permanent Facing | $15 - $25 |
| w/ Permanent Facing | N/A |
| Metal Bin Wall (H=10' [3m] to 30' [9m]) | $20 - $40 |
| Gabion Wall (H=6' [1.8m] to 21' [6.4m]) | $18 - $35 |
| Doublewall | $20 - $25 |
| Criblock | $12 - $25 |
| Permanent Tieback Walls | $45 - $60 |
| Soil Nailed Walls (U.S. Experience is Limited) | |
| Temporary | $17 - $25 |
| Permanent (Shotcrete Face) | $25 - $30 |
| Permanent (Architectural Facing) | $35 - $50 |
| Cantilever Soldier Pile w/Wood Lagging (H=5' [1.5m] to 15' [4.6m]) | $12 - $18 |
| Cantilever Sheet Pile Wall (H=5' (1.5m) to 15' [4.6m]) | $12 - $18 |
| Concrete Cantilever (H=6' [1.8m] to 30' [9.1m]) | $25 - $40 |

NOTES:

1. Average prices given above are <u>total</u> in-place costs including wall materials, erection, and select wall backfill. For mechanically stabilized earth walls, the select backfill is the backfill contained within the reinforced volume. Since excavation quantities can vary greatly job-to-job, excavation cost is <u>not</u> included.

2. Unless otherwise noted, the prices given cover walls in the 10'[3m] to 40' [12m] height range.

3. Alaska prices are higher.

4. Not all systems listed above are marketed in all states.

5. N/A denotes no cost experience to date in Region 10 [Northwest U.S.].

Estimation of the design life and a long term cost is more difficult for wall systems constructed with materials other than concrete, steel and wood. New wall materials such as plastics, fiber glass and other geosynthetics require in-depth study of available performance history, current technical guidance can be obtained by contacting the relevant technical groups such as the AASHTO sub-committee on new materials for the latest technical information and specifications [2.7].

4.2.9. Risk

An assessment of risk and safety aspects in both design and construction including impact on adjacent facilities is necessary prior to the wall type selection. Particular wall types are better suited to certain sub-surface conditions than others. An adequate sub-surface investigation must be completed at the project site to assess the applicability of individual systems. The owner must establish formal guidelines for the design of alternate walls to mitigate risk and to equate the design of different alternate wall systems. The owner should be responsible for long term stability of the ground surrounding the retaining structure. In recent years several spectacular failures have occurred involving alternate earth retaining systems. Subsequent analysis has shown that latent subsurface conditions, not involved with internal wall stability, caused the failures. Fig. 4.1 shows the failure which occurred on a highway project near Coos Bay, Oregon, involving a reinforced earth wall. The wall top moved out 18 feet and dropped 12 feet, while the wall base translated 23 feet. Investigations disclosed that the failure surface was along a thin inclined seam of soft clay in the natural deposits below the wall base. The owner/designer should establish design criteria such as minimum safety factors for overturning and sliding, design life, maximum allowable bearing pressure, embedment depth, earth pressure magnitude and distribution. A comprehensive technical review must be done for the proposed system in addition to comparisons of risk elements.

#### 4.2.10 Politics and Tradition

Political influences, trade practices and national policies affect the selection of structures. The selection also depends on the traditional practices in the region where the structure is to be constructed since the contractor in that region will be equipped to carry out the construction for the specific wall type under consideration. Moreover, construction familiarity and specialization on the designs of certain types of structures lead the contractor to select that particular structure type.

### 4.3. FORMULATION OF THE KNOWLEDGE BASE FOR RETAINING STRUCTURE SELECTION

#### 4.3.1 Introduction

Section 4.2 describes the factors affecting the selection of the most

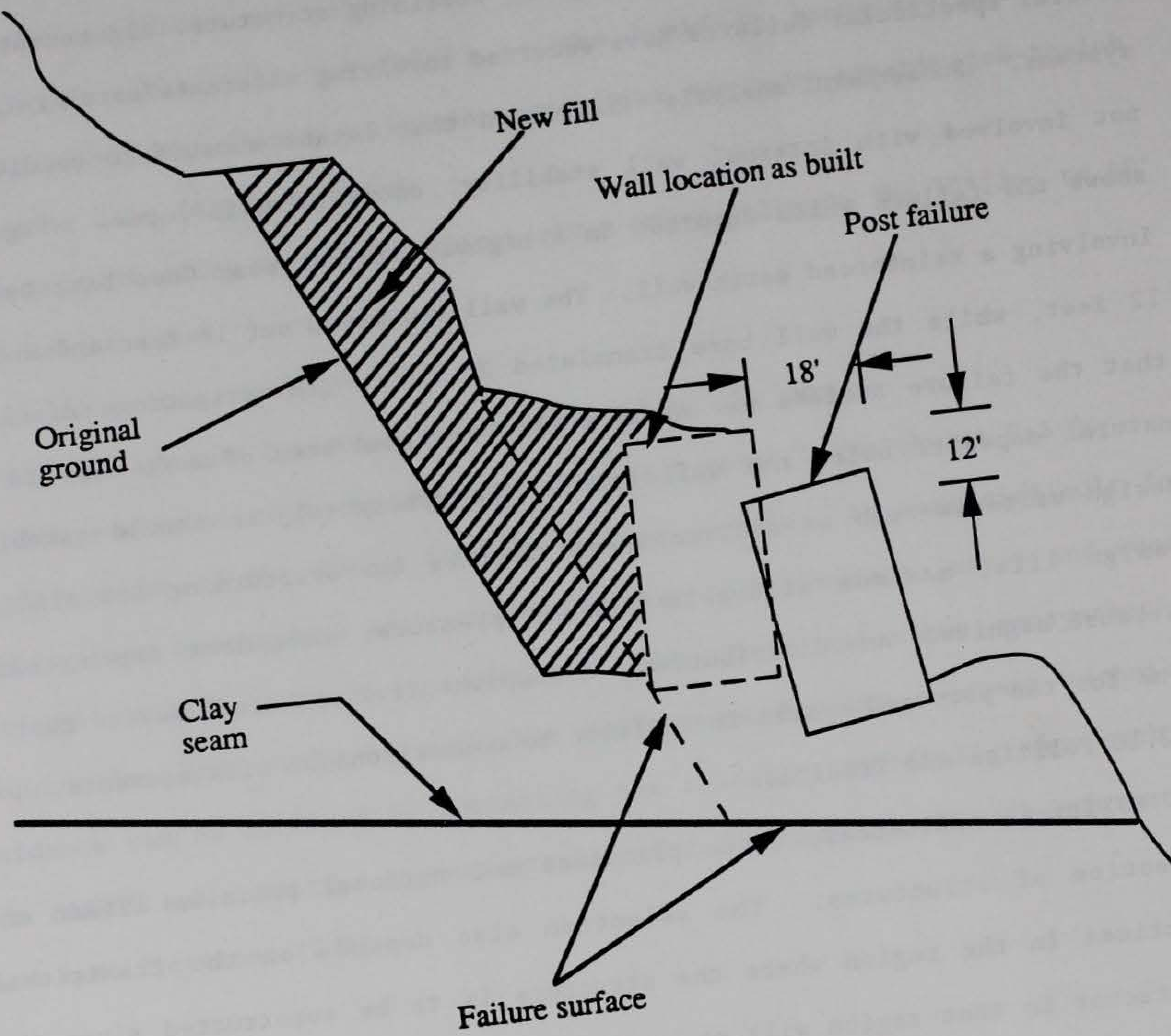



Figure 4.1 Coos Bay Failure [2.5]

appropriate retaining structure. The following section describes the methodology of the formulation of this knowledge into the system using the M.1 knowledge engineering shell. M.1 is a powerful PC- based knowledge system which is implemented in the C programming language. The knowledge base is prepared using any standard text editor. The retaining structure selection module, SELECTWALL is developed in M.1 syntax using Wordstar. The M.1 inference enginer primarily uses a backward-chaining reasoning process to reach conclusions, although it also uses the forward-chaining reasoning.

### 4.3.2 Methodology of Knowledge Formulation in SELECTWALL

The selection module consists of three phases. In the first phase of the selection process the system objective is to eliminate a few of the retaining structures and present the user with the possible structure types which are most suitable for the existing conditions. The second phase investigates the selected structures in a greater detail and the objective of this phase is to indicate the most appropriate structure from among the selected possible types. The last phase confirms this selection through queries on aesthetics and familiarity of construction of the selected structure. A schematic representation of the selection module is shown in Fig. 4.2.

Site location usually dictates the basic wall type and size. This portion of the knowledge is implemented in M.1 as shown in Fig. 4.3.

```
multivalued(location).

question(location) =

 'Which of the following list of sites best describes the
location of the proposed retaining structure?'.
legalvals(location) =
     [excavation,roadway,sidehill,abutment,forest-area,buiding-
related-excavation, mountainous-terrain,waterfront-area,railway,
building,marine].
automaticmenu(location).
enumeratedanswers(location).
```

Fig. 4.3. M.1 code for Site Location

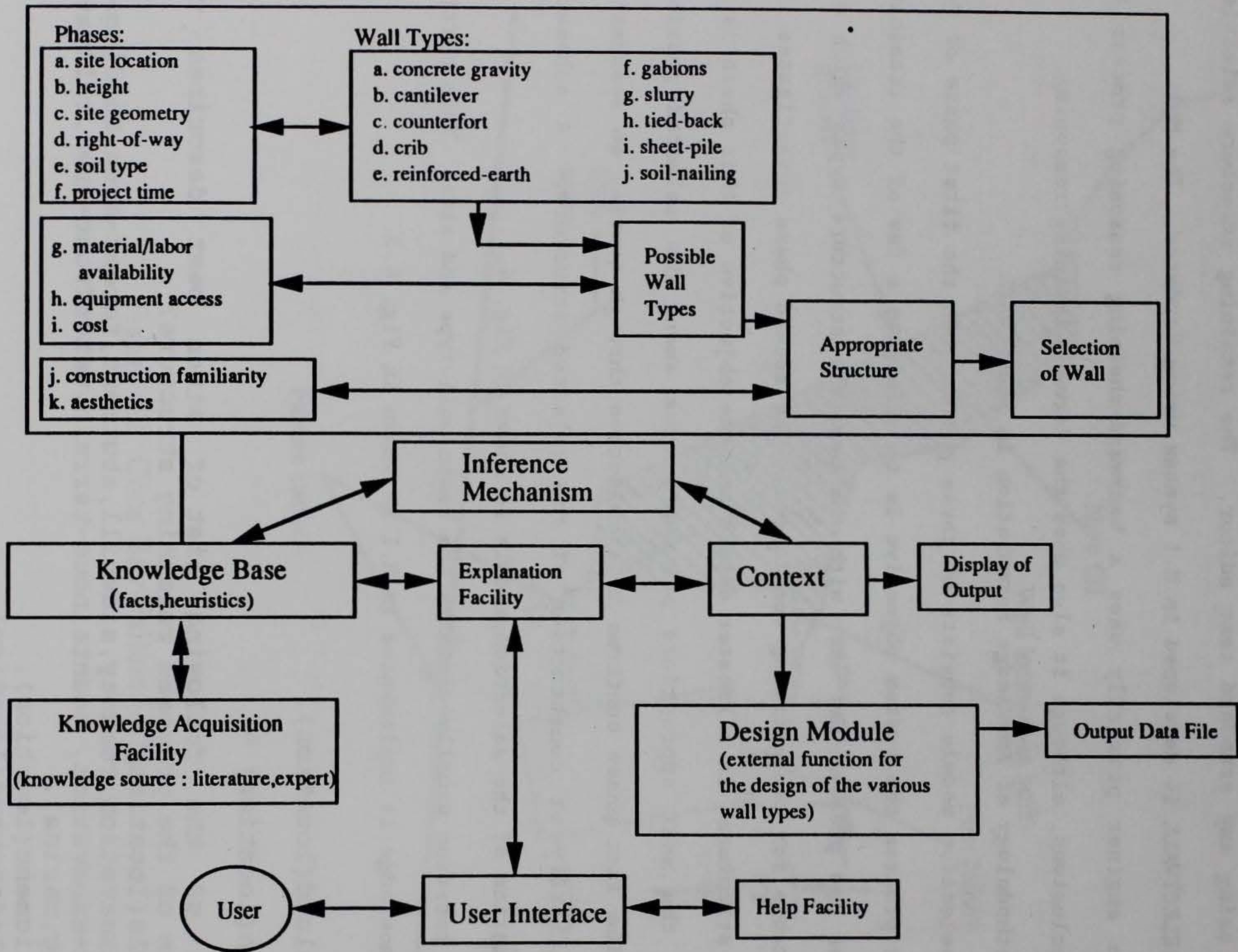



Figure 4.2 Basic Architecture of RETAININGEARTH

The features 'automaticmenu' and 'enumeratedanswers' enable the system to display an enumerated list of sites to the user. The user may input his choice by typing in either the number or sufficient alphabets to distinguish the response from the other sites displayed. The user could also seek an explanation to the question by typing 'why'. The explanation to question (location) is formulated in M.1. as shown below:

```
explanation(location) = [
        'The location where the structure is to be built is
a very important consideration in the selection of retaining
walls. You may choose (only one) location which is closest to the
above locations.'].
```

The next question queries the user whether the site is situated in a comparatively inaccessible region (such as mountainous terrain). If the region has a difficult topography and potential access problems, it is favorable to adopt a top-down staged construction like the soil nailing, tied-back wall or reinforced earth structure. The consultation continues and the user is asked whether the construction of retaining structure is for an emergency situation (e.g., landslide of mountainside blocking the traffic). Under these circumstances the reinforced earth structure and the tied back wall are favorable choices. In most cases the reinforced earth structure has been found to be economical when the soil conditions are not favorable for drilling for installation of tiebacks. The next two questions involve the important parameters, height and length of the earth to be retained which significantly control the cost of the structure. The wall height is an important variable for selecting the wall type since some structures have proven to be cost effective and structurally more compatible for a given height. The next few questions deal with the site geometry, right-of-way, soil type and project completion time. Site conditions usually dictate the basic wall type and size. The retaining structures are categorized broadly into two types (fills and cuts) as shown in Table 2.1. Right-of-way is also an important consideration in the selection of a structure especially in the case of systems which involve reinforcing elements (reinforced earth, tied back wall and soil nailing). Thus these structures cannot be considered if there is limited right-of-way at the site. The soil is also an important factor which influences the selection of a structure. The user is asked several questions related to the soil properties at the possible site. An example of a typical question in shown in Fig. 4.4.

```
question(soilb) =
     'Does the soil have the following properties:
1.Soils of high plasticity.
2.Granular soils with no coherance.
3.Soils with much water content.
4.Large size(boulders).-------(yes/no)?'.


legalvals(soilb) = [ yes,no ].


explanation(soilb) = [nl,
     'The data entered by you may lead to the selection of  tied-
back retaining structure,  subject to favorable soil condition.A
favorable soil for this type of retaining structure will NOT have
the properties stated in this question.',nl].
```

Fig. 4.4 M.1 Code for Soil Properties

The first ten rules in the ES lead to the selection of the 'possible-wall' types that are applicable for the given conditions. Each rule consists of all the aforementioned conditions. If these conditions are satisfied, the rule is activated or 'fired' and the particular 'possible-type' selected for consideration in the next phase. A typical example of a rule which leads to the possible-type structure selection is shown in Fig. 4.5.

```
rule-15: if site = fill and
            not(location = excavation) or
            not(location = building-related-excavation) or
            not(location = building) and
            height = H and
            (H > 6 and H < 34) and
            length = L and
            (L > 0) and
            rway = yes or
            rway = no and
            not(time = short) and
            soila = yes and
            soile = no and
            foundation = good or
            foundation = poor and
            vertload = yes or
            vertload = no

  then possible-wall = gabion-retaining-wall.
```

Fig. 4.5. Typical M.1 Rule for Possible Type Structure Selection

The rule-15 is activated and the gabion-retaining-wall is chosen as the 'possible-wall' if the following conditions in the above rule are satisfied:

i) if the site geometry belongs to the 'fill' category,

ii) if the construction of the structure does not involve an emergency situation;

iii) if the project location is not an excavation or building-related-excavation or building;

iv) if the site is accessible/inaccessible (loc 2) (this rule has both yes and no for this factor because the ES is framed in such a manner that all 'possible types' of walls are to be considered in the first phase of the consultation). The gabion structure is to be considered as a 'possible-type' even if the site is accessible, although it is suitable even when the site is inaccessible;

v) if the site involves a situation where traffic has to be maintained during construction;

vi) if the height of the earth to be retained is > 10 and ≤ 34 ft.;

vii) if the length of the earth to be retained is >0;

viii) if the right-of-way condition of the site is favorable or unfavorable;

ix) if the duration of the time of the project is not short;

x) if the soil is either firm or loose at the proposed site;

xi) if the in-situ soil does not have a high acidic content;

xii) if the foundation is good or poor;

and

xiii) if the proposed structure will be required to take vertical load/surcharge.

If the answers to all the questions related to the statements in rule-15 are favorable, then this rule is activated and the gabion structure is selected as a possible-wall type. The first phase may result in more than one possible-wall type depending upon the response of the user. Once the consultation in the first phase is over, the ES checks each rule to see if the conditions in the rule are satisfied. If so, the rule is then activated and the structure selected as a possible-type. Thus at the end of the first phase, the ES has one or a list of possible-wall types to be considered for selection in the next phase of the consultation.

A schematic procedure for the selection of the possible-type structures in the first phase of the selection module is shown in Fig. 4.6. If the list of possible-wall types selected consists only of one structure type, then the information related to the material required for the construction of this wall-type and the labor availability are asked by the system in the second phase of the selection process. This information is shown below (Fig. 4.7) in rule - 24, where gravity-retaining-wall is the possible type as recommended in the first phase; information regarding material and labor availability are sought by the system to justify its selection as the appropriate-structure.

```
rule-24: if possible-wall is sought and
            user-informed = yes and
            possible-wall = gravity-retaining-wall and
            materialcg = yes and
            labora = yes and
            listof(possible-wall) = [ONE]
         then appropriate-structure = gravity-retaining-wall.
```

Fig. 4.7 Typical M.1 Rule for Appropriate Wall Selection in the Second Phase

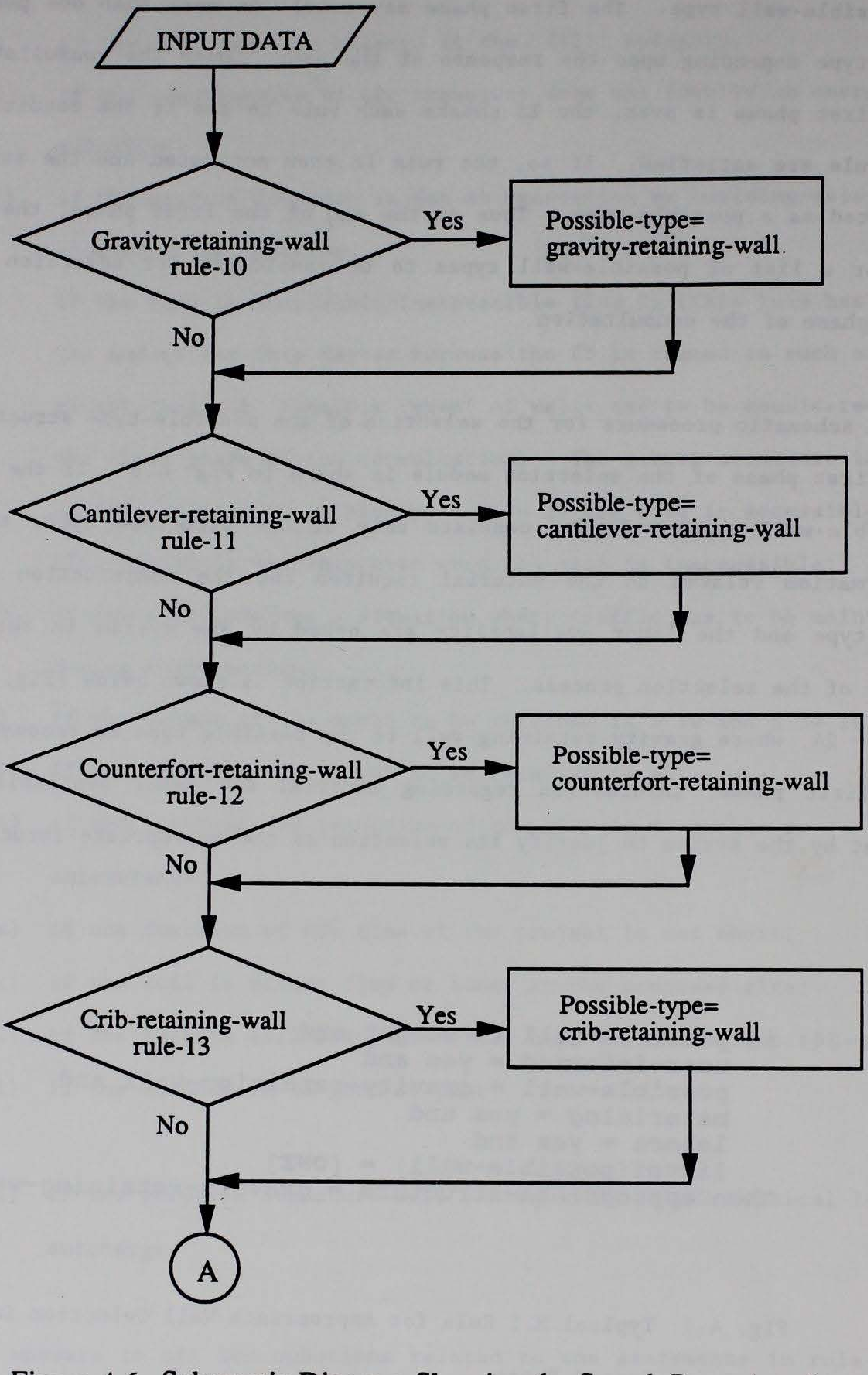



Figure 4.6 Schematic Diagram Showing the Search Procedure for the ' Possible-wall ' Types in Phase 1. ( continued )

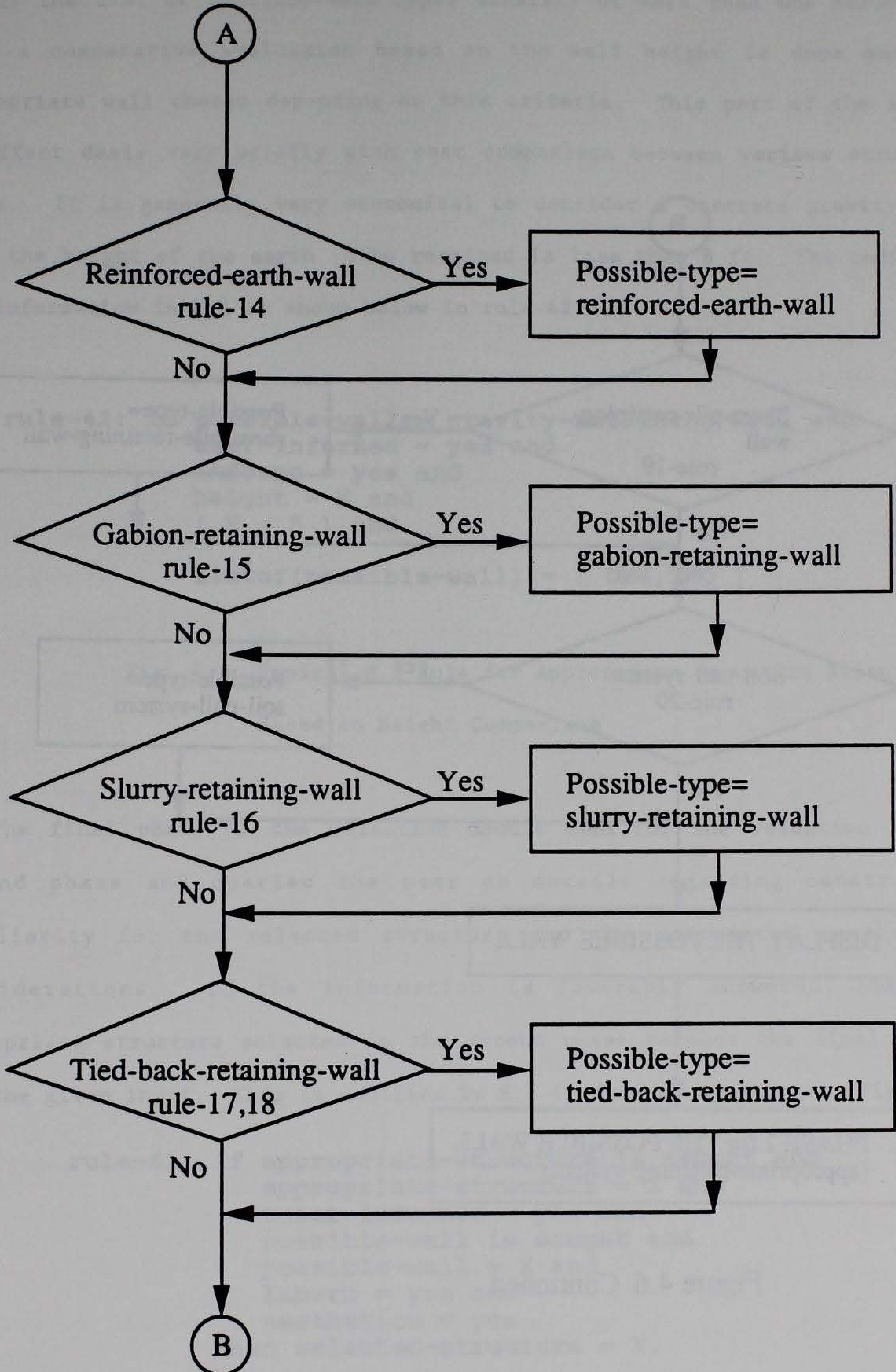



Figure 4.6 Continued

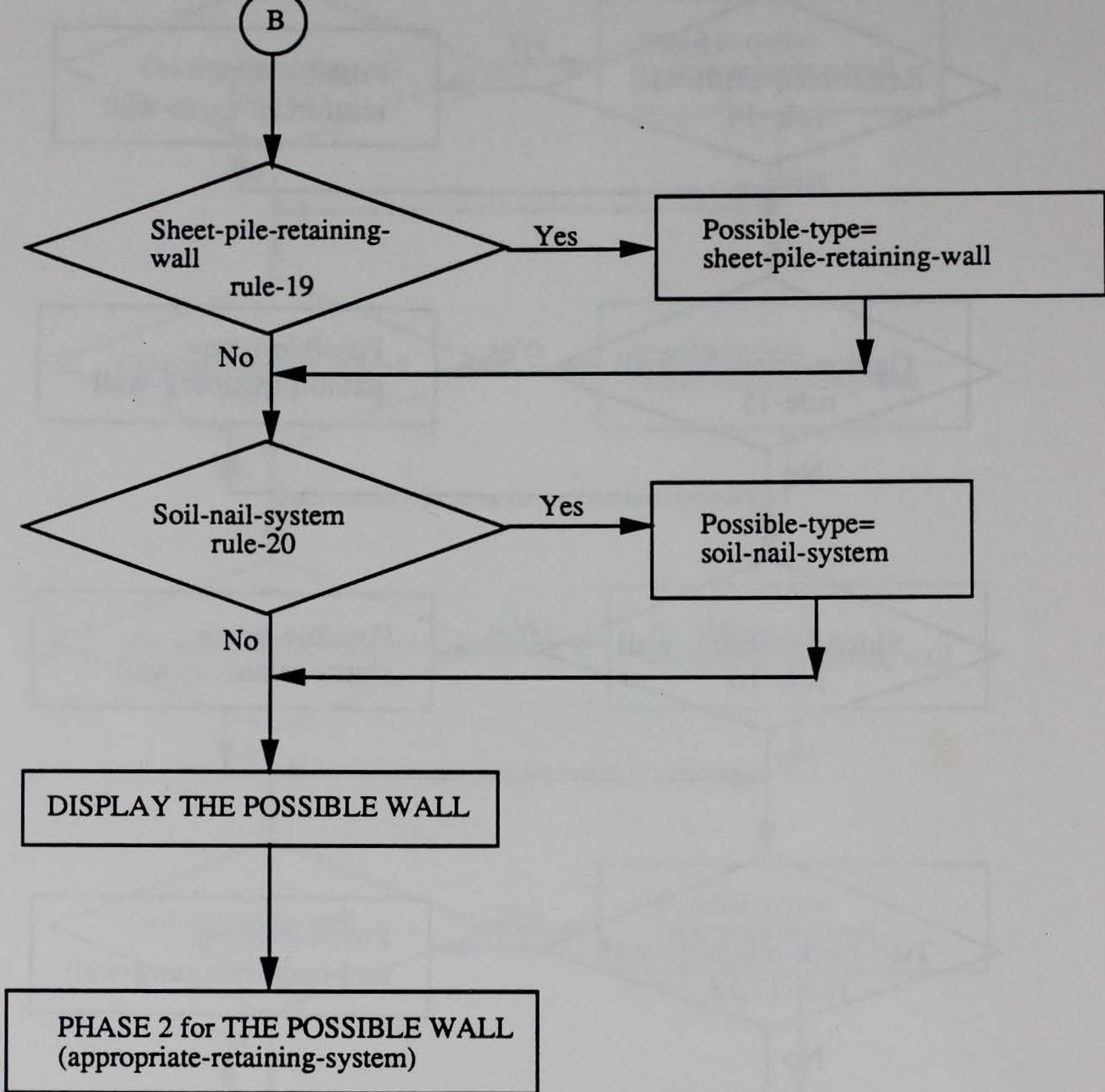



Figure 4.6 Continued

If the list of possible-wall types consists of more than one structure, then a comparative evaluation based on the wall height is done and the appropriate wall chosen depending on this criteria. This part of the system in effect deals very briefly with cost comparison between various structure types. It is generally very economical to consider a concrete gravity wall when the height of the earth to be retained is less than 8 ft. The coding of the information in M.1 is shown below in rule-42.

```
rule-42: if possible-wall = gravity-retaining-wall and
            user-informed = yes and
            informg = yes and
            height = H and
            ( H < 5 ) and

            listof(possible-wall) = [ ONE,TWO ]
```

Fig. 4.8 Typical M.1 Rule for Appropriate Structure Selection
Based on Height Comparison

The final phase of the selection module confirms the selection in the second phase and queries the user on details regarding construction familiarity for the selected structure and the associated aesthetical considerations. If the information is favorably answered, then the appropriate structure selected in the second phase becomes the final choice for the given input. This is codified in M.1 in rule-63 as shown in Fig. 4.9.

```
rule-63: if appropriate-structure is sought and
            appropriate-structure = X and
            tuser-informed = yes and
            possible-wall is sought and
            possible-wall = X and
            laborb = yes and
            aesthetics = yes
         then selected-structure = X.
```

Fig. 4.9. Typical M.1 Rule for Final Structure Choice

After the final structure choice the system asks the user if he is satisfied with the selection. If the user is satisfied the consultation would end, or else the list of structures available on the system is displayed to the user and the user makes his choice. This provision enables the system to be more flexible and allows the user to choose any desired structure. Once this is done, the consultation in M.1 ends and the system is linked to the design module where the design of the particular selected/chosen structure is then carried out. The details of the external code linkage between the selection and design modules are presented in Chapter 5. The complete listing of the M.1 code SELECTWALL for the selection process is given in Appendix A.

**REFERENCES**


4.1 GA Munfakh, Innovative Earth Retaining Structures: Selection, Design & Performance, GSP 25, Ed PC Lambe & LA Hansen, ASCE, New York, 1990, 52-65.

4.2 AR Schnore, Selecting Retaining Wall Type and Specifying Proprietary Retaining Walls in NYSDOT Practice, GSP 25, Ed PC Lambe & LA Hansen, ASCE, New York, 1990, 119-124.

4.3 M.1 Reference Manual for Software, Version 2.1, Teknowledge, Palo Alto, CA, Dec. 1988.

4.4 H Schnabel, A Contractor's Perspective on Wall Selection and Performance Monitoring, GSP 25, Ed. PC Lambe & LA Hansen, ASCE, New York, 1990, 67-84.

4.5 IK Lee, W White and OG Ingles, Retaining Structures, Geotechnical Engineering, Pitman, 1983, 245-278.

4.6 JE Bowles, Foundation Analysis and Design, 4th Ed., McGraw Hill Book Co., 1988, 530-642.

CHAPTER 5

# PROCEDURE FOR LINKING THE SELECTION AND DESIGN MODULES

## 5.1 INTRODUCTION

The expert system RETAININGEARTH consists of the selection and design modules. This chapter deals with the formulation of the control program which permits the M.1 selection module to link with the various design procedures in the design module, which enables the computation of the detailed design of the selected structure. Fig. 5.1 gives the general flowchart which shows the linkage of the five independent design programs in the design module with the M.1 selection module through a control program. The following section describes the linking procedure using the control program.

## 5.2 LINKING PROCEDURE USING THE CONTROL PROGRAM

The control program CALL is developed in C and employs the menu-driven technique which makes the system-user interface very effective and easy to follow. The photograph of the screen showing the various menus, shown in Figure 5.2, appears at the start of the consultation with ES RETAININGEARTH. After the consultation with M.1, the program control returns to CALL, along with the height and type of the selected structure. The import of the two parameters, namely the wall height and type, from the M.1 selection module is done by the external code EXT. EXT is developed in C conforming to the required M.1 format. Fig. 5.3 shows the external code EXT. Some of the rules in the knowledge base in the selection module have to be prepared to permit

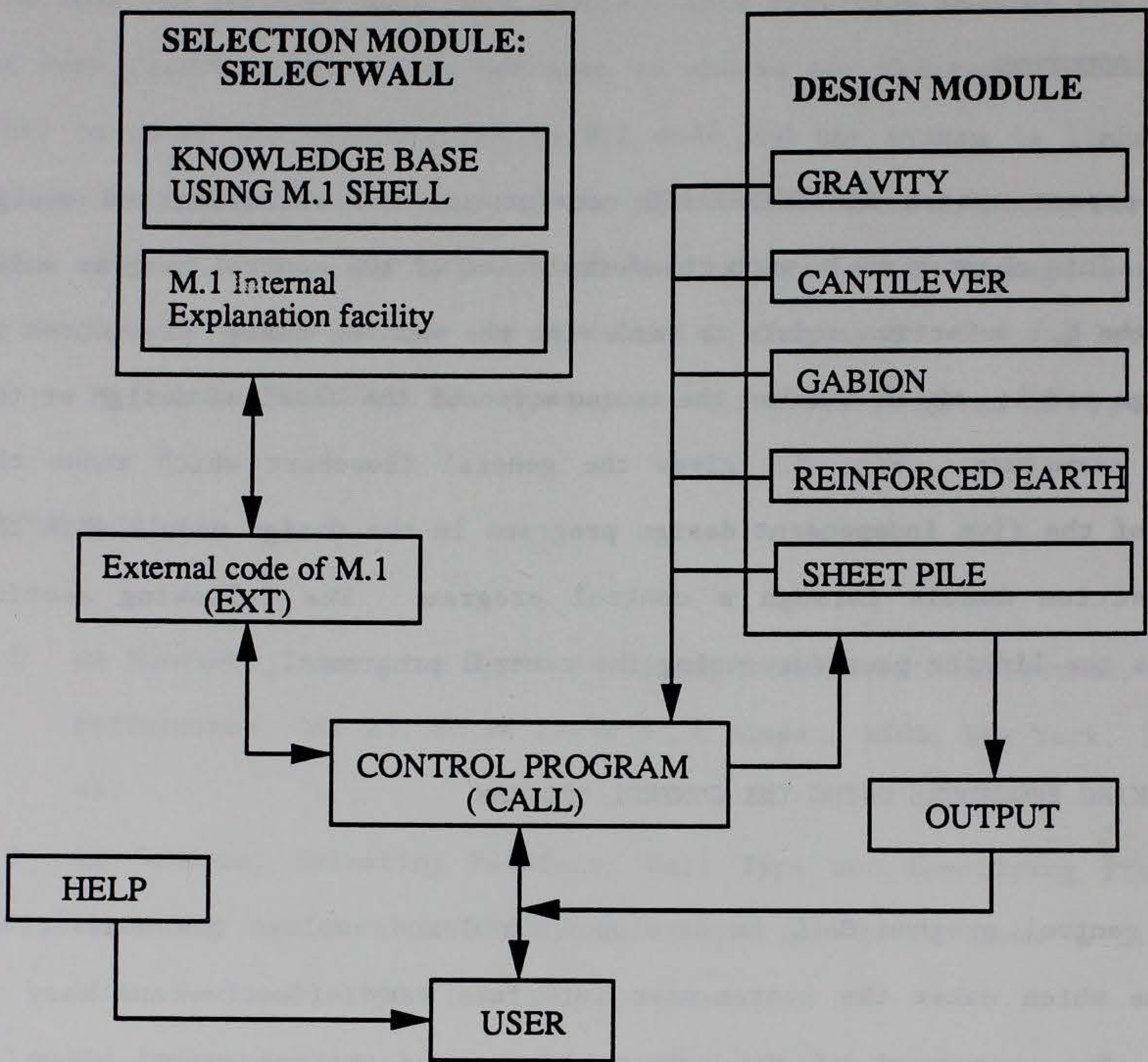



Figure 5.1 Flowchart of ES RETAININGEARTH, Showing Linking of the M.1 Selection Module with the Design Module.

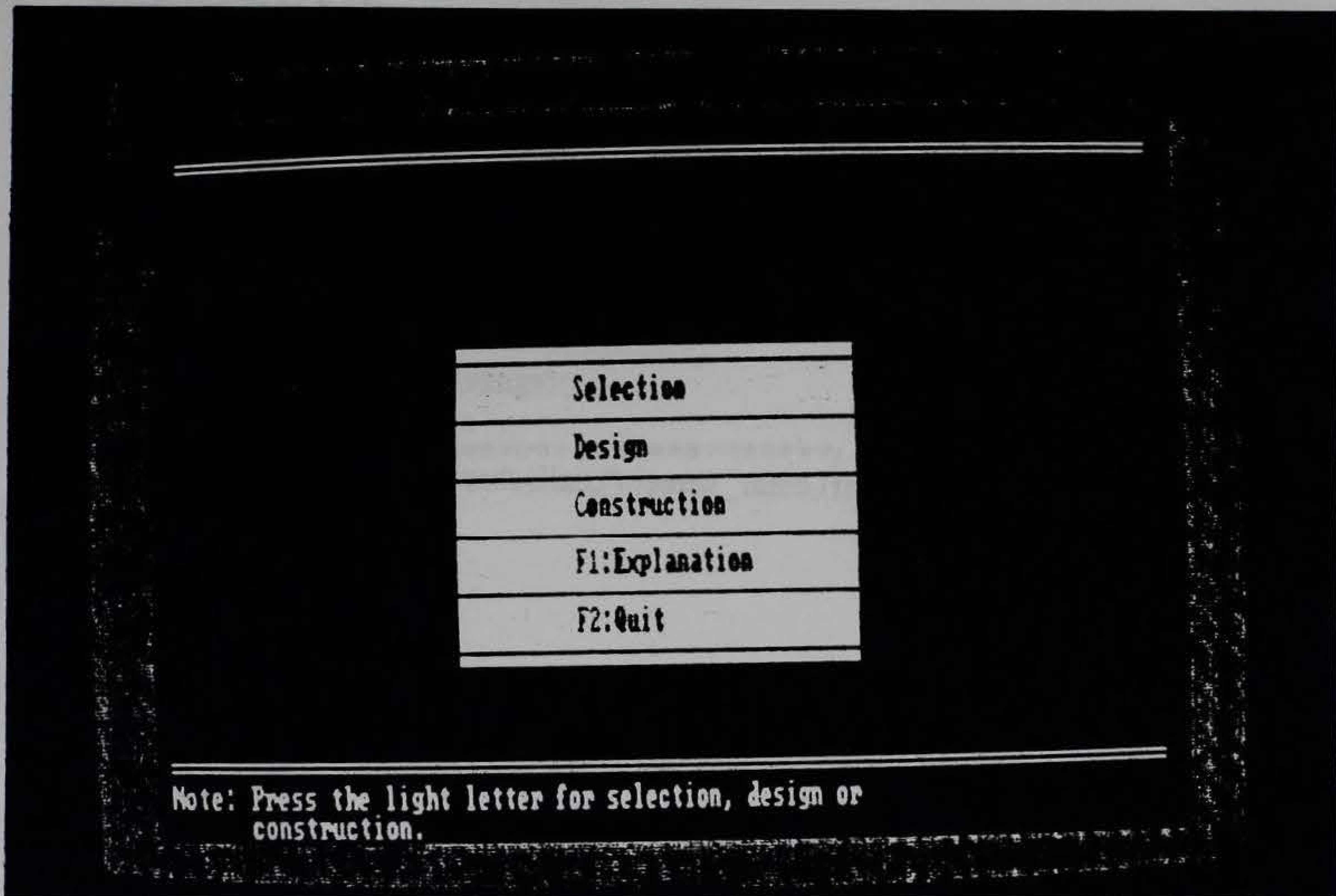



Figure 5.2 Photograph Showing the Various Menus at Start of RETAININGEARTH.

```
/* ---------------------------------------- */
/*    M.1 EXTERNAL CODE    */
/* ---------------------------------------- */

#include <stdio.h>
#include <signal.h>
#include <math.h>
#include <extif.h>
#include <ctype.h>
#include <string.h>
#include <graph.h>
#include <dos.h>
#include <stdlib.h>
#include <process.h>
char buf[4000];
extInit()
{
  extern int addFuction();
  extern int design();
  addFunction("design",1,design);
}
FILE *fp,*fopen();

extern int import();
extern int export();
extern int restoreScreen();
int get_mode();

void setscreen(int),set_page(int),clear_screen(),call(char *);
int page,color,screen;


design()
{
int ree;
float h;
long int type,y;
long int choice;
int oldmode;
saveScreen(buf);
oldmode=get_mode();
setscreen(3);
clear_screen();
_setvideomode(_DEFAULTMODE);
set_page(0);
import(1,FLOAT,&h,0);
import(2,LONG,&y,0);
choice=y;
export(1,FLOAT,&h);
fp=fopen("heitype","w");
fprintf(fp,"%.2f %d\n",h,y);
fclose(fp);
printf("\n\n The type of wall you have selected will be conveyed to design\n");
```

Figure 5.3 External Code EXT

```
printf(" part. Please go to the main menu if you want to design it\n");
printf("\n\n But first, please press any key to go back to M.1 \n");
while(kbhit()==0) {;}
setscreen(oldmode);
restoreScreen(buf);
return;
}
extExit()
{
}
/*******************************************/
int get_mode()
{
union REGS intregs,outregs;
intregs.h.ah=0x0f;
int86(0x10,&intregs,&outregs);
return(outregs.h.al);
}
/*******************************************/
void setscreen(n)
int n;
{
union REGS intregs,outregs;
screen=n;
intregs.h.ah=0;
intregs.h.al=n;
int86(0x10,&intregs,&outregs);
}
/*****************************************/


void set_page(p)
int p;
{
union REGS intregs,outregs;
extern int page;
page=p;
intregs.h.ah=5;
intregs.h.al=p;
int86(0x10,&intregs,&outregs);
}
/*****************************************/

void set_color(f,b)
int f,b;
{
color=(f & 143)+((b<<4) & 112);
}
/*****************************************/

void clear_screen()
{
union REGS inregs,outregs;
inregs.h.ah=6;
```

Figure 5.3 Continued

```
inregs.h.al=0;
inregs.h.bh=7;
inregs.h.ch=0;
inregs.h.cl=0;
inregs.h.dh=24;
inregs.h.dl=79;
int86(0x10,&inregs,&outregs);
}
/****************************************/
void call(s)
char *s;
{
char far *filename=s;
union REGS inregs,outregs;
struct SREGS insreg;
inregs.h.ah=0x4B;
inregs.h.al=0;
inregs.x.dx=FP_OFF(filename);
insreg.ds=FP_SEG(filename);
int86x(0x21,&inregs,&outregs,&insreg);
}


/**********************************************************************/
```

Figure 5.3 Continued

```
rule-64: if selected-structure is sought and
            selected-structure = X and
            listof(selected-structure) = [ONE] and
            display([nl,'The  selected  structure for  the  given condit
            continue and
            user-happy and
            display([nl,'I''m glad you are happy with my selection.',nl]
            list=A               and
            display(A)           and
            height=H             and
            choice1=Y            and
            design(H,Y)=DESIGN
            then end the consultation.
list=[
nl,
nl,'         1. Gravity;',
nl,'         2. Cantilever;',
nl,'         3. Counterfort;',
nl,'         4. Gabions;',
nl,'         5. Slurry;',
nl,'         6. Tied-back;',
nl,'         7. Reinforced-earth;',
nl,'         8. Sheet-piles;',
nl,'         9. Crib-wall;',
nl,'         10.Soil-nailing.',
nl,
nl].
nocache(list).
question(choice1)=[nl,' Please confirm the number of the wall which you

/*
multivalued(fault).
automaticmenu(fault).
enumeratedanswers(fault).
*/
question(fault) =
   ['Which wall do you think is best for these conditions?'].
/*
 legalvals(fault) =
[gravity,cantilever,counterfort,gabions,slurry,tied-
back,reinforced-earth,sheet-piles,crib-wall,soil-nailing].
*/
legalvals(fault)=integer.
rule-65: if user-informed and
            possible-wall is sought and
            user-informed and
            appropriate-structure is sought and
            tuser-informed and
            selected-structure is sought and
            not(user-happy) and
```

Figure 5.4 Example of Rules in M.1 for Export of Information

```
          list=F          and
          display(F)      and
          fault = Y               and
          height=H                and
          design(H,Y)=DESIGN
          then end the consultation.

/* ----------------------------------------------------------- */

rule-66: if external(design,[H,Y])=[DESIGN] and
          display(['The consultation is over. Press Alt + Q. Thank you
       then design(H,Y)=DESIGN.
/*************************************************************/
legalvals(choice1)=integer(1,10).
```

Fig. 5.4 Continued

```
int show(sss)
char *sss;
{
int c;
clear_screen();
get_lines(sss);
print_first_page();
locate(0,0);
while(kbhit()==0) {;}
c=getkey();
while(c!=ESCKEY && c!=F2KEY)
  {
  switch(c)
   {
   case UPARROW:
   scroll_page_down(1);
   locate(0,0);
   while(kbhit()==0) {;}
   c=getkey();
   break;
   case DOWNARROW:
   scroll_page_up(1);
   locate(0,0);
   while(kbhit()==0) {;}
   c=getkey();
   break;
   case PGUP:
   scroll_page_down(19);
   locate(0,0);
   while(kbhit()==0) {;}
   c=getkey();
   break;
   case PGDN:
   scroll_page_up(19);
   locate(0,0);
   while(kbhit()==0) {;}
   c=getkey();
   break;
   default:
   locate(0,0);
   while(kbhit()==0) {;}
   c=getkey();
   break;
   }
  }
return(c);
}
```

Figure 5.5 Function 'show' in Control Program CALL the EXT program to export information from the selection module SELECTWALL to the control program CALL. The representation of this information in M.1 is shown by the rules in Fig. 5.4. The external code EXT opens a file called 'heitype' and transports the wall height and the corresponding numbers representing the wall into 'heitype'. When the consultation in the M.1 selection module is over the user exits the module and the main menu is displayed to the user. If the user desires the design of the selected structure, he enters the appropriate design program through the control program CALL. CALL opens the file 'heitype' and sequentially searches the various programs in the design module for the corresponding selected wall type, and then interacts with the user in the consultation for the detailed design of the structure.

The other option available in the menu to the user is the help facility. If the user desires an explanation about the system, he could choose the 'help' menu. The help facility gives the user a brief description about the system and its operational features. This facility is made possible by the function 'show' residing in the CALL program shown in Fig. 5.5. The advantage of this facility is that the explanation resides in an independent file and hence does not occupy useful space in the design module. The listing of the control program CALL is given in Appendix B.

### 5.3 Overview of Program Operation

1. Make directory called WALL in root directory as follows:

   C> **mkdir wall** [enter]

2. C> **cd wall** [enter]

3. Copy all files from both the M.1 diskettes to the directory WALL:

   C> copy a:*.* [enter]

4. Copy the following files into the same directory WALL:

   (i) ml.c (ii) gravity.exe (iii) canwall.exe (iv) gabion.exe (v)earwall.exe (vi) sheet.exe

   (use same command as in 4).

5. Start the consultation with **RETAININGEARTH** as follows:

   C> ml ml.c [enter]

   Press [F9] key.

   Enter go to begin the consultation.

**REFERENCES**


5.1 M.1 Reference Manual for Software, Version 2.1, Teknowledge, Palo Alto, CA, Dec. 1988.

5.2 Microsoft C 5.1, Optimizing Compiler, User's Guide Language Reference, 1987.

# CHAPTER 6

# CONCRETE GRAVITY RETAINING STRUCTURES

## 6.1 INTRODUCTION

Retaining structures are generally used to hold back the earth and maintain a difference in the elevation of the ground surface as shown in Fig. 6.1. The retained material exerts a push on a structure and tends to overturn or slide it or both. In highway construction, the retaining structures are used along cuts or fills where space is inadequate for the appropriate side slopes. Bridge abutments and foundation walls which must support earth fills are also designed as retaining walls. A gravity wall, which is usually of plain concrete, depends entirely on its weight for stability. It is used for walls upto about 10 ft. high. The cantilever retaining structure is commonly used for walls in the range of 10 to 25 ft. in height. The stem, heel and toe are the structural components and each acts as a cantilever beam. Counterfort walls are often economical for heights over about 25 ft; the stem and the slab are tied together by counterforts spaced at intervals which act as tension ties to support the stem wall. A buttress wall, which is similar to a counterfort wall, has transverse support walls located on the side of the stem opposite to the backfill; these support walls act as compression struts. The counterfort is more popular than a buttress because the counterfort is hidden beneath the backfill whereas the buttress occupies valuable space in front of the wall. The bridge abutment acts in the same way as the cantilever retaining wall except that the bridge deck provides an additonal horizontal restraint at the top of the stem.

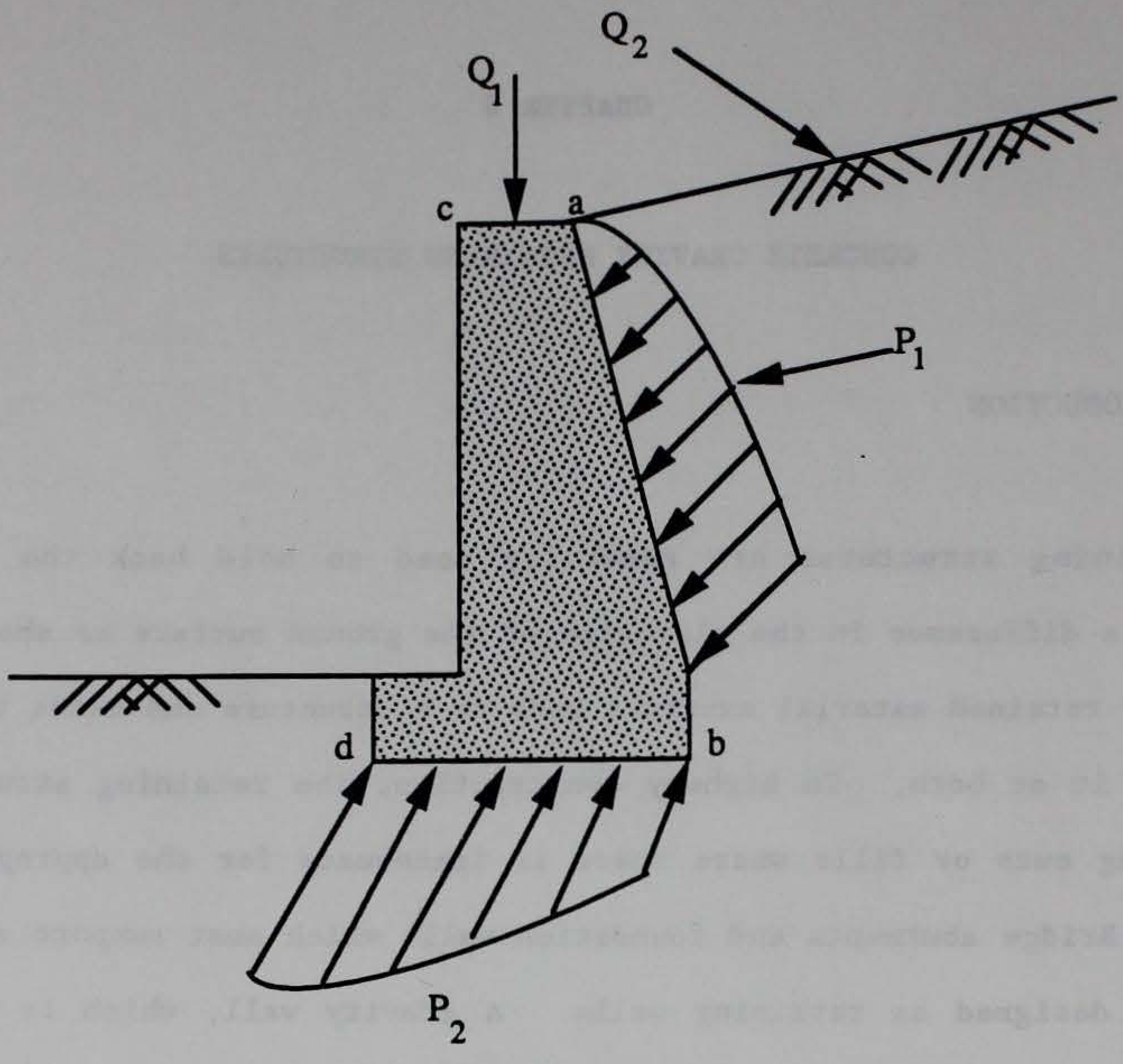



Figure 6.1 Loads on Retaining Walls [6.3]

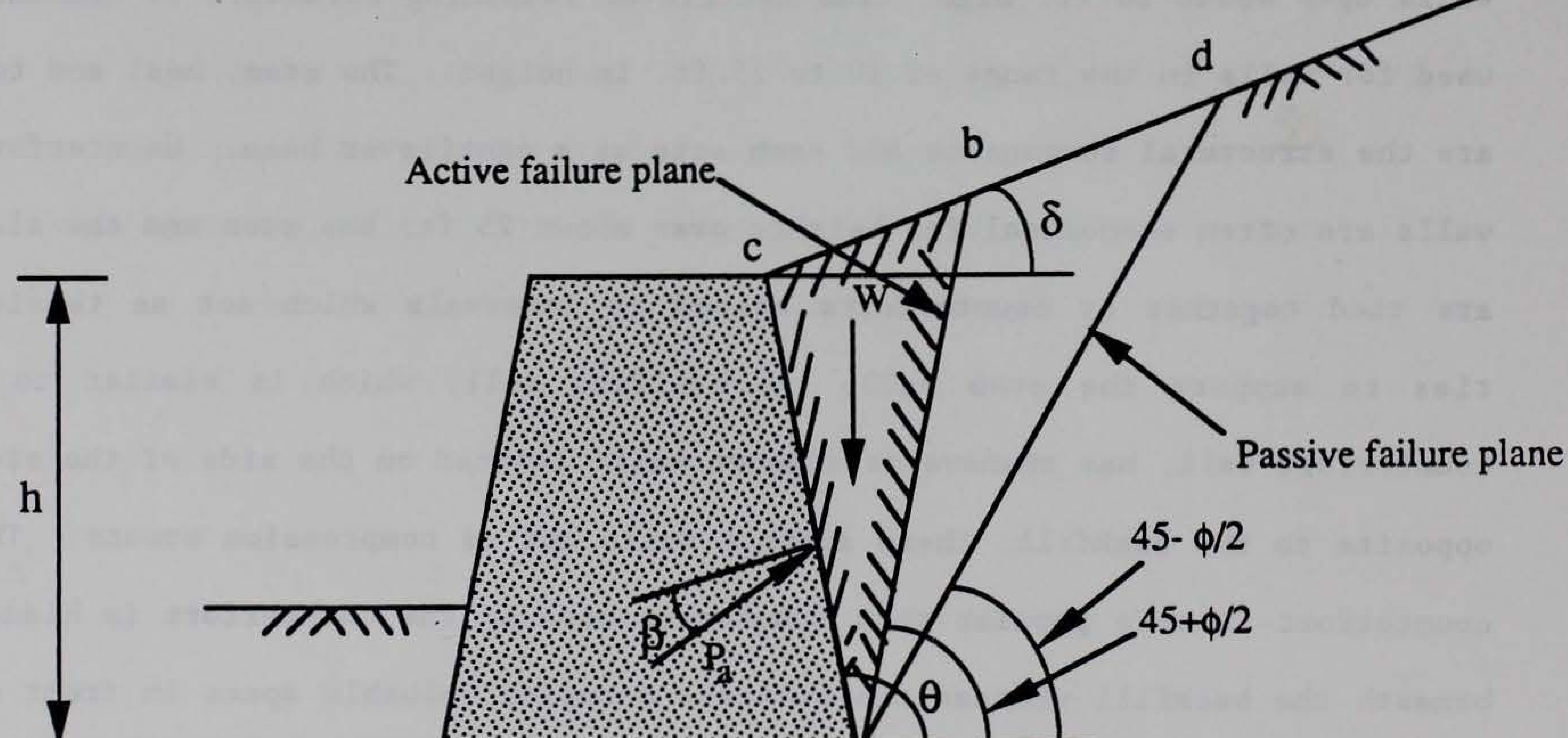



Figure 6.2 Basis of Active and Passive Earth Pressure Computations

The development of an integrated interactive microcomputer based analysis and design has been confined to typical concrete gravity and cantilever retaining walls.

## 6.2. DESIGN PRINCIPLES

### 6.2.1. Earth Pressure

The earth pressure exerted by the backfill on the retaining wall increases proportionally to the depth and its magnitude is expressed as

$$p_h = k_o \, wh \tag{6.1}$$

where

$k_o$ = a constant known as the coefficient of earth pressure at rest

$w$ = unit weight of the soil

and

$h$ = distance from the surface.

The value of $k_o$ is dependent on the nature of the backfill, the method of deposition and compaction. It ranges between 0.4 and 0.5 for uncompacted non-cohesive soils such as sands and gravels whereas the value could be as high as 0.8 for the same soil in a highly compacted state. $K_o$ could be of the order of 0.7 to 1.0 for cohesive soils.

The retaining walls which are constructed of elastic material deflect under the action of the pressure and since they rest on compressible soils they tilt and shift away from the fill. A sliding plane ab forms in the soil mass (Fig. 6.2) when the wall moves away from the fill. The wedge abc sliding along that plane exerts pressure against the wall. The angle $\phi$ whose tangent is equal to the coefficient of intergranular friction is known as the **angle of internal friction** and the corresponding pressure is known as the **active earth pressure**. If the wall is pushed against the fill, a sliding plane ad is formed and the wedge acd pushed upward by the wall along that plane. The pressure which this larger wedge exerts against the wall is known as the **passive earth pressure**.

The force due to **active pressure**, as determined from the Coulomb's equation can be written as

$$P_a = k_a \cdot \frac{wh^2}{2} \qquad \text{........ (6.2)}$$

$k_a$ = coefficient of active pressure

$$k_a = \frac{\sin^2(\theta - \phi)}{\sin^2\theta \sin(\theta + \delta)\left[1 + \sqrt{\dfrac{\sin(\phi + \delta)\sin(\phi - \beta)}{\sin(\theta + \delta)\sin(\theta - \beta)}}\right]^2} \qquad \text{...(6.3)}$$

$\theta$ = angle of the pressure surface measured counterclockwise from the horizontal,

$\phi$ = angle of internal friction of the soil,

$\delta$ = angle of friction along the pressure surface (between soil and concrete),

$\beta$ = angle of retained material with the horizontal,

w = unit weight of the backfill,

and

l = vertical projection of the pressure surface.

When the pressure surface is vertical ($\theta$ = 90°) and the friction on that surface is such that $\beta = \delta$, then $k_a$ becomes

$$k_a = \cos\beta \left[ \frac{\cos\beta - \sqrt{\cos^2\beta - \cos^2\phi}}{\cos\delta + \sqrt{\cos^2\beta - \cos^2\phi}} \right] \qquad \text{........ (6.4)}$$

If the backfill is level ($\beta = 0$) then the coefficient of active pressure becomes

$$k_a = \frac{1 - \sin\phi}{1 + \sin\phi} \qquad \text{.........(6.5)}$$

The coefficient of passive pressure, $k_p$ is given by

$$k_p = \cos\beta \left[ \frac{\cos\beta + \sqrt{\cos^2\beta - \cos^2\phi}}{\cos\beta - \sqrt{\cos^2\beta - \cos^2\phi}} \right] \qquad \text{.........(6.6)}$$

and for $\beta = 0$,

$$k_p = \frac{1 - \sin\phi}{1 + \sin\phi} \qquad \text{.........(6.7)}$$

Backfills behind retaining walls are rarely uniform and dry. Proper drainage of the fill is vitally important to reduce pressures, but the pressure will temporarily increase during heavy storms or sudden thaws even in a well-drained fill since the water movement through the fill toward the drains causes additional seepage pressure. Frost action and other influences may also temporarily increase its value over that of the theoretical active pressure. Representative values for w and $\phi$ often used in engineering practice are shown in Table 6.1.

The values for the coefficient of friction between concrete and various soils are also shown in the table above. Soils of type 1 or 2 should be used for backfills of retaining walls wherever possible. Under saturated conditions, clays and silts may become entirely liquid. (i.e., $\phi = 0$).

Table 6.1
Unit Weights, Effective Angles of Internal Friction $\phi$, and Coefficients of Friction With Concrete f

| Soil | Unit weight, pcf | $\phi$, degrees | f |
|---|---|---|---|
| 1. Sand or gravel without fine particles, highly permeable | 110-120 | 33-40 | 0.5-0.6 |
| 2. Sand or gravel with silt mixture, low permeability | 120-130 | 25-35 | 0.4-0.5 |
| 3. Silty sand, sand and gravel with high clay content | 110-120 | 23-30 | 0.3-0.4 |
| 4. Medium or stiff clay | 100-120 | 25-35* | 0.2-0.4 |
| 5. Soft clay, silt | 90-110 | 20-25* | 0.2-0.3 |

### 6.2.2 External Stability

The principles discussed below are applicable to both the gravity and the reinforced concrete cantilever walls. The retaining wall proportions are first established such that the structure stability is ensured under the action of earth forces (Fig. 6.3). Three requirements for the structure stability must be satisfied:

(i) the overturning moment $P_{ah}$ (h'/3) must be more than balanced by the resisting moment, $(W.x_1 + Pa_v.l)$ to ensure an adequate factor of safety against overturning (usually about 2.0);

(ii) The frictional resistance F in combination with any passive resistance $P_p$ against the toe must be greater than $P_{ah}$ providing an adequate factor of safety (usually 1.5) against sliding;

(iii) The base width l must be adequate to distribute the resultant vertical force R to the foundation soil without causing excessive settlement or rotation.

For the Rankine pressure distribution, shown in Fig. 6.3, the overturning

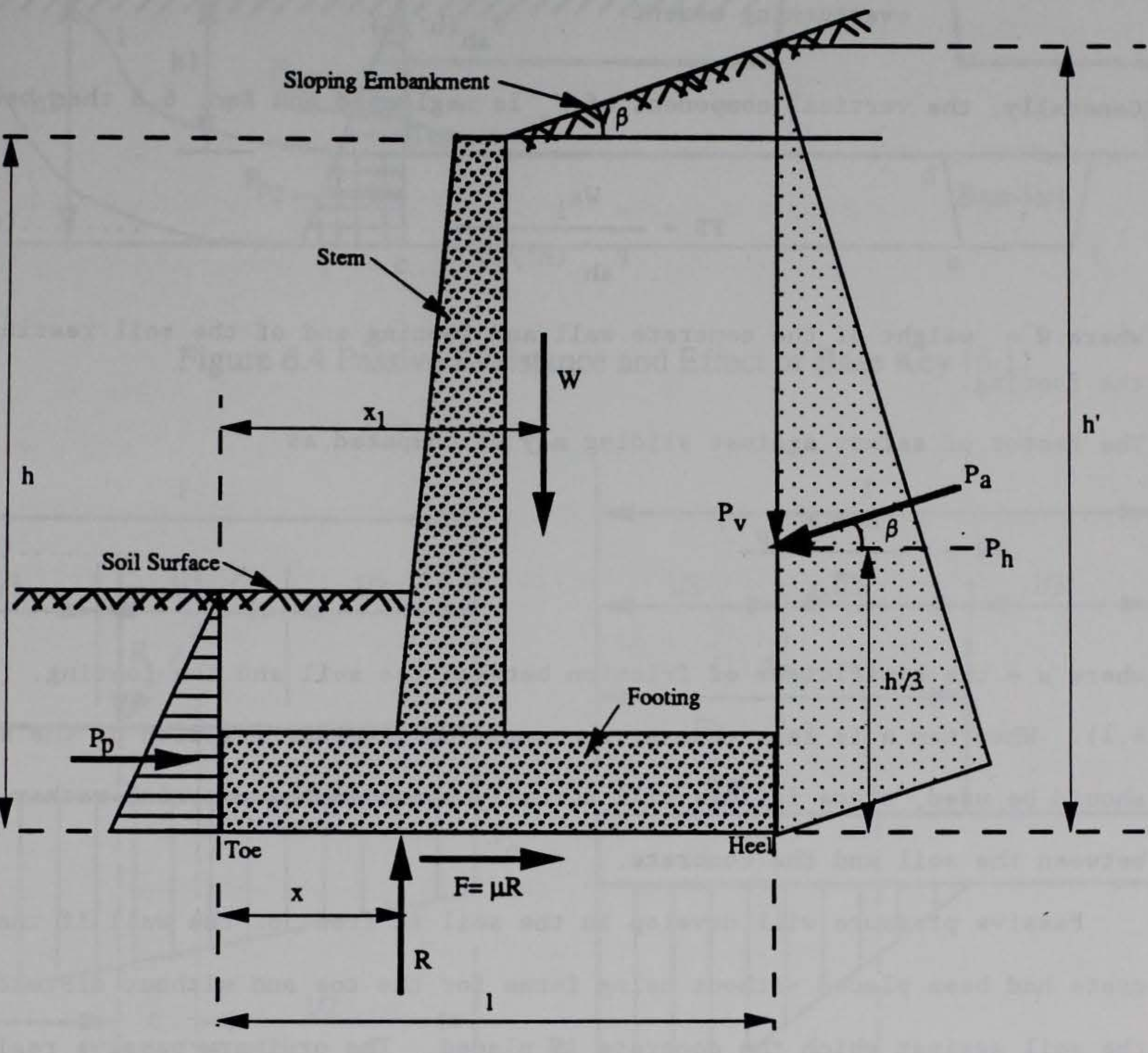



Figure 6.3. Cantilever Retaining Structure Under the Action of Earth Forces

factor of safety (FS) is computed as

$$FS = \frac{\text{resisting moment}}{\text{overturning moment}} = \frac{Wx_1 + P_{av} l}{P_{ah}(h'/3)} \quad \text{.........(6.8)}$$

Generally, the vertical component of $P_a$ is neglected and Eqn. 6.8 then becomes

$$FS = \frac{Wx_1}{P_{ah}\,(h'/3)} \quad \text{........(6.9)}$$

where W = weight of the concrete wall and footing and of the soil resting on the footing.

The factor of safety against sliding may be computed as

$$FS = \frac{\mu R + P_p}{P_{ah}} \quad \text{........(6.10)}$$

where $\mu$ = the coefficient of friction between the soil and the footing. (Table 6.1). When tan $\alpha$ is less than $\mu$, the angle of internal friction of the soil should be used, since failure will actually occur in the material rather than between the soil and the concrete.

Passive pressure will develop in the soil in front of the wall if the concrete had been placed without using forms for the toe and without disturbing the soil against which the concrete is placed. The ordinary passive resistance against the toe (Fig. 6.4) is expressed as

$$P_{p1} = \frac{1}{2}\, k_p \; wh_1^2 \quad \text{........(6.11)}$$

When the base key is placed in an unformed excavation against undisturbed material, an additional passive force $P_{p2}$ will be developed and a possible failure plane shifted from line 1 to line 2. This additional resistance is

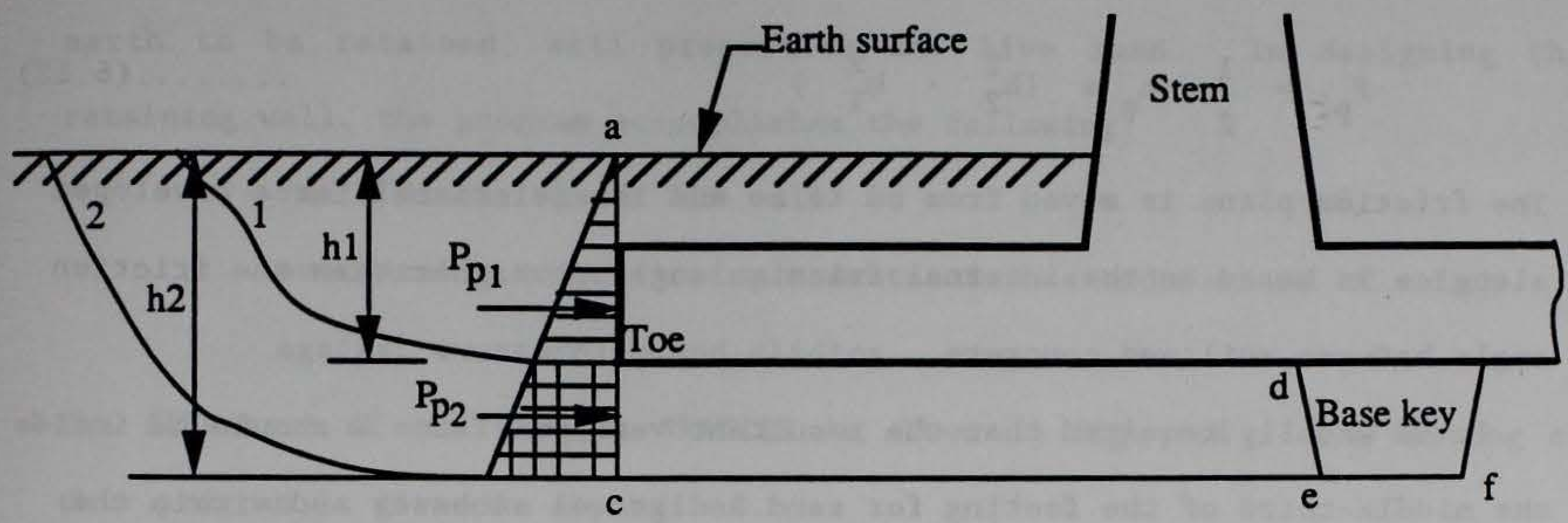
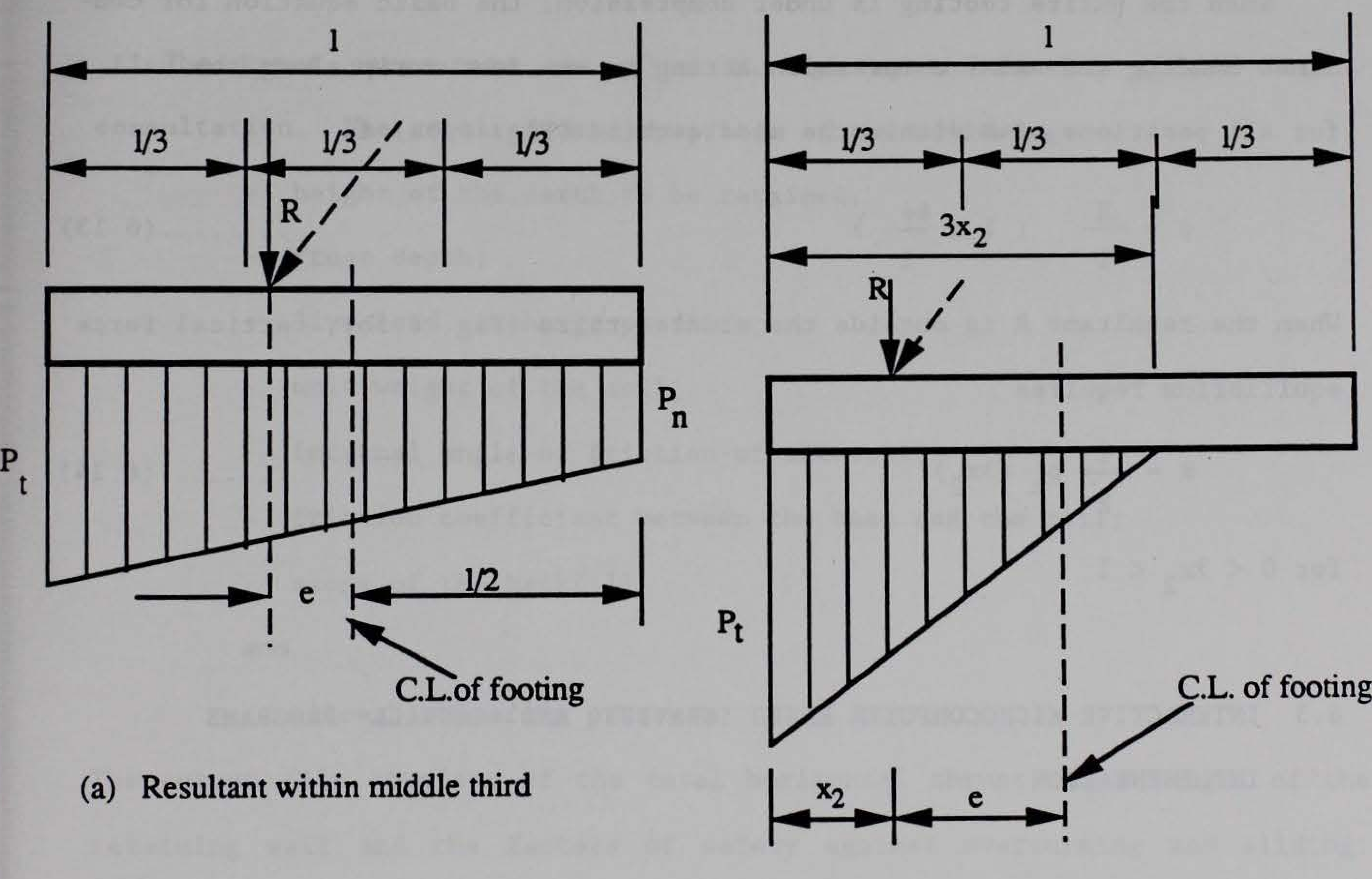



Figure 6.4 Passive Resistance and Effect of Base Key [6.1]




(a) Resultant within middle third

(b) Resultant outside middle third

Figure 6.5 Soil Pressure Distribution. [6.1] expressed as

$$P_{p2} = \frac{1}{2} c_p w (h_2^2 - h_1^2) \qquad \text{........(6.12)}$$

The friction plane is moved from bd to ce and the frictional force developed along ce is based on the internal friction angle $\phi$, rather than the friction angle between soil and concrete.

It is usually required that the resultant vertical force R should be inside the middle-third of the footing for sand and gravel subbases and within the middle-half for rock subbases and the maximum pressure may not exceed the allowable value.

When the entire footing is under compression, the basic equation for combined bending and axial compression acting on one foot strip along the wall for all positions of R within the middle-third (Fig. 6.5a) is

$$p = \frac{R}{l} \left( 1 \pm \frac{6e}{l} \right) \qquad \text{........(6.13)}$$

When the resultant R is outside the middle- third (Fig. 6.5b), vertical force equilibrium requires

$$R = \frac{1}{2} p_t (3x_2) \qquad \text{........(6.14)}$$

for $0 < 3x_2 < l$

## 6.3 INTERACTIVE MICROCOMPUTER BASED 'GRAVITY' AND 'CANWALL' PROGRAMS IMPLEMENTATION

### 6.3.1 Concrete Gravity Retaining Wall

#### 6.3.1.1 Program structure

The program GRAVITY provides the design of the concrete gravity retaining wall based on the first principles. The user must provide the height of the earth to be retained, soil properties and live load. In designing the retaining wall, the program accomplishes the following:

- main dimensions of the wall
- overturning and restoring moments, forces and factors of safety against overturning and sliding.

The flowchart of the program GRAVITY is shown in Fig. 6.6. The listing of this program is given in Appendix C.

6.3.1.2. Input and output

The input parameters are given through an interactive system - user consultation. The required data comprises of the following:

- height of the earth to be retained;
- frost depth;
- live load surcharge pressure;
- unit weight of the soil;
- internal angle of friction of the soil;
- friction coefficient between the base and the soil;
- slope of the backfill;

and

- allowable bearing pressure.

The output file consists of the total horizontal thrust, base length of the retaining wall and the factors of safety against overturning and sliding. Both the input and output are displayed on the screen and the design results can also be printed.

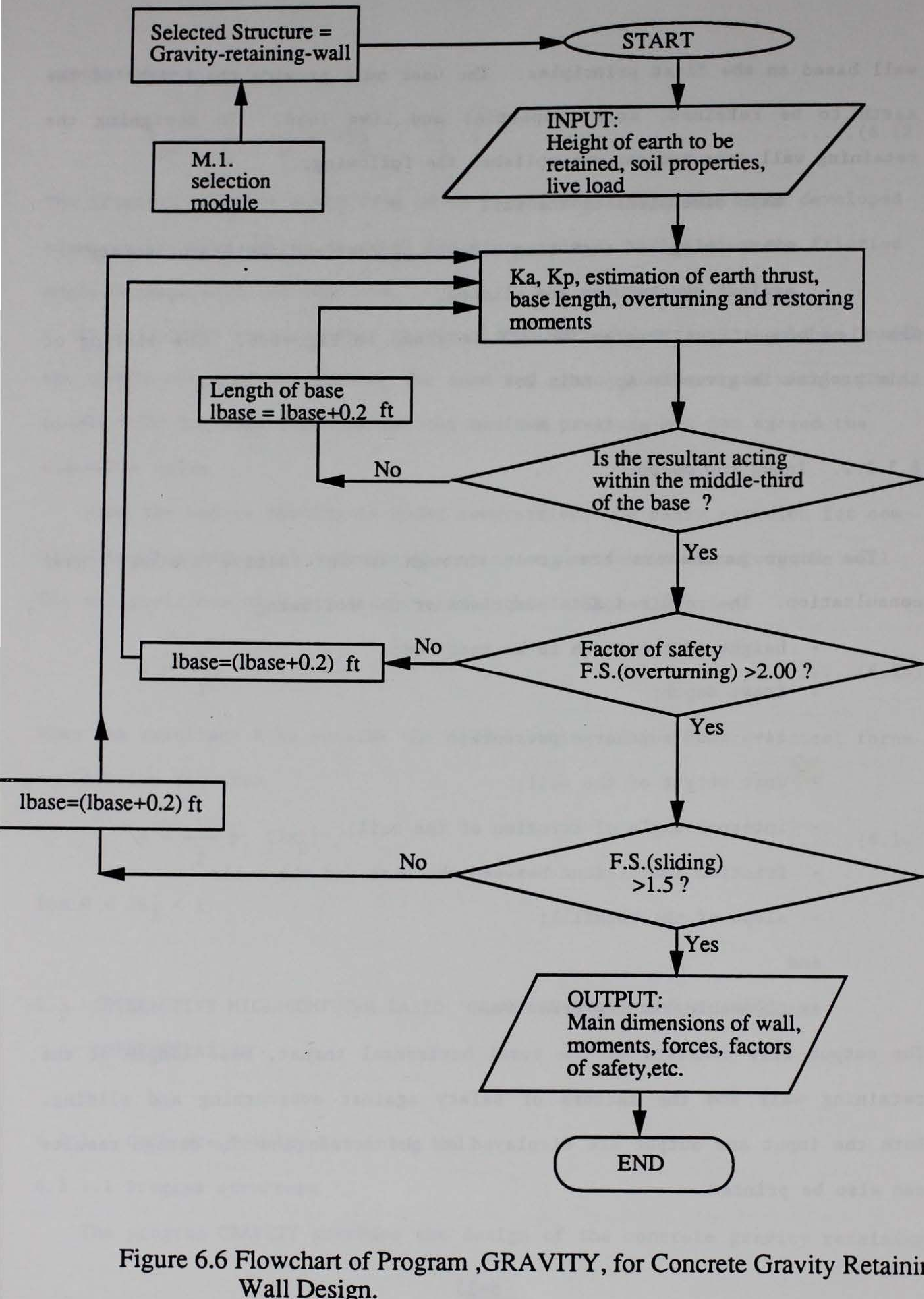



Figure 6.6 Flowchart of Program ,GRAVITY, for Concrete Gravity Retaining Wall Design.

6.3.1.3 Design example

The design example of a concrete gravity retaining wall is illustrated in this section. The user-system interactive consultation for the design input is shown in Fig. 6.7. Fig. 6.8 shows the input data as displayed on the screen.

The output file for the design example can be viewed on the screen as illustrated in Fig. 6.9.

6.3.2 Cantilever Retaining Wall

6.3.2.1 Program structure

The program CANWALL designs the cantilever retaining wall based on the earth pressure theories, stability requirements and the strength method of ACI 318-89. The user must provide strength properties for the concrete, reinforcing steel and soil. Additionally the user must specify the height of backfill, the surcharge, if any, and level of water table, etc. The program CANWALL gives the following final design results:

- main dimensions of the wall;
- determines the areas of steel required to resist flexure (stem and base);
- development length of bars;
- factors of safety against overturning and sliding.

The flowchart of the program CANWALL is shown in Fig. 6.10 and Appendix-D gives the listing of the program CANWALL.

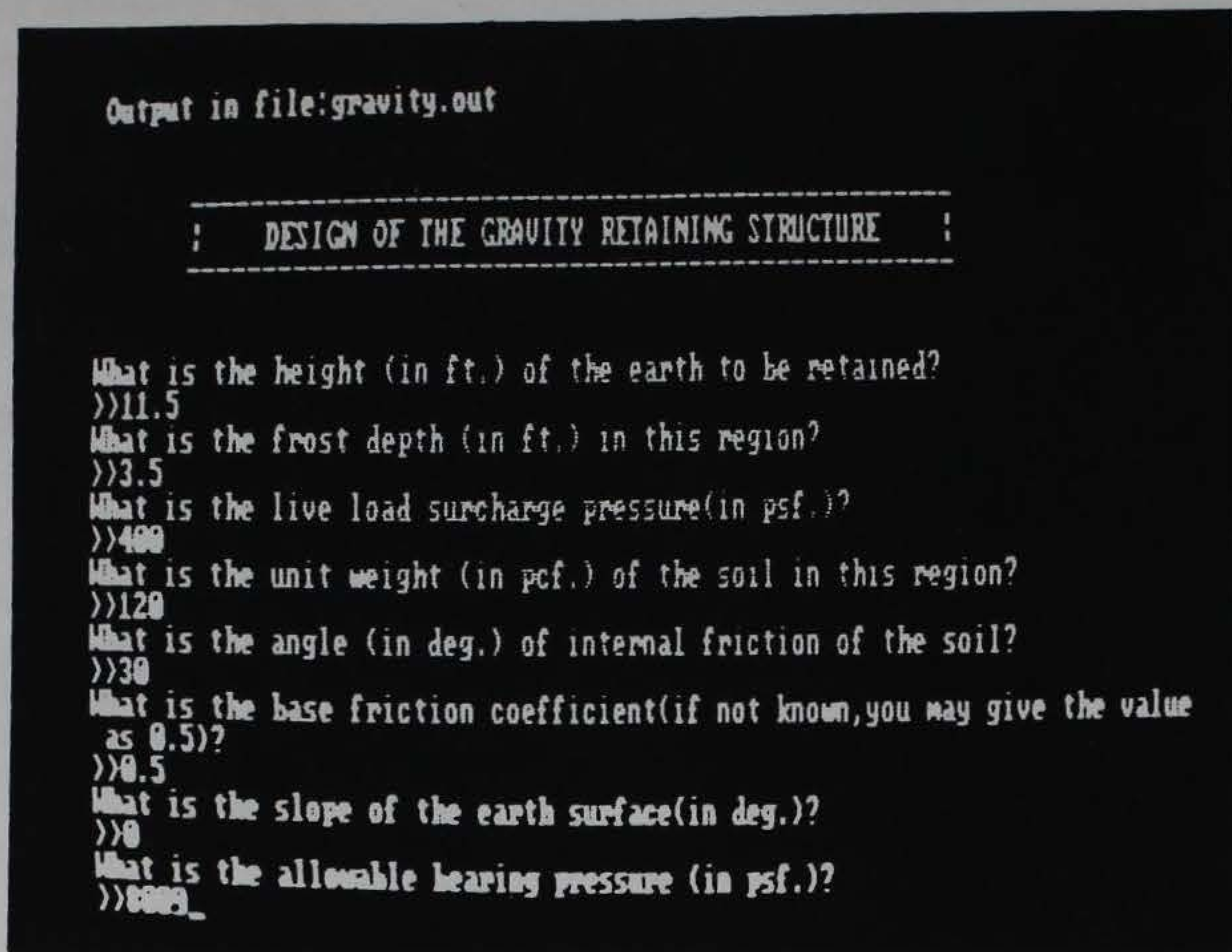



Figure 6.7 System-User Interactive Consultation for GRAVITY

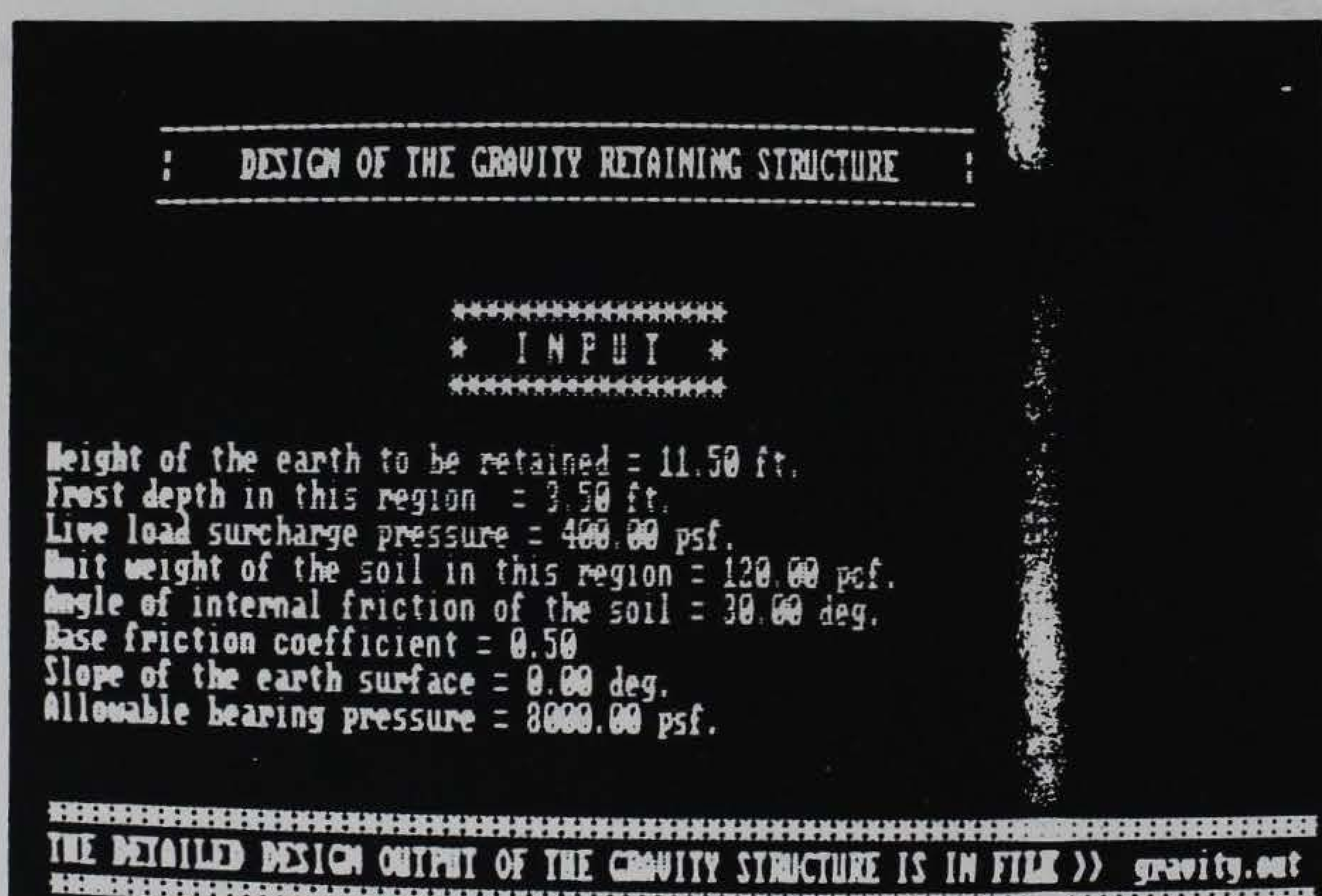



Figure 6.8 Screen Display of Input Data For GRAVITY

---

| DESIGN OF THE GRAVITY RETAINING STRUCTURE |

---

Coefficient of active pressure = 0.3333

Coefficient of passive pressure = 3.0000

Total earth thrust = 6500.00 lb.

Distance of earth's thrust from the base = 5.77 ft.

Length of the base = 9.75 ft.

Height of the structure = 15 ft.

Distance of the resultant from the front edge = 3.47 ft. --- OK (inside middle-third).

Overturning moment = 37500.00 ft.-lb.

Restoring moment = 116883.84 ft.-lb.

Maximum bearing pressure = 4377.27 psf.

Friction force = 11441.25 lb.

Passive pressure = 720.00 lb.

Factor of safety against overturning = 3.12 ---- > 2.0 ..... OK.

Factor of safety against sliding = 1.87 ---- > 1.5 ..... OK.

Figure 6.9 Output File of GRAVITY

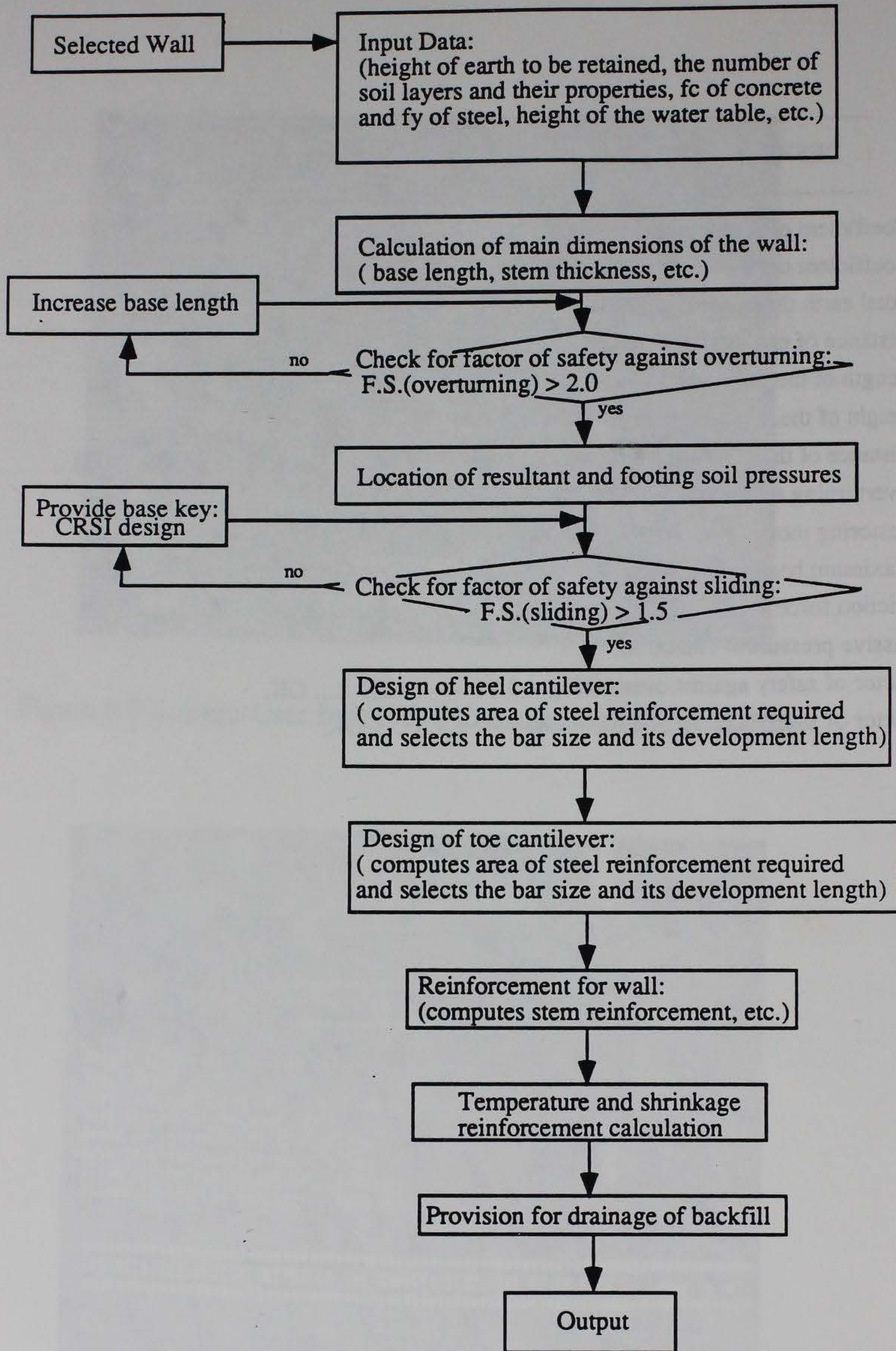



Figure 6.10. Flowchart of Program CANWALL for Cantilever Retaining Wall Design

6.3.2.2. Input and output

The input parameters are given through an interactive system-user consultation and the required data includes height of backfill to be retained, frost-depth, strength properties for the concrete and reinforcing steel, soil characteristics, surcharge load, backfill slope, number of soil strata and properties, groundwater table level, etc. The output file gives the dimensions of the stem and footing, the resisting and overturning moments and the associated factor of safety, sliding stability computations, areas of steel required to resist flexure in the stem, toe and heel of the base and the development lengths of the bars.

6.3.2.3. Design example

The design example of a cantilever retaining wall is illustrated in this section. The design input through the system-user interactive consultation is shown in Fig. 6.11. Fig. 6.12 shows the input data as seen on the screen. Fig. 6.13 illustrates the output file for the design example.

**REFERENCES**


6.1 CK Wang & CG Salmon, Reinforced Concrete Design, 3rd Ed, Harper & Row, NY, 1979, 359-387.

6.2 Building Code Requirements for Reinforced Concrete, ACI 318-89 Rev. 1989, Detroit, MI

6.3 HF Winterkorn & HY Fang, Foundation Engineering Handbook, Van Nostrand Reinhold Co. 1975, 402-443.

6.4 AH Nilson & G. Winter, Design of Concrete Structures, McGraw-Hill Book Co., 1986, 488-509.

6.5 GB Wynne, Reinforced Concrete Structures, Reston Publishing Co., 1981, 287-348.

6.6 JE Bowles, Foundation Analysis and Design, 4th Ed., McGraw Hill Book Co., 1988, 530-577.

```
C>canwall
Enter name of output file:canwall.out
Output in file:canwall.out

        : DESIGN OF CANTILEVER RETAINING STRUCTURE :

What is the height (in ft.) of the earth to be supported?
16
What is the angle (in deg.) which the backfill makes with the horizontal?
0
What is the height (in ft.) of the surcharge?
8
What is the density (pcf) of the surcharge material?
120
What is the angle (in deg.) of internal friction of the surcharge material?
35
What is the coefficient of friction between masonry and soil? (if unknown, you
may answer it as 0.40)
0.4
What is the compressive strength, fc (in ksi) of concrete?
3
What is the yield strength, fy (in ksi) of concrete?
40
What is the maximum soil pressure (in ksf)?
5
What is the height (in ft.) of the water table in this region?
0
How many layers of soil are there below the earth's surface(you are advised to g
ive the value of 1 or 2)?
1
What is the density of the soil(in pcf) below the earth's surface?
120
What is the angle(in deg.) of internal friction of the soil below the earth's su
rface?
35_
```

Figure 6.11System -User Interactive Consultation for CANWALL

```
***********
I N P U T
***********

Height of earth to be supported = 16.00 ft.
Angle which the backfill makes with the horizontal = 0.00 deg.
Height of the surcharge = 8.00 ft.
Density of the surcharge material = 120.00 pcf.
Angle of internal friction of the surcharge material = 35.00 deg.
Coefficient of friction between masonry and soil = 0.40
Compressive strength of concrete,fc = 3.00 ksi.
Yield strength of steel,fy = 40.00 ksi.
Maximum soil bearing pressure = 5.00 ksf.
Height of water table = 0.00 ft.
Layers of soil below the earth's surface = 1.0

******************************************************************
THE DETAILED DESIGN OF THE  CANTILEVER WALL IS IN FILE >> canwall.out
******************************************************************

C>_
```

Figure 6.12  Screen Display of Input Data for CANWALL

## | DESIGN OUTPUT OF CANTILEVER RETAINING STRUCTURE |

### | INPUT DATA |

Height of earth to be supported = 16.00 ft.
Angle which the backfill makes with the horizontal = 0.00 deg.
Height of the surcharge = 8.00 ft.
Density of the surcharge material = 120.00 pcf.
Angle of internal friction of the surcharge material = 35.00 deg.
Coefficient of friction between masonry and soil = 0.40
Compressive strength of concrete,fc = 3.00 ksi.
Yield strength of steel,fy = 40.00 ksi.
Maximum soil pressure = 5.00 ksf.
Height of water table = 0.00 ft.
Layers of soil below the earth's surface = 1.0

### | DESIGN OF CANTILEVER RETAINING STRUCTURE |

Coefficient of active pressure = 0.2710
Reinforcement ratio,Ro = 0.0139
Strength coefficient of resistance,Ru = 495.39 psi.
Total height of wall(including frost penetration) = 20.00 ft.
Thickness of footing = 2.00 ft.
Coefficient of active pressure = 0.2710
Base length of the footing = 11.30 ft.

STEM DIMENSIONS :

a) stem thickness at the bottom = 21 in.
b) stem thickness at the top = 12 in.

Overturning stability :

Resisting moment = 194.52 ft-kips
Overturning moment = 95.39 ft-kips
Factor of safety against overturning = 194.52 / 95.39 = 2.04 --OK-- (F.S. > 2.0)

The resultant soil pressure lies 0.3 in. outside the middle-third, however,it is very close, and the limiting condition of zero stress at the heel is considered adequate.

Sliding stability :

Sliding force = 11.71 kips
Resisting force = 10.59 kips
Factor of safety against sliding = 10.59 / 11.71 = 0.90
Key to be provided as factor of safety against sliding (0.90) is less than 1.5

Figure 6.13 Output File of CANWALL

Key dimensions(CRSI Handbook):The key has a square section (16.0in. x 16.0in.)

Generally it is desirable to place the front face of the key about 5 in. in front of the back face of the stem.This will permit anchoring the stem reinforcement, if present, in the key.

---

design of heel cantilever

---

Heel thickness = 29
Choose #8 @ 6 in.(As = 1.58 sq.in./ft.)
The required development length of the bars (at the top) is 32.31 in. into the toe of the footing measured from the stem reinforcement.Use an embedment length of 3.1 ft. from the backface of the wall.

---

design of toe cantilever

---

PhiVc(24.02 k.) > Vu(13.50 k.) ---- OK
Choose #7 @ 6 in.(As = 1.20 sq.in./ft.)
The required development length of the bottom bars is 24.54 in.
Use an embedment length of 2.5 ft. from the front face of the wall

---

design of reinforcement of wall

---

Choose #11 @ 6 in.(As = 3.12 sq.in./ft.)
The required development length of the bottom bars is 63.80 in.
Use an embedment length of 5.7 ft.

---

temperature and shrinkage reinforcement

---

Amount of horizontal temperature and shrinkage steel required on the front face = 0.33 sq.in.
Amount of horizontal temperature and shrinkage steel required on the back face = 0.16 sq.in.

For the face side of the wall:

Choose #6 @ 12 in.(As = 0.44 sq.in./ft.)

For the back side of the wall:

Choose #4 @ 12 in.(As = 0.20 sq.in./ft.)

Drainage :

Adequate drainage of backfill must be provided to prevent water from accumulating in the backfill material; a common minimum provision is to provide weep holes (4 in. diameter) every 10 to 15 ft. along the wall.

---

Figure 6.13 Continued

## CHAPTER 7

## GABION RETAINING STRUCTURES

### 7. 1 INTRODUCTION

Gabion walls are 'cellular' structures formed with rectangular cages made of zinc coated steel wire mesh, filled with stone of the proper size and mechanical characteristics. The individual units are firmly tied to each other with zinc coated wire so as to form a monolithic structure. The wire mesh must have high mechanical resistance and high resistance to corrosion and good deformability. Hexagonal double-twisted wire mesh, either zinc coated or coated with zinc plus PVC meets all the above requirements. The gabion structures, which have flexibility and permeability are specially applicable in the cases of sites with unconsolidated soils with modest mechanical strength and subject to settlement. A retaining wall built of box gabions is a homogeneous monolithic structure which can function under tension and absorb unforeseen stresses.

### 7.2. CHARACTERISTICS OF GABION AND FILLING MATERIALS

The gabions are subdivided into cells by inserting diaphragms spaced at 1 m. from each other to strengthen the structure and facilitate its assembly. The gabion's width would be 1 to 2 m. (Fig. 7.1a). The rectangular cages are made of double-twisted hexagonal mesh and high quality zinc coating. The double twisting ties together with the wires ensure that the mesh will not unravel even if one or more wires break (Fig. 7.1b). Zinc coating provides

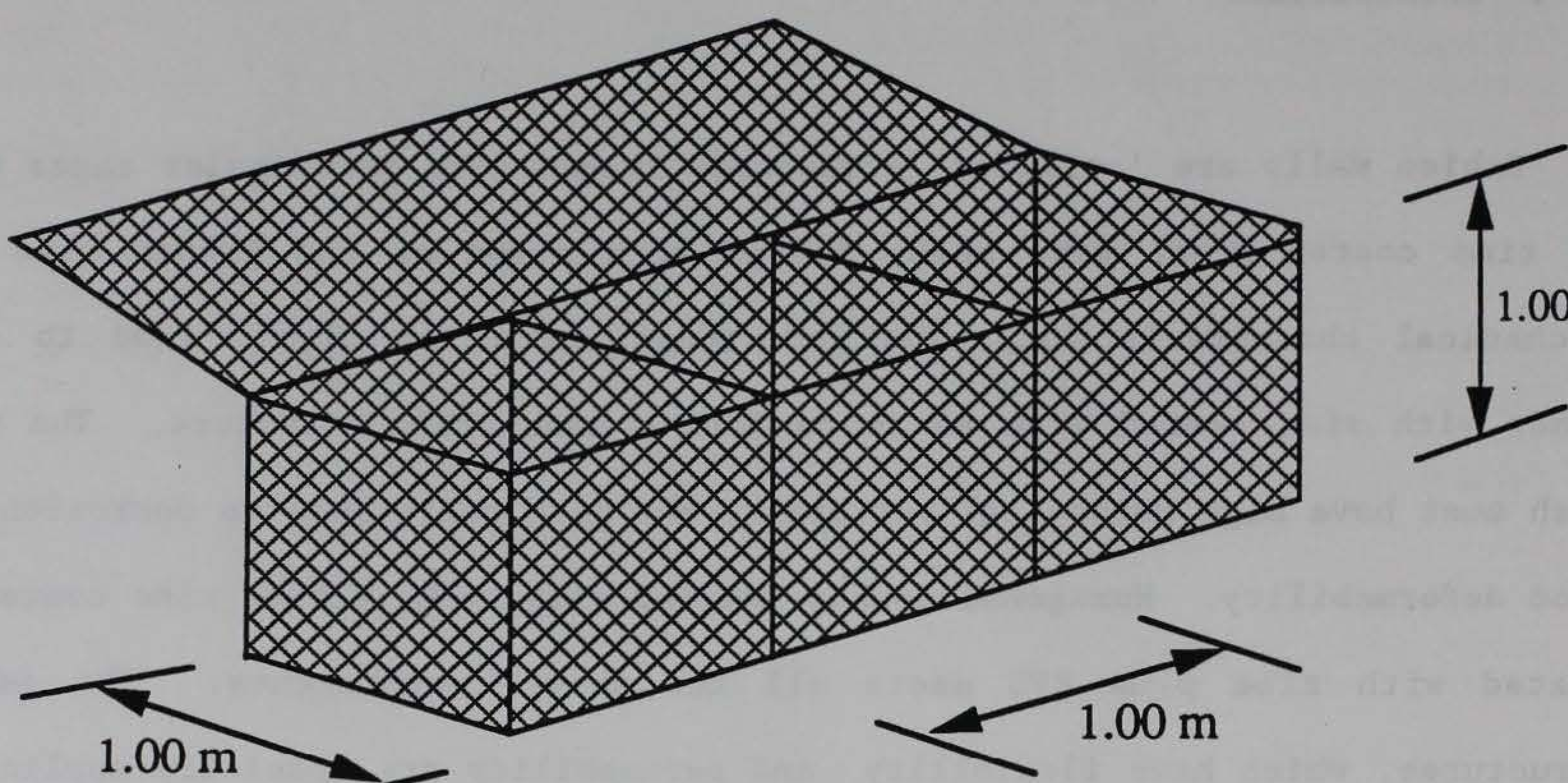



Figure 7.1a Typical Gabion Basket [7.1]

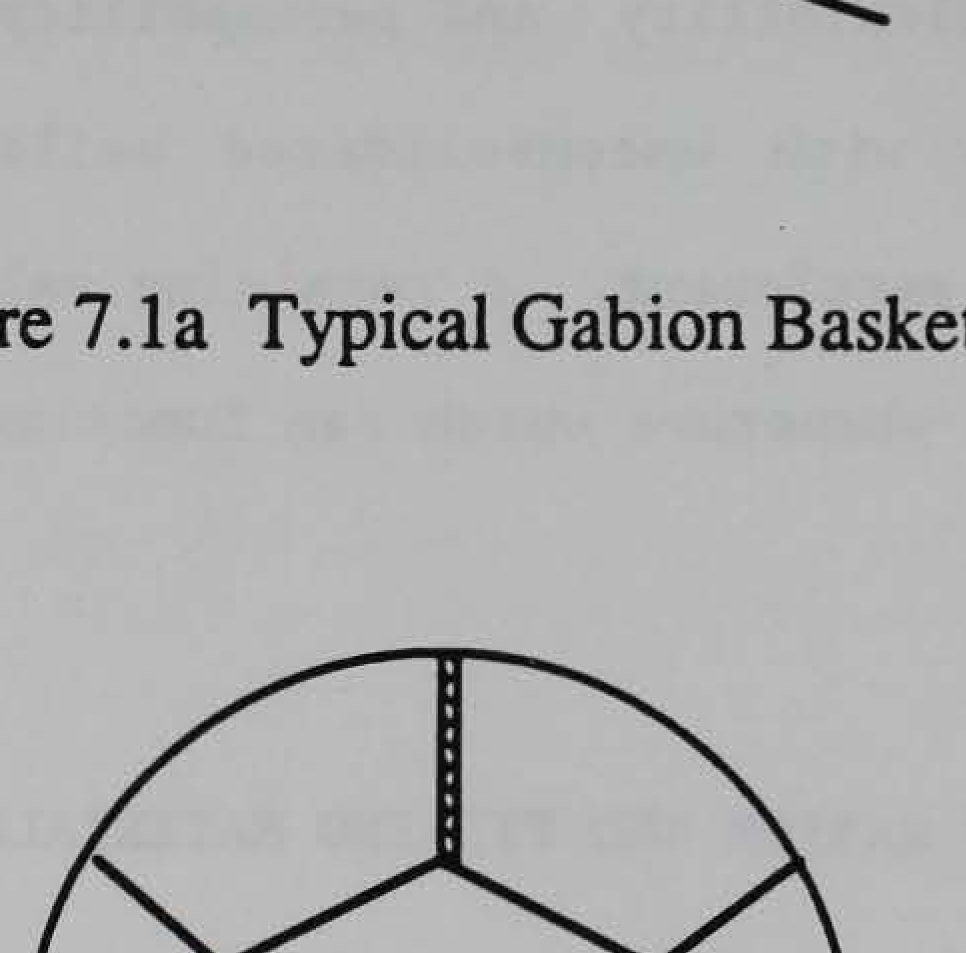



Figure 7.1b Detail of Wire Mesh. [7.1]

long term protection for steel wire against oxidation. The zinc coated wire, before being woven, can be coated with a special PVC (polyvinyl chloride) 0.4 to 0.6 mm thick and this additional coating protects from corrosion in marine or heavily polluted environment.

The most commonly used materials to fill the gabion are round or quarried stones. The stones must be weather resistant, non-friable, insoluble and sufficiently hard to ensure the structure durability. Table 7.1 shows the densities of materials which are particularly preferable for the gravity structure in submerged condition or exposed to running water. The porosity of the gabion, n, generally varies from 0.30 to 0.40 depending on the hardness and angularity of the stone. Fig. 7.2 shows the relationship for the determination of the apparent density of the filled gabion, $\gamma_g$ for a given density of the filling material, $\gamma'_s$ and the porosity of the gabion, n. The most appropriate size for stone varies from 1 to 2 times the dimension D of the mesh (Fig. 7.1b) to ensure that the stone should be large enough to prevent its escape through the mesh. The use of smaller size stone, 1 to 1.5D, enables an improved and more economical filling of the cage and also allows a better distribution of the imposed loads and structure adaptability to deformation.

## 7.3. CHARACTERISTICS OF GABION RETAINING STRUCTURES

The gabions are reinforced structures which are capable of resisting any type of stress and especially tension and shear. Particular attention should be given to details such as the selection of the proper type of gabion, careful filling and correct lacing methods. The gabion structures are also deformable and permeable. The structure has the characteristics of a complex

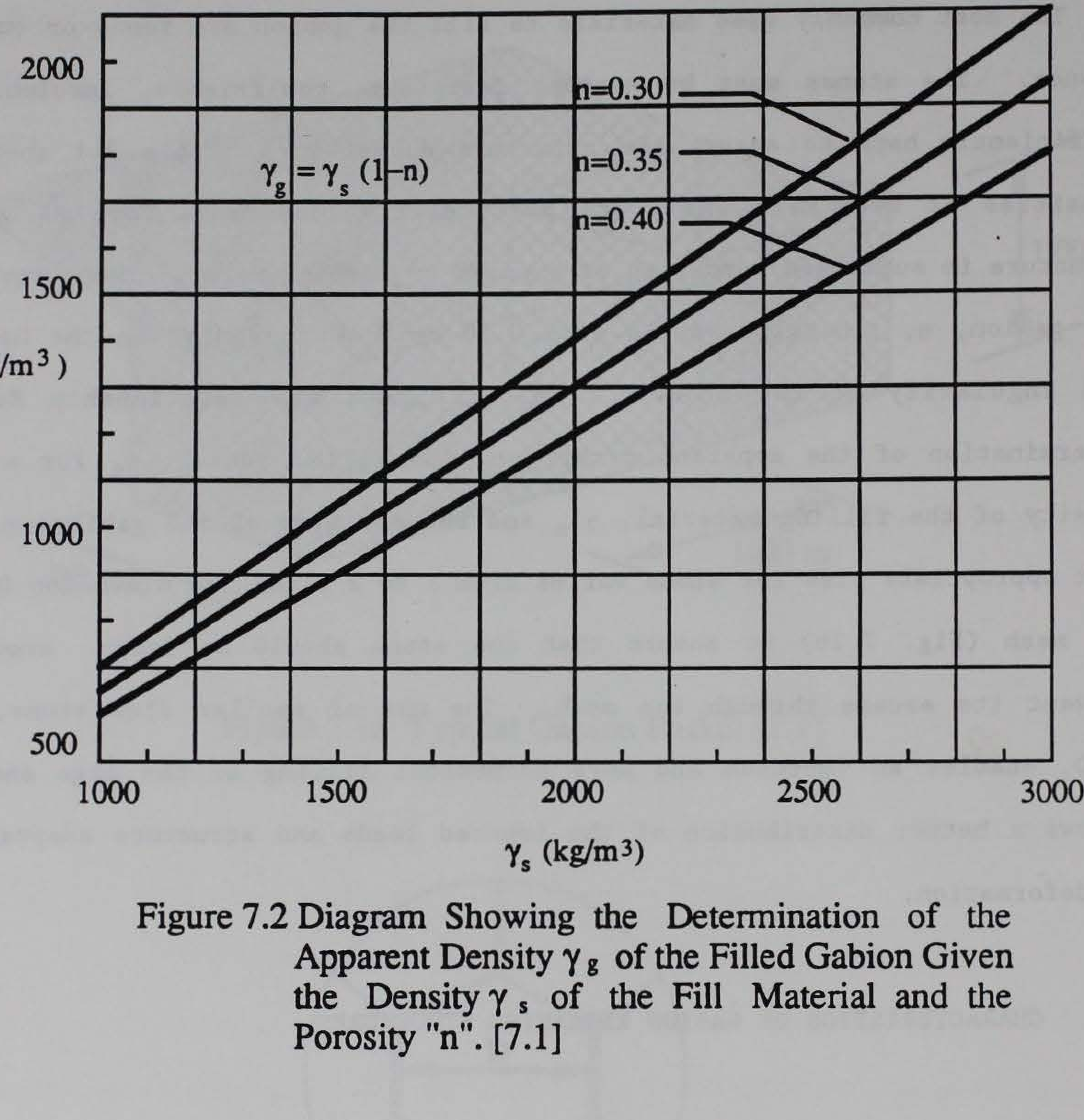



Figure 7.2 Diagram Showing the Determination of the Apparent Density $\gamma_g$ of the Filled Gabion Given the Density $\gamma_s$ of the Fill Material and the Porosity "n". [7.1]

reinforced one capable of adapting to the redistribution of loads due to the movement of the stonefill. The distribution and the final value of the soil stresses depend on the manner these structures deform and on the extent of the deformation. The gabion retaining structures are capable of collecting and carrying away ground water, thus eliminating/attenuating one of the principal causes of soil instability. Auxiliary means should be installed for the disposal of the water collected through the structure.

Gabion retaining structures have gained popularity due to i) long term durability, ii) ease of installation, iii) possible modification of the structure at a later date, iv) capacity for effective functioning and v) ease of maintenance.

Table 7.1. Density of Different Types of Stone

| Type of rock | Density $\gamma_s$ (kg/m$^3$) |
|---|---|
| Basalt | 2900 |
| Granite | 2600 |
| Hard limetone | 2600 |
| Trachytes | 2500 |
| Sandstone | 2300 |
| Soft limestone | 2200 |
| Tuff | 1700 |

## 7.4 DESIGN CRITERIA

### 7.4.1 Gabion Wall Types

Gabion retaining structures can be classified into the following four different types (Fig. 7.3):

i) gravity retaining structures,

ii) semi-gravity retaining structures,

iii) embankment stabilization structures,

and

iv) thin walls anchored back into the soil by panels of hexagonal mesh.

Gravity retaining structures are designed using traditional methods of analysis. Semi-gravity structures derive their resistance from the reinforcement provided by the mesh to an extent that they could be considered as a special type of reinforced structure. Structures for the embankment stabilization may be treated as gravity structures. For proper functioning of gabion retaining structures, it is desirable that the face should be battered at not less than 6 degrees, or should be stepped when the heights are greater than 3 m (~10 ft.).

### 7.4.2 Loads Imposed by the Backfill

The loads imposed by the soil on a retaining wall depend not only on the internal angle of friction and cohesion of the soil, but also on the extent to which the structure can deform. The active pressure in an earth retaining structure can only develop when the retaining structure has deformed sufficiently. Table 7.2 shows the lower limits of movement necessary to bring into action the active pressure.

Table 7.2 Minimum Displacement and Rotation Required to Develop the Active Pressure in Retaining Structures

| Type of Soil | Deflection $\delta$ | Rotation $\theta$ |
|---|---|---|
| i) Non cohesive | 0.001 H | 0.001 |
| ii) Cohesive | 0.004 H | 0.004 |

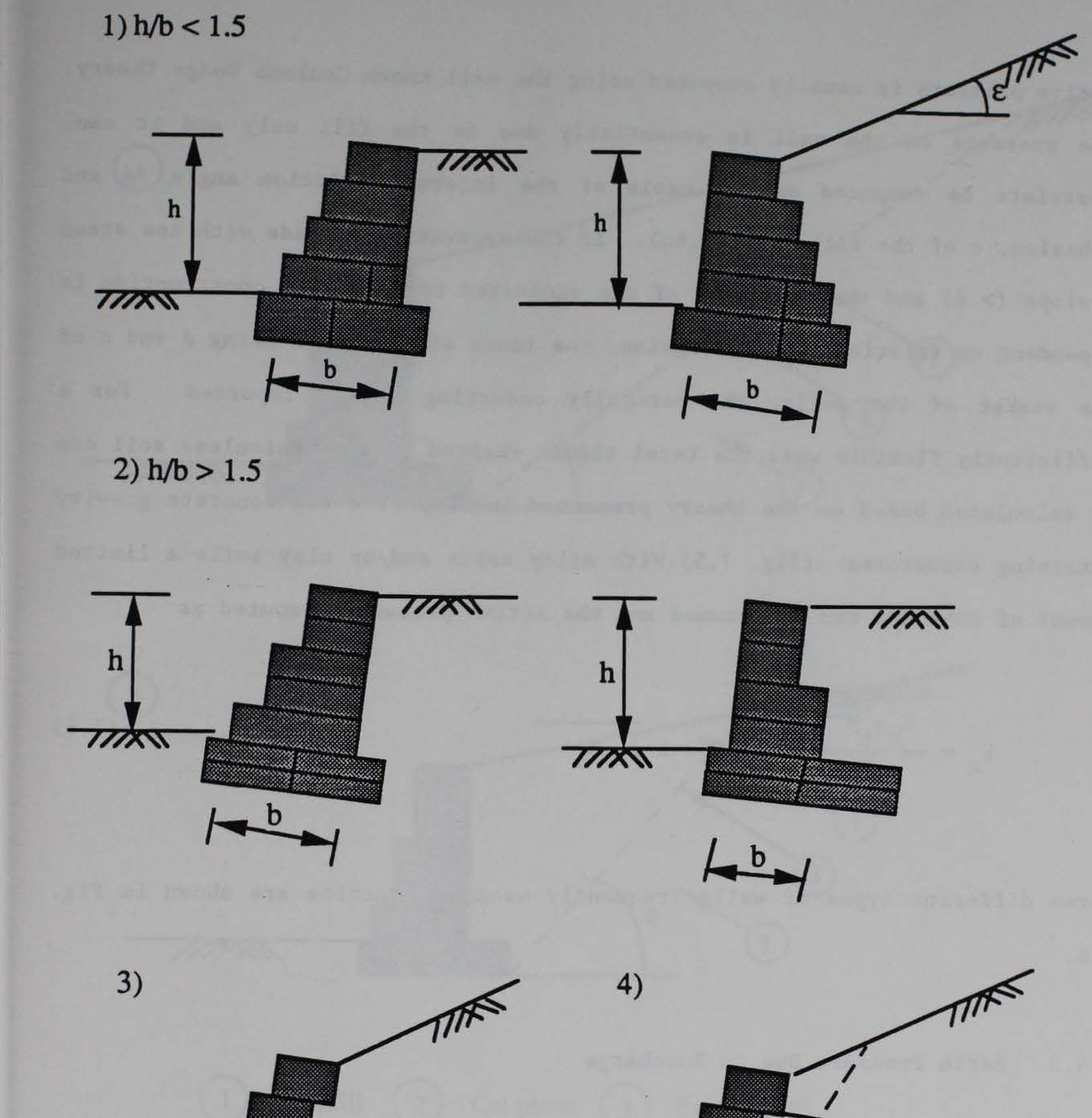



Figure 7.3 Types of Gabion Wall:
1) Gravity; 2) Semigravity; 3) Wall supporting sloping surcharge; 4) Thin walls with tie-back mesh panels. [7.1]

Active pressure is usually computed using the well known Coulomb Wedge Theory. The pressure on the wall is essentially due to the fill only and it can, therefore be computed on the basis of the internal friction angle, $\phi$ and cohesion, c of the fill (Fig. 7.4a). If the excavation is made with too steep a slope ($> \phi$) and the stability of the excavated force during construction is dependent on friction and/or cohesion, the loads are computed using $\phi$ and c of the weaker of the soils, the naturally occurring or the imported. For a sufficiently flexible wall the total thrust exerted by a cohesionless soil can be calculated based on the theory presented in Chapter 6 for concrete gravity retaining structures (Fig. 7.5). With silty soils and/or clay soils a limited amount of cohesion can be assumed and the active pressure computed as

$$S_a = \frac{K_a \gamma_t H^2}{2} - 2 c H \sqrt{K_a} \qquad \text{.........(7.1)}$$

Three different types of walls frequently used in practice are shown in Fig. 7.6.

7.4.3. Earth Pressure Due to Surcharge

The thrust from a distributed surcharge load on the horizontal backfill (Fig. 7.7) is computed as

$$S_a = p_o H K_a \qquad \text{.........(7.2)}$$

where

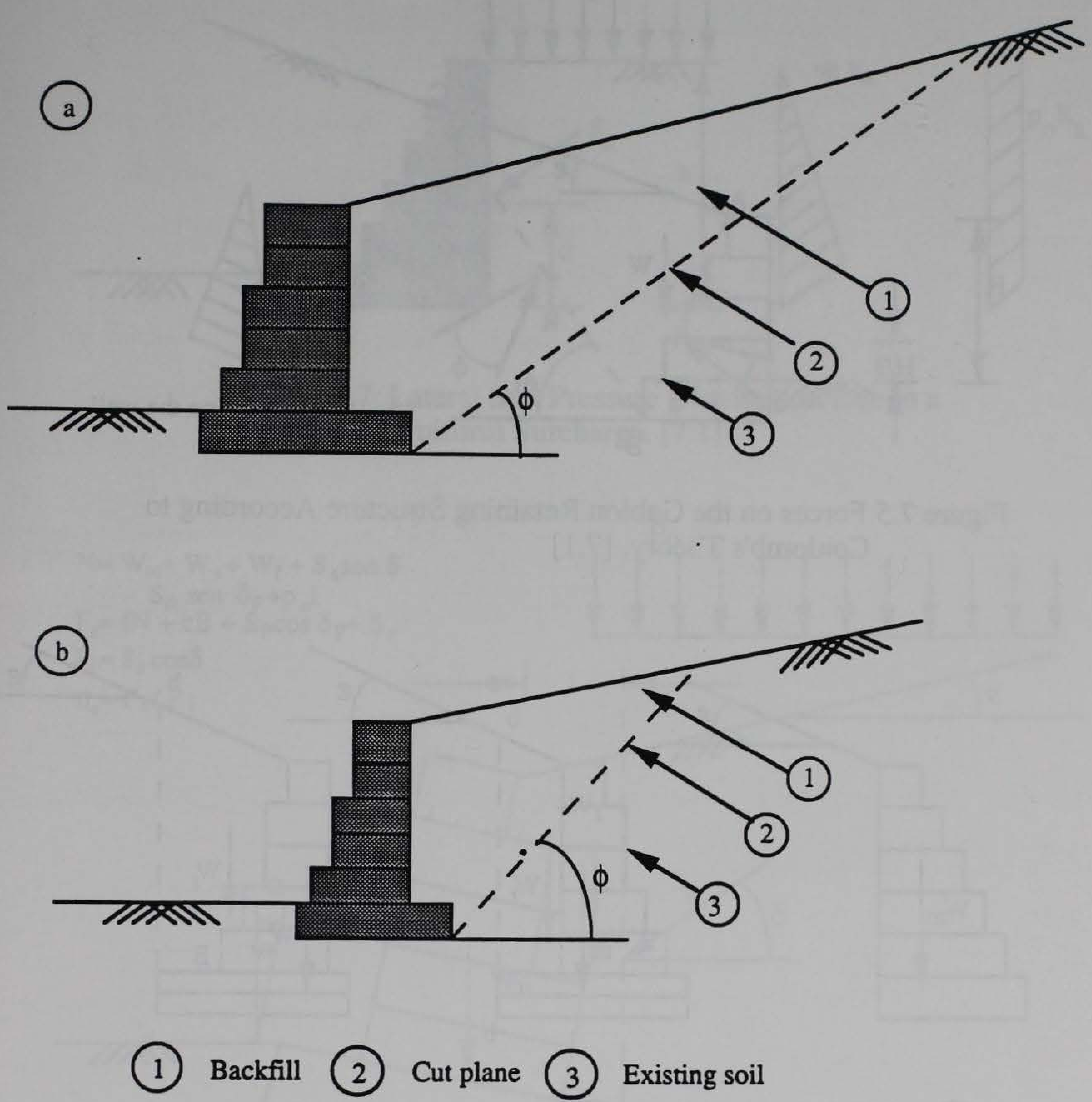



Figure 7.4 Schemes for the Backfilling of Gabion Structure. [7.1]

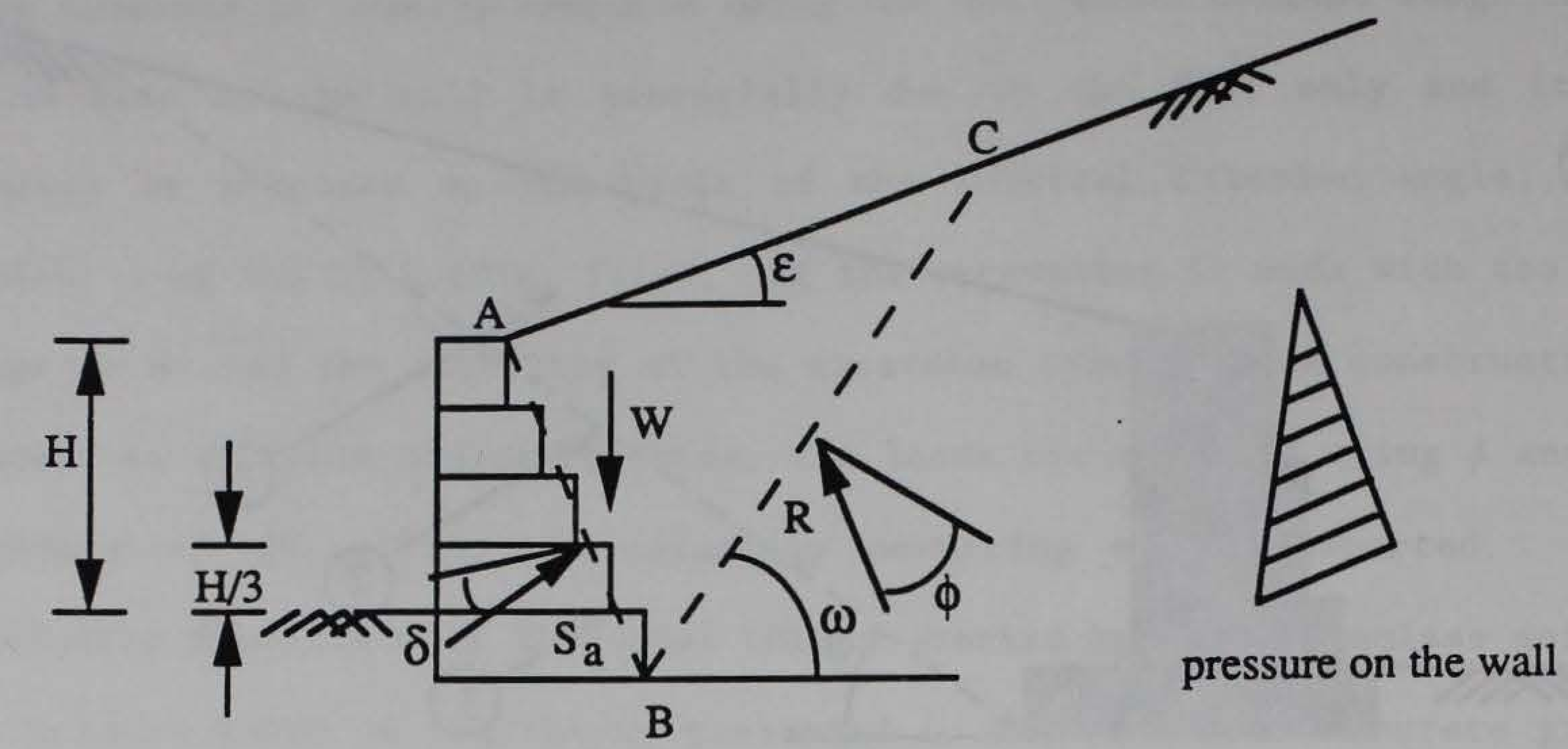



Figure 7.5 Forces on the Gabion Retaining Structure According to Coulomb's Theory. [7.1]

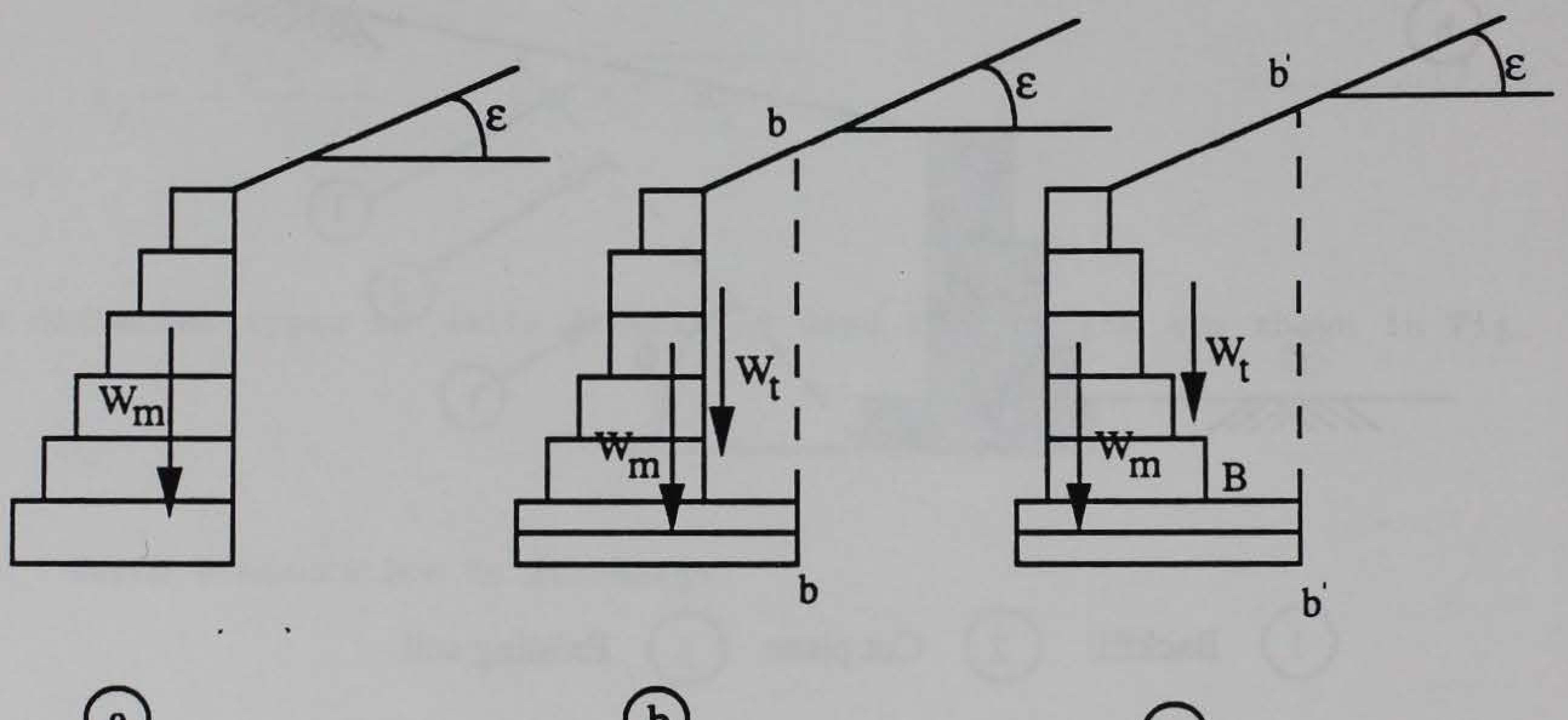



Figure 7.6 a) Gabion Gravity Wall; b) Semi-gravity Gabion Wall; c) Semi-gravity Gabion Wall with Anchorage Heel. [7.1]

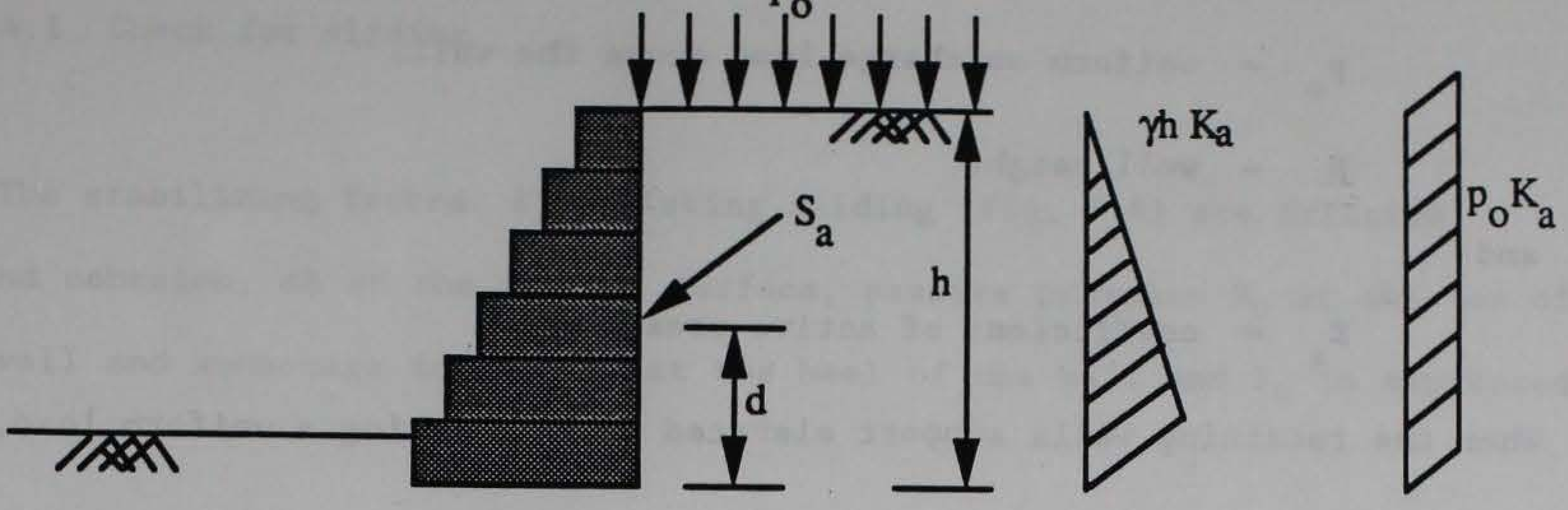



Figure 7.7 Lateral Soil Pressure Distribution Due to a Uniform Surcharge. [7.1]

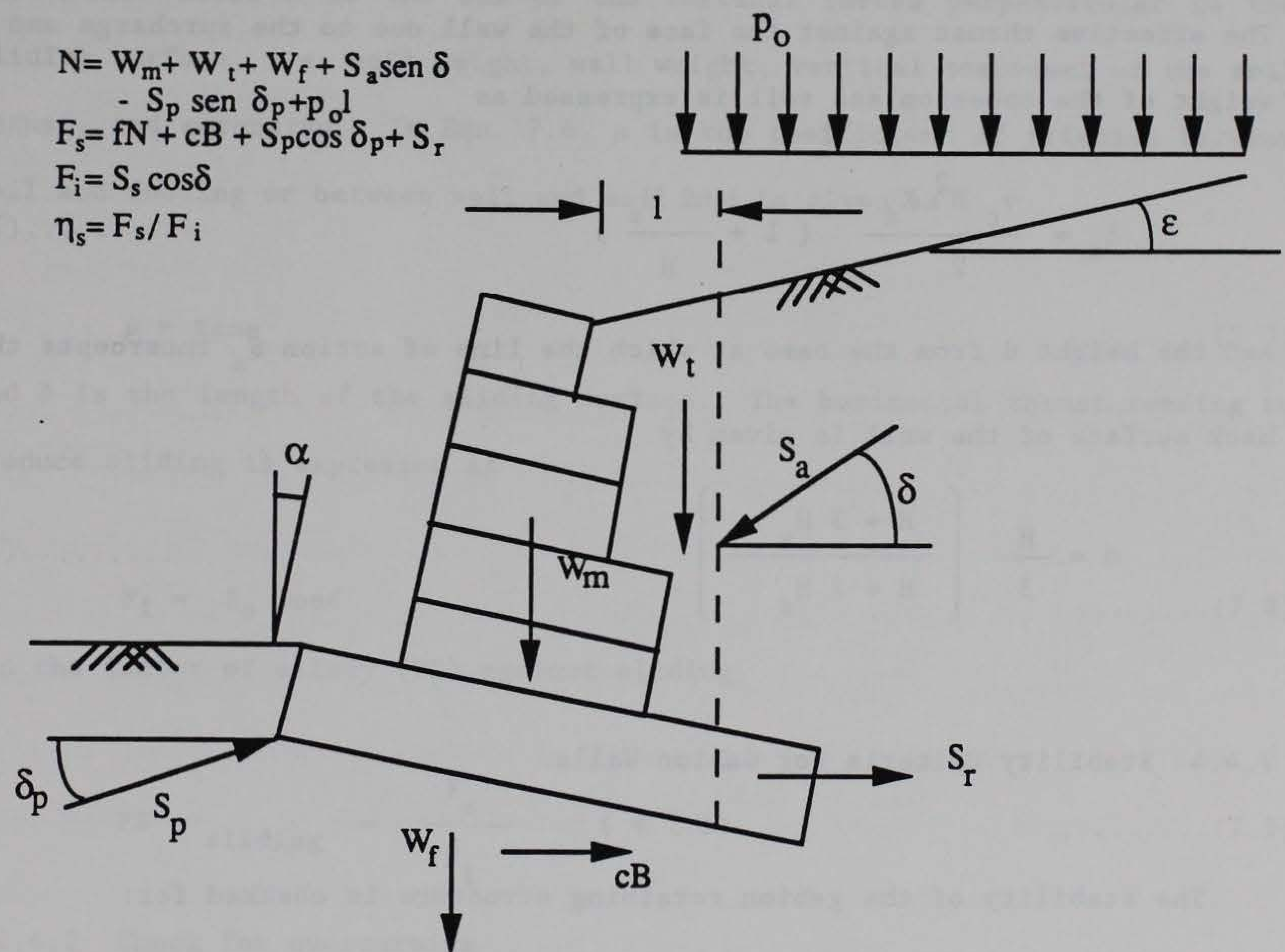



Figure 7.8 Force to be Considered in Analysing the Resistance to Sliding. [7.1]

$p_o$ = uniform surcharge load above the wall,

$H$ = wall height,

and

$K_a$ = coefficient of active pressure.

When the retaining walls support elevated roads exerting a uniform load, $p_o$ above the wall, the equivalent thickness of a soil layer, $H_s$ is given by

$$H_s = \frac{p_o}{\gamma_t} \qquad \text{.........(7.3)}$$

The effective thrust against the face of the wall due to the surcharge and the weight of the cohesionless soil is expressed as

$$S_a = \frac{\gamma_t H^2 K_a}{2} \left( 1 + \frac{2 H_s}{H} \right) \qquad \text{.........(7.4)}$$

and the height d from the base at which the line of action $S_a$ intercepts the back surface of the wall is given by

$$d = \frac{H}{3} \left[ \frac{H + 3 H_s}{H + 2 H_s} \right] \qquad \text{.........(7.5)}$$

7.4.4 Stability Criteria for Gabion Walls

The stability of the gabion retaining structure is checked for:

i) sliding,

ii) overturning,

and

iii) overall stability of the wall and surrounding soil.

7.4.4.1 Check for sliding

The stabilizing forces, $F_s$ resisting sliding (Fig. 7.8) are friction, $\mu N$ and cohesion, cB at the sliding surface, passive pressure $S_p$ at the toe of the wall and anchorage forces, $S_r$ at the heel of the wall and $F_s$ is expressed as

$$F_s = \mu N + cB + S_p \cos\delta + S_r \qquad \text{............(7.6)}$$

The normal force N is the sum of the vertical forces perpendicular to the sliding surface, i.e. soil weight, wall weight, vertical component of the soil thrust, and surcharge. In Eqn. 7.6, $\mu$ is the coefficient of friction between soil and footing or between soil and soil and is given as

$$\mu = \tan\phi \qquad \text{........(7.7)}$$

and B is the length of the sliding surface. The horizontal thrust tending to produce sliding is expressed as

$$F_i = S_a \cos\delta \qquad \text{........(7.8)}$$

and the factor of safety (FS) against sliding

$$FS_{\text{sliding}} = \frac{F_s}{F_i} \quad (\not< 1.5) \qquad \text{.........(7.9)}$$

7.4.4.2 Check for overturning

For a retaining wall which is resisting active earth pressure by virtue of its mass, the overturning and restoring moments are respectively given by

$$M_i = S_a l_a \qquad \text{........(7.10)}$$

$$M_s = W_{tot} l_w + S_p l_p \qquad \text{........(7.11)}$$

where $W_{tot}$ = resultant of the wall weight and the weight of the boxed soil above the heel.

$l_a$, $l_w$, $l_p$ = the lever arms for the forces $S_a$, $W_{tot}$, and $S_p$ respectively. The factor of safety against overturning is given by

$$FS_{\text{overturning}} = \frac{M_s}{M_i} \quad (\nless 1.5) \qquad \text{........(7.12)}$$

7.4.4.3. Check for overall stability

The analysis for the retaining wall failure on a semi-circular slip surface located within the soil below and behind the wall is made by a search for the slip circle, which has the smallest safety against failure. In the special situation where the fill behind the wall consists of cohesionless soil, the failure surface can be assumed to be a plane inclined at an angle $\omega$ with the horizontal. This surface is assumed to pass through point A, which is the lowest extreme point on the heel of the foundation as shown in Figure 7.9. The angle $\omega$ is made to vary to find the failure plane with the lowest factor of safety against sliding. The critical surface for the whole mass is the one with the smallest factor of safety given by

$$FS_{\text{overall stability}} = \frac{S_p}{S} = \text{min} \quad (\nless 1.2) \qquad \text{........(7.13)}$$

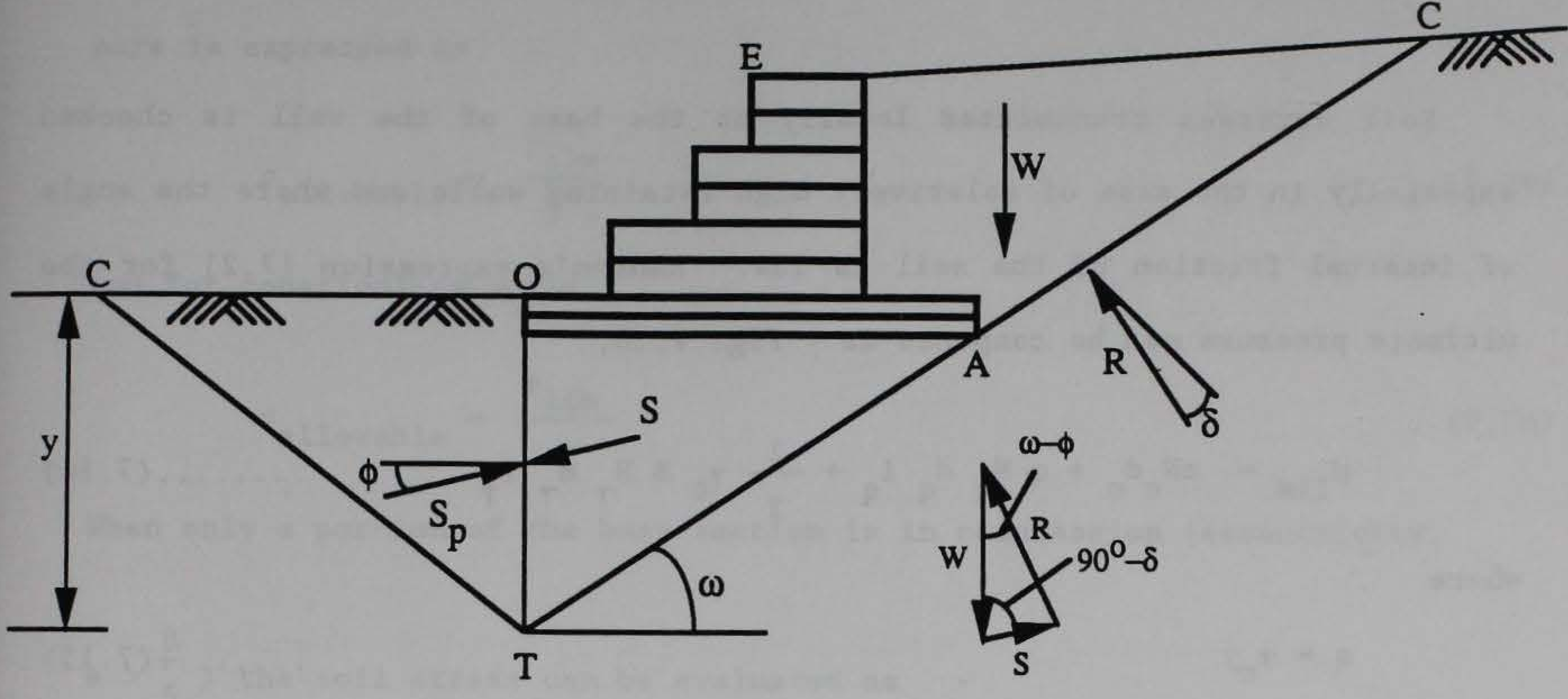



Figure 7.9 Diagram for the Analysis of Overall Stability. [7.1]

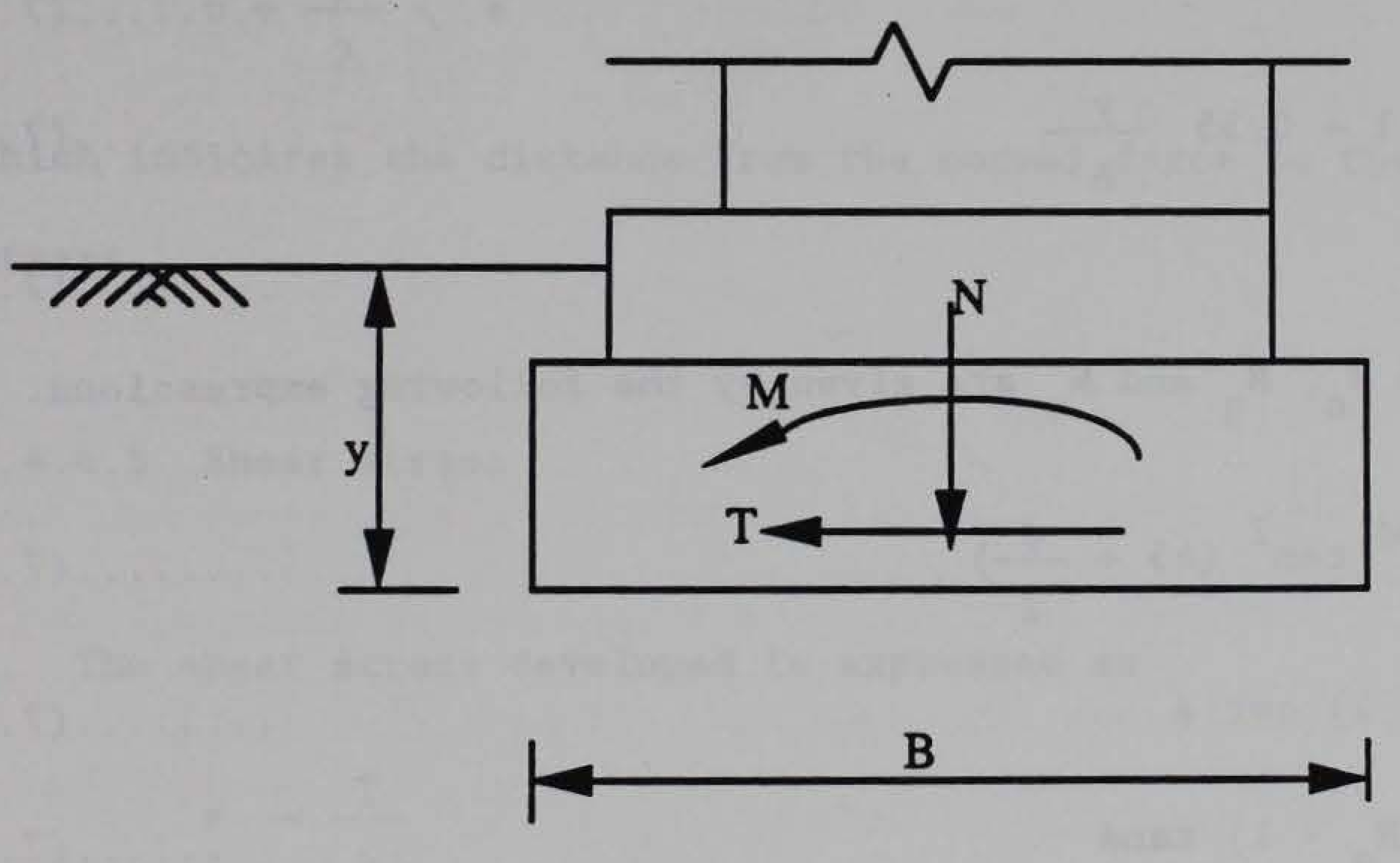



Figure 7.10 Characteristics of the Stresses Acting on the Foundation of a Gabion Structure. [7.1]

7.4.4.4 Check on foundation bearing pressure

Soil stresses transmitted locally at the base of the wall is checked especially in the case of relatively high retaining walls and where the angle of internal friction of the soil is low. Hanson's expression [7.2] for the ultimate pressure can be computed as - Fig. 7.10,

$$p_{lim} = cN_c d_c + q\, N_q\, d_q\, i_q + \frac{1}{2}\gamma_t\, B\, N_\gamma\, d_\gamma\, i_\gamma \qquad \text{........(7.14)}$$

where

$$q = \gamma_t y \qquad \text{........(7.15)}$$

$$i_q = 1 - 0.5\,\frac{T}{N} \qquad \text{........(7.16)}$$

$$i_\gamma = i_q^2 \qquad \text{........(7.17)}$$

$$d_c = d_q = 1 + 0.35\,\frac{y}{B} \qquad \text{........(7.18)}$$

$$d_\gamma = 1 \qquad \text{........(7.19)}$$

and the coefficients $N_c$, $N_q$ and $N_\gamma$ are given by the following expressions:

$$N_q = e^{\pi \tan\phi} \tan^2\left(45 + \frac{\phi}{2}\right) \qquad \text{........(7.20)}$$

$$N_c = (N_q - 1)\cot\phi \qquad \text{........(7.21)}$$

$$N_\gamma = 1.8\,(N_q - 1)\tan\phi \qquad \text{......(7.22)}$$

where the symbol e = 2.7183.

The maximum soil pressure developed at the wall foundation is given by

$$\sigma_{max} = \frac{N}{B}\left(1 + \frac{6e}{B}\right) \qquad \text{........(7.23)}$$

The allowable soil bearing stress which must be lower than the maximum pressure is expressed as

$$\sigma_{allowable} = \frac{p_{lim}}{3} \qquad \text{........(7.24)}$$

and for cohesionless soils

$$\sigma_{allowable} = \frac{p_{lim}}{2} \qquad \text{........(7.25)}$$

When only a portion of the base section is in compression (eccentricity, $e > \frac{B}{6}$ ) the soil stress can be evaluated as

$$\sigma_{max} = \frac{2N}{3u} \qquad \text{........(7.26)}$$

where

$$u = \frac{B}{2} - e \qquad \text{........(7.27)}$$

which indicates the distance from the normal force to the element under stress.

7.4.4.5 Shear stress

The shear stress developed is expressed as

$$\tau = \frac{T}{B} \qquad \text{..........(7.28)}$$

and the allowable shear stress is given by

$$\tau \leq \sigma \tan\phi^{*} + c_g \qquad \text{........(7.29)}$$

where

T = the tangential force ,

$\sigma$ = average normal stress at the section under consideration,

$\phi^*$ = internal 'fictitious' angle of friction of the aggregate and is given by

$$\phi^* = 25\ \gamma_g - 10° \qquad \text{........(7.30)}$$

in which $\gamma_g$ is the density of the filled gabions ($t/m^3$),

and

$$c_g = 0.03\ P_u - 0.05 \qquad \text{........(7.31)}$$

where

$P_u$ = weight of the metallic mix per cubic meter of the wall expressed in $kg/cm^3$ and $c_g$ is expressed in $kg/cm^2$.

## 7.5. INTERACTIVE MICROCOMPUTER BASED 'GABION' PROGRAM IMPLEMENTATION

### 7.5.1 Program Structure

The program GABION provides the design of the gabion retaining wall based on the lateral forces imposed by the backfill, earth pressure due to surcharge, stability analyses considering sliding and overturning, overall stability, and foundation on bearing pressure. In designing the gabion wall the program gives the designer the following:

- the geometry of the cross-section of the gabion structure
- factors of safety against sliding and overturning
- foundation bearing pressure

and

- check for overall stability.

The flowchart of the program GABION is given in Fig. 7.11 and the listing of the program in Appendix E.

### 7.5.2. Input and Output

The input parameters given through an interactive system-user consultation consist of the following:

- height of the earth to be retained;
- slope of the backfill
- internal angle of friction of the soil;
- unit weight of the backfill;
- density of the gabion;
- angle of batter of the gabion face;
- surcharge load;
- cohesion factor.

The output file consists of the number of gabion layers and their corresponding widths, coefficients of active and passive pressures, factors of safety against sliding and overturning, maximum soil pressure and shear stress and check on overall stability.

### 7.7.5.3 Design Example

The design example of a gabion retaining wall is illustrated in this section. The system-user interactive consultation for the design input is

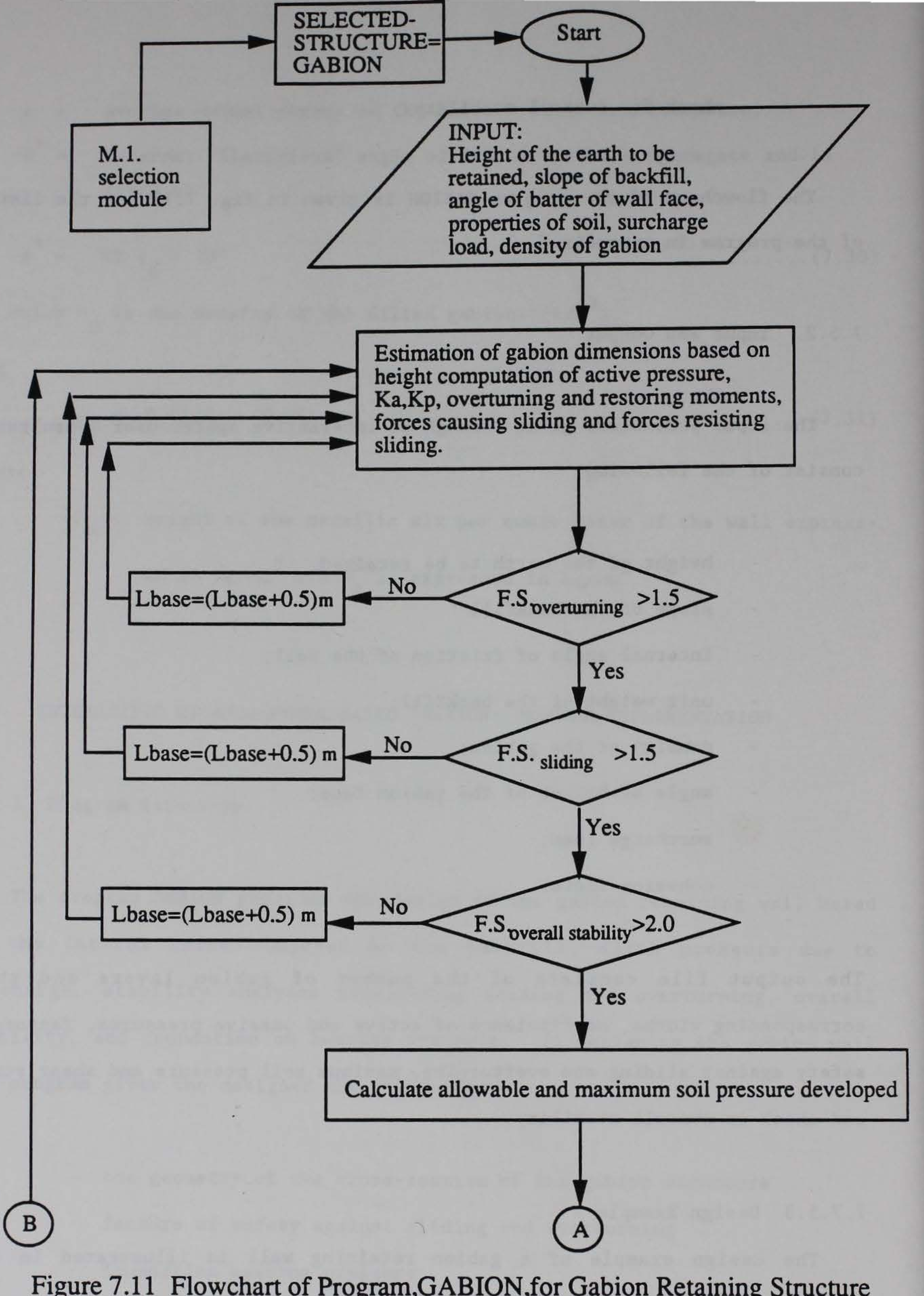



Figure 7.11 Flowchart of Program,GABION,for Gabion Retaining Structure Design(continued).

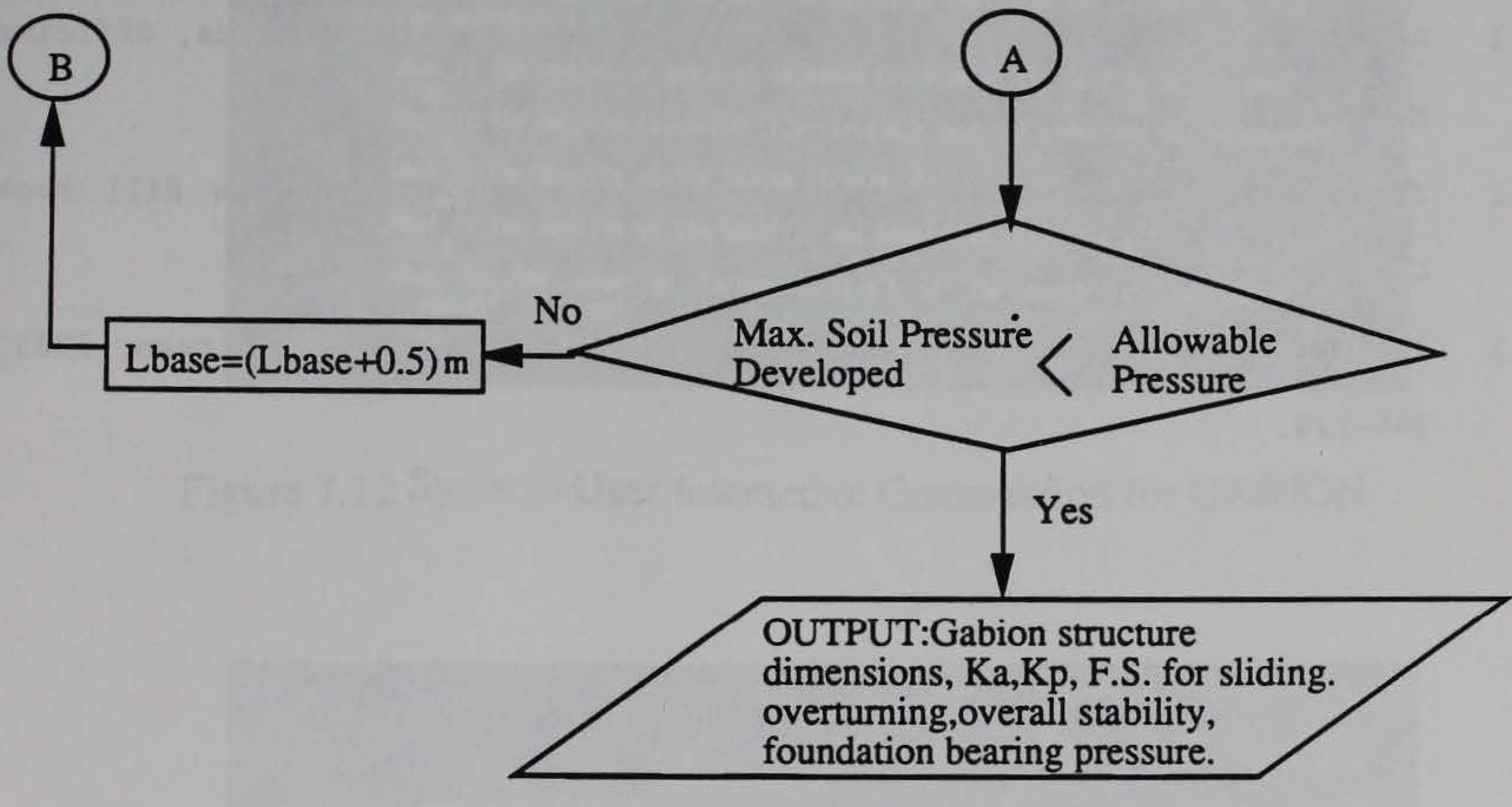



Figure7.11 Continued

shown in Fig. 7.12. Fig. 7.13 shows the input data as displayed on the screen. The output file for the design example as viewed on the screen is illustrated in Fig. 7.14.

**REFERENCES**


7.1 A Papetti, Flexible Gabion Structures in Earth Retaining Works, Officine Macaferri Sip. A., Bologna, Italy, 1987.

7.2 JE Bowles, Foundation Analysis and Design, 4th Ed., McGraw Hill Book Co., 1988, pp. 1007.

7.3 IK Lee, W. White and OG Ingles, Geotechnical Engineering, Pitman, 1983, 245-278.

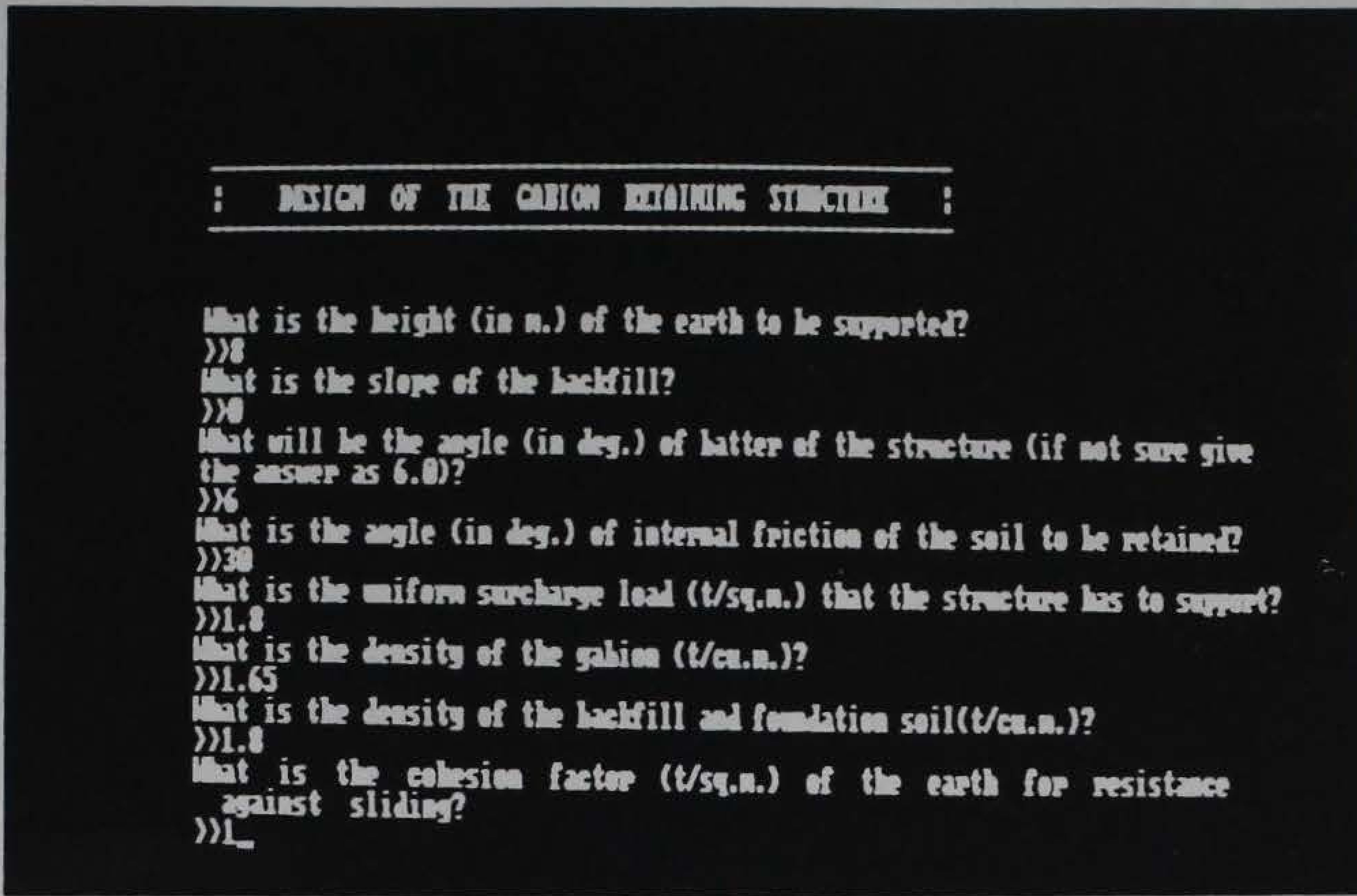



Figure 7.12 System -User Interactive Consultation for GABION

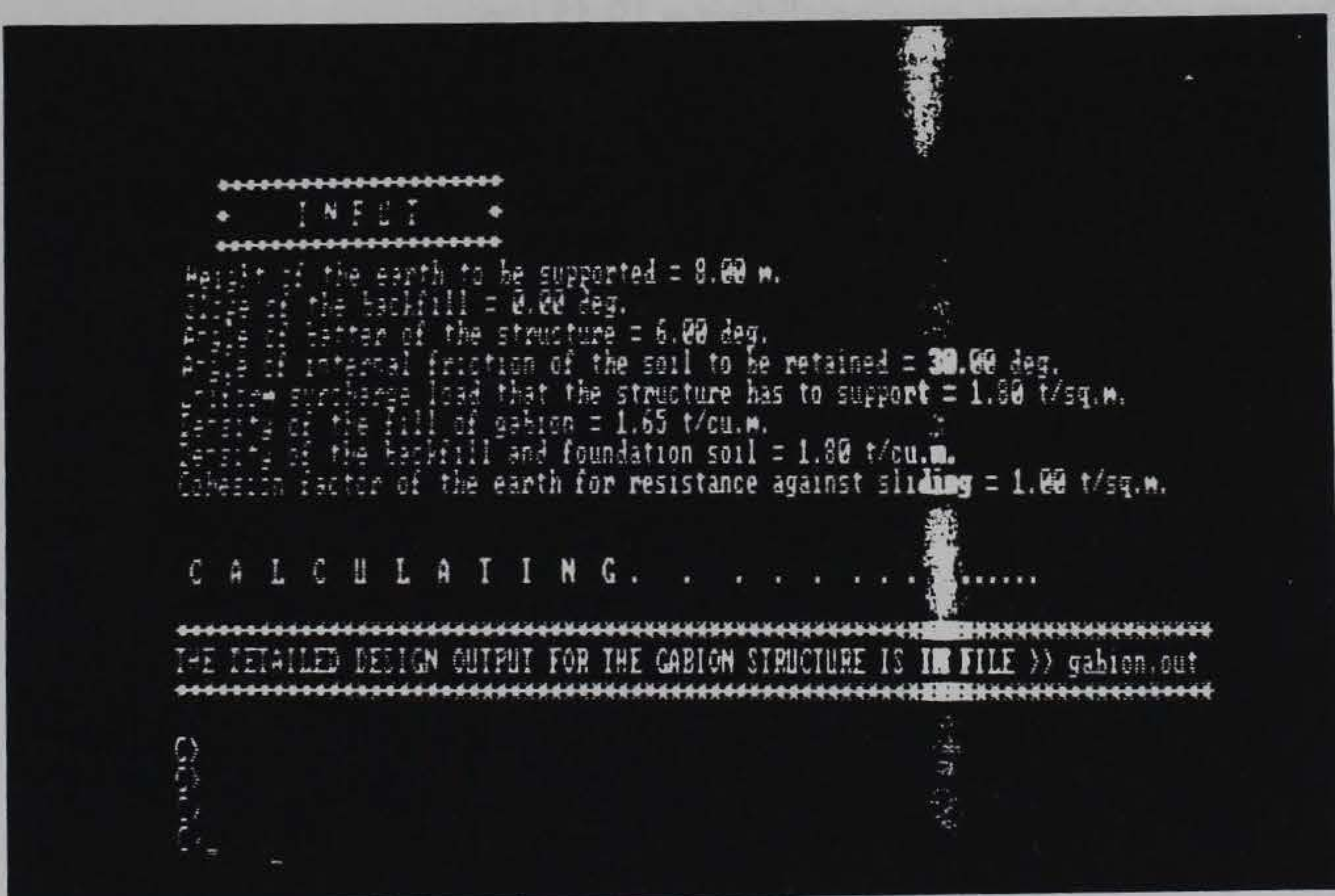



Figure 7.13 Screen Display of Input Data for GABION

| DESIGN OF THE GABION RETAINING STRUCTURE |

Number of gabion layers = 8.

| GABION # | WIDTH (m.) |
|---|---|
| 1 | 1.00 |
| 2 | 1.00 |
| 3 | 1.50 |
| 4 | 1.50 |
| 5 | 2.00 |
| 6 | 2.00 |
| 7 | 2.50 |
| 8 | 4.50 |

The position of the C.G. from the leftside of the bottom gabion = 2.00 m.
The position of the C.G. from the topside of the bottom gabion = 1.94 m.

The height of each gabion = 1.0 m.
Active earth pressure = 22.46 t/m.
Coefficient of active pressure,Ka = 0.2972
Coefficient of passive pressure,Kp = 10.0951

Factor of safety against sliding

The sum of the forces causing sliding = 19.45 t/m.
The sum of the forces resisting sliding = 51.37 t/m.

The factor of safety against sliding = 2.64 ----- OK, since FS > 1.5

Factor of safety against overturning

Sum of overturning moment = -2.09 t-m./m.
Sum of restoring moment = 147.47 t-m./m.

Factor of safety against overturning is NOT calculated as there is no overturning moment.

Check on foundation bearing pressure

Allowable soil bearing stress = 44.00 t/sq.m.
Maximum soil pressure developed = 19.31 t/sq.m.

Figure 7.14 Output File of GABION

Design safe as the maximum soil pressure is less than the allowable soil bearing pressure.

-------------------------------
Check for overall stability
-------------------------------

Passive resistance developed by toe prism = 320.46 t/m.
Thrust on the vertical plane of toe prism = 38.15 t/m.

Factor of safety for overall stability = 8.40 --- OK (since F.S. > 2.0 )

---------------------------------------------------------------------------

Figure 7.14 Continued

# CHAPTER 8

# REINFORCED EARTH WALLS

## 8.1 INTRODUCTION

The mechanically reinforced earth structure (Fig. 8.1) uses the principle of introducing reinforcing elements into a granular backfill via mechanical means such as metal strips and rods, geotextile strips and sheets, or wire grids. Reinforced embankments and reinforced soil structures must be designed to ensure both internal and external stability. Internal stability requires that the reinforced soil structures be coherent and self-supporting under the action of its own weight and any externally applied forces.

Reinforced soil structures are subject to the same external design criteria as conventional retaining walls. The structure must be stable against sliding because of the lateral pressure of the soil retained by the structure; besides it must resist overturning about its toe, must be safe against foundation failure, and have overall slope stability. Classical methods of soil mechanics have been found satisfactory for analysis of the external stability of reinforced soil structures.

## 8.2 EARTH REINFORCEMENT SYSTEMS

An earth reinforcement system has three major components: reinforcements, backfill or in situ ground and facing elements. The reinforcement materials can be both metallic and non-metallic while the reinforcement geometries can be broadly categorized as strips, grids, sheets, rods and fibers. The type of backfill is an important variable in determining the performance of the

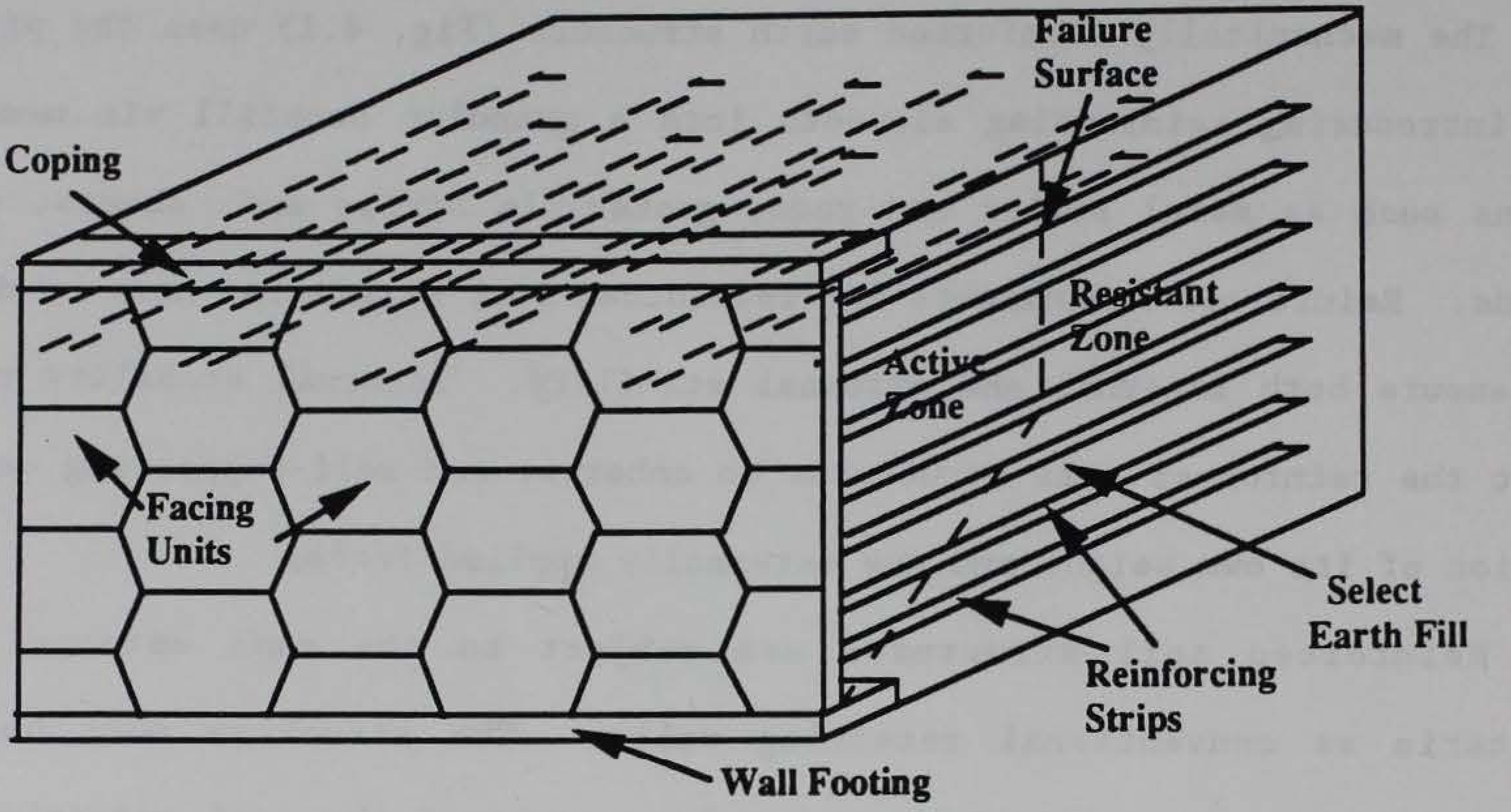



Figure 8.1 Schematic Diagram of a Reinforced Earth Structure Using Steel Strip Reinforcements.

composite reinforced soil structure. Facing elements are commonly provided to retain fill material at the face and prevent slumping and erosion of steep faces. Most common facing elements include precast concrete panels, prefabricated metal sheets and plates, gabions, welded wire mesh, shotcrete, inclusion of intermediate reinforcements between main reinforcement layers at the face, seeding of the exposed soil, and looping of geotextile reinforcements at the face [8.1].

### 8.2.1 Strip Reinforcement

The strip reinforcements, either metal or plastic, are normally placed in horizontal planes between successive lifts of soil backfill. The strip reinforcement system in the reinforced earth (Fig. 8.1) uses either ribbed or smooth prefabricated galvanized steel strips. Facing panels fastened to the strips consist usually of either precast concrete panels or prefabricated metal elements. Plastic strips have been introduced in an effort to avoid the problems of metal corrosion in adverse environments. The only commercially available nonmetallic strips are the Paraweb strip (Fig. 8.2) in which the fibers are made of high tenacity polyester or polyaramid. The strips are fastened to wall facings, typically consisting of precast concrete panels. Soil backfill is generally granular, ranging in size from sand to gravel.

### 8.2.2 Grid Reinforcement

Grid reinforcement systems consist of metallic or polymeric tensile resisting elements which are arranged in rectangular grids placed in horizontal planes in the backfill to resist outward movement of the

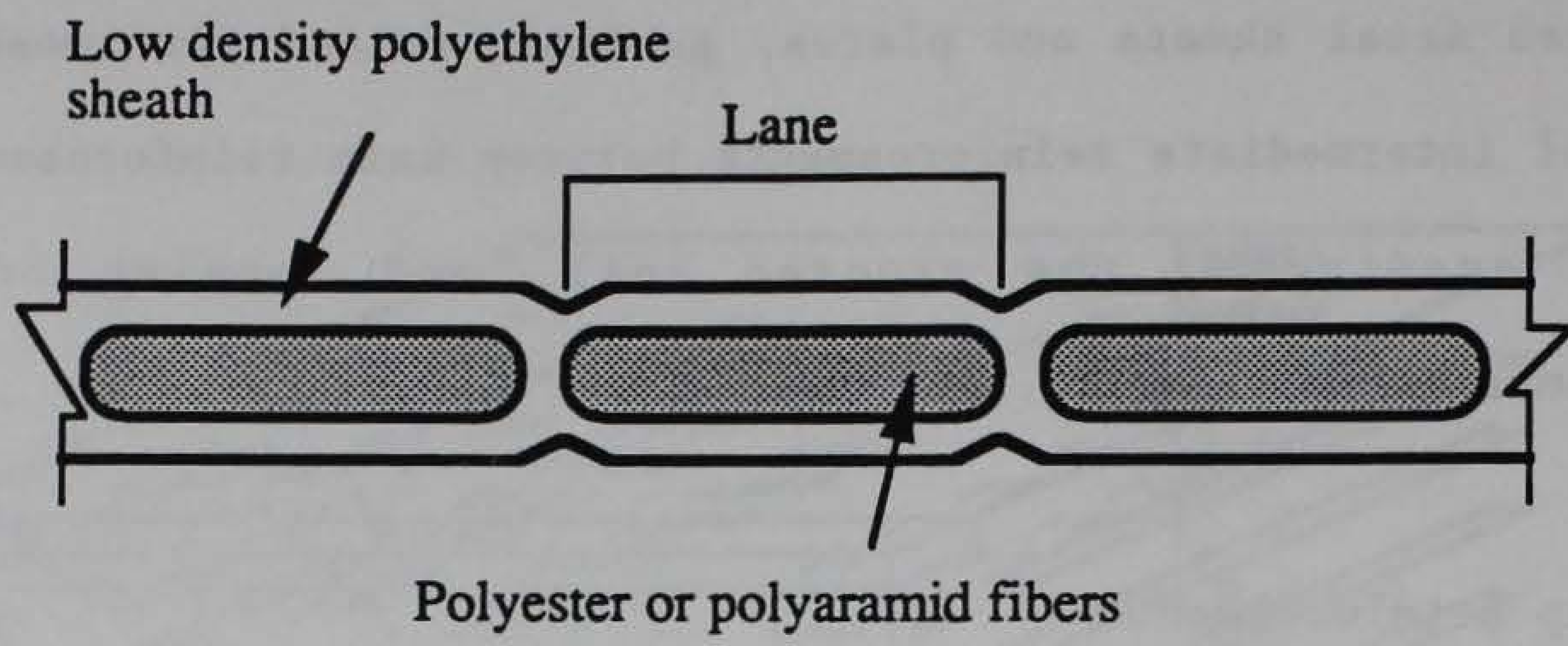



Figure 8.2 Cross-section of a Nonmetallic Paraweb Strip. [8.1]

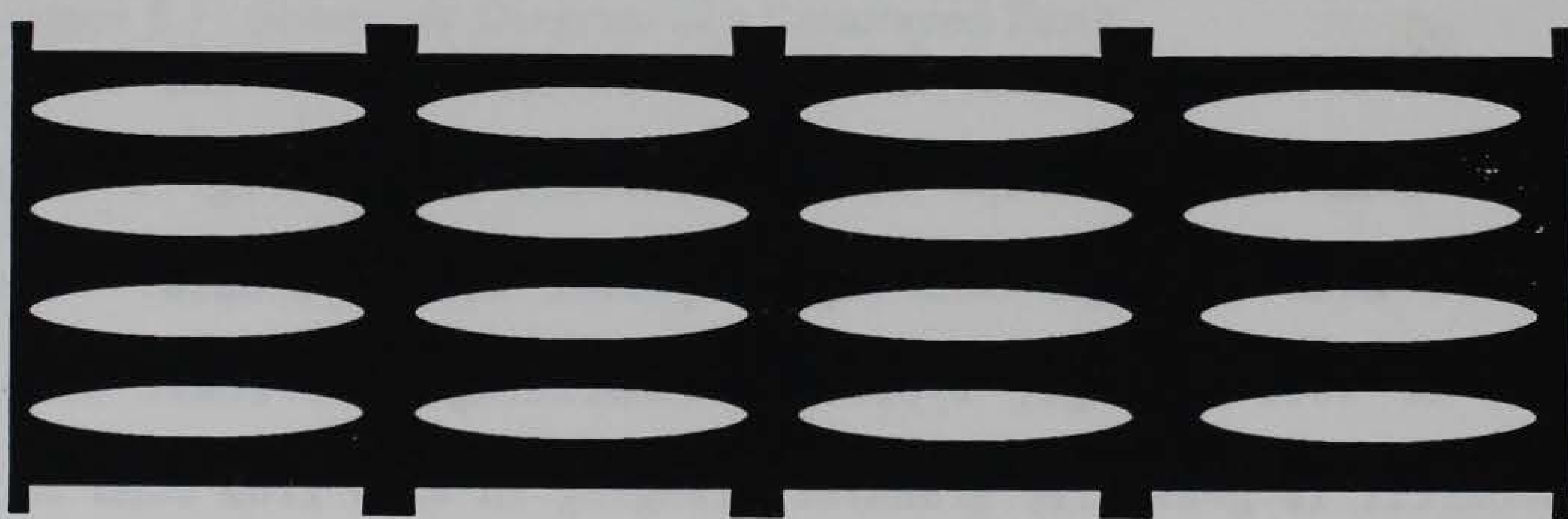


Figure 8.3 Plan View of a Geogrid Reinforcement. [8.1]

reinforcement soil mass. Grids transfer stress to the soil through passive soil resistance on transverse members of the grid and friction between the soil and horizontal surfaces of the grid. The Mechanically Stabilized Embankment (MSE) system employs prefabricated steel bar mat reinforcements at standard horizontal and vertical spacings and uses standard rectangular precast facing panels. The Welded Wire Wall and Reinforced Soil Embankment (RSE) systems employ standard welded wire mesh grid reinforcements within the backfill to constitute a reinforced soil structure. The two systems differ, however, in the facing arrangements. Grid reinforcement made of stable polymer materials may provide good resistance to deterioration in adverse soil and groundwater environments. Tensar Geogrids (Fig. 8.3) are high strength polymer grid reinforcements manufactured from high density polyethylene or polypropylene using a stretching process.

### 8.2.3 Sheet Reinforcement

Figure 8.4 shows continuous sheets of geotextiles laid down alternately with horizontal layers of soil to form a composite reinforced soil material. The mechanism of stress transfer between soil and sheet reinforcement is predominantly friction. Generally the geotextile fabrics used in soil reinforcement are made of polyester fibers. The backfill material typically consists of granular soil ranging from silty sand to gravel. Connection between the geotextile sheet and structural wall elements can be provided by casting the geotextile into the concrete, friction, nailing, overlapping, or other bonding methods.

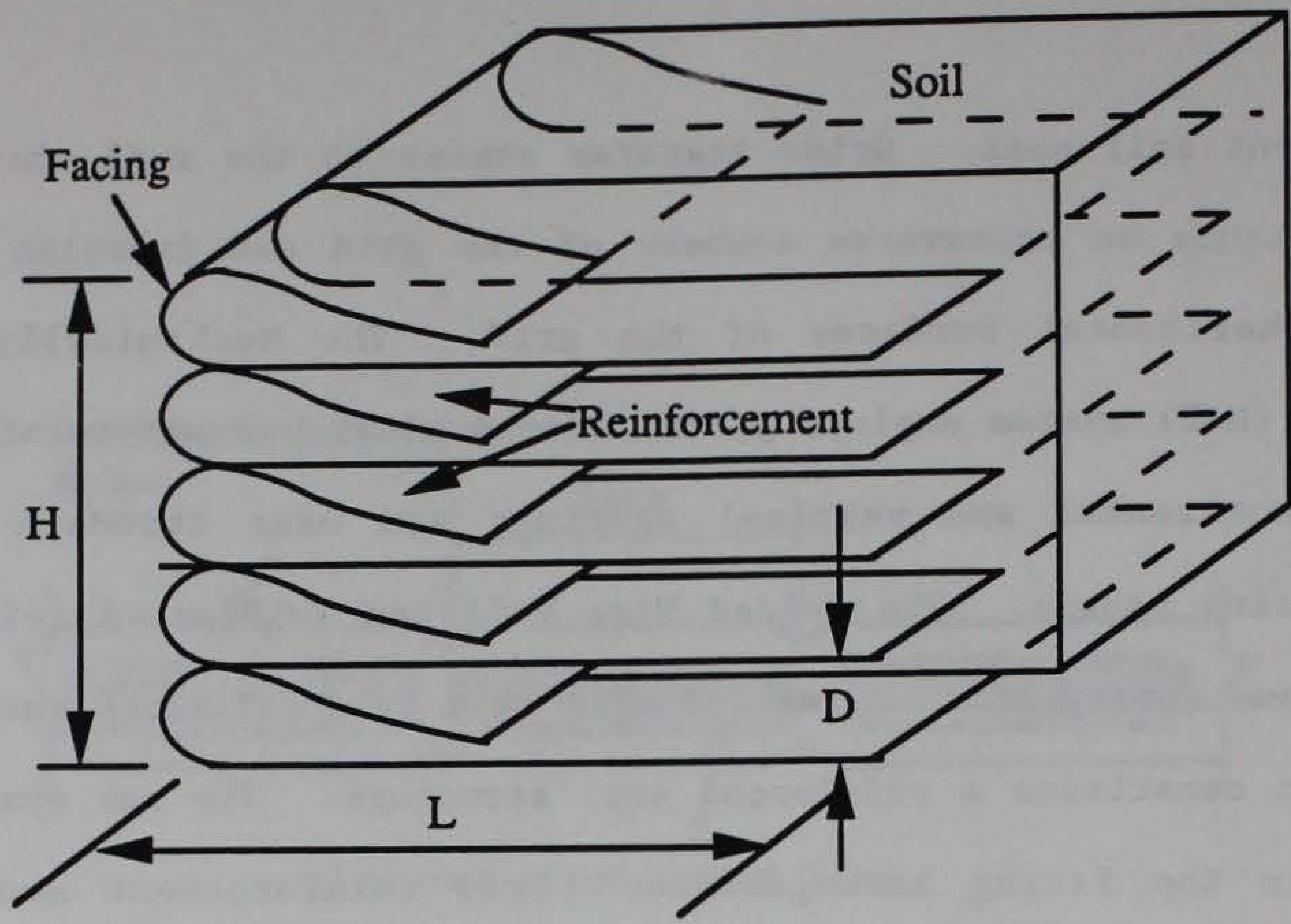



Figure 8.4 Schematic Diagram of a Reinforced Soil Wall Using Geotextile Sheet Reinforcements. [8.1]

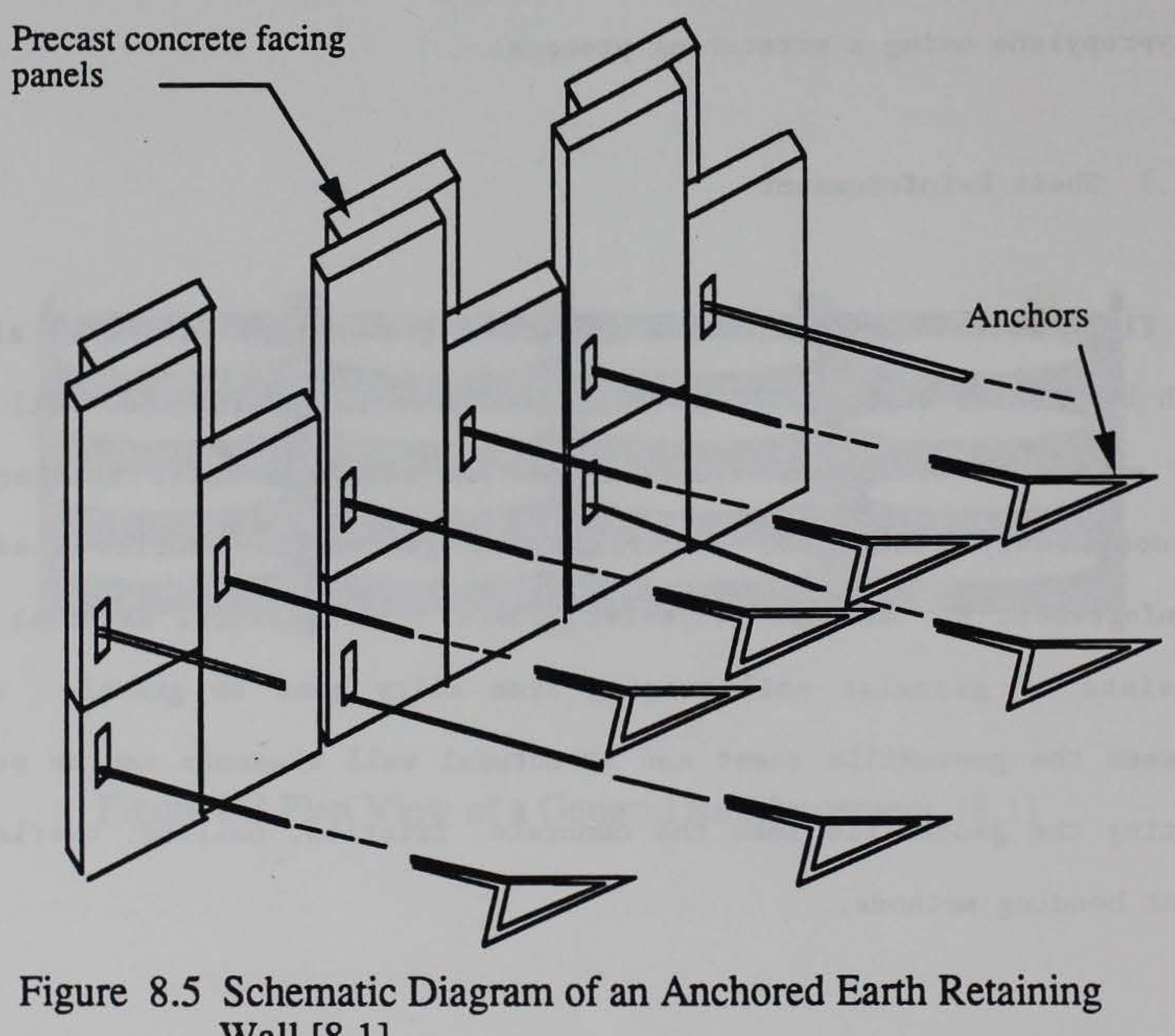



Figure 8.5 Schematic Diagram of an Anchored Earth Retaining Wall.[8.1]

### 8.2.4 Rod Reinforcement

Anchored Earth employs slender steel rod reinforcements bent at one end to form anchors (Fig. 8.5). Soil-to-reinforcement-stress transfer is primarily through passive resistance. This assumption employs that the system operates similar to tied-back retaining structures. Although Anchored Earth is not truly a reinforced soil system, friction should be developed along the length of the linear rod.

### 8.2.5 Fiber Reinforcement

Fibers can potentially provide reinforcement in three directions. The major limitation to use these fibers is the difficulty associated with efficient and economical mixing of the fibers uniformly into the backfill. The mixing process must be perfected before fiber inclusion can be used as an economically viable and routinely used reinforcement method. Materials for possible use include natural fibers (reeds and other plants), synthetic fibers (geotextile threads), and metallic fibers (small diameter metal threads).

## 8.3 DESIGN METHODOLOGY

### 8.3.1 Design Approaches

#### 8.3.1.1. Design based on at-failure conditions

Limit equilibrium analysis deals with the stability of a structure at incipient failure. The first type of limit analysis is based on the failure mechanism observed in laboratory models. The shape of the failure surface has conformed closely to a logarithmic spiral, and it coincides with locus of maximum tensile forces in the reinforcements. Failure is assumed to be caused by progressive rupture of reinforcing strips. Full shear resistance of the soil is assumed to be mobilized along the failure surface. In the second type of limit analysis, the shearing, tensile, and pullout resistance of the reinforcement are considered when the reinforcement crosses a potential sliding surface. The possible potential sliding surface could be planes, wedges, circles, logarithmic spirals and non-circular shapes.

8.3.1.1.1 Single-plane failure surfaces

The normal, N and tangential, T components of the force in the reinforcement are, added to the fully mobilized soil shear strength (Fig. 8.6a). Force in the reinforcement required to provide equilibrium is calculated for different slip plane angles [8.2]. The largest required force for maintaining equilibrium is determined together with the corresponding slip plane angle. This force is compared with the lesser of the reinforcement strength or the mobilized bond force to determine the factor of safety of the system.

8.3.1.1.2 Infinite slope failure surfaces

The tangential component of the force in the reinforcement is neglected in the infinite slope calculation procedure for analysis of the stability of long shallow slopes reinforced with tensile members. The normal component, N (Fig. 8.6a) is added to an available resisting force contributed by the soil shear

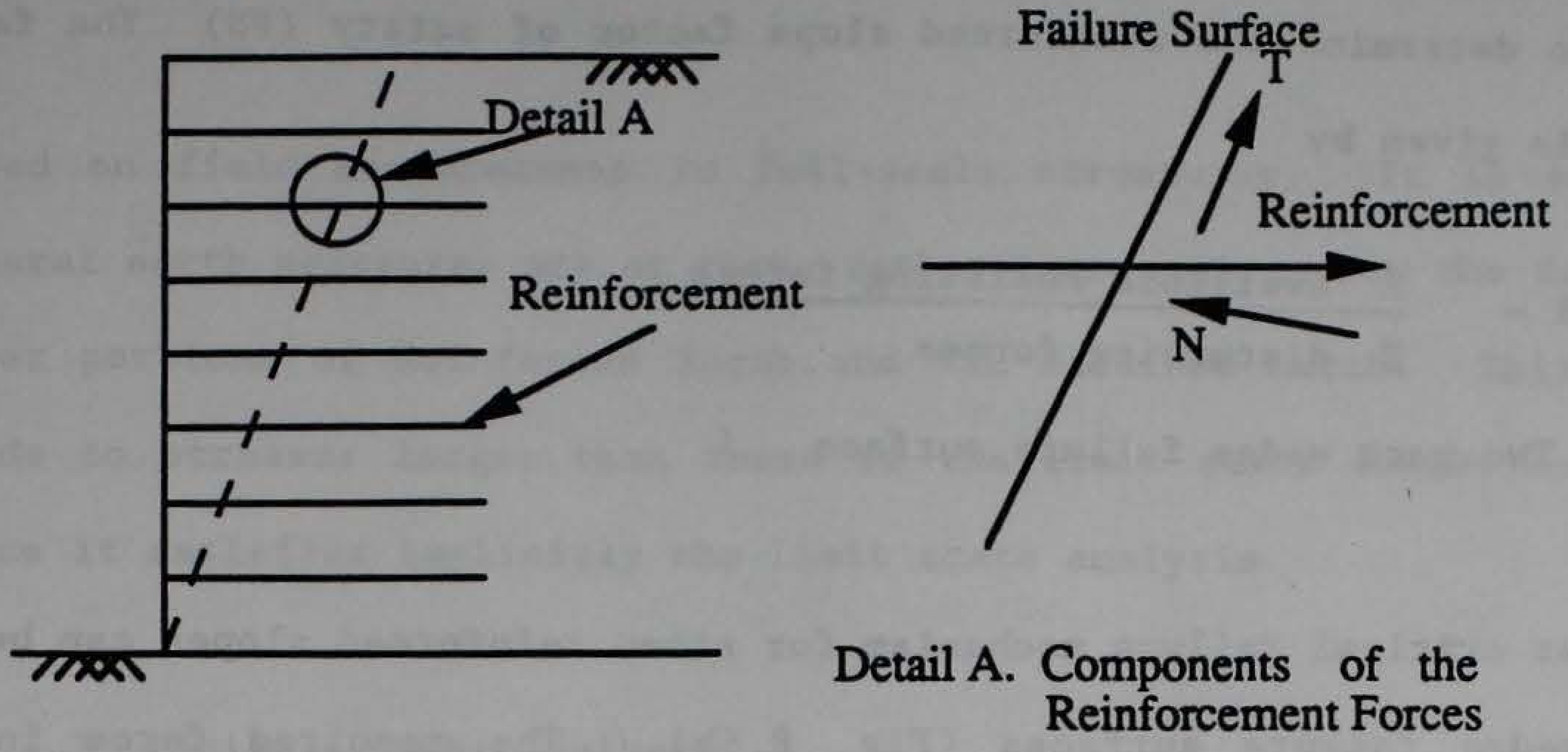



Figure 8.6a Single-plane Failure Surface (Limit Equilibrium Analysis).[8.1]

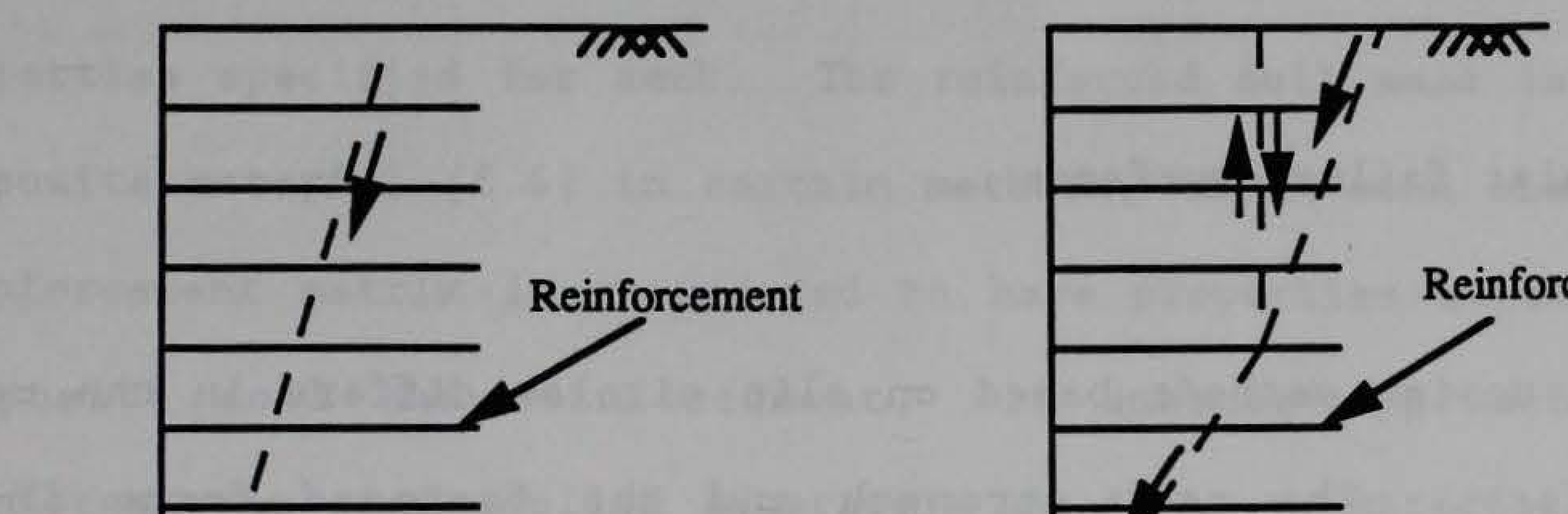



Figure 8.6b Toe-part Wedge Failure Surface. [8.1]

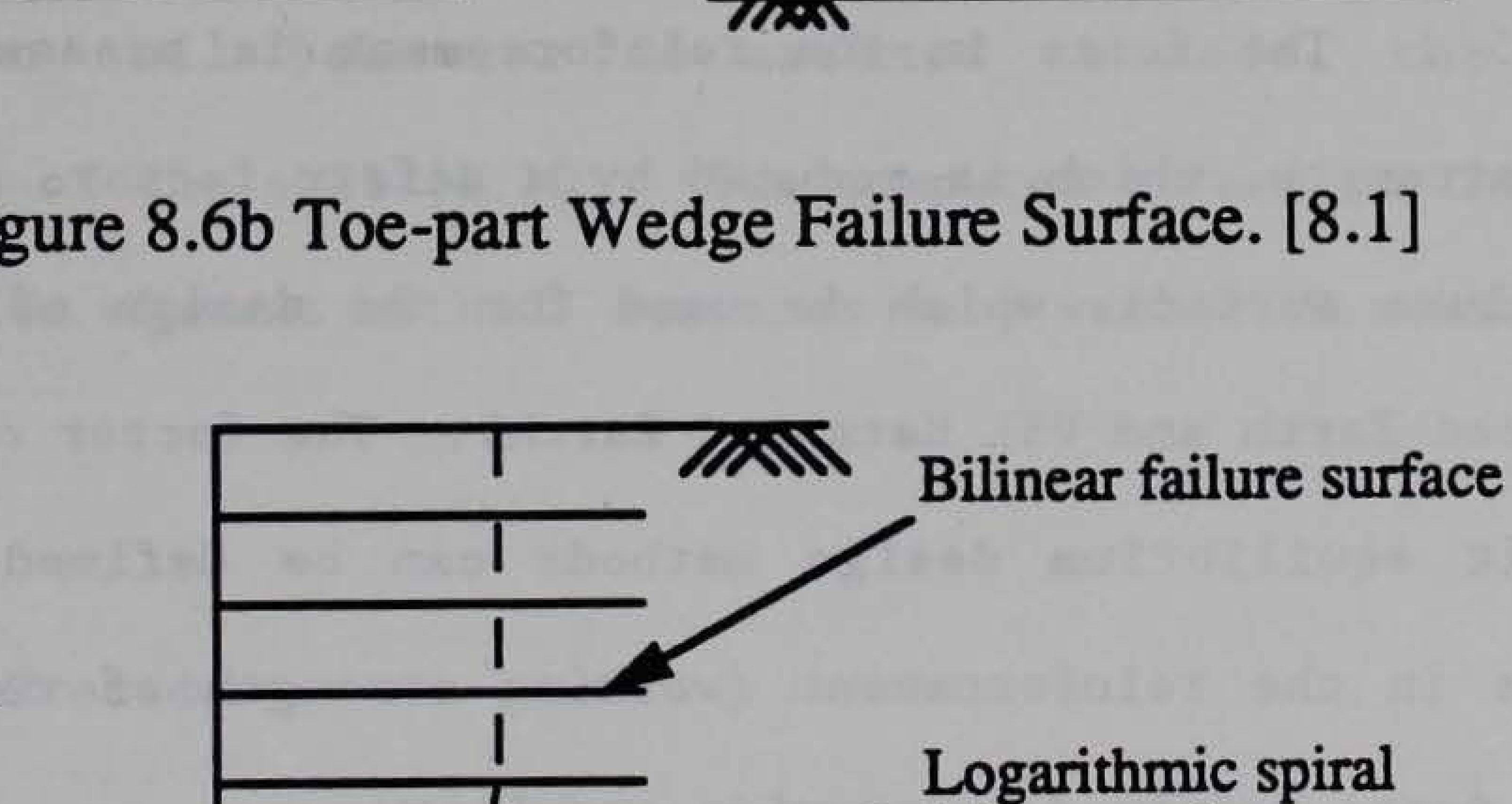



Figure 8.6c Logarithmic Spiral Failure Surface. [8.1]

strength to determine the reinforced slope factor of safety (FS) The factor of safety is given by

$$FS = \frac{\Sigma \text{ available resisting forces}}{\Sigma \text{ disturbing forces}}$$

8.3.1.1.3 Two-part wedge failure surface

The most critical failure mechanism for steep reinforced slopes can be the two-part wedge failure surfaces (Fig. 8.6b). The required force in the reinforcement for equilibrium is calculated [8.3 and 8.4] assuming full soil shear strength mobilization and compared with the maximum available force to obtain the factor of safety.

8.3.1.1.4 Circular failure surfaces

A number of design methods based on slip circles differs in the use of factors of safety. The soil strength and the factored force in the reinforcement are used to find the equilibrium safety factor for the embankment [8.5]. The force in the reinforcement is assessed based on the reinforcement strength, which is reduced by a safety factor. Fig. 8.6c shows a bilinear failure surface, which is used for the design of certain systems (e.g., Reinforced Earth and VSL Retained Earth). The factor of safety for the different limit equilibrium design methods can be defined as the maximum available force in the reinforcement (working strength of the reinforcement) divided by the force required to provide stability.

8.3.1.2 Working stress considerations

Design assumptions of the magnitudes and internal stress distributions are based on field measurements in full-scale structures. It is assumed that lateral earth pressures are at rest, rather than active, for the design of the upper portions of Reinforced Earth and VSL Retained Earth. This assumption leads to stresses larger than those of the limit state active condition and hence it satisfies implicitly the limit state analysis.

8.3.1.3 Finite element analysis

Different types of finite element analyses available for reinforced soil structures involve the subdivision of the soil into discrete elements, with properties specified for each. The reinforced soil mass is considered as a composite material [8.6] in certain methods, where each element in the soil reinforcement matrix is considered to have properties representative of the composite soil and reinforcement. In another approach, the soil and reinforcements are modelled separately as discrete materials [8.7 and 8.8]. The soil-reinforcement interaction is modelled using interface elements [8.9] and hyperbolic soil behavior is assumed to model the stress-strain characteristics of the soil [8.10]. The computed stress in the reinforcements using this finite element method, compares very well with measured values from the field.

8.3.2. Design Procedure

8.3.2.1 Site conditions and engineering properties of embankment material or in situ ground

Settlement analysis should be performed to determine anticipated wall settlements using measured or estimated soil properties. The foundation soil must be able to support the wall weight and any surcharge loading without a bearing capacity failure. The external stability conditions of the wall must be satisfied before a site can be considered acceptable for an earth wall.

The backfill material must satisfy certain minimum requirements to ensure adequate soil-to-reinforcement stress transfer. The grain size distribution, plasticity index, friction angle, and soil cohesion are often used to evaluate the reinforcement pullout resistance. The reinforcing system durability depends directly on the properties of the backfill. The change in pH, resistivity, chlorides, sulphates, grain size distribution, plasticity index, friction angle, cohesion, and unit weight must be assessed for the internal stability of an earth wall during the expected structure life. The external stability of an earth wall depends on the engineering properties of the in situ walls and the system geometry.

8.3.2.2 External stability

The reinforced soil mass in an earth wall should withstand the external loads, the horizontal earth pressure from the soil being retained behind the wall and loads applied to the top of the wall without failure by one of the following mechanisms:

i) sliding along the wall base or along any plane above the base;

ii overturning about the wall toe;

iii) bearing capacity failure of the foundation soil;

and

iv) general slope stability failure.

Fig. 8.7 shows the external loading of a typical earth wall.

8.3.2.2.1 Sliding failure along the wall base

The shear strength of the backfill material and foundation soil must be large enough to withstand the horizontal stresses applied to the reinforced soil mass from the retained soil and any additional live loads. Using a value of 1.5 for the factor of safety of an earth wall against sliding, the reinforcement length required for stability against sliding for a vertically faced wall with surcharge loading is determined for granular backfills and given by

$$L = \frac{1.5 K_a (q + (\gamma H/2))}{\gamma \tan\phi} \qquad \text{.........(8.1)}$$

where

$L$ = reinforcement length,

$K_a$ = active earth pressure coefficient

= $\tan^2 (45 - \phi_1/2)$,

$q$ = surcharge load,

$\gamma$ = unit soil weight,

$\phi$ = internal friction angle of foundation soil or backfill, whichever is smaller,

and

$\phi_1$ = internal friction angle of the soil retained by the wall.

For cohesive foundation soils the effective cohesion of the soil could be added to the denominator of Eqn. 8.1, if it can be relied on over sustained

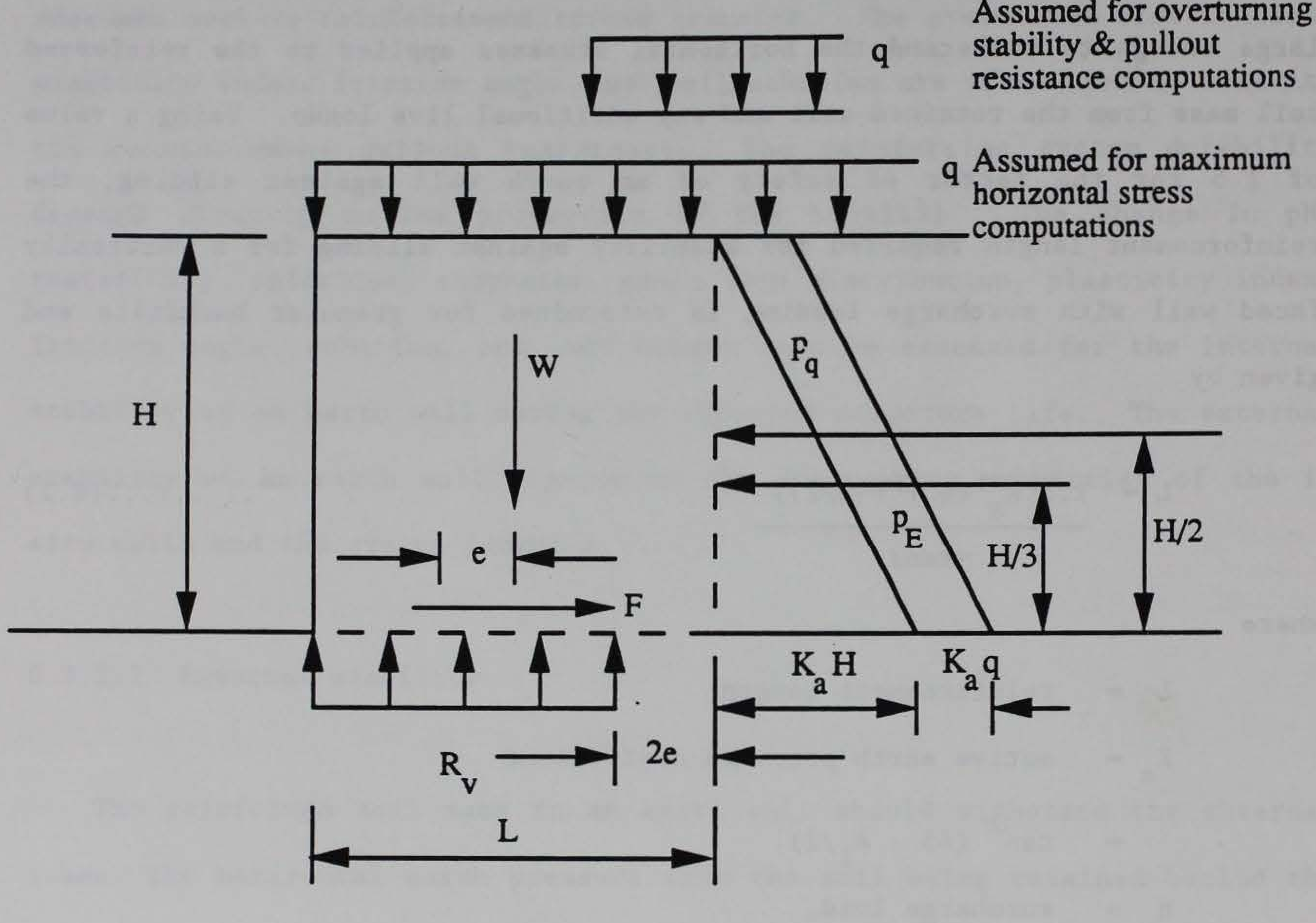



Figure 8.7 External Stability of an Earth Wall.[8.1]

time periods.

Traffic loads which are treated as uniform surcharge loads are assumed to act behind the reinforced soil mass for stability calculations and on top of the reinforced soil mass for maximum horizontal stress calculations (Fig. 8.7).

**Failure due to overturning about the wall toe**

The reinforced soil walls are designed to have a factor of safety of at least 2.0 with respect to overturning and the factor of safety (FS) is given as

$$FS \leq \frac{\Sigma M}{\Sigma M_D} \qquad \text{........(8.2)}$$

where

$\Sigma M_R$ - sum of the resisting moments,

and

$\Sigma M_D$ - sum of the driving moments.

For the case shown in Fig. 8.7,

$$\Sigma M_R = \frac{WL}{2} - \frac{\gamma H L^2}{2} \qquad \text{.........(8.3)}$$

and

$$\Sigma M_D = \frac{P_q H}{2} + \frac{P_E H}{3} \qquad \text{.........(8.4)}$$

where

$P_q$ - resultant horiztontal earth force from surcharge loading,

$P_E$ - resultant horizontal earth force from soil retained by earth wall,

and

$W$ - weight of reinforced soil mass.

**Bearing capacity failure**

The bearing capacity of the foundation soil must be checked to ensure that the vertical load exerted by the weight of the wall and surcharge is not excessive. The minimum factor of safety against this type of failure for reinforced soil walls is 2.0 and this is lower than that used for conventional reinforced concrete retaining walls because reinforced soil walls are flexible and able to function satisfactorily even after experiencing large differential settlements. This distribution of stress under a reinforced soil wall is represented using Meyerhof's stress distribution assumption of eccentrically loaded footings (Fig. 8.8).

The eccentricity, e, which is determined by setting the sum of the moments about the centerline of the reinforced soil mass equal to zero is given by (Fig. 8.7),

$$e = \frac{\Sigma M_D}{R_v} = \frac{P_q\ (H/2) + P_E\ (H/3)}{R_v} \quad .........(8.5)$$

where

$R_v$ - vertical reaction

- $W + q\ L$

The eccentricity, e, should be less than 1/6 the length of the

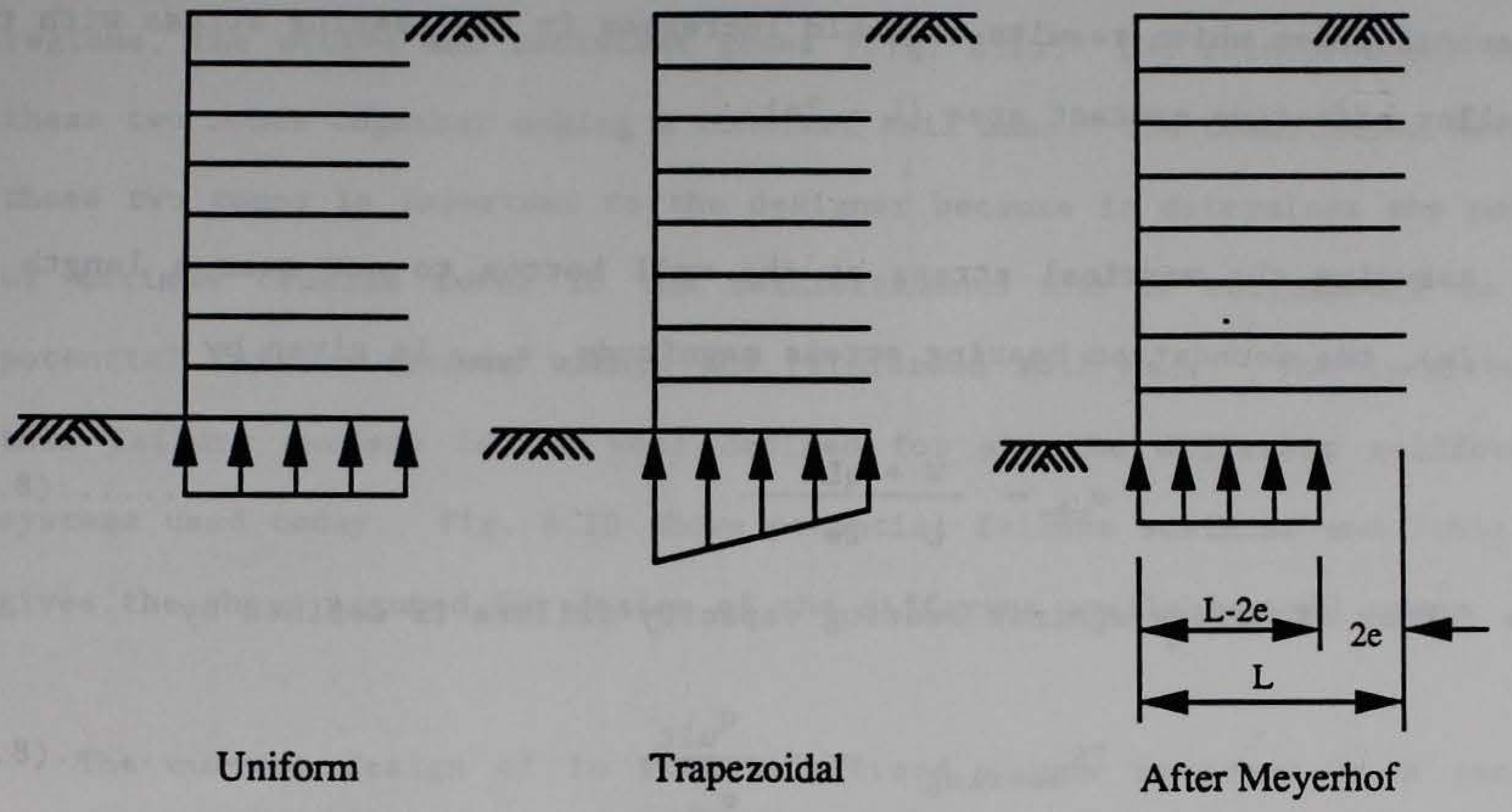



Figure 8.8 Stress Distribution Under an Earth Wall. [8.1]

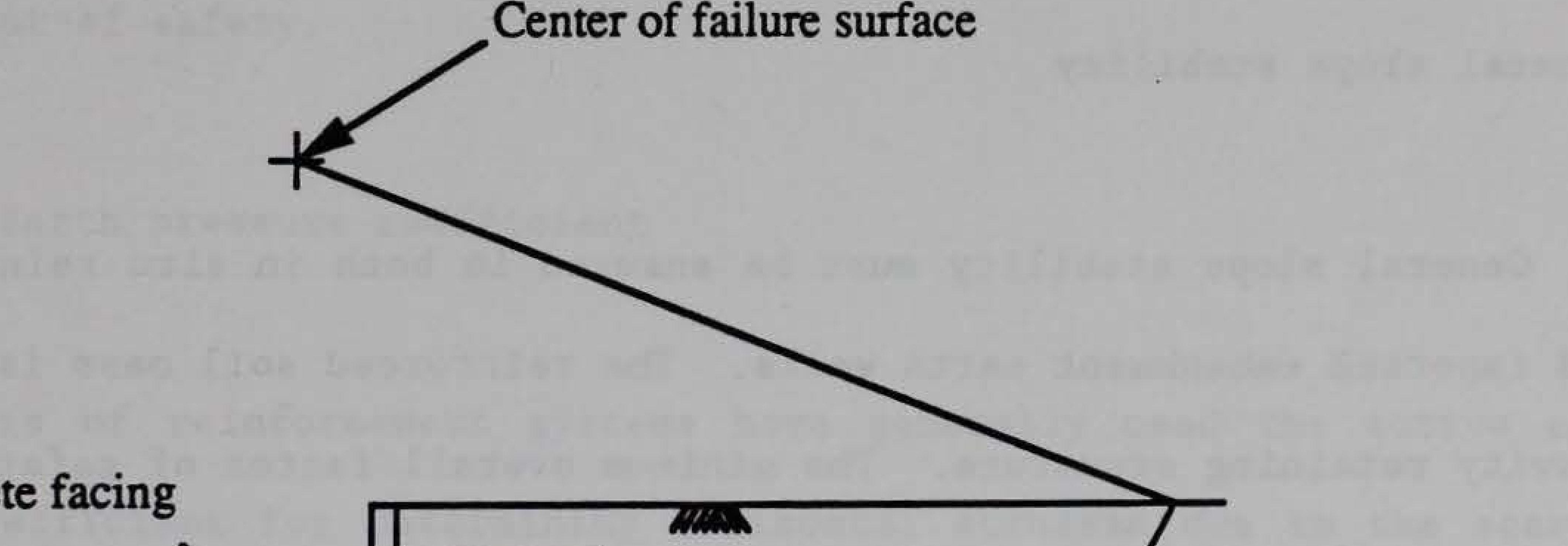



Figure 8.9 General Slope Stability of an Earth Wall. [8.1]

reinforcement, L when using Meyerhof's stress distribution. Meyerhof's assumption of a uniform stress distribution is less reasonable for greater eccentricities which results in rapid increases in the bearing stress with the smaller effective contact area (L - 2e).

Assuming the vertical stress at the wall bottom to act over a length of (L -2e), the foundation bearing stress magnitude, $\sigma_{vb}$, is given by

$$\sigma_{vb} = \frac{W + qL}{L - 2e} \qquad \text{.........(8.6)}$$

The factor of safety against bearing capacity failure is defined by

$$FS_{bearing} = \frac{q_{ult}}{\sigma_{vb}} \qquad \text{.........(8.7)}$$

where $q_{ult}$ = ultimate bearing capacity. The ultimate bearing capacity of the foundation soil is evaluated using classical soil mechanics methods.

**General slope stability**

General slope stability must be ensured in both in situ reinforced slopes and imported embankment earth walls. The reinforced soil mass is treated as a gravity retaining structure. The minimum overall factor of safety for general slope stability is determined using any appropriate slope stability analysis (Fig. 8.9).

### 8.3.2.3. Internal stability

#### 8.3.2.3.1 Failure surface

The reinforced soil mass in both imported embankment earth walls (vertical or sloping face) and in situ reinforced slopes can be divided into two regions, the active and resistant zones (Fig. 8.1). The reinforcement holds these two zones together making a coherent soil mass. The demarcation between these two zones is important to the designer because it determines the points of maximum tensile force in the reinforcements and it corresponds to the potential failure surface within the reinforced soil mass. The location of this failure surface is not well defined for all the different reinforcing systems used today. Fig. 8.10 shows potential failure surfaces and Table 8.1 gives the shear assumed for design of the different available wall types.

The current design of in situ reinforced slopes is based on a general stability analysis which considers potential failure surfaces which cross the reinforcements (Fig. 8.10). Numerous failure surfaces are analyzed and the potential failure surface is then defined as that surface which gives the lowest factor of safety.

8.3.2.3.2 Earth pressure coefficient

Designers of reinforcement systems have generally used the active earth pressure coefficient for determining horizontal stresses due to the scarcity of a large database of field instrumented walls. In Reinforced Earth and VSL Retained Earth systems a linear variation is assumed for the earth pressure coefficient, $K_o$ (coefficient of earth pressure at rest) at the top of the wall to $K_a$ (active earth pressure coefficient) at some depth (e.g., 20 ft.) below the top of the wall. A value of $K = 0.65$ is used in Welded Wire Wall; this value is larger than the at-rest earth pressure coefficient for the soils generally used in earth wall construction (Fig. 8.11). Table 8.1 also shows

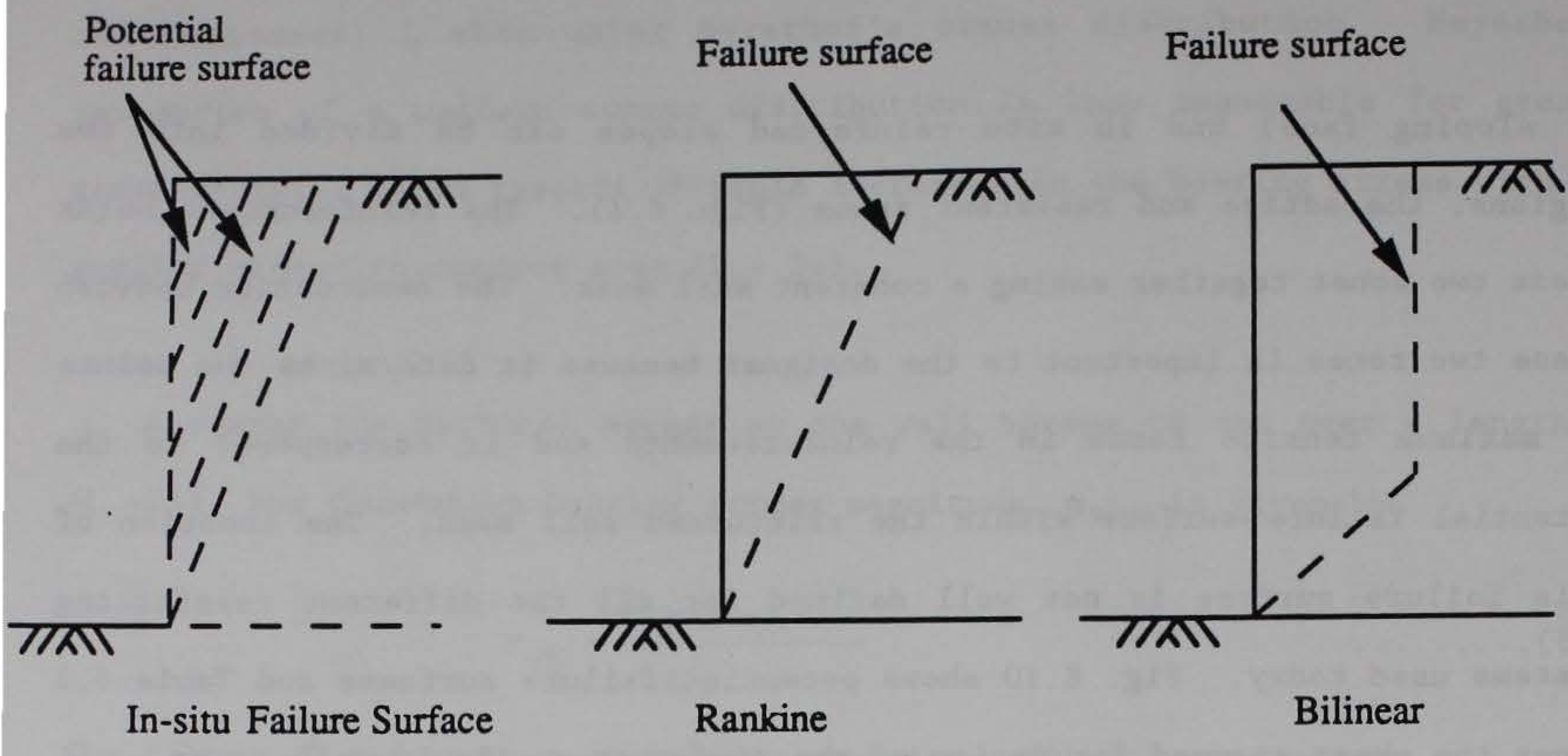



Figure 8.10 In-situ Reinforced Slope Potential Failure Surface Versus Rankine and Bilinear. [8.1]

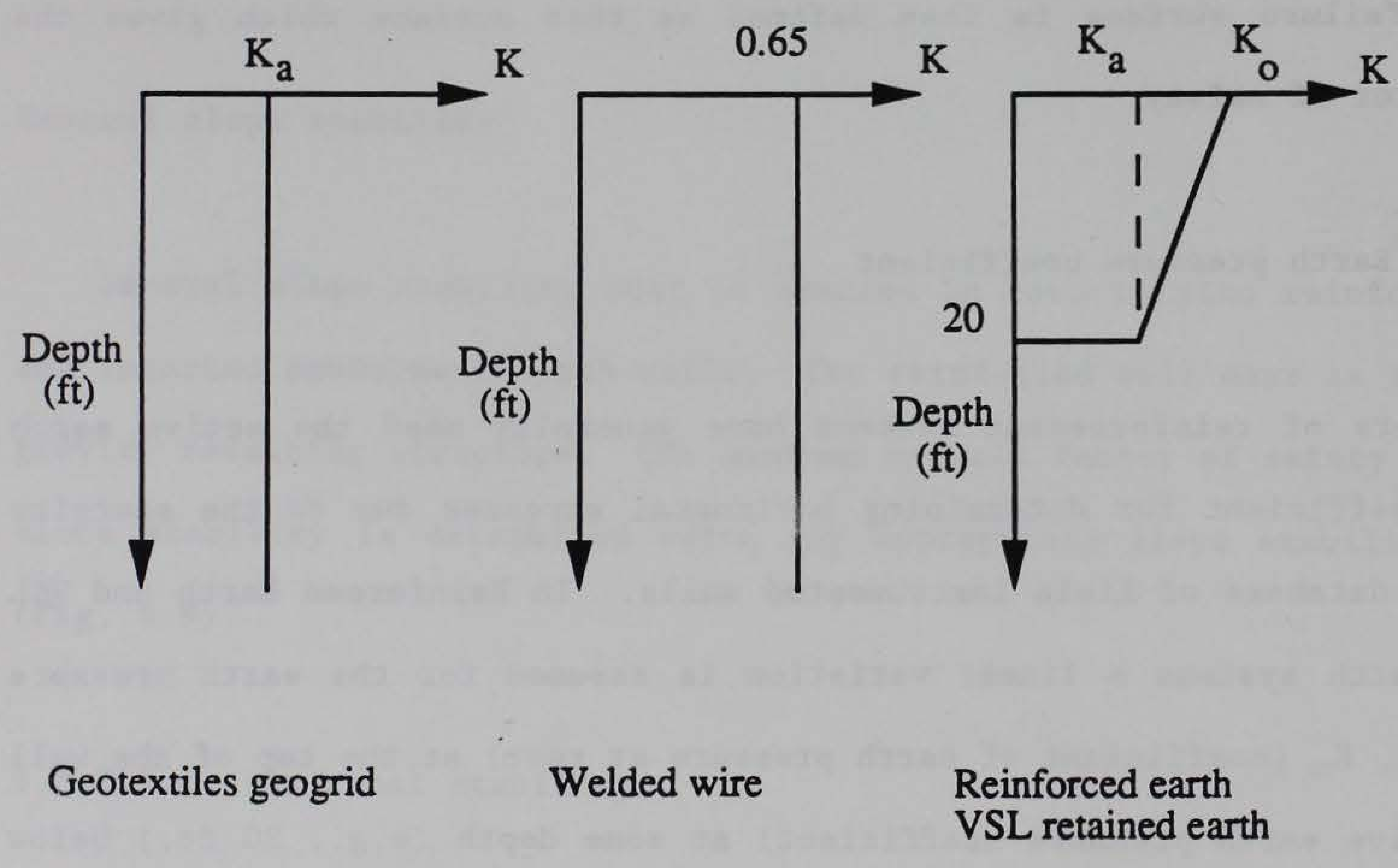



Figure 8.11 Assumed Variation in Earth Pressure Coefficient with Depth for Different Wall Types. [8.1]

Table 8.1. Internal Design Characteristics of Earth Walls

| Reinforcement Type | Trade Name | Failure Surface | | | Earth Pressure coefficient | | | | Durability | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Rankine | Bilinear | Wedge with varying angle from horizontal | $K_a$ | $K_o$ | $>K_o$ | Varying from $K_o$ at top of wall to $K_a$ 20 Ft below top | Reinforcement susceptible to corrosion | Degradation by ultraviolet light radiation |
| Strip Reinforcement | Reinforced Earth | | X | | | | | X | X | |
| | Plastics | | X | | | | | X | X | |
| Sheet Reinforcement | Geotextiles | X | | | | X | | | | X |
| Rod Reinforcement | Soil Nailing | | | X | X | | | | | |
| | Anchored Earth | | | X | X | | | | X | |
| Grid Reinforcement | VSL Retained Earth | | X | | | | | X | X | |
| | MSE, GASE | X | X | | X | | | | | |
| | Welded Wire Wall | X | | | | | X (0.65) | | X | |
| | Geogrid | X | | | X | | | | | X |

the earth pressure coefficient assumptions for design of the different reinforcing systems.

#### 8.3.2.3.3. Reinforcement rupture

The reinforcement system should withstand the stresses transferred to it from the soil without breaking. The horizontal stress, $\sigma_h$ at any depth within a wall is determined as,

$$\sigma_h = K \sigma_v \qquad \text{.........(8.8)}$$

where

$K$ = coefficient of horizontal earth pressure,

and

$\sigma_v$ = vertical stress at any depth

The vertical stress at any depth is determined by finding the eccentricity, e of the reactive force at that depth and then determining the vertical stress according to Meyerhof's stress distribution. The total horizontal load, FH to be resisted by a given reinforcement layer is expressed as

$$FH = \sigma_h \; S_H \; S_v \qquad \text{.........(8.9)}$$

where

$S_v$ = vertical spacing between reinforcements,

and

$S_H$ = center-to-center horizontal spacing between reinforcements.

The stress induced in the reinforcement from the horizontal load must be less than the allowable working stress of the tensile member to ensure adequate safety against rupture.

8.3.2.3.4 Pullout capacity

The pullout capacity of any reinforcement system can be determined based on the following:

i) analyses considering friction,

ii) analyses considering passive resistance,

and

iii) analyses considering both frictional and passive resistance.

The embedment length of the reinforcement, $L_e$ (i.e., the length of the reinforcement behind the potential failure surface-Fig. 8.1) must be large enough to ensure the transfer of stress from the reinforcement to the soil without reinforcement pullout. Table 8.2 gives the pullout equations used in the design of various reinforcing systems.

8.3.2.3.5 Durability

The most critical forms of degradation for earth structures result from corrosion and exposure to ultra violet light radiation. Reinforcements which are made from metal (Reinforced Earth, Anchored Earth, VSL Retained Earth, Welded Wire Walls, Soil Nailing, etc.) are susceptible to corrosion. The service life of these reinforcement systems can be extended by adding additional 'sacrificial' steel to the reinforcement members. The life of these materials can be extended by also using zinc galvanizing or epoxy coatings. Geotextile reinforcement is protected from prolonged exposure from ultraviolet light radiation by using asphalt and shotcrete as a protective coating. The long term resistance of geotextiles to chemical and biological

Table 8.2. Pullout Capacity Design Equations

| Reinforcement Type | Trade Name | Semi Empirical Equation for Pullout Capacity | | |
|---|---|---|---|---|
| | | Frictional | Passive | Frictional + Passive |
| Strip Reinforcement | Reinforced Earth | $P = \mu^* \gamma z L_e 2b$<br>$0.5 \le \mu^* \le 1.5$ | | |
| Sheet Reinforcement | Geotextiles | $P = \tan(2\phi/3)\gamma z L_e 2$ | | |
| Rod Reinforcement | Soil Nailing | $P = \pi d l_e (c+\gamma z t s n \phi z)$ | | |
| | Anchored Earth | | $P = \frac{K_p \gamma z \theta t_p}{\cos \alpha_1} e^{(2(3/4\pi - \alpha_1)\tan\phi')}$ | |
| Grid Reinforcement | VSL Retained Earth | | $P = A_c \gamma z d b n$<br>$15 \le A_c \le 40$ | |
| | Welded Wire Well | | | $P = (633+\gamma z d[\pi L_e M \tan\delta + 36.8n])$<br>for clean sands<br>$11° \le \delta \le 22°$ |
| | Tensar Geogrid | | | $P = L_e b \gamma z[(2\alpha_s \tan\delta) + \alpha_s + (\frac{b'}{\sigma_z} \frac{t}{s_x} \alpha_b)]$<br>$5 \le \frac{\sigma_b}{\sigma_z} \le 100$ |

attack is not known. Table 8.1 also gives the durability considerations pertinent to each type of reinforcement system.

### 8.4 INTERACTIVE MICROCOMPUTER BASED 'EARWALL' PROGRAM IMPLEMENTATION

#### 8.4.1 Program Structure

The program EARWALL provides the design of a typical reinforced earth retaining structure for the following four different cases of backfill and loading conditions:

i) wall with horizontal backfill,

ii) wall with sloping backfill,

iii) wall with uniform surcharge,

and

iv) wall with cohesive retained and foundation soils.

The principles of the general design methodology discussed in Section 8.3 has been used to formulate the interactive program, EARWALL. The user must provide the case pertaining to the problem in hand (one of the four cases mentioned above), the height of the earth to be retained, soil properties and reinforcing strip spacings. In designing the reinforced earth structure, the program gives an output which includes the reinforcing strip dimensions, the factors of safety against sliding and overturning and the soil bearing capacity. The program checks the internal stability requirements of the wall. The flowchart of the program EARWALL is shown in Fig. 8.12 and the listing of the program is given in Appendix F.

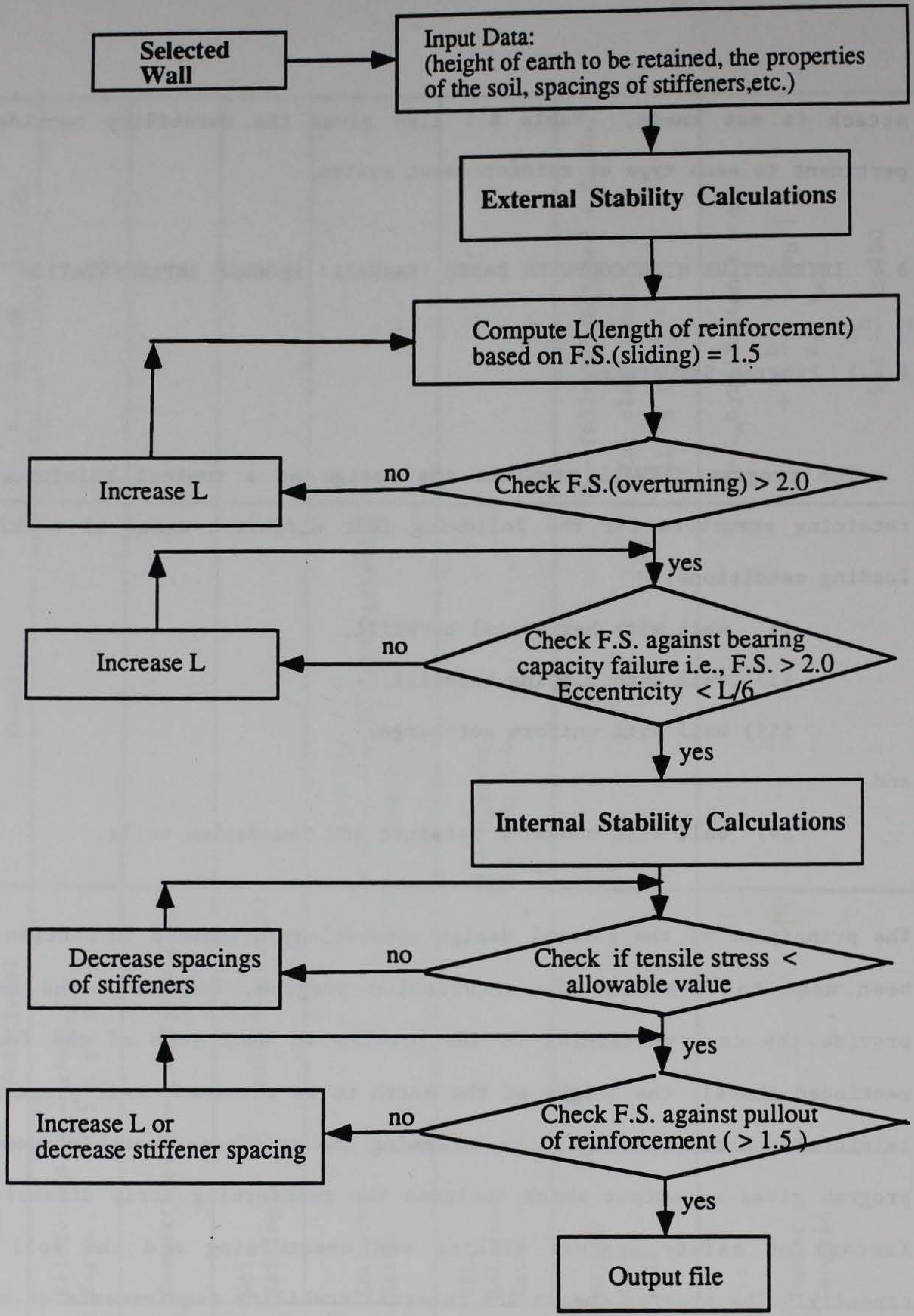



Figure 8.12. Flowchart of Program EARWALL for Reinforced Earth Design.

### 8.4.2. Input and Output

The input parameters are given through an interactive system-user consultation. The required input data are as follows:

- appropriate case relating to backfill/loading;
- height of the earth to be supported;
- internal angles of friction and specific weights of the special backfill and retained soil;
- size of the standard reinforcement strip used;
- horizontal and vertical spacings of the steel reinforcement strips;

and

- allowable tensile stress of the steel reinforcement strips.

The results in the output file are given for both external and internal stability of the wall. The external stability calculations include checks for minimum factor of safety against sliding and overturning. The internal stability computations give the reinforcement length details and the tensile forces to be resisted by the reinforcement. The design program also checks to ensure that the tensile stresses in the reinforcement and at the connections are within the allowable limits. The output file shows the intermediate results of all the interactions with the program, the design details and calculations in the form of tables.

### 8.4.3. Design Examples

Examples for the four different cases of backfill and loading conditions are presented in this section. The system-user interactive consultation for the design input, the input and output data as displayed on the screen are presented for each case.

**REFERENCES**


8.1 JK Mitchell & WCB Villet; Reinforcement of Earth Slopes and Embankments, TRB 290, Washington, D.C., June 1987, pp. 323.

8.2 UK Dept of Transportation.

8.3 MF Stocker, GW Korber, G Gassler, and G Gudehus, Soil Nailing, Proc. International Conference on Soil RTeinforcement: Reinforced Earth and Other Techniques. Paris, Vol. I (Mar. 1979) pp. 469-474.

8.4 KM Romstad, Z Al Yassin, LR Herrmann, and CK Shen, Stability Analysis of Reinforced Earth Retaining Structures. Proc. ASCE Symposium on Earth Reinforcement, Pittsburgh, Pennsylvania (1978) pp. 685-713.

8.5 TL Phan, P Segrestin, F Schlosser, and NT Long, Stability Analysis of Reinforced Earth Walls by Two Slip Circle Methods. Proc. International Conference Soil Reinforcement, Paris (1979) pp. 119-123.

8.6 KM Romstad, LR Herrmann, and CK Shen, Integrated Study of Reinforced Earth-1: Theoretical Formulation, ASCE J. Geotech Eng. Div., Vol. 102, No. GT5 (May 1976) pp. 457-471.

8.7 Al-Haussaini, and LD Johnson, Finite Element Analysis of a Reinforced Earth Wall, Technical Report S. 77, Soils and Pavements Design Laboratory, U.S. Army Engineer Waterways Station, Vicksburg, Miss. (May 1977).

8.8 LR Herrmann, and Z Al Yassin, Numerical Analysis of Reinforced Earth Systems, Proc. ASCE Symposium on Reinforced Earth, Pittsburgh, Penn. (Apr. 1978) preprint 3125, pp. 428-457.

8.9 RE Goodman, RL Taylor, and TL Brekke, A Model of the Mechanics of Jointed Rock. ASCE J. Soil Mechanics and Foundation Division, Vol. 94, No. SM3 (May 1968) pp. 637-659.

8.10 JM Duncan, KS Wong, Hyperbolic Stress-Strain Parameters for Nonlinear Finite Element Analyses of Stresses and Movements in Soil Masses. Report E-74-3, College of Engineering, University of California, Berkeley (July 1976).

8.11 GG Meyerhof, The Bearing Capacity of Foundations under Eccentric and Inclined Loads, Proc. 3rd International Conference on Soil Mechanics and Foundation Engineering, Zurich, Switzerland (1953) Vol. 1, pp. 440-445.

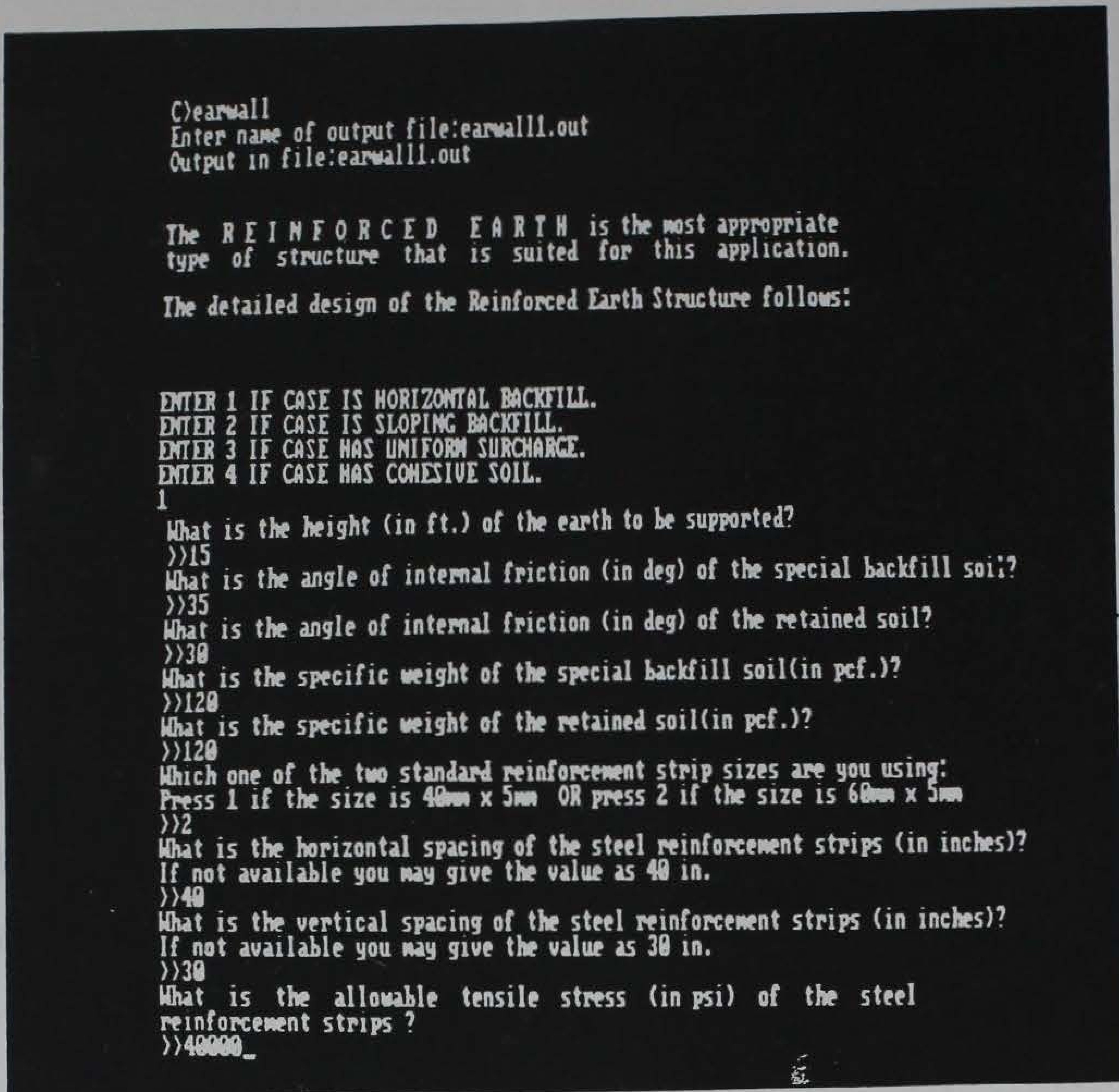



Figure 8.13 System -User Interactive Consultation for EARWALL - Case1

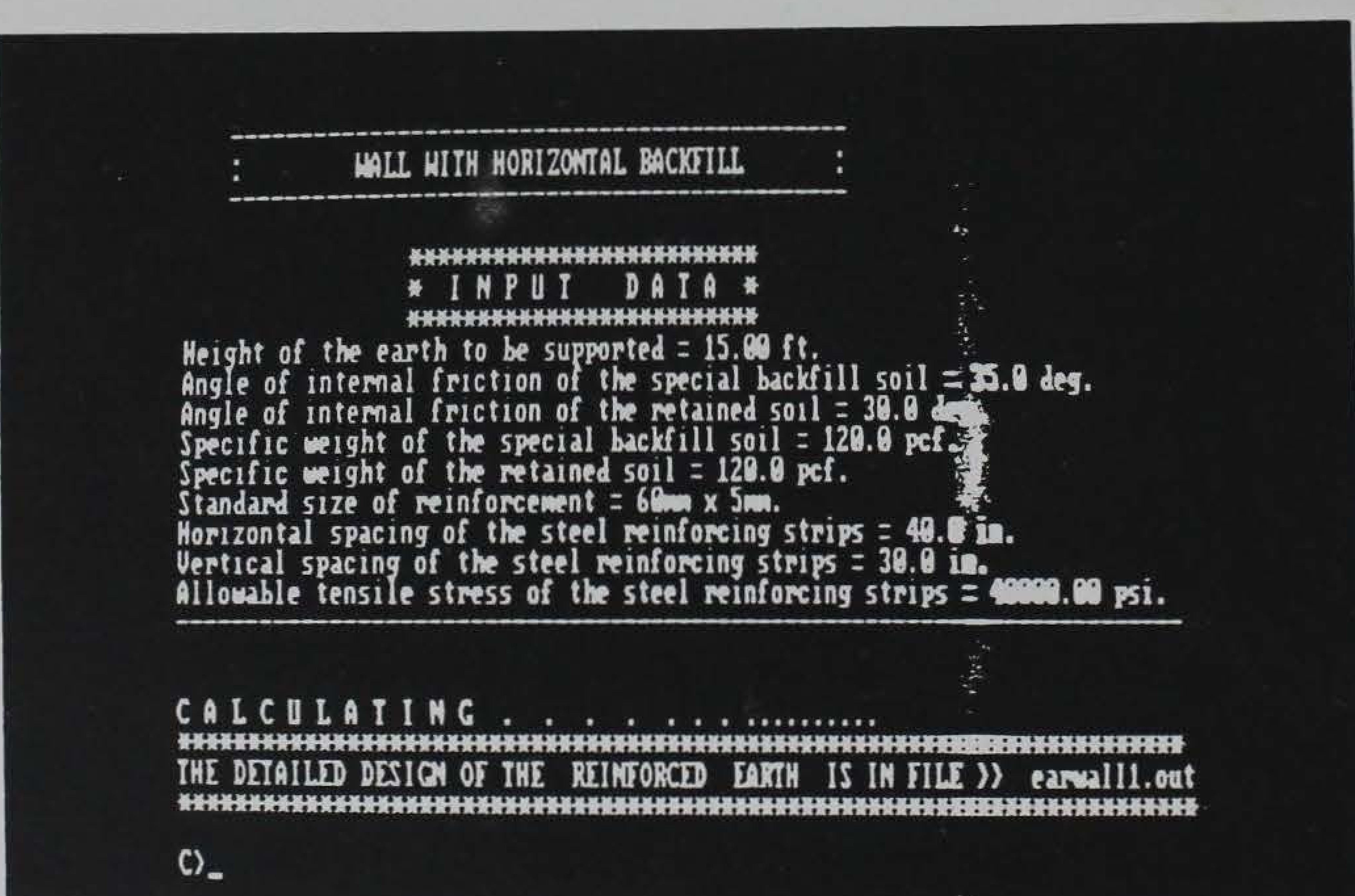



Figure 8.14 Screen Display of Input Data for EARWALL - Case 1

---

| DESIGN OF THE REINFORCED EARTH STRUCTURE |

---

: WALL WITH HORIZONTAL BACKFILL :

---

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

EXTERNAL STABILITY

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

1.Sliding on Base.

Coefficient of active pressure = 0.33
The total horizontal active earth force acting on the back of the wall,Pe = 4500.00 lb.
For a minimum factor of safety of 1.5 against sliding:
The minimum length of reinforcement,L = 6.50 ft.

2.Overturning about Toe.

For a minimum factor of safety of 2.0 against overturning:
The minimum base width of the wall = 7 ft.

3.Bearing Capacity Failure.

The eccentricity,e = 1.79 ft. is greater than the allowable value of e(=7/6.0) = 1.17.
Therefore the base width of the wall has to be increased.
Recomputing the eccentricity with base width of wall,Lb = 8
The eccentricity,e = 1.56 ft. is greater than the allowable value of e(=8/6.0) = 1.33.
Therefore the base width of the wall has to be increased.
Recomputing the eccentricity with base width of wall,Lb = 9
The eccentricity,e = 1.39 ft. is less than the allowable value of e(=Lb/6) = 1.50.----- OK.
Reqd. soil bearing capacity = 2603.57 lb./sq.ft.

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

INTERNAL STABILITY

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

| Layer # | Depth Z(ft.) |
|---|---|
| 1 | 1.25 |
| 2 | 3.75 |
| 3 | 6.25 |
| 4 | 8.75 |
| 5 | 11.25 |
| 6 | 13.75 |

Figure 8.15 Output File of EARWALL- Case 1

1.Tensile forces to be resisted by the reinforcement:

| Layer # | Depth [ft.] | Rv [lb/ft] | Pe [lb] | e [ft] | Sigv [lb/s.ft] | K | Sigh [lb/s.ft] | FH [lb] |
|---|---|---|---|---|---|---|---|---|
| 1 | 1.25 | 1350.0 | 31.2 | 0.01 | 150.3 | 0.4167 | 62.6 | 379.0 |
| 2 | 3.75 | 4050.0 | 281.2 | 0.09 | 458.9 | 0.3973 | 182.3 | 1102.9 |
| 3 | 6.25 | 6750.0 | 781.2 | 0.24 | 792.5 | 0.3779 | 299.4 | 1811.6 |
| 4 | 8.75 | 9450.0 | 1531.2 | 0.47 | 1173.2 | 0.3584 | 420.5 | 2544.1 |
| 5 | 11.25 | 12150.0 | 2531.2 | 0.78 | 1633.6 | 0.3390 | 553.8 | 3350.4 |
| 6 | 13.75 | 14850.0 | 3781.2 | 1.17 | 2227.8 | 0.3196 | 711.9 | 4307.0 |

2.Tensile stress in reinforcement:

| Strip # | Tensile Stress [psi.] |
|---|---|
| 1 | 815.0 |
| 2 | 2371.8 |
| 3 | 3895.8 |
| 4 | 5471.1 |
| 5 | 7205.1 |
| 6 | 9262.5 |

TENSILE STRESS IN STRIP < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.

3.Tensile stress at connection:

| Strip # | Tensile Stress [psi.] |
|---|---|
| 1 | 910.0 |
| 2 | 2648.1 |
| 3 | 4349.8 |
| 4 | 6108.6 |
| 5 | 8044.7 |
| 6 | 10341.8 |

TENSILE STRESS AT CONNECTION < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.

Figure 8.15 Continued

4.Pullout of reinforcement:

| Strip # | Le [ft.] | P [lb.] | FH [lb.] | F.S. |
|---|---|---|---|---|
| 1 | 4.50 | 385.4 | 379.0 | 1.02 |
| 2 | 4.50 | 1076.6 | 1102.9 | 0.98 |
| 3 | 4.50 | 1661.4 | 1811.6 | 0.92 |
| 4 | 5.75 | 2732.7 | 2544.1 | 1.07 |
| 5 | 7.05 | 3934.6 | 3350.4 | 1.17 |
| 6 | 8.35 | 5154.6 | 4307.0 | 1.20 |

RECALCULATING WITH Lb = 10 as F.S.(PULLOUT) < 1.5

1.Tensile forces to be resisted by the reinforcement:

| Layer # | Depth [ft.] | Rv [lb/ft] | Pe [lb] | e [ft] | Sigv [lb/s.ft] | K | Sigh [lb/s.ft] | FH [lb] |
|---|---|---|---|---|---|---|---|---|
| 1 | 1.25 | 1500.0 | 31.2 | 0.01 | 150.3 | 0.4167 | 62.6 | 378.8 |
| 2 | 3.75 | 4500.0 | 281.2 | 0.08 | 457.1 | 0.3973 | 181.6 | 1098.8 |
| 3 | 6.25 | 7500.0 | 781.2 | 0.22 | 784.0 | 0.3779 | 296.2 | 1792.3 |
| 4 | 8.75 | 10500.0 | 1531.2 | 0.43 | 1147.6 | 0.3584 | 411.3 | 2488.6 |
| 5 | 11.25 | 13500.0 | 2531.2 | 0.70 | 1570.9 | 0.3390 | 532.5 | 3221.8 |
| 6 | 13.75 | 16500.0 | 3781.2 | 1.05 | 2088.8 | 0.3196 | 667.5 | 4038.4 |

2.Tensile stress in reinforcement:

| Strip # | Tensile Stress [psi.] |
|---|---|
| 1 | 814.7 |
| 2 | 2362.9 |
| 3 | 3854.4 |
| 4 | 5351.8 |
| 5 | 6928.5 |
| 6 | 8684.7 |

TENSILE STRESS IN STRIP < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.

Figure 8.15 Continued

3.Tensile stress at connection:

| Strip # | Tensile Stress [psi.] |
|---|---|
| 1 | 909.6 |
| 2 | 2638.3 |
| 3 | 4303.5 |
| 4 | 5975.4 |
| 5 | 7735.9 |
| 6 | 9696.7 |

TENSILE STRESS AT CONNECTION < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.

4.Pullout of reinforcement:

| Strip # | Le [ft.] | P [lb.] | FH [lb.] | F.S. |
|---|---|---|---|---|
| 1 | 5.50 | 471.1 | 378.8 | 1.24 |
| 2 | 5.50 | 1315.8 | 1098.8 | 1.20 |
| 3 | 5.50 | 2030.6 | 1792.3 | 1.13 |
| 4 | 6.75 | 3208.3 | 2488.6 | 1.29 |
| 5 | 8.05 | 4492.9 | 3221.8 | 1.39 |
| 6 | 9.35 | 5772.0 | 4038.4 | 1.43 |

RECALCULATING WITH Lb = 11 as F.S.(PULLOUT) < 1.5

1.Tensile forces to be resisted by the reinforcement:

| Layer # | Depth [ft.] | Rv [lb/ft] | Pe [lb] | e [ft] | Sigv [lb/s.ft] | K | Sigh [lb/s.ft] | FH [lb] |
|---|---|---|---|---|---|---|---|---|
| 1 | 1.25 | 1650.0 | 31.2 | 0.01 | 150.2 | 0.4167 | 62.6 | 378.7 |
| 2 | 3.75 | 4950.0 | 281.2 | 0.07 | 455.9 | 0.3973 | 181.1 | 1095.7 |
| 3 | 6.25 | 8250.0 | 781.2 | 0.20 | 777.9 | 0.3779 | 293.9 | 1778.3 |
| 4 | 8.75 | 11550.0 | 1531.2 | 0.39 | 1129.4 | 0.3584 | 404.8 | 2449.1 |
| 5 | 11.25 | 14850.0 | 2531.2 | 0.64 | 1527.5 | 0.3390 | 517.8 | 3132.8 |
| 6 | 13.75 | 18150.0 | 3781.2 | 0.95 | 1996.6 | 0.3196 | 638.1 | 3860.2 |

Figure 8.15 Continued

2.Tensile stress in reinforcement:

| Strip # | Tensile Stress [psi.] |
|---|---|
| 1 | 814.4 |
| 2 | 2356.4 |
| 3 | 3824.3 |
| 4 | 5266.8 |
| 5 | 6737.2 |
| 6 | 8301.5 |

TENSILE STRESS IN STRIP < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.

3.Tensile stress at connection:

| Strip # | Tensile Stress [psi.] |
|---|---|
| 1 | 909.3 |
| 2 | 2631.0 |
| 3 | 4269.9 |
| 4 | 5880.5 |
| 5 | 7522.3 |
| 6 | 9268.9 |

TENSILE STRESS AT CONNECTION < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.

4.Pullout of reinforcement:

| Strip # | Le [ft.] | P [lb.] | FH [lb.] | F.S. |
|---|---|---|---|---|
| 1 | 6.50 | 556.7 | 378.7 | 1.47 |
| 2 | 6.50 | 1555.1 | 1095.7 | 1.42 |
| 3 | 6.50 | 2399.8 | 1778.3 | 1.35 |
| 4 | 7.75 | 3683.8 | 2449.1 | 1.50 |
| 5 | 9.05 | 5051.2 | 3132.8 | 1.61 |
| 6 | 10.35 | 6389.4 | 3860.2 | 1.66 |

RECALCULATING WITH Lb = 12 as F.S.(PULLOUT) < 1.5

Figure 8.15 Continued

1.Tensile forces to be resisted by the reinforcement:

| Layer # | Depth [ft.] | Rv [lb/ft] | Pe [lb] | e [ft] | Sigv [lb/s.ft] | K | Sigh [lb/s.ft] | FH [lb] |
|---|---|---|---|---|---|---|---|---|
| 1 | 1.25 | 1800.0 | 31.2 | 0.01 | 150.2 | 0.4167 | 62.6 | 378.6 |
| 2 | 3.75 | 5400.0 | 281.2 | 0.07 | 454.9 | 0.3973 | 180.7 | 1093.5 |
| 3 | 6.25 | 9000.0 | 781.2 | 0.18 | 773.3 | 0.3779 | 292.2 | 1767.8 |
| 4 | 8.75 | 12600.0 | 1531.2 | 0.35 | 1115.9 | 0.3584 | 400.0 | 2419.8 |
| 5 | 11.25 | 16200.0 | 2531.2 | 0.59 | 1496.1 | 0.3390 | 507.2 | 3068.4 |
| 6 | 13.75 | 19800.0 | 3781.2 | 0.88 | 1931.8 | 0.3196 | 617.3 | 3734.9 |

2.Tensile stress in reinforcement:

| Strip # | Tensile Stress [psi.] |
|---|---|
| 1 | 814.2 |
| 2 | 2351.5 |
| 3 | 3801.7 |
| 4 | 5203.9 |
| 5 | 6598.6 |
| 6 | 8032.0 |

TENSILE STRESS IN STRIP < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.

3.Tensile stress at connection:

| Strip # | Tensile Stress [psi.] |
|---|---|
| 1 | 909.1 |
| 2 | 2625.5 |
| 3 | 4244.7 |
| 4 | 5810.3 |
| 5 | 7367.5 |
| 6 | 8968.0 |

TENSILE STRESS AT CONNECTION < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.

Figure 8.15 Continued

4.Pullout of reinforcement:

| Strip # | Le [ft.] | P [lb.] | FH [lb.] | F.S. |
|---|---|---|---|---|
| 1 | 7.50 | 642.4 | 378.6 | 1.70 |
| 2 | 7.50 | 1794.3 | 1093.5 | 1.64 |
| 3 | 7.50 | 2769.0 | 1767.8 | 1.57 |
| 4 | 8.75 | 4159.4 | 2419.8 | 1.72 |
| 5 | 10.05 | 5609.5 | 3068.4 | 1.83 |
| 6 | 11.35 | 7006.8 | 3734.9 | 1.88 |

Figure 8.15 Continued

```
Output in file:earwall2.out

The R E I N F O R C E D  E A R T H is the most appropriate
type of structure that is suited for this application.

The detailed design of the Reinforced Earth Structure follows:


ENTER 1 IF CASE IS HORIZONTAL BACKFILL.
ENTER 2 IF CASE IS SLOPING BACKFILL.
ENTER 3 IF CASE HAS UNIFORM SURCHARGE.
ENTER 4 IF CASE HAS COHESIVE SOIL.
 2
 What is the height (in ft.) of the earth to be supported?
 >>15
 What is the angle of internal friction (in deg) of the special backfill soil?
 >>35
 What is the angle of internal friction (in deg) of the retained soil?
 >>30
 What is the specific weight of the special backfill soil(in pcf.)?
 >>120
 What is the specific weight of the retained soil(in pcf.)?
 >>120
 What is the angle (in deg.) that the backfill soil makes with the horizontal?
 >>26
 Which one of the two standard reinforcement strip sizes are you using:
 Press 1 if the size is 40mm x 5mm  OR press 2 if the size is 60mm x 5mm
 >>2
 What is the horizontal spacing of the steel reinforcement strips (in inches)?
 If not available you may give the value as 40 in.
 >>40
 What is the vertical spacing of the steel reinforcement strips (in inches)?
 If not available you may give the value as 30 in.
 >>30
 What  is  the  allowable  tensile  stress  (in psi)  of  the  steel
 reinforcement strips ?
 >>40000_
```

Figure 8.16 System -User Interactive Consultation for EARWALL - Case 2

```
 ---------------------------------------------------
 :          WALL WITH SLOPING BACKFILL            :
 ---------------------------------------------------
              *************************
              *  I N P U T    D A T A *
              *************************
Height of the earth to be supported = 15.00 ft.
Angle that the backfill makes with the horizontal = 26.0 deg.
Angle of internal friction of the special backfill soil = 35.0 deg.
Angle of internal friction of the retained soil = 30.0 deg.
Specific weight of the special backfill soil = 120.0 pcf.
Specific weight of the retained soil = 120.0 pcf.
Standard size of reinforcement = 60mm x 5mm.
Horizontal spacing of the steel reinforcing strips = 40.0 in.
Vertical spacing of the steel reinforcing strips = 30.0 in.
Allowable tensile stress of the steel reinforcing strips = 40000.00 psi.
--------------------------------------------------------------------------

C A L C U L A T I N G . . . . . . . ..........
**************************************************************************
THE DETAILED DESIGN OF THE REINFORCED EARTH IS IN FILE >> earwall2.out
**************************************************************************

C>
C>_
```

Figure 8.17 Screen Display of Input Data for EARWALL- Case 2

| DESIGN OF THE REINFORCED EARTH STRUCTURE |

: WALL WITH SLOPING BACKFILL

^^^^^^^^^^^^^^^^^^^^^^^^^^

EXTERNAL STABILITY

^^^^^^^^^^^^^^^^^^^^^^^^^^

1.Sliding on Base.

Horizontal component of the active earth pressure coeff.,Kah = 0.4668
For a minimum factor of safety of 1.5 against sliding:
The minimum length of reinforcement,L = 18 ft.

2.Overturning about Toe.

For the reinforcement length = 18 ft.:
The factor of safety against overturning = 3.16-------- OK (since > 2.0)

3.Bearing Capacity Failure.

The moment around the centerline at the base = 99899.12 ft.lb.
Vertical rection = 42120.00 lb.
Eccentricity,e = 2.37
The length L = 18 ft. is sufficient to maintain the resultant within the middle third ----- OK
(since e = 2.37 ft. < L / 6.0 = 3.00 ft..

^^^^^^^^^^^^^^^^^^^^^^^^^^

INTERNAL STABILITY

^^^^^^^^^^^^^^^^^^^^^^^^^^

1.Rupture of reinforcement:

| Strip # | Depth [ft.] | Vertical Load[lb.] | Pe [lb.] | Eccentricity [ft.] | Sigv [psf.] | Sigh [psf.] | FH [lb.] |
|---|---|---|---|---|---|---|---|
| 1 | 1.25 | 12420.0 | 2942.6 | 0.00 | 589.3 | 224.9 | 1360.9 |
| 2 | 3.75 | 17820.0 | 4553.0 | 0.00 | 932.9 | 338.0 | 2045.0 |
| 3 | 6.25 | 23220.0 | 6513.5 | 0.17 | 1314.9 | 450.8 | 2727.6 |
| 4 | 8.75 | 28620.0 | 8824.2 | 0.81 | 1746.3 | 564.8 | 3417.2 |
| 5 | 11.25 | 34020.0 | 11484.9 | 1.42 | 2244.5 | 682.4 | 4128.4 |
| 6 | 13.75 | 39420.0 | 14495.7 | 2.05 | 2835.5 | 807.0 | 4882.1 |

Figure 8.18 Output File of EARWALL- Case 2

2.Tensile stress in reinforcement and at the connection & pullout of reinforcement:

| Strip # | Depth [ft.] | Stress in Strip [psi.] | Stress @ Connection[psi] | Mu* | Le [ft] | P [lb] | F.S. |
|---|---|---|---|---|---|---|---|
| 1 | 1.25 | 2926.7 | 3267.8 | 1.270 | 12.15 | 4190.9 | 3.08 |
| 2 | 3.75 | 4397.7 | 4910.2 | 1.170 | 12.15 | 5539.7 | 2.71 |
| 3 | 6.25 | 5865.7 | 6549.2 | 1.070 | 13.45 | 7305.3 | 2.68 |
| 4 | 8.75 | 7348.8 | 8205.2 | 0.970 | 14.75 | 8953.1 | 2.62 |
| 5 | 11.25 | 8878.3 | 9912.9 | 0.870 | 16.05 | 10388.1 | 2.52 |
| 6 | 13.75 | 10499.1 | 11722.5 | 0.770 | 17.35 | 11518.0 | 2.36 |

TENSILE STRESS IN STRIP < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.
TENSILE STRESS AT CONNECTION < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.

Figure 8.18 Continued

```
Output in file:earwall3.out

The  R E I N F O R C E D   E A R T H  is the most appropriate
type  of  structure  that  is  suited  for  this  application.

The detailed design of the Reinforced Earth Structure follows:


ENTER 1 IF CASE IS HORIZONTAL BACKFILL.
ENTER 2 IF CASE IS SLOPING BACKFILL.
ENTER 3 IF CASE HAS UNIFORM SURCHARGE.
ENTER 4 IF CASE HAS COHESIVE SOIL.
 3
 What is the height (in ft.) of the earth to be supported?
 >>15
 What is the angle of internal friction (in deg) of the special backfill soil?
 >>35
 What is the angle of internal friction (in deg) of the retained soil?
 >>30
 What is the specific weight of the special backfill soil(in pcf.)?
 >>120
 What is the specific weight of the special backfill soil(in pcf.)?
 >>120
 What is the uniform vertical surcharge loading at the top of the wall(in psf.)?
 >>600
 Which one of the two standard reinforcement strip sizes are you using:
 Press 1 if the size is 40mm x 5mm  OR press 2 if the size is 60mm x 5mm
 >>2
 What is the horizontal spacing of the steel reinforcement strips (in inches)?
 If not available you may give the value as 40 in.
 >>40
 What is the vertical spacing of the steel reinforcement strips (in inches)?
 If not available you may give the value as 30 in.
 >>30
 What  is  the  allowable  tensile  stress  (in psi)  of  the  steel
 reinforcement strips ?
 >>40000_
```

Figure 8.19 System -User Interactive Consultation for EARWALL - Case 3

```
 ------------------------------------------
 :          WALL WITH SURCHARGE           :
 ------------------------------------------
            ************************
            *  I N P U T   D A T A *
            ************************
Height of the earth to be supported = 15.00 ft.
Uniform vertical surcharge at the top of the wall = 600.0 psf.
Angle of internal friction of the special backfill soil = 35.0 deg.
Angle of internal friction of the retained soil = 30.0 deg.
Specific weight of the special backfill soil = 120.0 pcf.
Specific weight of the retained soil = 120.0 pcf.
Standard size of reinforcement = 60mm x 5mm.
Horizontal spacing of the steel reinforcing strips = 40.0 in.
Vertical spacing of the steel reinforcing strips = 30.0 in.
Allowable tensile stress of the steel reinforcing strips = 40000.00 psi.
---------------------------------------------------------------------------
C A L C U L A T I N G . . . . . . . . .........
***************************************************************************
THE DETAILED DESIGN OF THE  REINFORCED  EARTH  IS IN FILE >>  earwall3.out
***************************************************************************

C>
C>
C>_
```

Figure 8.20 Screen Display of Input Data for EARWALL-Case 3

```
------------------------------------------------------------------------------
| DESIGN OF THE REINFORCED EARTH STRUCTURE |
------------------------------------------------------------------------------

      ---------------------------------------------
      : WALL WITH SURCHARGE :
      ---------------------------------------------

      ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
      EXTERNAL STABILITY
      ^^^^^^^^^^^^^^^^^^^^^^^^^^^^

 1.Sliding on Base.
 ----------------------
 Coefficient of active pressure = 0.33
 External horizontal force = 4500.00 lb./ft.
 Resultant horizontal force from surcharge loading = 3000.00 lb./ft.

 For a minimum factor of safety of 1.5 against sliding:
 The minimum length of reinforcement,L = 8.1 ft.

 2.Overturning about Toe.
 ------------------------------
 F.S.(overturning) = 1.76 ----NG (since FS < 2.0). Recalculate with L = 8.4
 F.S.(overturning) = 1.87 ----NG (since FS < 2.0). Recalculate with L = 8.6
 F.S.(overturning) = 1.98 ----NG (since FS < 2.0). Recalculate with L = 8.9
 F.S.(overturning) = 2.10 ---- OK (since FS > 2.0)

 3.Bearing Capacity Failure.
 --------------------------------
 Vertical reaction,Rv = 21285.57 lb.
 Eccentricity = 2.11 ft.
 Resultant outside middle third.(since 2.11 > (8.9/6.0 = 1.48))

 Recompute with L = 9.1:
 Vertical reaction,Rv = 21885.57 lb.
 Eccentricity = 2.06 ft.
 Resultant outside middle third.(since 2.06 > (9.1/6.0 = 1.52))

 Recompute with L = 9.4:
 Vertical reaction,Rv = 22485.57 lb.
 Eccentricity = 2.00 ft.
 Resultant outside middle third.(since 2.00 > (9.4/6.0 = 1.56))

 Recompute with L = 9.6:
 Vertical reaction,Rv = 23085.57 lb.
 Eccentricity = 1.95 ft.
 Resultant outside middle third.(since 1.95 > (9.6/6.0 = 1.60))

 Recompute with L = 9.9:
 Vertical reaction,Rv = 23685.57 lb.
 Eccentricity = 1.90 ft.
 Resultant outside middle third.(since 1.90 > (9.9/6.0 = 1.64))
```

Figure 8.21 Output File of EARWALL-Case 3

Recompute with L = 10.1:
Vertical reaction,Rv = 24285.57 lb.
Eccentricity = 1.85 ft.
Resultant outside middle third.(since 1.85 > (10.1/6.0 = 1.69))

Recompute with L = 10.4:
Vertical reaction,Rv = 24885.57 lb.
Eccentricity = 1.81 ft.
Resultant outside middle third.(since 1.81 > (10.4/6.0 = 1.73))

Recompute with L = 10.6:
Vertical reaction,Rv = 25485.57 lb.
Eccentricity = 1.77 ft.
Resultant inside middle third.(since 1.77 < (10.6/6.0 = 1.77))--- OK.
The ultimate bearing capacity of the foundation = 7191 psf.

^^^^^^^^^^^^^^^^^^^^^^^^^^^
INTERNAL STABILITY
^^^^^^^^^^^^^^^^^^^^^^^^^^^

1.Tensile stress in reinforcement and at connection :

FOR RUPTURE:

| Reinf. Strip # | Depth [ft.] | e [ft.] | Sigv. [psf.] | K | Sigh. [psf.] | FH [lb.] | FT [lb.] | FTc [lb.] | F.S. |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.25 | 0.02 | 753.0 | 0.417 | 313.8 | 1898.4 | 4082.6 | 4558.3 | 8.78 |
| 2 | 3.75 | 0.16 | 1082.1 | 0.397 | 429.9 | 2600.9 | 5593.4 | 6245.2 | 6.40 |
| 3 | 6.25 | 0.39 | 1455.8 | 0.378 | 550.1 | 3328.1 | 7157.1 | 7991.1 | 5.01 |
| 4 | 8.75 | 0.69 | 1897.2 | 0.358 | 680.0 | 4114.0 | 8847.4 | 9878.3 | 4.05 |
| 5 | 11.25 | 1.07 | 2441.9 | 0.339 | 827.8 | 5008.2 | 10770.2 | 12025.3 | 3.33 |
| 6 | 13.75 | 1.52 | 3149.7 | 0.320 | 1006.5 | 6089.5 | 13095.7 | 14621.7 | 2.74 |

TENSILE STRESS IN STRIP < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.
TENSILE STRESS AT CONNECTION < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.

2.Pullout of reinforcement.

FOR PULLOUT:

| Reinf. Strip # | Depth [ft.] | e [ft.] | Sigv. [psf.] | Sigh. [psf.] | FH [lb.] | Mu* | Le [ft.] | P [lb.] | F.S. |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.25 | 0.11 | 153.1 | 63.8 | 385.9 | 1.5 | 6.12 | 523.8 | 1.36 |
| 2 | 3.75 | 0.37 | 483.5 | 192.1 | 1162.1 | 1.4 | 6.12 | 1463.2 | 1.26 |
| 3 | 6.25 | 0.69 | 862.9 | 326.1 | 1972.7 | 1.3 | 6.12 | 2258.0 | 1.14 |
| 4 | 8.75 | 1.09 | 1320.4 | 473.2 | 2863.2 | 1.2 | 7.37 | 3500.9 | 1.22 |
| 5 | 11.25 | 1.54 | 1904.1 | 645.5 | 3905.0 | 1.1 | 8.67 | 4836.0 | 1.24 |
| 6 | 13.75 | 2.07 | 2702.8 | 863.7 | 5225.5 | 1.0 | 9.97 | 6151.0 | 1.18 |

Figure 8.21 Continued

Recalculating with L = 10.87 ft.as F.S.(pullout) < 1.5:

1.Tensile stress in reinforcement and at connection :

FOR RUPTURE:

| Reinf. Strip # | Depth [ft.] | e [ft.] | Sigv. [psf.] | K | Sigh. [psf.] | FH [lb.] | FT [lb.] | FTc [lb.] | F.S. |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.25 | 0.02 | 752.9 | 0.417 | 313.7 | 1898.1 | 4081.9 | 4557.5 | 8.78 |
| 2 | 3.75 | 0.15 | 1080.6 | 0.397 | 429.3 | 2597.3 | 5585.7 | 6236.5 | 6.41 |
| 3 | 6.25 | 0.38 | 1450.7 | 0.378 | 548.1 | 3316.2 | 7131.7 | 7962.7 | 5.02 |
| 4 | 8.75 | 0.68 | 1884.4 | 0.358 | 675.4 | 4086.2 | 8787.5 | 9811.5 | 4.08 |
| 5 | 11.25 | 1.05 | 2414.2 | 0.339 | 818.4 | 4951.4 | 10648.1 | 11888.9 | 3.36 |
| 6 | 13.75 | 1.48 | 3093.5 | 0.320 | 988.6 | 5980.8 | 12861.8 | 14360.6 | 2.79 |

TENSILE STRESS IN STRIP < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.
TENSILE STRESS AT CONNECTION < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.

2.Pullout of reinforcement.

FOR PULLOUT:

| Reinf. Strip # | Depth [ft.] | e [ft.] | Sigv. [psf.] | Sigh. [psf.] | FH [lb.] | Mu* | Le [ft.] | P [lb.] | F.S. |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.25 | 0.10 | 152.9 | 63.7 | 385.5 | 1.5 | 6.37 | 545.2 | 1.41 |
| 2 | 3.75 | 0.36 | 481.9 | 191.4 | 1158.2 | 1.4 | 6.37 | 1522.9 | 1.31 |
| 3 | 6.25 | 0.68 | 857.1 | 323.8 | 1959.2 | 1.3 | 6.37 | 2350.3 | 1.20 |
| 4 | 8.75 | 1.06 | 1305.1 | 467.8 | 2830.0 | 1.2 | 7.62 | 3619.7 | 1.28 |
| 5 | 11.25 | 1.51 | 1869.2 | 633.6 | 3833.5 | 1.1 | 8.92 | 4975.5 | 1.30 |
| 6 | 13.75 | 2.02 | 2626.6 | 839.4 | 5078.1 | 1.0 | 10.22 | 6305.3 | 1.24 |

Recalculating with L = 11.12 ft.as F.S.(pullout) < 1.5:
1.Tensile stress in reinforcement and at connection :

FOR RUPTURE:

| Reinf. Strip # | Depth [ft.] | e [ft.] | Sigv. [psf.] | K | Sigh. [psf.] | FH [lb.] | FT [lb.] | FTc [lb.] | F.S. |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.25 | 0.02 | 752.7 | 0.417 | 313.7 | 1897.7 | 4081.2 | 4556.7 | 8.78 |
| 2 | 3.75 | 0.15 | 1079.2 | 0.397 | 428.8 | 2594.0 | 5578.4 | 6228.5 | 6.42 |
| 3 | 6.25 | 0.37 | 1445.9 | 0.378 | 546.3 | 3305.3 | 7108.1 | 7936.4 | 5.04 |
| 4 | 8.75 | 0.66 | 1872.6 | 0.358 | 671.2 | 4060.5 | 8732.3 | 9749.9 | 4.10 |
| 5 | 11.25 | 1.02 | 2389.0 | 0.339 | 809.8 | 4899.5 | 10536.5 | 11764.3 | 3.40 |
| 6 | 13.75 | 1.45 | 3042.7 | 0.320 | 972.3 | 5882.7 | 12651.0 | 14125.2 | 2.83 |

TENSILE STRESS IN STRIP < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.

Figure 8.21 Continued

TENSILE STRESS AT CONNECTION < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.

2.Pullout of reinforcement.

FOR PULLOUT:

| Reinf. Strip # | Depth [ft.] | e [ft.] | Sigv. [psf.] | Sigh. [psf.] | FH [lb.] | Mu* | Le [ft.] | P [lb.] | F.S. |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.25 | 0.10 | 152.8 | 63.7 | 385.2 | 1.5 | 6.62 | 566.6 | 1.47 |
| 2 | 3.75 | 0.35 | 480.4 | 190.8 | 1154.6 | 1.4 | 6.62 | 1582.7 | 1.37 |
| 3 | 6.25 | 0.66 | 851.7 | 321.8 | 1946.9 | 1.3 | 6.62 | 2442.5 | 1.25 |
| 4 | 8.75 | 1.04 | 1291.1 | 462.8 | 2799.8 | 1.2 | 7.87 | 3738.5 | 1.34 |
| 5 | 11.25 | 1.48 | 1837.7 | 623.0 | 3769.0 | 1.1 | 9.17 | 5115.0 | 1.36 |
| 6 | 13.75 | 1.98 | 2559.2 | 817.8 | 4947.9 | 1.0 | 10.47 | 6459.6 | 1.31 |

Recalculating with L = 11.37 ft.as F.S.(pullout) < 1.5:
1.Tensile stress in reinforcement and at connection :

FOR RUPTURE:

| Reinf. Strip # | Depth [ft.] | e [ft.] | Sigv. [psf.] | K | Sigh. [psf.] | FH [lb.] | FT [lb.] | FTc [lb.] | F.S. |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.25 | 0.02 | 752.6 | 0.417 | 313.6 | 1897.4 | 4080.5 | 4556.0 | 8.78 |
| 2 | 3.75 | 0.15 | 1077.9 | 0.397 | 428.2 | 2590.8 | 5571.7 | 6220.9 | 6.43 |
| 3 | 6.25 | 0.36 | 1441.4 | 0.378 | 544.6 | 3295.1 | 7086.2 | 7912.0 | 5.06 |
| 4 | 8.75 | 0.65 | 1861.6 | 0.358 | 667.2 | 4036.9 | 8681.4 | 9693.0 | 4.13 |
| 5 | 11.25 | 1.00 | 2365.8 | 0.339 | 802.0 | 4852.0 | 10434.4 | 11650.3 | 3.43 |
| 6 | 13.75 | 1.42 | 2996.8 | 0.320 | 957.7 | 5793.9 | 12460.0 | 13912.0 | 2.88 |

TENSILE STRESS IN STRIP < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.
TENSILE STRESS AT CONNECTION < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.

2.Pullout of reinforcement.

FOR PULLOUT:

| Reinf. Strip # | Depth [ft.] | e [ft.] | Sigv. [psf.] | Sigh. [psf.] | FH [lb.] | Mu* | Le [ft.] | P [lb.] | F.S. |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.25 | 0.10 | 152.7 | 63.6 | 384.9 | 1.5 | 6.87 | 588.0 | 1.53 |
| 2 | 3.75 | 0.34 | 478.9 | 190.3 | 1151.2 | 1.4 | 6.87 | 1642.5 | 1.43 |
| 3 | 6.25 | 0.65 | 846.7 | 319.9 | 1935.5 | 1.3 | 6.87 | 2534.8 | 1.31 |
| 4 | 8.75 | 1.02 | 1278.4 | 458.2 | 2772.1 | 1.2 | 8.12 | 3857.3 | 1.39 |
| 5 | 11.25 | 1.44 | 1809.3 | 613.3 | 3710.7 | 1.1 | 9.42 | 5254.5 | 1.42 |
| 6 | 13.75 | 1.93 | 2499.3 | 798.7 | 4832.1 | 1.0 | 10.72 | 6613.8 | 1.37 |

Figure 8.21 Continued

Recalculating with L = 11.62 ft.as F.S.(pullout) < 1.5:
1.Tensile stress in reinforcement and at connection :

FOR RUPTURE:

| Reinf. Strip # | Depth [ft.] | e [ft.] | Sigv. [psf.] | K | Sigh. [psf.] | FH [lb.] | FT [lb.] | FTc [lb.] | F.S. |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.25 | 0.02 | 752.5 | 0.417 | 313.6 | 1897.2 | 4079.9 | 4555.3 | 8.78 |
| 2 | 3.75 | 0.14 | 1076.7 | 0.397 | 427.8 | 2587.9 | 5565.4 | 6213.9 | 6.44 |
| 3 | 6.25 | 0.35 | 1437.3 | 0.378 | 543.1 | 3285.6 | 7065.8 | 7889.2 | 5.07 |
| 4 | 8.75 | 0.63 | 1851.5 | 0.358 | 663.6 | 4014.9 | 8634.3 | 9640.4 | 4.15 |
| 5 | 11.25 | 0.98 | 2344.5 | 0.339 | 794.8 | 4808.4 | 10340.5 | 11545.5 | 3.46 |
| 6 | 13.75 | 1.39 | 2955.1 | 0.320 | 944.3 | 5713.2 | 12286.4 | 13718.1 | 2.92 |

TENSILE STRESS IN STRIP < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.
TENSILE STRESS AT CONNECTION < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.

2.Pullout of reinforcement.

FOR PULLOUT:

| Reinf. Strip # | Depth [ft.] | e [ft.] | Sigv. [psf.] | Sigh. [psf.] | FH [lb.] | Mu* | Le [ft.] | P [lb.] | F.S. |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.25 | 0.10 | 152.6 | 63.6 | 384.6 | 1.5 | 7.12 | 609.4 | 1.58 |
| 2 | 3.75 | 0.34 | 477.6 | 189.8 | 1148.0 | 1.4 | 7.12 | 1702.3 | 1.48 |
| 3 | 6.25 | 0.64 | 842.0 | 318.2 | 1924.9 | 1.3 | 7.12 | 2627.0 | 1.36 |
| 4 | 8.75 | 0.99 | 1266.6 | 454.0 | 2746.7 | 1.2 | 8.37 | 3976.2 | 1.45 |
| 5 | 11.25 | 1.41 | 1783.5 | 604.6 | 3657.7 | 1.1 | 9.67 | 5394.0 | 1.47 |
| 6 | 13.75 | 1.89 | 2445.7 | 781.6 | 4728.5 | 1.0 | 10.97 | 6768.1 | 1.43 |

Recalculating with L = 11.87 ft.as F.S.(pullout) < 1.5:
1.Tensile stress in reinforcement and at connection :

FOR RUPTURE:

| Reinf. Strip # | Depth [ft.] | e [ft.] | Sigv. [psf.] | K | Sigh. [psf.] | FH [lb.] | FT [lb.] | FTc [lb.] | F.S. |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.25 | 0.02 | 752.4 | 0.417 | 313.5 | 1896.9 | 4079.3 | 4554.7 | 8.78 |
| 2 | 3.75 | 0.14 | 1075.6 | 0.397 | 427.3 | 2585.2 | 5559.5 | 6207.3 | 6.44 |
| 3 | 6.25 | 0.35 | 1433.4 | 0.378 | 541.6 | 3276.8 | 7046.9 | 7868.0 | 5.08 |
| 4 | 8.75 | 0.62 | 1842.1 | 0.358 | 660.3 | 3994.6 | 8590.5 | 9591.6 | 4.17 |
| 5 | 11.25 | 0.96 | 2324.9 | 0.339 | 788.1 | 4768.1 | 10254.1 | 11448.9 | 3.49 |
| 6 | 13.75 | 1.36 | 2917.0 | 0.320 | 932.2 | 5639.5 | 12128.0 | 13541.3 | 2.95 |

TENSILE STRESS IN STRIP < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.
TENSILE STRESS AT CONNECTION < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.

Figure 8.21 Continued

2.Pullout of reinforcement.

-------------------------------

FOR PULLOUT:

| Reinf. Strip # | Depth [ft.] | e [ft.] | Sigv. [psf.] | Sigh. [psf.] | FH [lb.] | Mu* | Le [ft.] | P [lb.] | F.S. |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.25 | 0.10 | 152.4 | 63.5 | 384.3 | 1.5 | 7.37 | 630.8 | 1.64 |
| 2 | 3.75 | 0.33 | 476.4 | 189.3 | 1145.1 | 1.4 | 7.37 | 1762.1 | 1.54 |
| 3 | 6.25 | 0.62 | 837.8 | 316.5 | 1915.1 | 1.3 | 7.37 | 2719.3 | 1.42 |
| 4 | 8.75 | 0.97 | 1255.8 | 450.1 | 2723.2 | 1.2 | 8.62 | 4095.0 | 1.50 |
| 5 | 11.25 | 1.38 | 1759.9 | 596.6 | 3609.4 | 1.1 | 9.92 | 5533.5 | 1.53 |
| 6 | 13.75 | 1.85 | 2397.5 | 766.2 | 4635.3 | 1.0 | 11.22 | 6922.4 | 1.49 |

Recalculating with L = 12.12 ft.as F.S.(pullout) < 1.5:
1.Tensile stress in reinforcement and at connection :

-------------------------------------------------

FOR RUPTURE:

| Reinf. Strip # | Depth [ft.] | e [ft.] | Sigv. [psf.] | K | Sigh. [psf.] | FH [lb.] | FT [lb.] | FTc [lb.] | F.S. |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.25 | 0.02 | 752.3 | 0.417 | 313.5 | 1896.6 | 4078.8 | 4554.1 | 8.78 |
| 2 | 3.75 | 0.14 | 1074.5 | 0.397 | 426.9 | 2582.6 | 5554.0 | 6201.1 | 6.45 |
| 3 | 6.25 | 0.34 | 1429.8 | 0.378 | 540.3 | 3268.5 | 7029.1 | 7848.2 | 5.10 |
| 4 | 8.75 | 0.61 | 1833.4 | 0.358 | 657.1 | 3975.7 | 8549.9 | 9546.2 | 4.19 |
| 5 | 11.25 | 0.94 | 2306.8 | 0.339 | 782.0 | 4731.0 | 10174.2 | 11359.8 | 3.52 |
| 6 | 13.75 | 1.33 | 2882.1 | 0.320 | 921.0 | 5572.1 | 11983.0 | 13379.3 | 2.99 |

TENSILE STRESS IN STRIP < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.
TENSILE STRESS AT CONNECTION < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.

2.Pullout of reinforcement.

-------------------------------

FOR PULLOUT:

| Reinf. Strip # | Depth [ft.] | e [ft.] | Sigv. [psf.] | Sigh. [psf.] | FH [lb.] | Mu* | Le [ft.] | P [lb.] | F.S. |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.25 | 0.09 | 152.3 | 63.5 | 384.1 | 1.5 | 7.62 | 652.3 | 1.70 |
| 2 | 3.75 | 0.32 | 475.3 | 188.8 | 1142.4 | 1.4 | 7.62 | 1821.8 | 1.59 |
| 3 | 6.25 | 0.61 | 833.8 | 315.0 | 1906.0 | 1.3 | 7.62 | 2811.6 | 1.48 |
| 4 | 8.75 | 0.95 | 1245.9 | 446.5 | 2701.6 | 1.2 | 8.87 | 4213.8 | 1.56 |
| 5 | 11.25 | 1.35 | 1738.4 | 589.3 | 3565.2 | 1.1 | 10.17 | 5673.0 | 1.59 |
| 6 | 13.75 | 1.81 | 2354.0 | 752.3 | 4551.1 | 1.0 | 11.47 | 7076.6 | 1.55 |

Figure 8.21 Continued

Recalculating with L = 12.37 ft.as F.S.(pullout) < 1.5:
1.Tensile stress in reinforcement and at connection :

FOR RUPTURE:

| Reinf. Strip # | Depth [ft.] | e [ft.] | Sigv. [psf.] | K | Sigh. [psf.] | FH [lb.] | FT [lb.] | FTc [lb.] | F.S. |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.25 | 0.02 | 752.2 | 0.417 | 313.5 | 1896.4 | 4078.3 | 4553.5 | 8.78 |
| 2 | 3.75 | 0.14 | 1073.5 | 0.397 | 426.5 | 2580.2 | 5548.8 | 6195.4 | 6.46 |
| 3 | 6.25 | 0.33 | 1426.4 | 0.378 | 539.0 | 3260.8 | 7012.5 | 7829.7 | 5.11 |
| 4 | 8.75 | 0.59 | 1825.3 | 0.358 | 654.2 | 3958.1 | 8512.0 | 9503.9 | 4.21 |
| 5 | 11.25 | 0.92 | 2290.0 | 0.339 | 776.3 | 4696.6 | 10100.2 | 11277.2 | 3.55 |
| 6 | 13.75 | 1.30 | 2850.0 | 0.320 | 910.8 | 5510.1 | 11849.8 | 13230.6 | 3.02 |

TENSILE STRESS IN STRIP < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.
TENSILE STRESS AT CONNECTION < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.

2.Pullout of reinforcement.

FOR PULLOUT:

| Reinf. Strip # | Depth [ft.] | e [ft.] | Sigv. [psf.] | Sigh. [psf.] | FH [lb.] | Mu* | Le [ft.] | P [lb.] | F.S. |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.25 | 0.09 | 152.2 | 63.4 | 383.8 | 1.5 | 7.87 | 673.7 | 1.76 |
| 2 | 3.75 | 0.32 | 474.2 | 188.4 | 1139.8 | 1.4 | 7.87 | 1881.6 | 1.65 |
| 3 | 6.25 | 0.60 | 830.1 | 313.6 | 1897.5 | 1.3 | 7.87 | 2903.8 | 1.53 |
| 4 | 8.75 | 0.93 | 1236.6 | 443.2 | 2681.6 | 1.2 | 9.12 | 4332.7 | 1.62 |
| 5 | 11.25 | 1.33 | 1718.6 | 582.6 | 3524.7 | 1.1 | 10.42 | 5812.5 | 1.65 |
| 6 | 13.75 | 1.78 | 2314.5 | 739.6 | 4474.7 | 1.0 | 11.72 | 7230.9 | 1.62 |

***********************************************************************

Figure 8.21 Continued

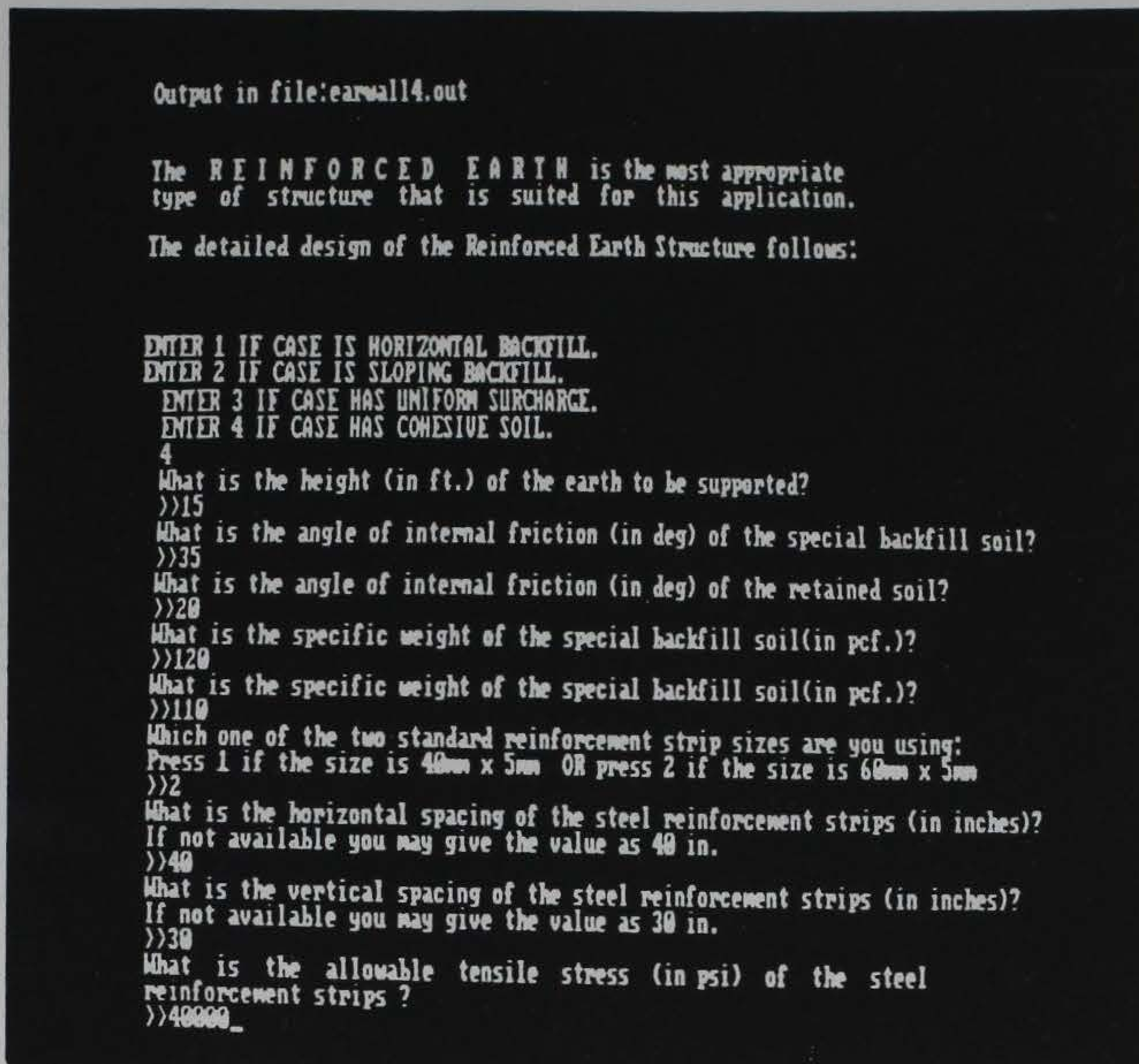



Figure 8.22 System -User Interactive Consultation for EARWALL - Case 4

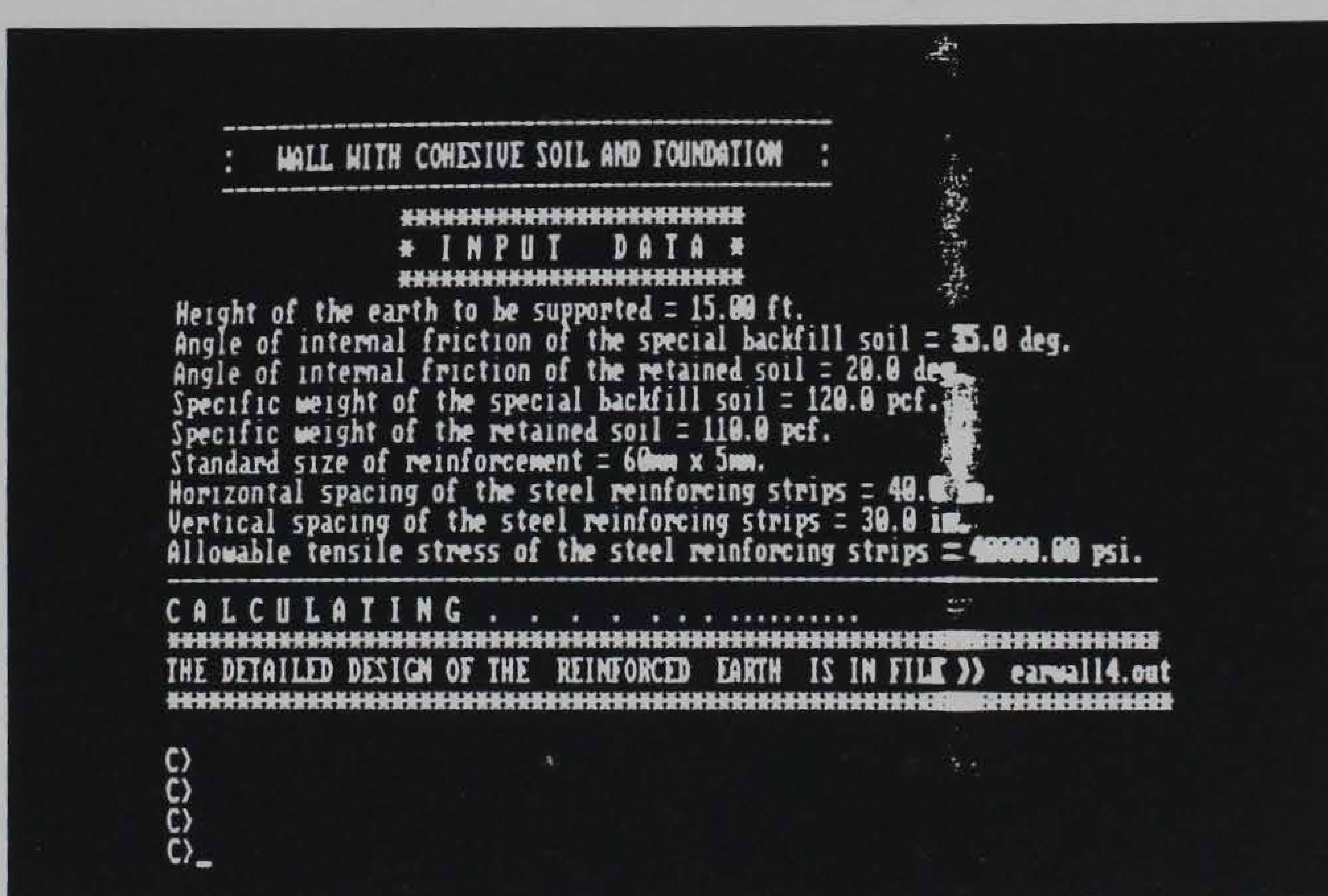



Figure 8.23 Screen Display of Input Data for EARWALL- Case 4

```
---------------------------------------------------------------------------------
| DESIGN OF THE REINFORCED EARTH STRUCTURE |
---------------------------------------------------------------------------------

---------------------------------------------------------------------------
: WALL WITH COHESIVE SOIL AND FOUNDATION :
---------------------------------------------------------------------------

^^^^^^^^^^^^^^^^^^^^^^^^^^^^
EXTERNAL STABILITY
^^^^^^^^^^^^^^^^^^^^^^^^^^^^

1.Sliding on Base.
----------------------
Coefficient of active pressure = 0.49
External horizontal force = 6067.35 lb./ft.
For a minimum factor of safety of 1.5 against sliding:
The minimum length of reinforcement,L = 13.9 ft.

2.Overturning about Toe.
-----------------------------
F.S.(overturning) = 5.73 ---- OK (since FS > 2.0)

3.Bearing Capacity Failure.
---------------------------------
Vertical reaction,Rv = 25454.85 lb.
Eccentricity = 1.19 ft.
Resultant inside middle third.(since 1.19 < (14.1/6.0 = 2.36))--- OK.
The ultimate bearing capacity of the foundation = 4329 psf.

^^^^^^^^^^^^^^^^^^^^^^^^^^^^
INTERNAL STABILITY
^^^^^^^^^^^^^^^^^^^^^^^^^^^^

---------------------------------
 Layer #    Depth Z(ft.)
---------------------------------
   1          1.25
   2          3.75
   3          6.25
   4          8.75
   5         11.25
   6         13.75
---------------------------------
```

Figure 8.24 Output File of EARWALL- Case 4

1.Tensile forces to be resisted by the reinforcement:

| Layer # | Depth [ft.] | Rv [lb/ft] | Pe [lb] | e [ft] | Sigv [lb/s.ft] | K | Sigh [lb/s.ft] | FH [lb] |
|---|---|---|---|---|---|---|---|---|
| 1 | 1.25 | 2121.2 | 46.0 | 0.01 | 150.2 | 0.42 | 62.6 | 378.6 |
| 2 | 3.75 | 6363.7 | 413.7 | 0.08 | 455.2 | 0.40 | 180.9 | 1094.2 |
| 3 | 6.25 | 10606.2 | 1149.1 | 0.23 | 774.7 | 0.38 | 292.7 | 1771.0 |
| 4 | 8.75 | 14848.7 | 2252.3 | 0.44 | 1120.1 | 0.36 | 401.5 | 2428.8 |
| 5 | 11.25 | 19091.1 | 3723.1 | 0.73 | 1505.7 | 0.34 | 510.4 | 3088.1 |
| 6 | 13.75 | 23333.6 | 5561.7 | 1.09 | 1951.5 | 0.32 | 623.6 | 3773.0 |

2.Tensile stress in reinforcement:

| Strip # | Tensile Stress [psi.] |
|---|---|
| 1 | 814.29 |
| 2 | 2353.05 |
| 3 | 3808.67 |
| 4 | 5223.30 |
| 5 | 6641.10 |
| 6 | 8113.92 |

TENSILE STRESS IN STRIP < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.

3.Tensile stress at connection:

| Strip # | Tensile Stress [psi.] |
|---|---|
| 1 | 909.18 |
| 2 | 2627.24 |
| 3 | 4252.48 |
| 4 | 5831.96 |
| 5 | 7414.98 |
| 6 | 9059.42 |

TENSILE STRESS AT CONNECTION < ALLOWABLE TENSILE STRESS(40000.00 psi.) ---- OK.

Figure 8.24 Continued

4.Pullout of reinforcement:

| Strip # | Le [ft.] | P [lb.] | FH [lb.] | F.S. |
|---|---|---|---|---|
| 1 | 9.64 | 825.82 | 378.65 | 2.18 |
| 2 | 9.64 | 2306.66 | 1094.17 | 2.11 |
| 3 | 9.64 | 3559.74 | 1771.03 | 2.01 |
| 4 | 10.89 | 5177.82 | 2428.84 | 2.13 |
| 5 | 12.19 | 6805.05 | 3088.12 | 2.20 |
| 6 | 13.49 | 8328.92 | 3772.98 | 2.21 |

*********************************************************************

Figure 8.24 Continued

# CHAPTER 9

# SHEET PILE WALLS

## 9.1 INTRODUCTION

Sheet pile walls are widely used for both large and small waterfront structures. Sheetpiling is also used for beach erosion protection, to assist in stabilizing ground slopes, for shoring walls of trenches and other excavations, and for cofferdams. When the wall is under about 3 m in height it may be cantilevered, however, for larger wall heights it is usually anchored- termed anchored sheetpiling or anchored bulkheads. Sheet piles generally resist **horizontal** pressures due to soil and water and derive their stability from the **horizontal** resistance of the ground into which they are driven and also from the horizontal support provided by any anchors, ties, or struts placed at a higher level. Fig. 9.1 shows typical sheetpiling structures.

Sheetpiling materials may be of timber, reinforced concrete or steel. Timber piling is often used as temporary braced sheeting in trenches to prevent side cave-ins during installation of deep water and sewer lines. If timber sheeting is used in permanent structures above water level, preservative treatment is necessary, and even so the useful life is seldom over 10 to 15 years.

Reinforced concrete sheet piles are precast concrete members usually with a tongue-and-groove joint. They are designed for the computed service

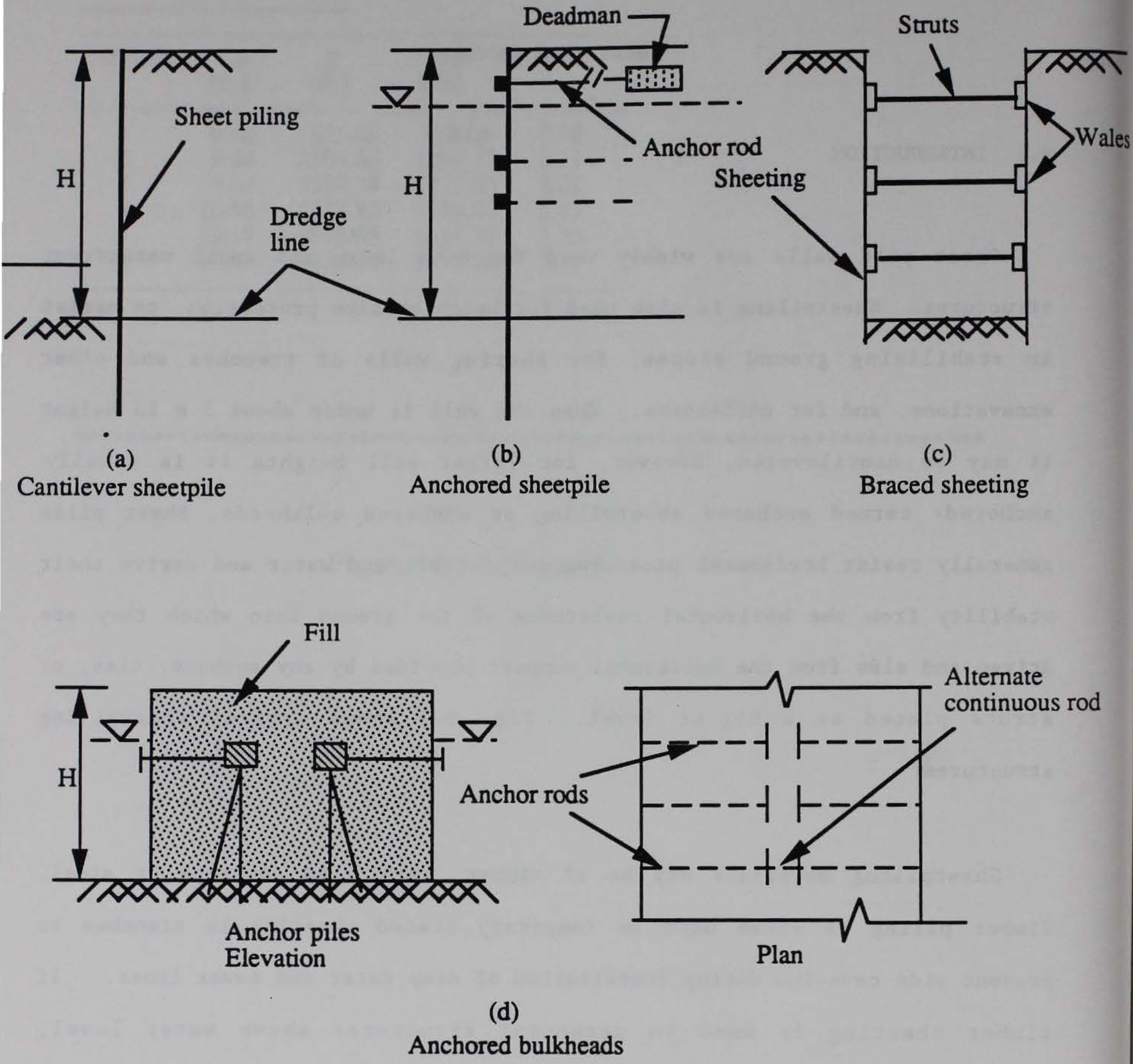



Figure 9.1 Typical Sheetpile Structures. [9.1]

stresses, but handling and driving stresses which are significant due to their weight must also be taken into account. They are relatively large in size; coupled with the high unit weight of concrete, the piles are quite heavy and may not be competitive with other types of piles unless they are cast in close proximity to the jobsite.

Steel sheetpiling is the most common type used because of the following advantages over other materials:

i) it is resistant to high driving stresses as developed in hard or rocky material;

ii) it is of relatively light weight;

iii) it may be reused several times;

iv) it has a long service life either, above or below water with modest protection;

v) the pile length can be increased by either welding or bolting;

vi) joints are less likely to deform when wedged full with soil and stones during driving.

In situations where large bending moments are to be resisted, the Z piles are used in anchored or cantilever walls. In cellular coffer dams, straight-web sheet piles are used where the web is subjected to tension.

## 9.2 BASIC THEORY

### 9.2.1. Cantilever Sheetpiling

Cantilever sheet piles depend on adequate embedment into the dredge line soil so that the wall is essentially a wide cantilever beam of length H, as shown in Fig. 9.1a. The stability of the cantilever sheet pile is largely dependent on embedment depth and hence it is important that the soil is stable below the dredge line.

Current analysis methods may be divided into two groups:

i) Discrete element methods - the finite difference and finite element approaches.

ii) Classical methods - procedures which involve simplifying assumptions and rigid body statics.

In this presentation, the classical method is used to analyse the cantilever sheet pile. The analysis of the cantilever sheet pile is dependent on the embedment soil type, viz., granular or cohesive. The following sections describe the designs of the cantilever sheet pile in granular and cohesive soils.

#### 9.2.1.1 Cantilever sheetpiling in granular soils

The pile is subjected to active pressure to the dredge line, on the backfill side. Due to the active pressure the wall tends to rotate and develop passive pressure in front of the wall and active pressure behind the wall. At the pivotal point b of Fig. 9.2, the soil behind the wall goes from active to passive pressure, with active pressure in front of the wall for the remainder of the distance to the pile bottom. The soil mass above the point under consideration can be treated as a surcharge for computing active and passive pressures. It is assumed that the soil below the dredge line has the same internal angle of friction, $\phi$ as the backfill above the dredge line.

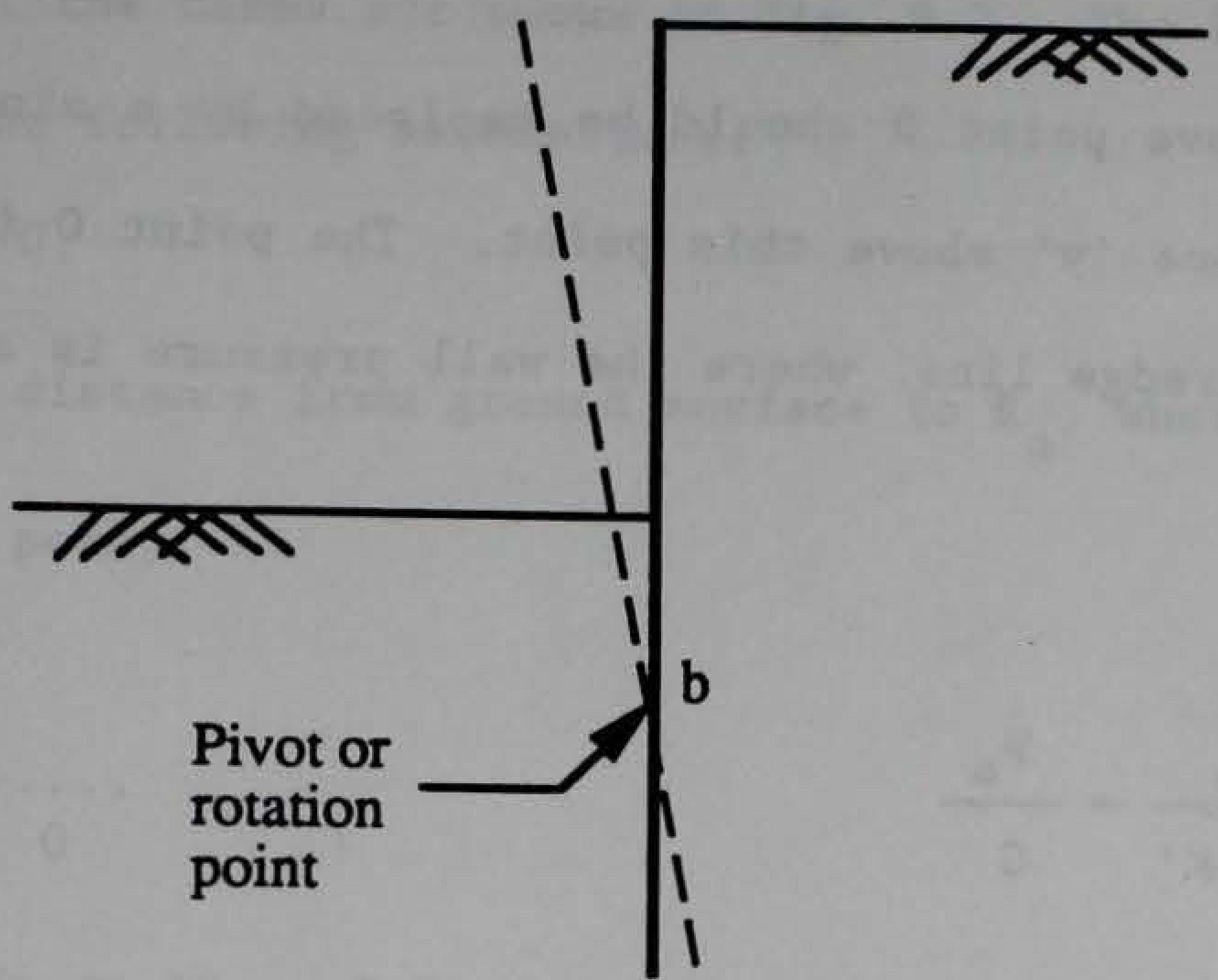
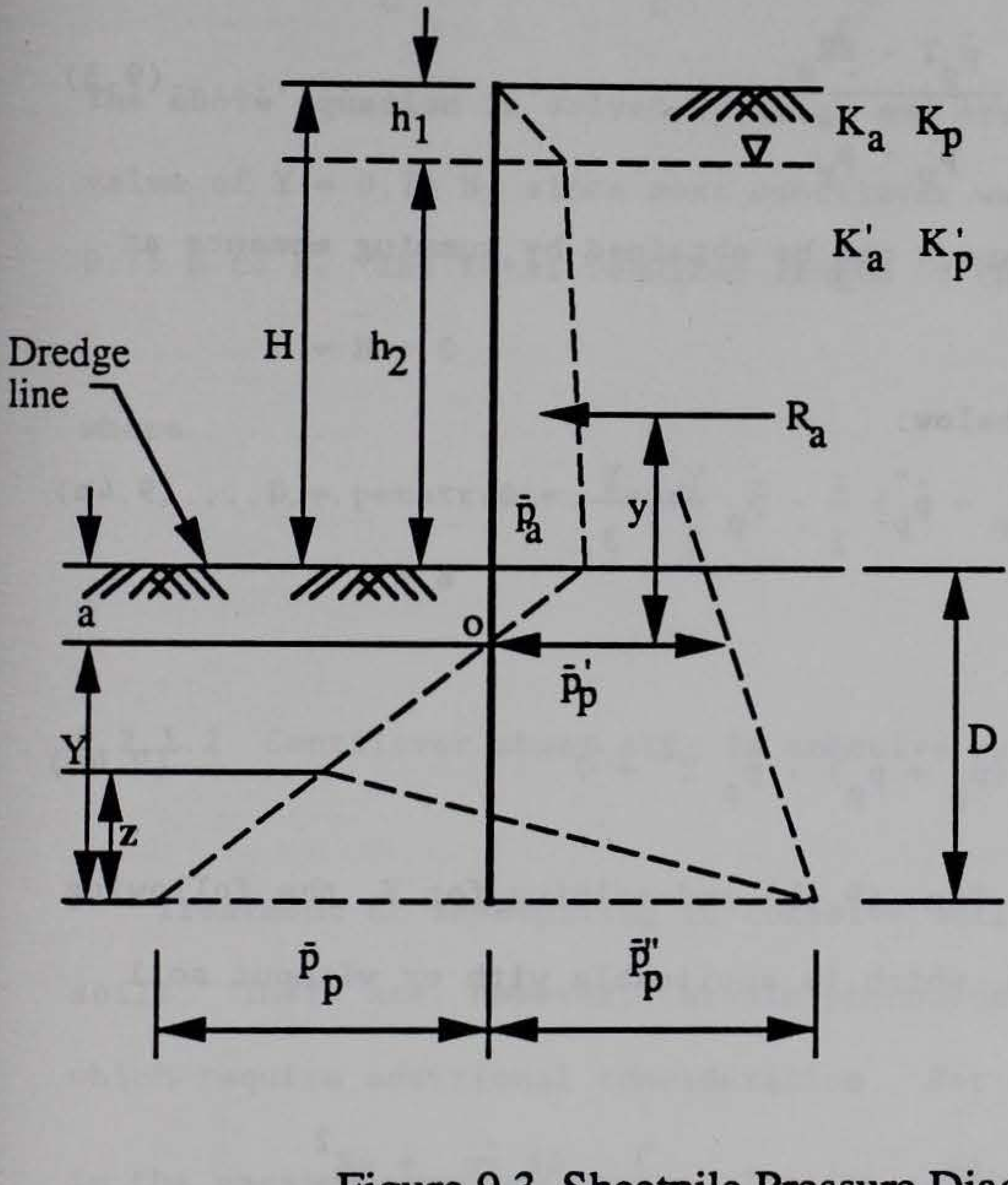



Figure 9.2 Assumed Elastic Line of Sheetpile. [9.1]




If $\phi$ below the water table is the same as above, then $K'_p = K_p$ and $K'_a = K_a$

$$a = \frac{\bar{p}_a}{\gamma'(K'_a - K'_p)} = \frac{\bar{p}_a}{C}$$

$$\bar{p}_p = CY$$

$$\bar{p}'_p = \gamma h_1 K_p + (h_2 + a)\gamma' K'_p - \gamma' a K'_a$$

$$\bar{p}''_p = \bar{p}'_p + CY$$

$$K_a = \sin(2\pi/2 + \phi) \,/\, \sin(\pi/2 - \delta)(1 + A)^2$$

$$K_p = \sin(2\pi/2 - \phi) \,/\, \sin(\pi/2 + \phi)(1 - A)^2$$

$$A = [\sin(\phi + \delta)\sin\phi \,/\, \sin(\pi/2 + \delta)]^{1/2}$$

Figure 9.3 Sheetpile Pressure Diagram for Granular Soil. [9.1]

Fig. 9.3 shows the cantilever sheet pile pressure diagram for granular soil.

First, all the forces above point 0 should be replaced by a single force resultant $R_a$ located a distance 'y' above this point. The point 0 is located at a distance 'a' below the dredge line, where the wall pressure is assumed to be zero, and is given by

$$a = \frac{\bar{p}_a}{\gamma' (K'_p - K'_a)} = \frac{\bar{p}_a}{\gamma' K'} = \frac{\bar{p}_a}{C} \quad \text{.........(9.1)}$$

The distance z can be found in terms of Y by statics ($\Sigma F_H = 0$) to obtain

$$R_a + (\bar{p}_p + \bar{p}_p'') \frac{z}{2} - \bar{p}_p \frac{Y}{2} = 0 \quad \text{.........(9.2)}$$

and solving for z,

$$z = \frac{\bar{p}_p Y - 2R_a}{\bar{p}_p + \bar{p}_p''} \quad \text{.........(9.3)}$$

An additional equation in Y and z can be obtained by summing moments at the bottom of the pile as shown below:

$$R_a (Y + \bar{y}) + \frac{z}{3} (\bar{p}_p + \bar{p}_p'') \frac{z}{2} - \bar{p}_p \frac{Y}{2} \frac{Y}{3} = 0 \quad \text{........(9.4a)}$$

Simplifying,

$$6R_a ( Y + \bar{y}) + z^2 (\bar{p}_p + \bar{p}_p'') - \bar{p}_p Y^2 = 0 \quad \text{........(9.4b)}$$

Substituting Eqn. (9.3) into Eqn. (9.4b) and solving for Y, the following fourth-degree equation is formed, which is applicable with or without soil water:

$$Y^4 + Y^3 \frac{\bar{p}_p'}{C} - Y^2 \frac{8R_a}{C} - Y \left[ \frac{6R_a}{C^2} (2\bar{y}C + \bar{p}_p') \right] - \frac{6R_a \bar{y}\bar{p}_p + 4R_a^2}{C^2} = 0 \quad \text{.........(9.5)}$$

where all the terms are shown in Fig. 9.3. The Eqn. 9.5 is solved by making the following assumptions:

$a = 0$;

$\bar{y}$ = distance from ground surface to $R_a$, where $R_a$ is the force against pole;

and

$\bar{p}'_p = 0$

Eqn. 9.5 reduces to

$$Y^4 - Y^2 \frac{8R_a}{C} - Y \frac{12R_a \bar{y}}{C} - \frac{4R_a^2}{C^2} = 0 \quad \text{.........(9.6)}$$

The above equation is solved by trial and error, with an initial starting value of Y = 0.75 H, since most cantilever walls require embedment depths of 0.75 H to H. The total required length of the pile can be computed as

$$L = H + D \quad \text{.........(9.7)}$$

where

D = penetration depth

= Y + a

9.2.1.2 Cantilever sheet pile in cohesive soils ($\phi$ = o)

Treatment of sheetpiling in cohesive soils is similar to that in granular soils. There are, however, certain phenomena associated with cohesive soils which require additional consideration. For example, consolidation may occur in the passive-pressure zones. Tension cracks may form in the active zone and become filled with water, thus increasing the lateral pressure considerably, as well as changing the location of the resultant. The clay may shrink and pull away from the wall, which also increases the lateral pressure. This latter problem may be allowed for in the design by neglecting the theoretical benefits of the wall adhesion.

Referring to Fig. 9.4,

$$\sigma_a = \bar{q}\ K_a - 2\ c\ \sqrt{K_a} \qquad \text{.........(9.8)}$$

$$\sigma_p = \bar{q}\ K_p + 2\ c\sqrt{K_p} \qquad \text{.........(9.9)}$$

At point A on the left side of the sheetpiling at the dredge line $\bar{q} = 0$ and the net pressure at A is given by

$$\sigma_p - \sigma_a = 2c - (\bar{q} - 2c) = 4\ c - \bar{q} \qquad \text{.......(9.10)}$$

which acts to the right, since $K_a = K_p = 1.0$.

At point B the pressures are given by

$$\sigma_p = \bar{q} + \gamma_e\ D + 2\ c \qquad \text{........(9.11)}$$

which acts to the left, and

$$\sigma_a = \gamma_e D - 2\ c \qquad \text{.......(9.12)}$$

which acts to the right.

Summing pressures $(\sigma_p - \sigma_a)$, and equating to Eqn. 9.10,

$$\gamma_e\ D + \bar{q} + 2\ c - (\gamma_e\ D - 2\ c) = 4c + \bar{q} \qquad \text{........(9.13)}$$

which acts to the left.

$\Sigma\ F_H = 0$ for wall stability gives

$$R_a + \frac{z}{2}\ (4\ c - \bar{q} + 4\ c + \bar{q}) - D\ (4\ c - \bar{q}\ ) = 0 \qquad \text{.........(9.14)}$$

Solving for z,

$$z = \frac{D\ (4c - \bar{q}) - R_a}{4c} \qquad \text{........(9.15)}$$

Summation of the moments about the base gives

$$R_a\ (\bar{y} + D) - \frac{D^2}{2}\ (4c - \bar{q}) + \frac{z^2}{3}\ (4c) = 0 \qquad \text{........(9.16)}$$

Substituting Eqn. (9.15) into Eqn. (9.16) and collecting like terms and simplifying,

$$D^2\ (4c - \bar{q}) - 2DR_a - \frac{R_a(12\ c\bar{y} + R_a)}{2c + \bar{q}} = 0 \qquad \text{.......(9.17)}$$

All the terms are identified in Fig. 9.4 where q = the effective pressure at dredge line.

### 9.2.2 Anchored Sheetpiling

The **free-earth** and **fixed-earth** methods are the two classical methods used to design anchored sheet pile structures. In the design program SHEETPILE, developed for the anchored sheetpile design, the free-earth support method is used and hence discussion is limited to this method only. The free-earth method assumes that the piling is rigid and may rotate at the anchor-rod level, with failure occurring by rotation about the fixed anchor rod. Passive pressure develops in the soil in front of the piling, and active pressure develops behind the wall. After the theoretical embedment is computed, the value may be increased 20 to 40 percent or $K_p$ may be divided by an appropriate safety factor prior to computation of the embedment length. The assumed pressure diagrams and definition of terms are illustrated in Fig. 9.5a and 9.5b. From Fig. 9.5a the distance 'a' from the dredge line to the point of zero pressure is given by

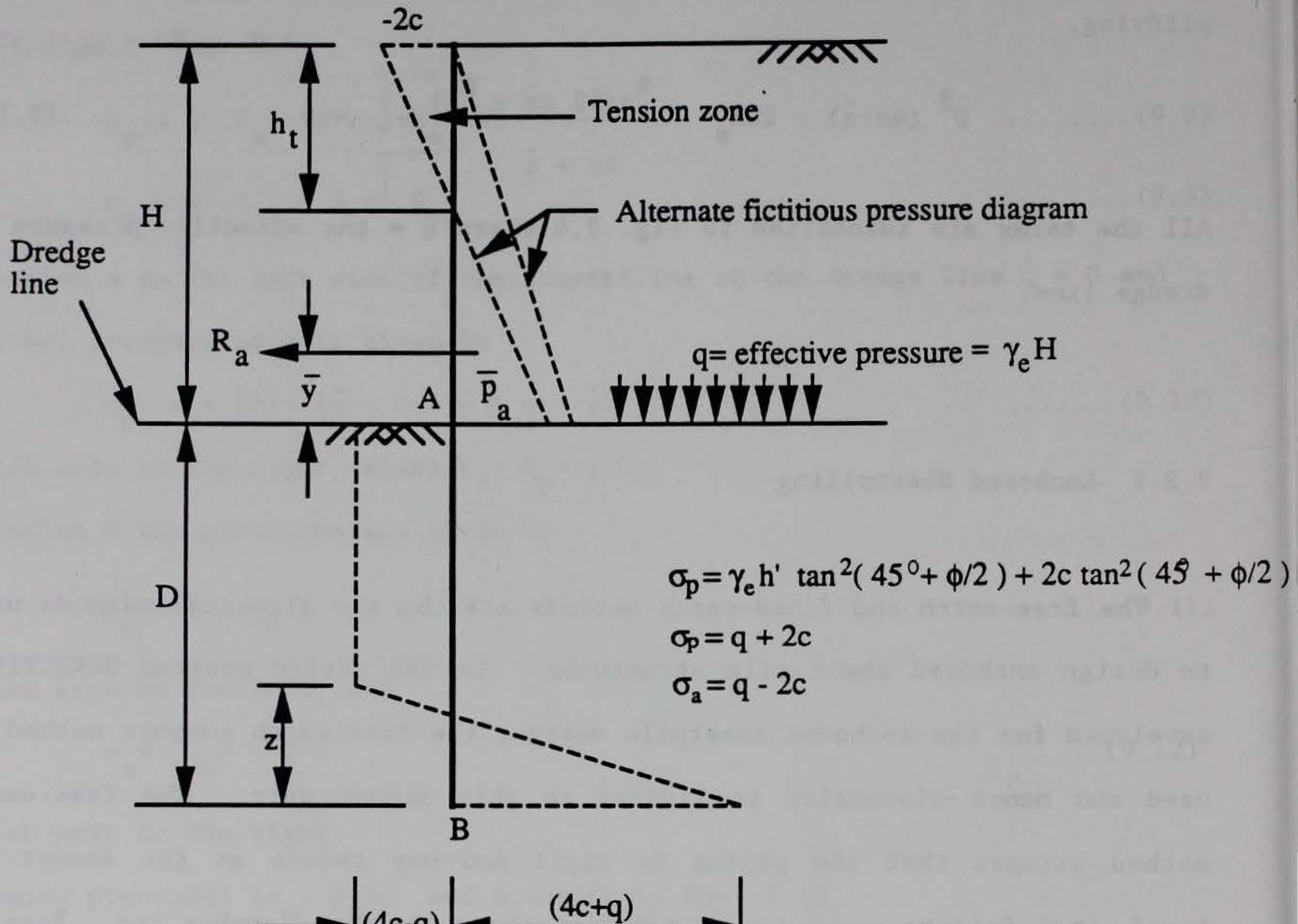



Figure 9.4 Sheetpile in Cohesive Soil. [9.1]

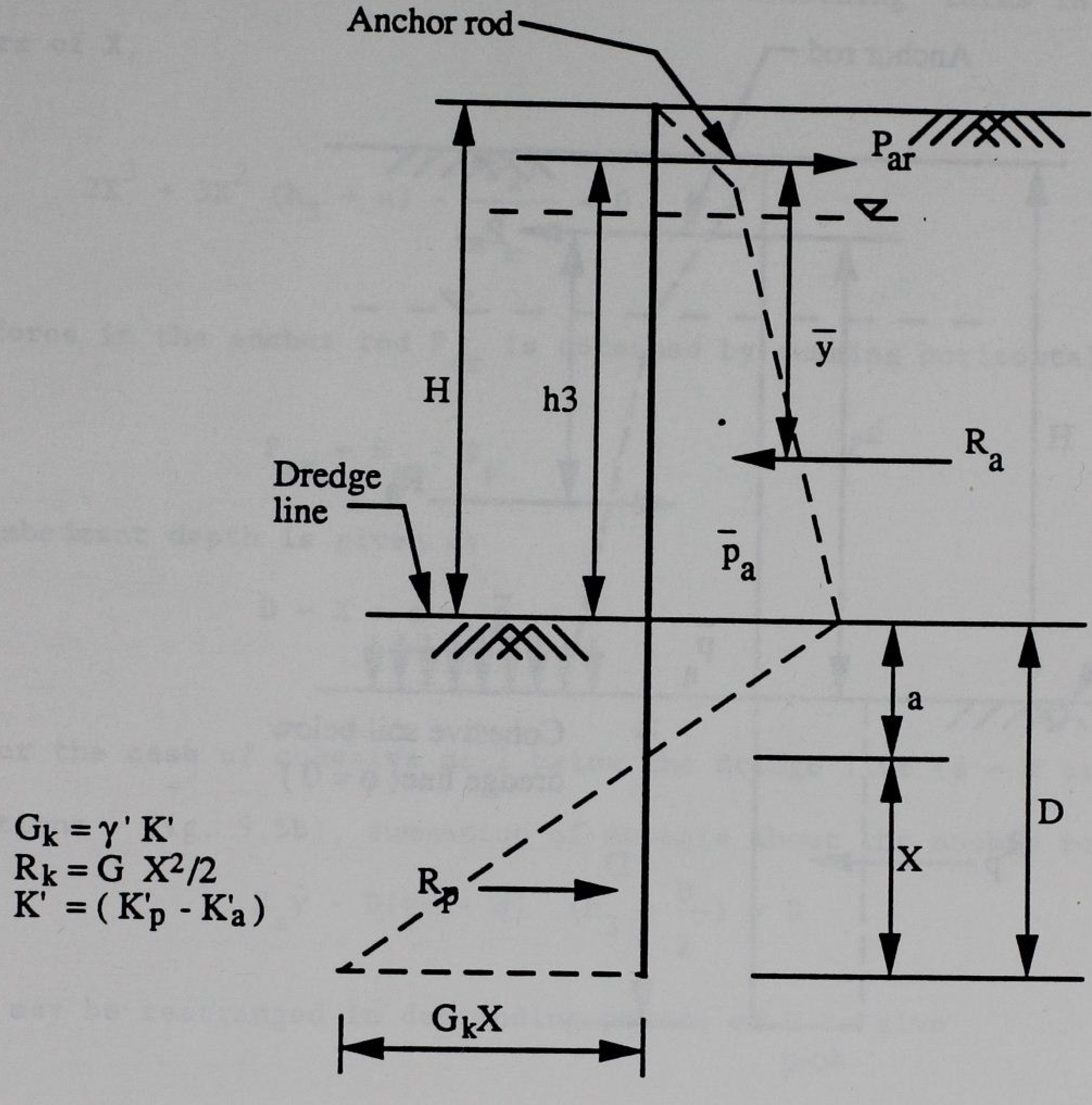



Figure 9.5a Sheetpile in Granular Soil. [9.1]

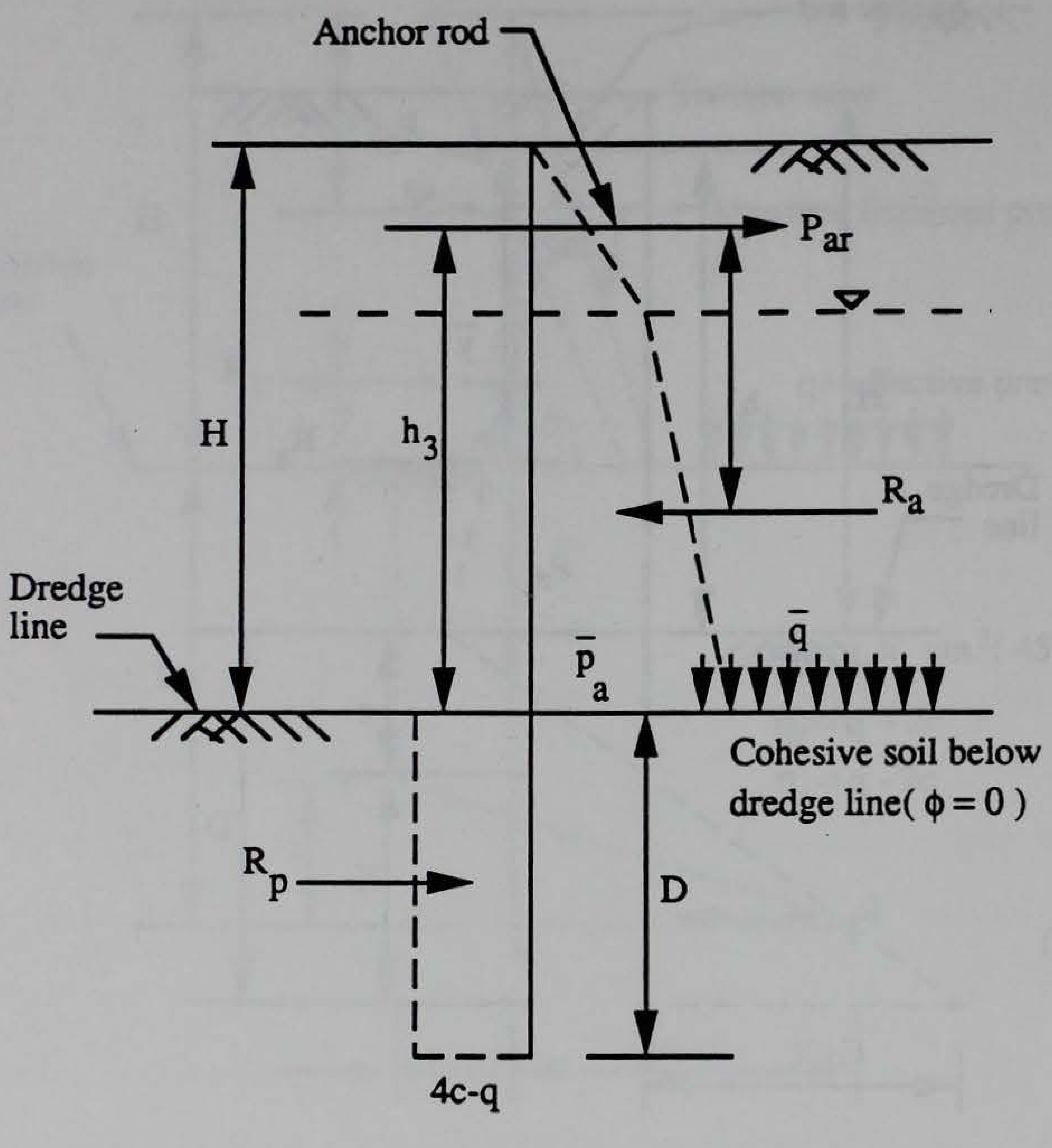



Figure 9.5b Anchored Sheetpile in Granular Soil Backfill and Cohesive Base. [9.1]

$$a = \frac{\bar{P}_a}{\gamma' K'} = \frac{\bar{P}_a}{G_k} \qquad \text{........(9.18)}$$

Summing the moments about the anchor rod and combining terms in descending powers of X,

$$2X^3 + 3X^2 (h_3 + a) - \frac{6R_a\bar{y}}{G_k} = 0 \qquad \text{........(9.19)}$$

The force in the anchor rod $P_{ar}$ is obtained by summing horizontal forces as

$$P_{ar} = R_a - R_p \qquad \text{........(9.20)}$$

The embedment depth is given as

$$D = X + a \qquad \text{........(9.21)}$$

For the case of cohesive soil below the dredge line ($\phi = 0$ or undrained conditions - Fig. 9.5b), summation of moments about the anchor rod gives

$$R_a\bar{y} - D(4c - \bar{q}) \quad (h_3 + \frac{D}{2}) = 0 \qquad \text{........(9.22)}$$

which may be rearranged in descending powers of D to give

$$D^2 + 2Dh_3 - \frac{2\bar{y}\ R_a}{4c - \bar{q}} = 0 \qquad \text{........(9.23)}$$

The forces in the anchor rod can be computed by Eqn (9.20). From inspection of Fig. 9.5b it can be seen that if the passive pressure $\sigma_p \leq 0$ the wall is unstable. This occurs at

$$\frac{c}{q} = \frac{c}{\gamma H} \leq 0.25 \qquad \text{........(9.24)}$$

For a given backfill material, there is a critical value of H beyond which a stable wall cannot be constructed in clay. The ratio of $c/\bar{q}$ is termed the stability number.

When the wall adhesion $c_a$ is taken into account the stability number is approximated by

$$S_n = \frac{c}{\bar{q}} \sqrt{1 + \frac{c_a}{c}} \qquad \text{.........(9.25)}$$

For $c_a = 0.56c$, the stability number becomes

$$S_n = \frac{1.25c}{\bar{q}} \qquad \text{........(9.26)}$$

For a safety factor (FS) of 1 and $\frac{c}{\bar{q}} = 0.25$, the stability number becomes approximately 0.31 considering the wall adhesion. In sheet pile design in clay, the wall should have a stability number given by

$$S_n = 0.3 \times FS \qquad \text{........(9.27)}$$

9.2.2.1 Rowe's moment reduction method applied to anchored sheetpiling

Rowe has proposed moment reductions for sheet pile designs based on the free-earth method. This technique may be used for uniform medium dense to dense silty sand or sand deposits. The Rowe moment-reduction theory is based on the following factors:

i) The relative density of granular soil;

ii) The stability number of cohesive soils given by Eqn. 9.26;

iii) A flexibility number (derived for FPS units)

$$\rho = \frac{H^4}{EI} \left[ \frac{ft^4}{lb\text{-}in^2} \right]$$

where H = total length of piling in ft.;

E = modulus of elasticity in psi and I = moment of inertia in $in^4$, for a unit width of wall

iv) The relative height of piling $\alpha$ and the relative freeboard of the piling $\beta$ as shown in Fig. 9.6.

Curves derived from experimental data [Rowe (1952, 1957)] have been published for selected values of $S_n$, $\alpha$, log $\rho$, and moment ratios. These curves are necessary for a design using this method, and presented in Fig. 9.6a and b for both sand and clay soils.

Design by the moment-reduction method proceeds by first performing a free-earth analysis for the maximum bending moment $M_o$ and the length of pile. Next the appropriate moment-reduction curve from Fig. 9.6 depending on the anchor-rod location, pile length, and soil type, is selected and this curve plotted. From a table of sheetpiling sections, the actual bending moments M of the piles can be computed as M = allowable steel stress x section modulus/unit width. The flexibility coefficient ($\rho = H^4/EI$) is also computed for the corresponding pile section. Using the $M/M_o$ ratio and $\rho$, a second curve is plotted for the various sections and superimposed on the standard curve. The intersection of the two curves may not coincide with a pile section, but any pile section lying above the intersection is satisfactory to use, the pile closest to the intersection being the most economical.

9.2.2.2. Design of wales

Wales are longitudinal members running parallel with and in close contact

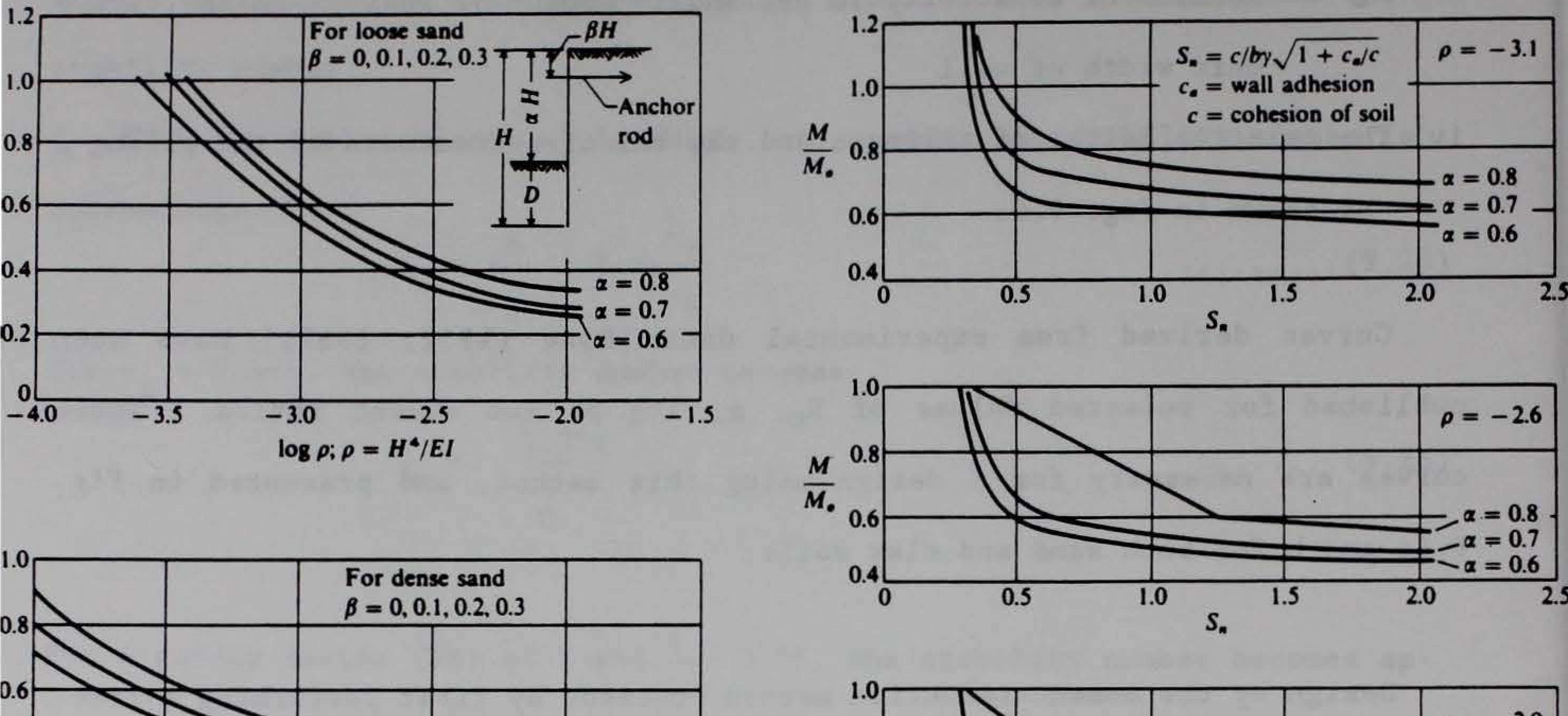



Figure 9.6 Rowe's Moment-reduction Curves for the " Free-earth " Support Method. (a) Sheet Piles in Granular Soil; (b) Sheet Piles in Cohesive Soil. [9.1]

with the wall as shown in Fig. 9.7. They may be located on either the front or backface of the wall. The backface location is desirable in certain cases for both appearance and clearance but will require adequate attachment to the wall by bolting or welding to support the anchor rod pull. Adequate attachment is usually obtained by field welding the wales to the backface. Wales on the front face are somewhat easier to install but also require a hole for the anchor rod through the wall. Wales are usually made of a pair of channels back to back with spacing for the anchor rod. Sometimes a pair of I beams are used, however, wide flange shapes are usually not suitable. Ten cross-sections of steel sections for steel pile are included in the data base.

## 9.3 INTERACTIVE MICROCOMPUTER BASED 'SHEETPILE' PROGRAM IMPLEMENTATION

### 9.3.1. Program Structure

SHEETPILE is an interactive program which gives the design of sheet pile structures and wales. The input data consist of the soil properties and height of earth to be retained; these are used in the computation of bending and toe failure resistances of the structure.

After the user inputs the data, the system checks whether the data conforms to the following facts:

i) If the height of the earth to be retained is less than 10 ft., the cantilever sheetpile is selected and designed; if the height is greater than 10 ft. then the system chooses the anchored sheetpile for further detailed design computations;

ii) The shear strength c divided by the effective pressure q at dredge line should be larger than 0.25;

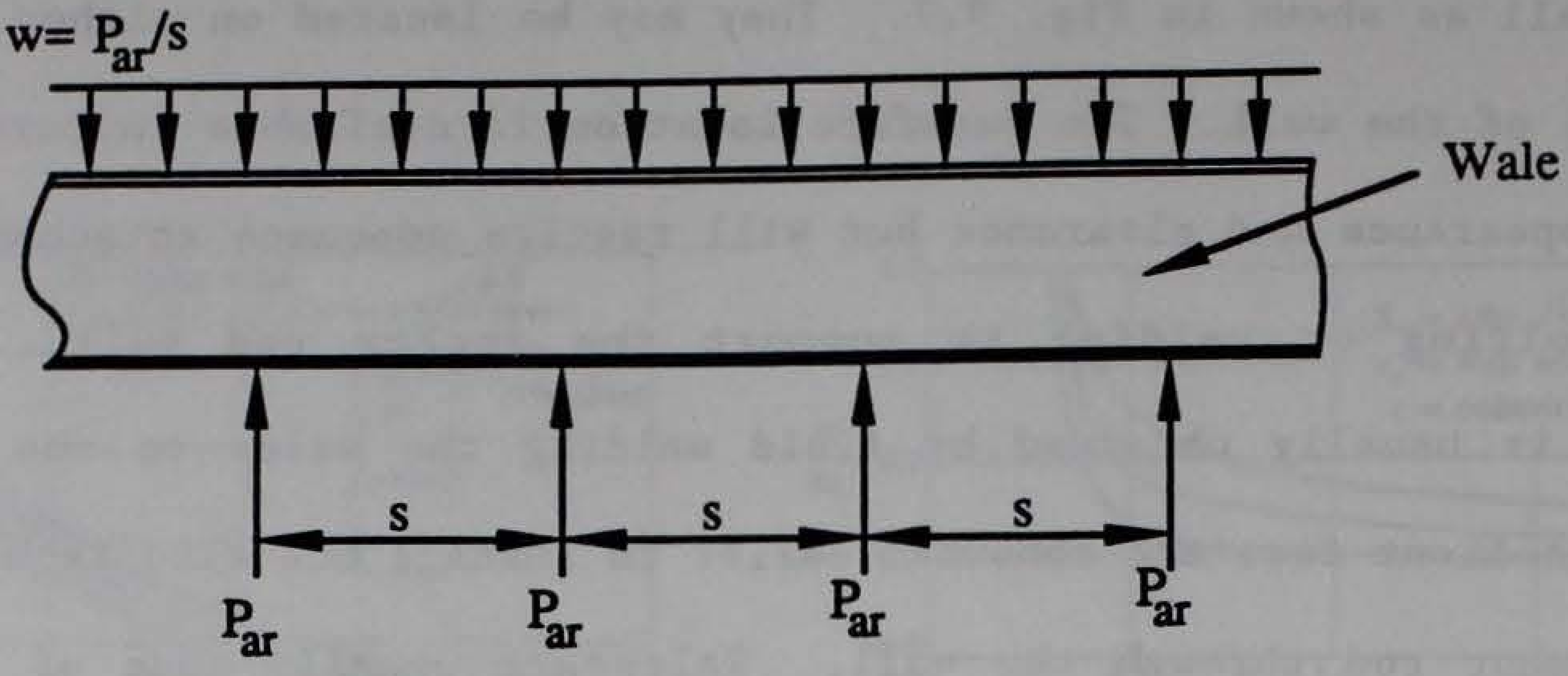



Figure 9.7a Wale Design for Bending Assuming a Uniform Pressure from Anchor Rod. [9.1]

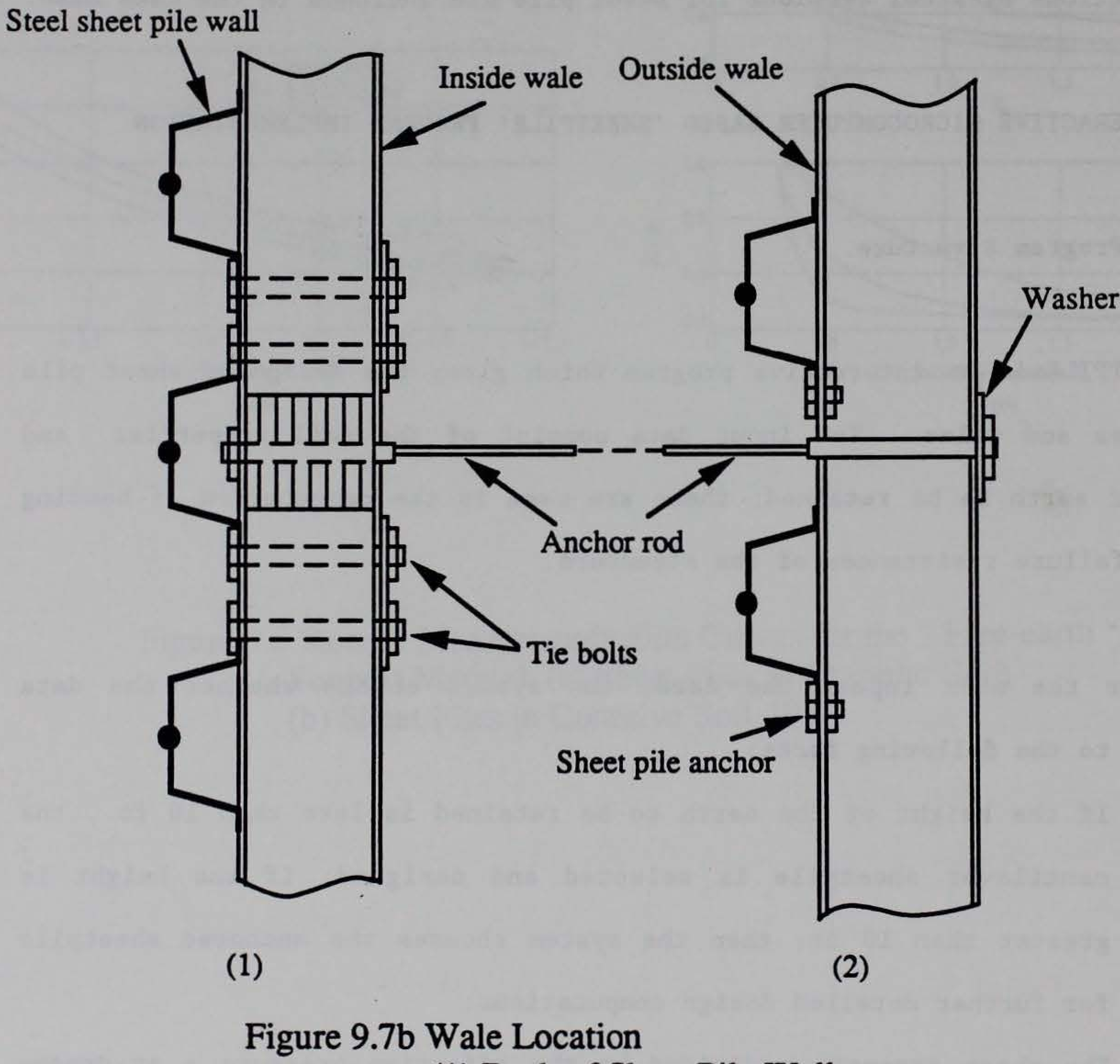



Figure 9.7b Wale Location
(1) Back of Sheet Pile Wall
(2) Front of Sheet Pile Wall. [9.1]

iii) The anchor should be placed above the water table because of erosion problems;

iv) The internal angle of friction $\phi$ is to be between 30° and 36°;

v) The angle of friction $\delta$, between soil and the wall, is to be 0.6 to 0.8 times $\phi$.

If the aforementioned values are not satisfied then the system asks the user to repeat the input. This feature has been incorporated in SHEETPILE to eliminate the possibility of getting unreasonable values in the output. The database includes four cross sections of sheet piles viz. PZ40, PZ35, PZ27 and PZ22 and their section properties and also the cross sectional details of ten wale sections.

The output consists of the dimensions of the structure and the minimum depth below the dredge line. The flowchart of the program SHEETPILE is given in Fig. 9.8. and the listing of SHEETPILE in Appendix G.

9.3.2. Input and Output

The input data is given through an interactive system-user consultation as follows:

- height of the earth to be retained;
- properties of backfill and foundation soil;
- surcharge load;

and

- height of water table.

The output file consists of coefficients of both active and passive

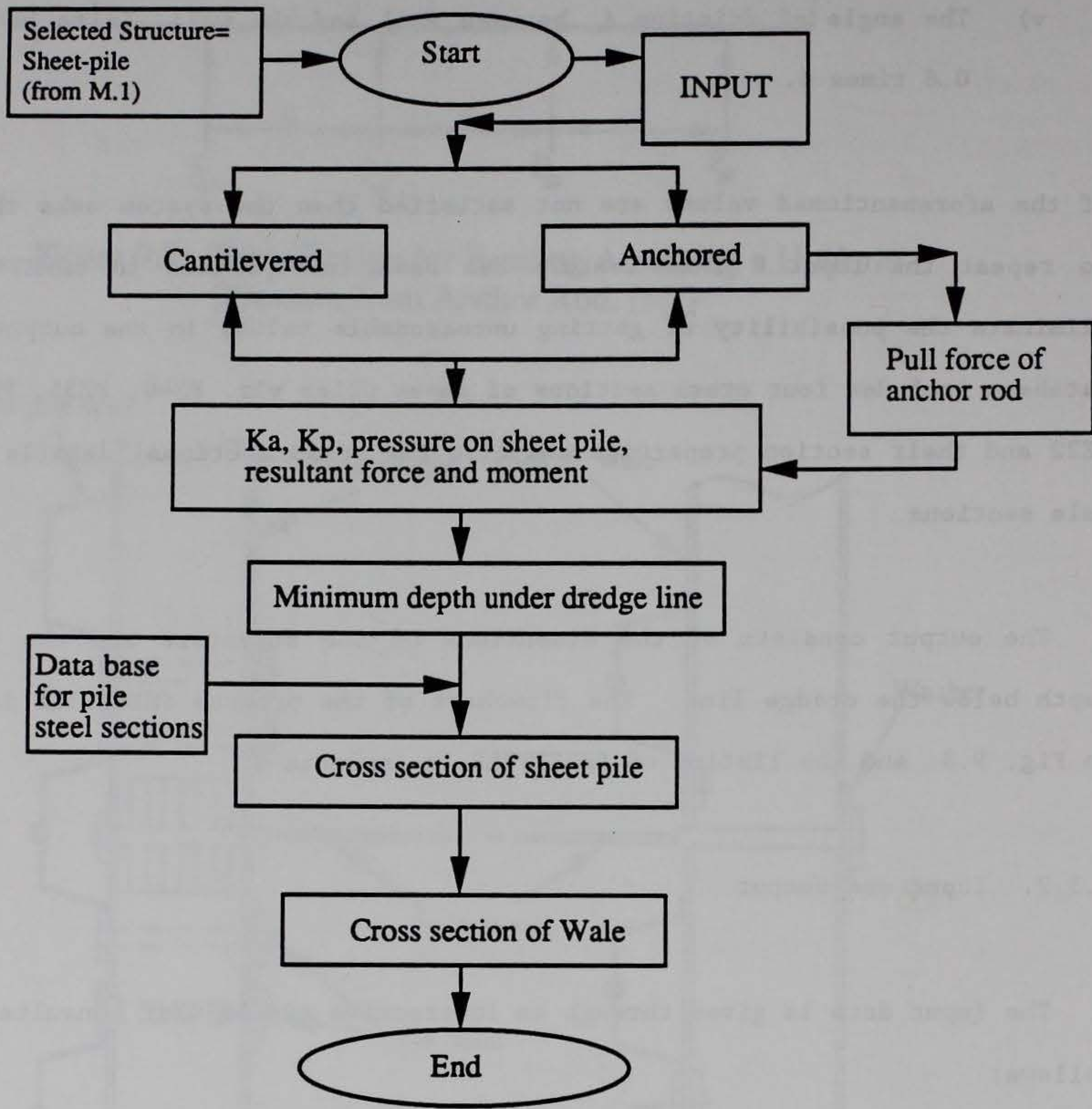



Figure 9.8 Flowchart of Program SHEETPILE for Sheetpile Design

pressure, sections of sheet pile and wales, safety factors against bending and toe failure, maximum moment along sheet pile and pull force of anchor rod.

### 9.3.3 Design Example

The design of an anchored sheetpile is illustrated here. The interface between the user and the system is implemented through a set of pre-defined questions shown in Fig. 9.9. Fig. 9.10 shows the input data as displayed on the screen. The design output is shown in Fig. 9.11.

## REFERENCES


9.1 JE Bowles, Foundation Analysis and Design, 4th Ed., McGraw Hill, 1988, 580-642.

9.2 RB Peck, WE Hanson and TH Thornburn, Foundation Engineering, 2nd Ed., John Wiley & Sons, 1974, 514.

9.3 IK Lee, W. White and OG Ingles, Retaining Structures, Geotechnical Engineering, Pitman, 1983, 245-278.

```
C>sheet

The following consultation is for the design of sheet
pile structures.
Please answer the following questions.

Is this    1.Cantilevered sheet pile;         or
           2. Anchored sheet pile.
If the height of earth to be retained is  greater  than 10 ft,
you should choose anchored sheet pile, otherwise the system
will change it automatically.
>>2
  1. Sand;
  2. Cohesive soil;
  3. Sand backfill on cohesive soil.

  From the given soil types, enter the number corresponding !
  to the type of insitu soil.
>>1
 What is the height of the wall (in ft) ?
>>30
 What is the height of the water table (in ft) ?
>>22
 What is the density of the soil above the water table (in kcf) ?
>>0.11
 What is the internal friction angle of the soil (in degree ) ?
>>30
 What is the frictional angle between soil and wall(in degree) ?
 This value could be 0.6 — 0.8 of the internal friction angle.
>>20
 What is the height of the anchor (in ft) ?
>>26
 What is the surcharge (in ksf) ?
>>0.5_
In this system, there are four cross sections for your sheetpile

1. PZ40   Sx=60.70          Ix=490.80
2. PZ35   Sx=48.40          Ix=360.60
3. PZ27   Sx=30.20          Ix=184.20
4. PZ22   Sx=18.10          Ix=84.50
Please input the yeilding stress!(in ksi)
>>40_
   ratio of moment (in Y axis)=1.56, log(H/EI)(in X axis)=-3.[illegible]
   ratio of moment (in Y axis)=0.97, log(H/EI)(in X axis)=-3.1[illegible]
   ratio of moment (in Y axis)=0.58, log(H/EI)(in X axis)=-2.[illegible]

  The sand is dense or loose (d/l) ?
 >>

  The most appropriate cross section is No. 4 with Sx=18.10  Ix=84.50
  The factor of safety is 1.86

  In this system, there are 10 cross sections for your wale .

  1.  C10x30      Sx=20.70   Ix=103.00
  2.  C12x20.7    Sx=21.50   Ix=129.00
  3.  C12x25      Sx=24.10   Ix=144.00
  4.  C12x30      Sx=27.00   Ix=162.00
  5.  MC12x30.9   Sx=30.60   Ix=183.00
  6.  MC12x32.9   Sx=31.80   Ix=191.00
  7.  MC12x37     Sx=34.20   Ix=205.00
  8.  MC12x35     Sx=36.10   Ix=216.00
  9.  MC12x40     Sx=39.00   Ix=234.00
 10.  MC12x45     Sx=42.00   Ix=252.00

 Please input the yield stress of the wale.(in ksi)
>>40
 The most appropriate cross section for wale is a pair of
 No. 1     Sx=20.70     Ix=103.00
 channels back-to-back with spacing for the anchor rod.

 The factor of safety is 1.80

 Press any key to continue.
 _
```

Figure 9.9 User-System Interactive Consultation for SHEETPILE


Figure 9.10 Screen Display of Input Data for SHEETPILE

```
/*-----------------------------------------*/
/* |        DESIGN OF SHEET PILES        | */
/*-----------------------------------------*/


 *********************
*      OUTPUT      *
*********************
The minimum depth under dredge line: 9.72  ft
The maximum moment:              67.38  kips-ft/ft
The maximum moment occures at:     20.43  ft(from the top)
The pull force at anchor rod:       8.46  kips/ft
------------------------------------------------------------------------

The height of wall:                    30.00  ft
The height of water table:             22.00  ft
The density of soil:                    0.11  kcf
The internal friction angle of soil:   30     degree
The friction angle between
soil and wall:                         20     degree
The height of anchor:                  26.00  ft
The surcharge:                          0.50  ksf

The following are the cross sections of sheet pile available in this system :

 1. PZ40  Sx=60.70      Ix=490.80
 2. PZ35  Sx=48.40      Ix=360.60
 3. PZ27  Sx=30.20      Ix=184.20
 4. PZ22  Sx=18.10      Ix=84.50

The section suitable for this sheet pile is #4 (PZ22).


The following are the cross sections of wale available in this system :

 1. C10x30     Sx=20.70   Ix=103.00
 2. C12x20.7   Sx=21.50   Ix=129.00
 3. C12x25     Sx=24.10   Ix=144.00
 4. C12x30     Sx=27.00   Ix=162.00
 5. MC12x30.9  Sx=30.60   Ix=183.00
 6. MC12x32.9  Sx=31.80   Ix=191.00
 7. MC12x37    Sx=34.20   Ix=205.00
 8. MC12x35    Sx=36.10   Ix=216.00
 9. MC12x40    Sx=39.00   Ix=234.00
10. MC12x45    Sx=42.00   Ix=252.00

The section suitable for  the wale is #1 (C 10x30).


************************************************************
```

Figure 9.11 Output File of SHEETPILE

## CHAPTER 10

## CONCLUSIONS

The expert system (ES) RETAININGEARTH, is a microcomputer-based developmental prototype which has the capability to select the most appropriate retaining structure from a list of ten typical structures for a given set of user input conditions. The organization of the knowledge base in the selection module, SELECTWALL permits the addition of more knowledge to the knowledge base.

RETAININGEARTH is a rule-based expert system developed using the M.1 knowledge engineering shell. The rules in this knowledge base show that rule-based systems provide an effective and easily readable means of encoding knowledge. The formulation of the knowledge base SELECTWALL and its link to the design module are presented together with the source code. A concise treatment of the design methodologies of the retaining structures, viz. the concrete gravity and cantilever walls, gabions, reinforced earth and sheet-pile structures is presented together with the design flowcharts. The development of a design module which consists of codes for the designs of five different structures, is presented with illustrations of case studies.

The expert system RETAININGEARTH in its present form is a developmental prototype and addition of more heuristic knowledge into the knowledge base is desirable, as and when more knowledge becomes available, and this will make the system a more effective tool for the designer. The ES clearly demonstrates that both algorithmic and heuristic approaches involved in engineering design and decision making processes can be effectively coded into facts and rules in the knowledge base of the system. The degree of uncertainty associated with parameters, viz. material/labor availability, equipment access to site, and influence of climatic conditions favorable for project construction, can be taken into account in the structure selection process, using fuzzy logic in the knowledge-based expert system.

# APPENDIX A

## LISTING OF M.1 SELECTWALL

```
prefix begin.
prefix the.
prefix end.

initialdata = [ begin the consultation, end the consultation ].

begin message = [
  nl,'    ---------------------------------------------------- ',
  nl,'    |    SELECTION AND DESIGN OF RETAINING STRUCTURE   | ',
  nl,'    ---------------------------------------------------- ',
  nl,
  nl,'    This is an expert system for the selection and design',
  nl,'    of retaining structures. This system helps to select',
  nl,'    the most appropriate type(s) of retaining structure(s)',
  nl,'    and then design the chosen structure.',
nl,'      You may type ''why''to get the explanation to any question.',
nl,
nl,
nl,
nl].

nocache(begin message).

question(begin signal) =
   [nl,'Are you ready to begin the consultation(yes/no)?',nl].

legalvals(begin signal) = [yes,no].

nocache(begin signal).


question(continue) =
   [nl,'Please type ''yes'' to continue !',nl].

legalvals(continue) = [yes].

nocache(continue).


rule-1: if begin message = M and
           display(M) and
           begin signal and
           display('\f')
        then begin the consultation.

rule-2: if not begin signal and
  display("Issue the 'go' command when you are ready to begin.") and
  do(abort)
        then begin the consultation.

/*-------------------------LOCATION----------------------------*/

multivalued(location).

question(location) =
```

```
'Which of the following list of sites best describes the
location of the proposed retaining structure?'.
legalvals(location) =
     [excavation,roadway,sidehill,abutment,forest-area,buiding-
related-excavation, mountainous-terrain,waterfront-area,railway,
building,marine].
automaticmenu(location).
enumeratedanswers(location).

explanation(location) = [
          'The location where the structure is to be built is
a very important consideration in the selection of retaining
walls. You may choose (only one) location which is closest to the
above locations.'].

/*-------------------CONSULTATION------------------------------*/

legalvals(height)=real.
question(height) =
  'What is the height of the earth to be retained (in feet)?'.

question(length) =
  'What is the length of the structure (in feet)?'.

/*--------------------SITE-CONDITION----------------------*/

question(site) =
      'Is the site a fill or cut situation ?'.
legalvals(site) = [ fill,cut ].
automaticmenu(site).
enumeratedanswers(site).

explanation(site) = [
     'Your response to this question leads you to the two major
catagories under which the retaining structures are
grouped. The fill catagory consists of the gravity,cantilever,
counterfort,reinforced-earth,gabions and the crib walls.The cut
situation consists of the sheet-pile,slurry wall, soil-nailed-
system and the tied-back wall.'].

question(bsiteb) =
'Does the site in question have buildings around it (yes/no)?'.

legalvals(bsiteb) = [ yes,no ].

/*----------------------MATERIAL---------------------------*/
question(materialcg) =
      'Is concrete and steel locally available(yes/no)?'.
legalvals(materialcg) = [ yes,no ].

explanation(materialcg) = [nl,
     'The data entered by you may lead to the selection of
concrete gravity, counterfort or cantilever retaining structures,
subject to the availability of material in this region.
     Even if local material is not available, if the cost of
```

```
material is low and the transportation easily done, then you  may
answer YES to this question.',nl].

question(materialcrb) =
      'Is  granular backfill(minimum angle of internal friction=28
deg. or noncohesive clean broken stone or coarse sand and gravel
soil locally available (yes/no)?'.
legalvals(materialcrb) = [ yes,no ].

explanation(materialcrb) = [nl,
         'The  data entered by you may lead to the selection  of
the  crib wall,  subject to the availability of material in  this
region.
         Even if local material is not available, if the cost of
the  material is low and the transportation easily done,then  you
may answer YES to this question.',nl].

question(materialreinf) =
      'Are  the following materials locally available:
1.Medium  to  fine  grained  sand,  or  silty  sand  to  crushed
sandstone,  crushed  dolerite,  greystone  gravel  or  screened
ironstone.
2.Backfill has not more than 15% passing 200um sieve.
3.The angle of internal friction of backfill > 25 deg.
4.Cranes for excavation.
5.Compacting equipments.
6.Thin metal strips or rods or geotextile strips,  sheets or wire
grids or other reinforcing strips.
7.Facing units to prevent soil erosion between the  reinforcement
and for better appearance. ---- (yes,no)?'.

legalvals(materialreinf) = [ yes,no ].

explanation(materialreinf) = [nl,
         'The  data entered by you may lead to the selection  of
the  reinforced-earth  retaining  structure,  subject  to  the
availability of materials in this region.
         Even if local material is not available, if the cost of
material is low and the transportation easily done,  then you may
answer YES to this question.',nl].

question(materialgab) =
    'Are the following materials locally available:
         1.  Gabion  baskets(mesh)  which  must  satisfy  high
mechanical  resistance,  high  resistance  to  corossion,  good
deformability and should not unravel easily.
         2.  4"  to  10" diameter stone  which  is  non-friable,
weather-resistant, and of high density.
         3.Bucket  loader,  back hoe,  or clamshell for  dumping
stone into basket. ---- (yes/no)?'.
legalvals(materialgab) = [ yes,no ].

explanation(materialgab) = [nl,
    'The data entered by you may lead to the selection of gabion
retaining  structure,  subject to the availability of material in
```

```
this region.
        Even if local material is not available, if the cost of the
material is low and the transportation easily done, then you may
answer YES to this question.If it is a large project, you are
advised to answer NO to this question, if the aforementioned
materials are not available.',nl].

question(materialslur) =
        'Are the following materials locally available :
            1.Heavy density and sufficiently viscous liquid - like
a mixture of barium sulphate and bentonite clay.
            2.Trench        excavation        equipments        - includes
grab, scoop, drilling rig, clamshell bucket, etc.----(yes/no)?'.
legalvals(materialslur) = [ yes,no ].

explanation(materialslur) = [nl,
        'The data entered by you may lead to the selection of
slurry-wall, subject to the availability of material in this
region.
        Even if local material is not available, if the cost of the
material is low and the transportation easily done, then you may
answer YES to this question.',nl].

question(materialtied) =
        'Are soldier pile, auger, grouted material and tied material
locally available (yes/no)?'.
legalvals(materialtied) = [ yes,no ].

explanation(materialtied) = [nl,
        'The data entered by you may lead to the selection of the
tied-back wall, subject to the availability of material in this
region.
        Even if local material is not locally available, if the cost
of material is low and the transportation easily done, then you
may answer YES to this question.',nl].

question(materialpile) =
        'Are sheet piles of sufficient length available? Also, are
equipments    needed    for    driving    piles ( like    sheet-pile
hammer,    vibratory    driving    device,    crane,    etc. )
available (yes/no)?'.
legalvals(materialpile) = [ yes,no ].

explanation(materialpile) = [nl,
            'The data entered by you may lead to the selection of
the sheet-pile retaining structure, subject to the availability
of construction material in this region.
            Even if local material is not available, if the cost of
the material is low and the transportation easily done,then you
may answer YES to this question.',nl].

question(materialnail) =
'Are the materials related to soil nailing available (yes/no)?'.
legalvals(materialnail) = [yes,no].
```

```
explanation(materialnail) = [nl,
        'The  data entered by you may lead to the selection  of
the  soil-nailed system,  subject to the availability of constru-
ction material in this region.
        Even if local material is not available, if the cost of
the  material is low and the transportation easily done,then  you
may answer YES to this question.',nl].

/*---------------------FOUNDATION---------------------------*/
question(foundation) =
     'How would you rate the foundation of the site?'.
legalvals(foundation) =
     [ good,poor ].
automaticmenu(foundation).
enumeratedanswers(foundation).
/*----------------------LABOR-----------------------------*/

question(labora) =
     'Is skilled labor ( concretor,  formworker,  reinforcer  and
laborers) available (yes/no) ?'.
legalvals(labora) = [ yes,no ].

question(laborb) =
'Is  this  the type of wall that is locally  constructed  in this
region  under similar conditions (You  are advised  to  read  the
explanation  (type ''why'')  before  responding to this question)
(yes/no) ?'.
legalvals(laborb) = [ yes,no ].
explanation(laborb) = [nl,
     'This question is asked because if a similar wall is already
existing  in this region,  then the contractor in this region has
the experience to carry out the work. Besides this reason he will
also   be   well   equipped  for   undertaking   this   kind   of
construction.If   this  is  a  remote  region  and  no  retaining
structure exists in the nearby region, then too you should answer
YES to this question.',nl].

/*----------------------VERTICAL LOAD--------------------*/
question(vertload) =
     'Will  the  proposed  wall  be  required  to  take  vertical
load(yes,no) ?'.
legalvals(vertload) = [ yes,no ].
automaticmenu(vertload).
enumeratedanswers(vertload).

 /*--------------------AESTHETICS----------------------*/

question(aesthetics) =
     'Is the building of the retaining structure aesthetically
 acceptable in this region(yes,no)?'.

legalvals(aesthetics) = [ yes,no ].

explanation(aesthetics) = [nl,
        'It  is  necessary  that  the  retaining  structure  be
```

```
aesthetically acceptable.It should fit in the existing structure
and environment.
        Generally, concrete structures do not fit well in
rural settings.However, if appearance of the structure and
landcape is not very important,you may answer YES to this
question.Gabions have the added advantage of fitting in well both
in urban as well as in rural settings - since their fillers are
stones which allow natural vegetation.',nl].

/* ------------------------right-of-way----------------------*/

question(rway) =
     'Is there a space restriction behind the structure face
(yes/no) ?'.

legalvals(rway) = [ yes,no ].

explanation(rway) = [nl,
     'When considering compacted earth systems, a relatively
large space is required behind the structure face as compared to
that needed for construction of conventional walls(the length of
the reinforcing elements is at least 0.7 times the wall height).
Right-of-way is a very important consideration in the selection
of retaining structures.',nl].

/*-----------------------soil-condition-----------------------*/

question(soila) =
     'Is the soil condition firm at the proposed site(yes/no)?'.

legalvals(soila) = [ yes,no ].

explanation(soila) = [nl,
     'Since concrete gravity wall is a mass intensive
structure,soil needs to have enough bearing capacity to withstand
settlement.A firm soil is considered to be suitable for this
type of retaining structure.',nl].

question(soilb) =
     'Does the soil have the following properties:
1.Soils of high plasticity.
2.Granular soils with no coherance.
3.Soils with much water content.
4.Large size(boulders).-------(yes/no)?'.

legalvals(soilb) = [ yes,no ].

explanation(soilb) = [nl,
     'The data entered by you may lead to the selection of tied-
back retaining structure, subject to favorable soil condition.A
favorable soil for this type of retaining structure will NOT have
the properties stated in this question.',nl].

question(soilc) =
     'Is the subsoil suitable for driving sheet-piles (yes/no)?'.
```

```
legalvals(soilc) = [ yes,no ].

explanation(soilc) = [nl,
      'The data you have entered may lead to the selection of the
sheet-pile   retaining  structure,   subject  to  favorable  soil
condition.If  the  subsoil  contains many boulders  and  is  very
dense,then this type of soil is not suitable for this  structure.
However, if a system of soldier beams and lagging can be used then
you may answer YES to this question.',nl].

question(soild) =
      'Is the soil capable of retaining itself unsupported (around
5 ft. at a time)(yes/no)?'.

legalvals(soild) = [ yes,no ].


question(soile) =
   'Do the in-situ soils have a highly acidic nature (yes/no)?'.
legalvals(soile) = [ yes,no ].

/*------------------------construction-time-------------------*/
multivalued(time).

question(time) =
      'Which  of the following  briefly  states the best relative
construction time for the proposed project?'.

legalvals(time) =  [ short,medium,long ].

automaticmenu(time).
enumeratedanswers(time).

/*---------------------noise-pollution------------------------*/

question(noise) =
      'Is  the site in the premises of a hospital or similar  area
where  the noise of construction ( like drilling,hammmering,etc.)
is a critical issue (yes/no) ?'.

legalvals(noise) = [ yes,no ].

explanation(noise) = [nl,
      'Any  present-day  construction  involves the  use  of  heavy
machinery  which  make a lot of noise while  being  operated.  In
certain  areas,  like in hospital premises,  it is mandatory that
the  surrounding region be kept tranquil and  peaceful.  In  such
cases  operations  which  make  a  lot of noise  is  to  be  kept
minimum.',nl].

/*--------SELECTION OF POSSIBLE TYPE OF RETAINING WALLS-------*/

multivalued(possible-wall).

rule-10: if site = fill and
```

```
            not(location = excavation) or
            not(location = forest-area) or
            not(location = building-related-excavation) or
            not(location = mountainous-terrain) and
            (height = H and H <= 5) and
            (length = L and L <= 150) and
            rway = yes or
            rway = no and
            time = long and
            soila = yes and
            not(foundation = poor) and
            vertload = yes or
            vertload = no and
            noise = yes or
            noise = no
    then possible-wall = gravity-retaining-wall.

rule-11: if site = fill and
            not(location = excavation) or
            not(location = forest-area) or
            not(location = building-related-excavation) or
            not(location = mountainous-terrain) and
            height = H and ( H > 5 and  H < 10 ) and
            (length = L and L <= 150) and
            rway = yes or
            rway = no and
            time = long and
            soila = yes and
            not(foundation = poor) and
            vertload = yes or
            vertload = no and
            noise = yes or
            noise = no
         then possible-wall = cantilever-retaining-wall.

rule-12: if site = fill and
            not(location = excavation) or
            not(location = forest-area) or
            not(location = building-related-excavation) or
            not(location = mountainous-terrain) and
            (height = H and H > 15) and
            (length = L and L <=100) and
            rway = yes or
            rway = no and
            time = long and
            soila = yes and
            not(foundation = poor) and
            vertload = yes or
            vertload = no and
            noise = yes or
            noise = no
      then possible-wall = counterfort-retaining-wall.

rule-13: if site = fill and
            not(location = excavation) or
```

```
        not(location = forest-area) or
        not(location = building-related-excavation) or
        not(location = waterfront-area) or
        not(location = mountainous-terrain) and
        height = H and
        H < 6 and
        length = L and
        L < 75 and
        rway = no and
        time = short or
        time = medium or
        time = long and
        soila = yes or
        soila = no and
        foundation = good or
        foundation = poor and
        vertload = no and
        noise = yes or
        noise = no
     then possible-wall = crib-retaining-wall.

rule-14: if site = fill and
        not(location = excavation) or
        not(location = building-related-excavation) and
        (height = H and H < 20) and
        (length = L and L > 75) and
        rway = no and
        soile = no and
        time = medium or
        time = long and
        soila = yes and
        foundation = good or
        foundation = poor and
        vertload = yes or
        vertload = no and
        noise = no
     then  possible-wall  =  reinforced-earth-wall.

rule-15: if site = fill and
        not(location = excavation) or
        not(location = building-related-excavation) or
        not(location = building) and
        height = H and
        (H > 6 and H < 34) and
        length = L and
        (L > 0) and
        rway = yes or
        rway = no and
        not(time = short) and
        soila = yes and
        soile = no and
        foundation = good or
        foundation = poor and
        vertload = yes or
        vertload = no
```

```
then possible-wall = gabion-retaining-wall.

rule-16: if site = cut and
           not(location = roadway) or
           not(location = railway) or
           not(location = forest-area) or
           not(location = mountainous-terrain) or
           not(location = waterfront-area) and
           (height = H and H > 10) and
           soila = yes and
           not(time = short) and
           soile = no and
           bsiteb = yes or
           bsiteb = no
        then possible-wall = slurry-retaining-wall.

rule-17: if site = cut and
           not(location = abutment) or
           not(location = building-related-excavation) or
           not(location = building) or
           not(location = waterfront-area) and
           (height = H and H >= 15) and
           rway = no and
           not(time = short) and
           soilb = no
        then possible-wall = tied-back-retaining-wall.

rule-18: if site = cut and
           location = excavation and
           time = short or
           time = long or
           time = medium and
           rway = no and
           soilb = no
        then possible-wall = tied-back-retaining-wall.

rule-19: if site = cut and
           not(location = building) or
           not(location = forest-area) and
           (height = H and H <= 20) and
           rway = no and
           noise = no and
           not(time = short) and
           soilc = yes and
           bsiteb = no
        then possible-wall = sheet-pile-retaining-wall.

rule-20: if site = cut and
           not(location = waterfront-area) or
           not(location = marine) and
           noise = no or
           noise = yes and
           rway = yes or
           rway = no and
           height = H and
```

```
            ( H > 5 ) and
            not(time = short) and
            soild = yes
         then possible-wall = soil-nailed-system.

rule-21: if possible-wall is sought and
         listof(possible-wall) = [ONE] and
display([nl,'The  type  of retaining wall which  may  be suitable for th
continue and
display('\f')
         then user-informed.

rule-22: if possible-wall is sought and
            listof(possible-wall) = [ONE,TWO] and
         display([nl,'The possible types of retaining walls which suit t
         continue and
         display('\f')
     then user-informed.

rule-23: if possible-wall is sought and
      listof(possible-wall) = [ONE,TWO,THREE] and
     display([nl,'The  types of retaining walls which  suit  the mention
     continue and
     display('\f')
        then user-informed.

rule-3: if possible-wall is sought and
      listof(possible-wall) = [ONE,TWO,THREE,FOUR] and
     display([nl,'The  types  of retaining walls which  suit  the
mentioned conditions are ',ONE,',',TWO,',',THREE,' and ',FOUR,'. M.1 is
     continue and
     display('\f')
        then user-informed.

rule-4: if possible-wall is sought and
      listof(possible-wall) = [ONE,TWO,THREE,FOUR,FIVE] and
     display([nl,'The  types  of retaining walls which  suit  the mentio
     continue and
     display('\f')
        then user-informed.

rule-5: if possible-wall is sought and
      listof(possible-wall) = [ONE,TWO,THREE,FOUR,FIVE,SIX] and
     display([nl,'The  types  of retaining walls which  suit  the mentio
     continue and
     display('\f')
        then user-informed.


/*---------------USER HAPPY WITH SELECTION-----------------*/

question(user-happy) = [ 'Are you happy with this selection (yes/no)?'].
legalvals(user-happy) = [yes,no].
```

```
/*------------ USER INFORMED OF SELECTED WALL-----------------*/

rule-24: if possible-wall is sought and
            user-informed = yes and
            possible-wall = gravity-retaining-wall and
            materialcg = yes and
            labora = yes and
            listof(possible-wall) = [ONE]
         then appropriate-structure = gravity-retaining-wall.

rule-25: if materialcg = yes and
            labora = yes
         then informg.

rule-26: if possible-wall = cantilever-retaining-wall and
            user-informed = yes and
            informg = yes and
            listof(possible-wall) = [ ONE ]
         then appropriate-structure = cantilever-retaining-wall.

rule-27: if possible-wall = counterfort-retaining-wall and
            user-informed = yes and
            informg = yes and
            listof(possible-wall) = [ ONE ]
         then appropriate-structure = counterfort-retaining-wall.

rule-28: if possible-wall = crib-retaining-wall and
            user-informed = yes and
            materialcrb = yes and
            labora = yes and
            listof(possible-wall) = [ ONE ]
         then appropriate-structure = crib-retaining-wall.

rule-29: if possible-wall = reinforced-earth-wall and
            user-informed = yes and
            materialreinf = yes and
            labora = yes and
            listof(possible-wall) = [ ONE ]
         then appropriate-structure = reinforced-earth-wall.

rule-30: if possible-wall = gabion-retaining-wall and
            user-informed = yes and
            materialgab = yes and
            labora = yes and
            listof(possible-wall) = [ONE]
         then appropriate-structure = gabion-retaining-wall.

rule-31: if possible-wall = tied-back-retaining-wall and
            user-informed = yes and
            materialtied = yes and
            labora = yes and
            listof(possible-wall) = [ONE]
         then appropriate-structure = tied-back-retaining-wall.

rule-32: if possible-wall = slurry-retaining-wall and
```

```
        user-informed = yes and
        materialslur = yes and
        labora = yes and
        listof(possible-wall) = [ONE]
     then appropriate-structure = slurry-retaining-wall.

rule-33: if possible-wall = sheet-pile-retaining-wall and
        user-informed = yes and
        materialpile = yes and
        labora = yes and
        listof(possible-wall) = [ONE]
     then appropriate-structure = sheet-pile-retaining-wall.

rule-34: if possible-wall = soil-nailed-system and
        user-informed = yes and
        materialnail = yes and
        labora = yes and
        listof(possible-wall) = [ONE]
     then appropriate-structure = soil-nailed-system.

rule-35: if materialcrb = yes and
        labora = yes
     then informcrb.

rule-36: if materialpile = yes and
        labora = yes
     then informpile.

rule-37: if materialslur = yes and
        labora = yes
     then informslur.

rule-38: if materialnail = yes and
        labora = yes
     then informnail.

rule-39: if materialreinf = yes and
        labora = yes
     then informerth.

rule-40: if materialtied = yes and
        labora = yes
     then informtied.

rule-41: if materialgab = yes and
        labora = yes
     then informgab.

rule-42: if possible-wall = gravity-retaining-wall and
        user-informed = yes and
        informg = yes and
        height = H and
        ( H < 5 ) and

        listof(possible-wall) = [ ONE,TWO ]
```

```
        then appropriate-structure = gravity-retaining-wall.

rule-43: if possible-wall = gravity-retaining-wall and
            user-informed = yes and
            informg = yes and
            informerth = yes and
            height = H and
            ( H < 5 ) and
            listof(possible-wall) = [ ONE,TWO,THREE ]
        then appropriate-structure = gravity-retaining-wall.

rule-44: if possible-wall = cantilever-retaining-wall and
            user-informed = yes and
            informg = yes and
            height = H and
            ( H < 8 ) and
            listof(possible-wall) = [ ONE,TWO ]
        then appropriate-structure = cantilever-retaining-wall.

rule-45: if possible-wall = cantilever-retaining-wall and
            user-informed = yes and
            informg = yes and
            height = H and
            ( H < 8 ) and
            listof(possible-wall) = [ ONE,TWO,THREE ]
        then appropriate-structure = cantilever-retaining-wall.

rule-46: if possible-wall = counterfort-retaining-wall and
            user-informed = yes and
            informg = yes and
            height = H and
            ( H > 25 ) and
            listof(possible-wall) = [ ONE,TWO ]
        then appropriate-structure = counterfort-retaining-wall.

rule-47: if possible-wall = counterfort-retaining-wall and
            user-informed = yes and
            informg = yes and
            height = H and
            ( H > 25 ) and
            listof(possible-wall) = [ ONE,TWO,THREE ]
        then appropriate-structure = counterfort-retaining-wall.

rule-48: if possible-wall = gabion-retaining-wall and
            user-informed = yes and
            informgab = yes and
            listof(possible-wall) = [ ONE,TWO ]
        then appropriate-structure = gabion-retaining-wall.

rule-49: if possible-wall = gabion-retaining-wall and
            user-informed = yes and
            informgab = yes and
            listof(possible-wall) = [ ONE,TWO,THREE ]
        then appropriate-structure = gabion-retaining-wall.
```

```
rule-50: if possible-wall = reinforced-earth-wall and
            user-informed = yes and
            informerth = yes and
            listof(possible-wall) = [ ONE,TWO ]
         then appropriate-structure = reinforced-earth-wall.

rule-51: if possible-wall = reinforced-earth-wall and
            user-informed = yes and
            informerth = yes and
            listof(possible-wall) = [ ONE,TWO,THREE ]
         then appropriate-structure = reinforced-earth-wall.

rule-52: if possible-wall = crib-retaining-wall and
            user-informed = yes and
            informcrb = yes and
            listof(possible-wall) = [ ONE,TWO ]
         then appropriate-structure = crib-retaining-wall.

rule-53: if possible-wall = crib-retaining-wall and
            user-informed = yes and
            informcrb = yes and
            listof(possible-wall) = [ ONE,TWO,THREE ]
         then appropriate-structure = crib-retaining-wall.

rule-54: if possible-wall = tied-back-retaining-wall and
            user-informed = yes and
            informtied = yes and
            listof(possible-wall) = [ ONE,TWO ]
         then appropriate-structure = tied-back-retaining-wall.

rule-55: if possible-wall = tied-back-retaining-wall and
            user-informed = yes and
            informtied = yes and
            listof(possible-wall) = [ ONE,TWO,THREE ]
         then appropriate-structure = tied-back-retaining-wall.

rule-56: if possible-wall = slurry-retaining-wall and
            user-informed = yes and
            informslur = yes and
            listof(possible-wall) = [ ONE,TWO ]
         then appropriate-structure = slurry-retaining-wall.

rule-57: if possible-wall = slurry-retaining-wall and
            user-informed = yes and
            informslur = yes and
            listof(possible-wall) = [ ONE,TWO,THREE ]
         then appropriate-structure = slurry-retaining-wall.

rule-58: if possible-wall = sheet-pile-retaining-wall and
            user-informed = yes and
            informpile = yes and
            listof(possible-wall) = [ ONE,TWO ]
         then appropriate-structure = sheet-pile-retaining-wall.

rule-59: if possible-wall = sheet-pile-retaining-wall and
```

```
         user-informed = yes and
         informpile = yes and
         listof(possible-wall) = [ ONE,TWO,THREE ]
      then appropriate-structure = sheet-pile-retaining-wall.

rule-60: if possible-wall = soil-nailed-system and
         user-informed = yes and
         informnail = yes and
         listof(possible-wall) = [ ONE,TWO ]
      then appropriate-structure = soil-nailed-system .

rule-61: if possible-wall = soil-nailed-system  and
         user-informed = yes and
         informnail = yes and
         listof(possible-wall) = [ ONE,TWO,THREE ]
      then appropriate-structure = soil-nailed-system .

rule-62: if appropriate-structure is sought and
         appropriate-structure = X and
         possible-wall is sought and
         possible-wall = X and
         listof(appropriate-structure) = [ONE] and
         display([nl,'The appropriate wall for the given conditions
         continue and
         display('\f')
      then tuser-informed.

rule-63: if appropriate-structure is sought and
         appropriate-structure = X and
         tuser-informed = yes and
         possible-wall is sought and
         possible-wall = X and
         laborb = yes and
         aesthetics = yes
      then selected-structure = X.

rule-64: if selected-structure is sought and
         selected-structure = X and
         listof(selected-structure) = [ONE] and
         display([nl,'The  selected  structure for  the  given condit
         continue and
         user-happy and
         display([nl,'I''m glad you are happy with my selection.',nl]
         list=A              and
         display(A)          and
         height=H            and
         choice1=Y           and
         design(H,Y)=DESIGN
         then end the consultation.
list=[
nl,
nl,'     1. Gravity;',
nl,'     2. Cantilever;',
nl,'     3. Counterfort;',
nl,'     4. Gabions;',
```

```
nl,'          5. Slurry;',
nl,'          6. Tied-back;',
nl,'          7. Reinforced-earth;',
nl,'          8. Sheet-piles;',
nl,'          9. Crib-wall;',
nl,'          10.Soil-nailing.',
nl,
nl].
nocache(list).
question(choice1)=[nl,' Please confirm the number of the wall which you

/*
multivalued(fault).
automaticmenu(fault).
enumeratedanswers(fault).
*/
question(fault) =
   ['Which wall do you think is best for these conditions?'].
/*
 legalvals(fault) =
[gravity,cantilever,counterfort,gabions,slurry,tied-
back,reinforced-earth,sheet-piles,crib-wall,soil-nailing].
*/
legalvals(fault)=integer.
rule-65: if user-informed and
            possible-wall is sought and
            user-informed and
            appropriate-structure is sought and
            tuser-informed and
            selected-structure is sought and
            not(user-happy) and
            list=F          and
            display(F)      and
            fault = Y               and
            height=H                and
            design(H,Y)=DESIGN
            then end the consultation.

/* ------------------------------------------------------- */

rule-66: if external(design,[H,Y])=[DESIGN] and
            display(['The consultation is over. Press Alt + Q. Thank you
         then design(H,Y)=DESIGN.
/**********************************************************/
legalvals(choice1)=integer(1,10).
```

# APPENDIX B

# LISTING OF CONTROL PROGRAM CALL

```
/*------------------------------------------------------------------------------------------*/
/*                    CONTROL PROGRAM                                                        */
/*------------------------------------------------------------------------------------------*/
#include <stdio.h>
#include <stdlib.h>
#include <dos.h>
#include <ctype.h>
#include <string.h>
#include <process.h>
#include <signal.h>
#include <graph.h>
#include <math.h>
#define HOMEKEY  0x47
#define ENDKEY   0x4f
#define UPARROW  0x48
#define DOWNARROW 0x50
#define PGUP     0x49
#define PGDN     0x51
#define ESCKEY   0x1b
#define F1KEY    0x3b
#define F2KEY    0x3c
#define ALTP     0x19
#define ALTE     0x12
#define ALTS     0x1f
#define MAX_LINE 600
#define MAX_CHAR 80
#define F3KEY    0x3d
#define F4KEY    0x3e
#define F5KEY    0x3f
#define F6KEY    0x40
/***********************************************************/

void setscreen(int),set_color(int,int),locate(int,int);
void set_page(int),graph_print(int,int),printstrg(int,int,char *);
void clear_screen(),scroll_window(int),get_lines(char *),call(char *);
void print_first_page(),scroll_page_up(),scroll_page_down(),title();
void input_time();
void waitt();
int  show(char *);
FILE *fp,*fp1,*fopen();
int keykb(),getkey();
int page,color,c,last_line,first_line,cur_first_line,cur_last_line;
char line[MAX_LINE][MAX_CHAR];
int screen,xloc,yloc;
long int couleur=7;
float hh;
int ty;

char *name[10]={"Gravity","Cantilever","Counterfort","Gabions","Slurry",
            "Tied-back","Reinforced-earth","Sheetpile","Crib wall",
            "Soil-nailing"};

/**************************************************/
main()
```

```
{
int i,j,k;
int c,cc,ccc;
_setvideomode(_MRES4COLOR);
title();
_setbkcolor(_LIGHTBLUE);
set_page(1);
locate(0,0);
set_color(couleur,0);
locate(23,1);
graph_print(205,77);
locate(2,1);
graph_print(205,77);
set_color(0,11);
printstrg(10,24,"       Selection          ");
printstrg(12,24,"       Design             ");
printstrg(14,24,"       Construction        ");
printstrg(16,24,"       F1:Explanation      ");
printstrg(18,24,"       F2:Quit            ");
set_color(4,11);
printstrg(10,33,"S");
printstrg(12,33,"D");
printstrg(14,33,"C");
printstrg(16,33,"F1");
printstrg(18,33,"F2");
for(i=5;i<=10;i++)
{
j=2*i-1;
locate(j,23);
graph_print(196,31);
}
set_color(7,0);
printstrg(24,1," Note: Press the light letter for selection, design or");
printstrg(25,1,"       construction.      ");
locate(0,0);
while(kbhit()==0) {;}
c=getkey();
k=0;
while(c!=F2KEY)
{
switch(c)
  {
  case ESCKEY:
  _setvideomode(_MRES4COLOR);
  setscreen(3);
  _setbkcolor(_LIGHTBLUE);
  set_page(1);
  set_color(couleur,0);
  locate(23,1);
  graph_print(205,77);
  locate(2,1);
  graph_print(205,77);
  set_color(0,11);
  printstrg(10,24,"       Selection          ");
```

```
printstrg(12,24,"        Design          ");
printstrg(14,24,"        Construction    ");
printstrg(16,24,"        F1:Explanation  ");
printstrg(18,24,"        F2:Quit         ");
set_color(4,11);
printstrg(10,33,"S");
printstrg(12,33,"D");
printstrg(14,33,"C");
printstrg(16,33,"F1");
printstrg(18,33,"F2");
for(i=5;i<=10;i++)
{
j=2*i-1;
locate(j,23);
graph_print(196,31);
}
set_color(7,0);
printstrg(24,1," Note: Press the lighted letter for selection, design or");
printstrg(25,1,"       construction. ");
locate(0,0);
while(kbhit()==0) {;}
c=getkey();
break;
case 's':
_setvideomode(_DEFAULTMODE);
system("m1");
printf("\n\n Please press Esc key to continue our consultation !\n");
while(kbhit()==0) {;}
c=getkey();
k=1;
break;
case 'd':
clear_screen();


if(k!=0)
{
fp=fopen("heitype","r");
fscanf(fp,"%f %d",&hh,&ty);
fclose(fp);
j=ty;
if(fp!=NULL)
printf("\n\n HEIGHT=%.2f TYPE=%d  [ %s ]\n",hh,ty,name[j-1]);
}
else goto s1;
/*
printf("\n\n Please press any key to continue !\n");
while(kbhit()==0) {;}
if(fp==NULL)
{
printf("\n Choose one type of wall you want to design\n");
printf(" 1.  Gravity wall;\n");
printf(" 2.  Cantilever wall;\n");
printf(" 3.  Countrefort wall;\n");
```

```
printf(" 4.  Gabions;\n");
printf(" 5.  Slurry;\n");
printf(" 6.  Tied-back earth;\n");
printf(" 7.  Reinforced-earth;\n");
printf(" 8.  Sheetpile;\n");
printf(" 9.  Crib wall;\n");
printf(" 10. Soil nailing;\n>>");
scanf("%d",&ty);
}
*/
switch(ty)
  {
case 8:
printf(" \n DESIGN OF SHEETPILE\n");
system("sheet");
printf(" Press Esc key to return to main menu\n");
while(kbhit()==0) {;}
c=getkey();
break;
case 1:
printf(" \n DESIGN OF GRAVITY WALL\n");
system("gravity");
printf(" Press Esc key to return to main menu\n");
while(kbhit()==0) {;}
c=getkey();
break;
case 2:
printf(" DESIGN OF CANTILEVER WALL\n");
system("cant");
printf(" Press Esc key to return to main menu\n");
while(kbhit()==0) {;}
c=getkey();
break;
case 4:
printf(" DESIGN OF GABION\n");
system("gabion");
printf(" Press Esc key to return to main menu\n");
while(kbhit()==0) {;}
c=getkey();
break;
case 7:
printf(" DESIGN OF REINFORCED EARTH\n");
system("reinf");
printf(" Press Esc key to return to main menu\n");
while(kbhit()==0) {;}
c=getkey();
break;
case3:
printf(" The program for this wall has not been installed.\n");
printf(" Press Esc key to return to main menu\n");
while(kbhit()==0) {;}
c=getkey();
break;
case 5:
```

```
printf(" The program for this wall has not been installed.\n");
printf(" Press Esc key to return to main menu\n");
while(kbhit()==0) {;}
c=getkey();
break;
case 6:
printf(" The program for this wall has not been installed.\n");
printf(" Press Esc key to return to main menu.\n");
while(kbhit()==0) {;}
c=getkey();
break;
case 9:
printf(" The program for this wall has not been installed.\n");
printf(" Press Esc key to return to main menu\n");
while(kbhit()==0) {;}
c=getkey();
break;
case 10:
printf(" The program for this wall has not been installed.\n");
printf(" Press Esc key to return to main menu\n");
while(kbhit()==0) {;}
c=getkey();
break;
default:
while(kbhit()==0) {;}
c=getkey();
break;
   }
s1:if(k==0)
  {
  printf("\7");
  printf("\n\n You are advised to go through the selection process!\n");
  printf(" Press Esc key.\n");
  while(kbhit()==0) {;}
  c=getkey();
  }
  break;
  case 'c':
  clear_screen();
  printf("\n\n The part is under development !\n");
  printf(" Press Esc key to return to main menu\n");
  while(kbhit()==0) {;}
  c=getkey();
  break;
  case F1KEY:
  c=show("exp.cal");
  set_color(0,11);
  break;
  default:
  while(kbhit()==0) {;}
  c=getkey();
  break;
  }
}
```

```
if(fp!=NULL);
system("del heitype");

_setvideomode(_DEFAULTMODE);
}
/***********************************************************/

void printstrg (x,y,s)
int x,y;
char *s;
{
union REGS intregs,outregs;
extern int page,color;
while(*s!='\0')
 {
 intregs.h.ah=2;
 intregs.h.bh=page;
 intregs.h.dh=x-1;
 intregs.h.dl=y-1;
 int86(0x10,&intregs,&outregs);
 intregs.h.ah=9;
 intregs.h.al=*s&55;
 intregs.h.bh=page;
 intregs.h.bl=color;
 intregs.x.cx=0;
 int86(0x10,&intregs,&outregs);
 y++;
 s++;
 }
intregs.h.ah=1;
intregs.h.bh=page;
intregs.h.dh=x-1;
intregs.h.dl=y-1;
int86(0x10,&intregs,&outregs);
}
/***********************************************************/

void setscreen(n)
int n;
{
union REGS intregs,outregs;
screen=n;
intregs.h.ah=1;
intregs.h.al=n;
int86(0x10,&intregs,&outregs);
}
/****************************************/

void locate(x,y)
int x,y;
{
union REGS intregs,outregs;
xloc=x;
```

```
yloc=y;
intregs.h.ah=1;
intregs.h.bh=page;
intregs.h.dh=xloc-1;
intregs.h.dl=yloc;
int86(0x10,&intregs,&outregs);
}
/*****************************************/

void graph_print(s,n_times)
int s,n_times;
{
union REGS intregs,outregs;
intregs.h.ah=9;
intregs.h.al=s;
intregs.h.bh=0;
intregs.h.bl=color;
intregs.x.cx=n_times;
int86(0x10,&intregs,&outregs);
}
/****************************************/

void set_page(p)
int p;
{
union REGS intregs,outregs;
extern int page;
page=p;
intregs.h.ah=5;
intregs.h.al=p;
int86(0x10,&intregs,&outregs);
}
/****************************************/

void set_color(f,b)
int f,b;
{
color=(f & 143)+((b<<4) & 112);
}
/****************************************/

void clear_screen()
{
union REGS inregs,outregs;
inregs.h.ah=5;
inregs.h.al=0;
inregs.h.bh=6;
inregs.h.ch=1;
inregs.h.cl=0;
inregs.h.dh=24;
inregs.h.dl=79;
int86(0x10,&inregs,&outregs);
}
/**********************************************/
```

```
void scroll_window(n)
int n;
{
union REGS inregs,outregs;
if (n>=20) n=0;
inregs.h.ah=((n>=0)? 6:7);
inregs.h.al=((n>=0)? n:-n);
inregs.h.bh=6;
inregs.h.ch=0;
inregs.h.cl=1;
inregs.h.dh=20;
inregs.h.dl=79;
int86(0x10,&inregs,&outregs);
}
/*****************************************************************/
int keykb()
{
union REGS inregs,outregs;
inregs.h.ah=0;
return (int86(0x16,&inregs,&outregs));
}
/**********************************************/
int getkey()
{
char al,ah,c;
int key;
key=keykb();
ah=(key >> 8) & 0x00ff;
al=key & 0x00ff;
if(al==0)
c=ah;
else
c=al;
return(c);
}
/************************************/
void get_lines(s)
char *s;
{
int i,c;
fp=fopen(s,"r");
first_line=0;
last_line=0;
while((c=getc(fp))!='0')
 {
 ++last_line;
 i=0;
  while(c!='\n')
 {
 line[last_line-1][i]=s;
 i++;
 c=getc(fp);
 }
 line[last_line-1][i]='\n';
```

```
}
line[last_line][0]='0';
fclose(fp);
}
/*********************************************************/
void print_first_page()
{
int i,c;
cur_first_line=first_line;
cur_last_line=-1;
set_color(couleur,0);
locate(1,1);
graph_print(196,72);
while(
(cur_last_line<last_line) && (cur_last_line<20))
{
++cur_last_line;
locate((cur_last_line+2),1);
i=0;
  while((c=(line[cur_last_line][i] & 255))!='0' && c!='\n')
  {
   graph_print(c,0);
   ++yloc;
   locate(xloc,yloc);
   ++i;
  }
  if(c=='\n')
  graph_print(32,72-i);
  if (c=='0')
  break;
}
locate(cur_last_line+3,1);
graph_print(196,72);
}
/*********************************************************/
int show(sss)
char *sss;
{
int c;
clear_screen();
get_lines(sss);
print_first_page();
locate(0,0);
while(kbhit()==0) {;}
c=getkey();
while(c!=ESCKEY && c!=F2KEY)
 {
 switch(c)
  {
  case UPARROW:
  scroll_page_down(1);
  locate(0,0);
  while(kbhit()==0) {;}
  c=getkey();
```

```
break;
case DOWNARROW:
scroll_page_up(1);
locate(0,0);
while(kbhit()==0) {;}
c=getkey();
break;
case PGUP:
scroll_page_down(19);
locate(0,0);
while(kbhit()==0) {;}
c=getkey();
break;
case PGDN:
scroll_page_up(19);
locate(0,0);
while(kbhit()==0) {;}
c=getkey();
break;
default:
locate(0,0);
while(kbhit()==0) {;}
c=getkey();
break;
}
}
return(c);
}
/****************************************************************/
void scroll_page_down(n)
int n;
{
int c,i;
while( (n>0) && (cur_first_line > first_line ))
{
--cur_first_line;
if((cur_last_line - cur_first_line) > 20)
--cur_last_line;
--n;
scroll_window(-1);
locate(2,1);
i=0;
while(
( c= (line[cur_first_line][i] & 255 )) !='\n' && c!='0')
{
graph_print(c,0);
++yloc;
locate(xloc,yloc);
++i;
}
if(c=='\n')
graph_print(32,72-i);
if(c=='0')
graph_print(32,72);
```

```
 }
}
/***********************************************************/
void scroll_page_up(n)
int n;
{
int c,i;
while( (n>0) && (cur_last_line < last_line))
 {
 ++cur_last_line;
 if(cur_last_line-cur_first_line>180)
 ++cur_first_line;
 --n;
 scroll_window(1);
 locate(21,1);
 i=0;
 while((c=(line[cur_last_line][i] & 255)) !='\n' && c!='0')
  {
  graph_print(c,1);
  ++yloc;
  locate(xloc,yloc);
  ++i;
  }
 if( c=='\n')
 graph_print(32,72-i);
 if(c=='@')
 graph_print(32,72);
 }
}
/*****************************************************/
void title()
{
int i,delay1,delay2,delay3,delay4,delay5,delay6,delay7,delay8,delay9;
int delay10,delay11,delay12,delay13,delat14;
setscreen(3);
_setbkcolor(_LIGHTGREEN);
set_color(4,11);
for(i=0;i<20;i++)
printstrg(1+i,1,"                                        ");
set_color(4,7);
printstrg(1,1,"                                   ");
printstrg(2,1,"                                   ");
printstrg(3,1,"                                   ");
printstrg(4,1,"            RETAININGEARTH                     ");
printstrg(5,1,"                                   ");
printstrg(6,1,"     Expert System for Selection, Design and       ");
printstrg(7,1,"      Construction of Retaining Structures       ");
printstrg(8,1,"                                   ");
printstrg(9,1,"      DEPARTMENT OF OCEAN ENGINEERING              ");
printstrg(10,1,"        FLORIDA ATLANTIC UNIVERSITY            ");
printstrg(11,1,"             JULY  1990             ");
printstrg(12,1,"                                   ");
printstrg(13,1,"                                   ");
printstrg(14,1,"                                   ");
```

```
printstrg(15,1,"                                    ");
locate(0,0);
waitt();
waitt();
waitt();
waitt();
waitt();
waitt();
waitt();
waitt();
waitt();
}
/*********************************************/
/*******************************************************/
void waitt()
{
int delay1,delay2,delay3;
for(delay1=0;delay1<30000;delay1++) {;}
for(delay2=0;delay2<30000;delay2++) {;}
for(delay3=0;delay3<30000;delay3++) {;}
}
```

# APPENDIX C

# LISTING OF PROGRAM GRAVITY

```
/*   -------------------------------------------------------------------------   */
/*   |  DESIGN OF THE GRAVITY RETAINING STRUCTURE  |   */
/*   -------------------------------------------------------------------------   */

#include <stdio.h>
#include <math.h>
#include <ctype.h>

main()
{
FILE *fopen(),*fp_out;
char fn_out[12];

float
h,surchp,gamat,angphi,angphi1,mu,angdel,angdel1,bearingp,ca1,ca2,ca3,ca4,ca,cp;
float surcht,p,y,overnom,lbase,h1,w1,l2,l3,h2,h3,w2,w3,w4,w5,w6,wtot,x1,x2,x3,
   x4,x5,x6;
float resmom,a,a11,a22,fsover,maxbearp,fricf,passpr,slidefs,ss;
int ht;

extern double sin();
extern double cos();

printf("Enter name of output file:");
scanf("%s",fn_out,"w");
fp_out = fopen(fn_out,"w");
printf("Output in file:%s\n\n\n",fn_out);

fprintf(fp_out,"   -------------------------------------------------------------------------  \n");
fprintf(fp_out,"   |  DESIGN OF THE GRAVITY RETAINING STRUCTURE  |
\n");
fprintf(fp_out,"   -------------------------------------------------------------------------  \n");

printf("   -------------------------------------------------------------------------  \n");
printf("   |  DESIGN OF THE GRAVITY RETAINING STRUCTURE  |  \n");
printf("   -------------------------------------------------------------------------  \n\n\n");
printf("What is the height (in ft.) of the earth to be retained?\n>>");
scanf("%f",&h);
printf("What is the frost depth (in ft.) in this region?\n>>");
scanf("%f",&ss);
printf("What is the live load surcharge pressure(in psf.)?\n>>");
scanf("%f",&surchp);
printf("What is the unit weight (in pcf.) of the soil in this region?\n>>");
scanf("%f",&gamat);
printf("What is the angle (in deg.) of internal friction of the soil?\n>>");
scanf("%f",&angphi1);
printf("What is the base friction coefficient(if not known,you may give the value\n as
0.5)?\n>>");
scanf("%f",&mu);
printf("What is the slope of the earth surface(in deg.)?\n>>");
scanf("%f",&angdel1);
printf("What is the allowable bearing pressure (in psf.)?\n>>");
scanf("%f",&bearingp);
printf("\n\n C A L C U L A T I N G . . . . . . . ........\n\n\n");
```

```
angphi = (3.1415926 / 180.0) * angphi1;
angdel = (3.1415926 / 180.0) * angdel1;

if (angdel !=0)
        goto ang1;
else ca = (1 - sin(angphi)) / (1 + sin(angphi));
cp = 1/ca;
goto ang2;
ang1:ca1 = pow((cos(angdel)),2);
ca2 = pow((cos(angphi)),2);
ca3 = pow((ca1 - ca2),0.5);
ca4 = cos(angdel);
ca = ca4 * (ca4 - ca3) / (ca4 + ca3);
cp = ca4 * (ca4 + ca3) / (ca4 - ca3);
ang2:surcht = surchp / gamat;
ht = h + ss;
p = 0.5 * ca * gamat * ht * (ht + 2*surcht);
y = ((ht*ht) + (3*ht*surcht)) / (3*(ht + 2*surcht));
ovemom = p * y;
/*---------------------------------------------------*/
/*  calculation of the weight and restoring moment   */
/*---------------------------------------------------*/
lbase = 0.65 * ht;
h1 = 2.00;
h11:w1 = lbase * h1 * 150;
l2 = 1.5;
l3 = lbase - l2 - 0.75;
h2 = ht - h1;
h3 = h2;
w2 = l2 * h2 * 150;
w3 = l3 * h3 * 150 * 0.5;
w4 = l3 * h3 * gamat * 0.5;
w5 = 0.75 * h3 * gamat;
w6 = (lbase - 3) * surchp;
wtot = w1 + w2 + w3 + w4 + w5 + w6;
x1 = lbase * 0.5;
x2 = 1.5;
x3 = (0.333 * l3) + l2 + 0.75;
x4 = (0.667 * l3) + l2 + 0.75;
x5 = lbase - 0.375;
x6 = (lbase - 2.25) * 0.5;
resmom = (w1 * x1) + (w2 * x2) + (w3 * x3) + (w4 * x4) + (w5 * x5) + (w6 * x6);
a = (resmom - ovemom)/wtot;
a11 = lbase / 3.0;
a22 = 2 * a11;
if (a11 <= a  &&  a <= a22)
        goto mp;
lbase = lbase + 0.2;
goto h11;
mp:fsover = resmom / ovemom;
if (fsover > 2.00)
        goto mp1;
lbase = lbase + 0.2;
goto h11;
```

```
mp1:maxbearp = ((4*lbase)-(6*a))*wtot/(lbase*lbase);
fricf = 0.5*wtot;
passpr = 6 * gamat;
slidefs = (fricf + passpr)/p;
if (slidefs > 1.5)
        goto fin;
lbase = lbase + 0.2;
goto h11;
fin:fprintf(fp_out,"Coefficient of active pressure = %.4f\n",ca);
fprintf(fp_out,"Coefficient of passive pressure = %.4f\n",cp);
fprintf(fp_out,"Total earth thrust = %.2f lb.\n",p);
fprintf(fp_out,"Distance of earth's thrust from the base = %.2f ft.\n",y);
fprintf(fp_out,"Length of the base  = %.2f ft.\n",lbase);
fprintf(fp_out,"Height of the structure = %d ft.\n",ht);
fprintf(fp_out,"Distance of the resultant from the front edge =  %.2f ft. --- OK (inside
middle-third).\n",a);
fprintf(fp_out,"Overturning moment = %.2f ft.-lb.\n",ovemom);
fprintf(fp_out,"Restoring moment = %.2f ft.-lb.\n",resmom);
fprintf(fp_out,"Maximum bearing pressure = %.2f psf.\n",maxbearp);
fprintf(fp_out,"Friction force = %.2f lb.\n",fricf);
fprintf(fp_out,"Passive pressure = %.2f lb.\n",passpr);  ·
fprintf(fp_out,"Factor of safety against overturning = %.2f  ---- > 2.0  .....
OK.\n",fsover);
fprintf(fp_out,"Factor of safety against sliding = %.2f ----  > 1.5  ..... OK.\n",slidefs);
printf("     --------------------------------------------------  \n");
printf("     |  DESIGN OF THE GRAVITY RETAINING STRUCTURE   |  \n");
printf("     --------------------------------------------------  \n\n\n");
printf("                *****************\n");
printf("                *  I N P U T  *\n");
printf("                *****************\n\n");

printf("Height of the earth to be retained = %.2f ft.\n",h);
printf("Frost depth in this region  = %.2f ft.\n",ss);
printf("Live load surcharge pressure = %.2f psf.\n",surchp);
printf("Unit weight of the soil in this region = %.2f pcf.\n",gamat);
printf("Angle of internal friction of the soil = %.2f deg.\n",angphi1);
printf("Base friction coefficient = %.2f\n",mu);
printf("Slope of the earth surface = %.2f deg.\n",angdel1);
printf("Allowable bearing pressure = %.2f psf.\n\n\n",bearingp);
printf("*****************************************************************
**************\n");
printf("THE DETAILED DESIGN OUTPUT OF THE GRAVITY STRUCTURE IS IN
FILE >>  %s\n",fn_out);
printf("*****************************************************************
**************\n");

fclose(fp_out);
return(0);
}
```

# APPENDIX D

# LISTING OF PROGRAM CANWALL

```
/*  ---------------------------------------------------------------------------  */
/*   | DESIGN OF CANTILEVER RETAINING STRUCTURE |                                */
/*  ---------------------------------------------------------------------------  */
float table[5][3] = {
                     { 0.0278, 0.0206, 0.0160 },
                     { 0.0325, 0.0241, 0.0187 },
                     { 0.0371, 0.0275, 0.0214 },
                     { 0.0437, 0.0324, 0.0252 },
                     { 0.0491, 0.0364, 0.0283 }
                   };
#include <stdio.h>
#include <ctype.h>
#include <math.h>
main()
{
FILE *fopen(),*fp_out;
char fn_out[15];
float h,surcht,gsur,g1,alphas,alpha1,miu,mu,maxspr,fc,fy,hw;
float ca,alpha,deltab,u,m,layer,wsub1,ru,ro,romax;
float h1,t,x1,x2,p1,p2,p,x,lbase,y,m1,m2,t1,w1,w2,w3,w4,w5,wtot;
float lev1,lev2,lev3,lev4,lev5,lev6,lev7,lev8,mw1,mw2,mw3,mw4,mw5,mwt,ywm,
resmom,ovemom,fsover;
float xbar,xbar1,xbar2,xbar3,xbar4,fsldg,ffric,fssldg,ke,yy1,yy2,yy3,yy4;
float stemvu,sphivc,wu,wearth,wsurch,wfooting,muw,vuw,dreqd,minro,reqdas,phivcw;
float run,ron,a11,a12,a13,a14,a15,a16,a17,phivc;
float df,ld1,ld2,ldf,ld,ldl,p11,p12,p13,x4,vutoe,mu11,mu12,mu13,d22;
float ruwall,ronwall,wreqdas,reqdas3,aalpha,ddeltab,csd,csa,ca1,ca2;
float gw,gb1,h11,p3,p4,m3,m4,tbt,w6,w7,mw6,mw7,wearth1,wearth2,x3;
float g2,alpha2,h12,x21,x22,x23,x8,ast,asfront,asback,a01,aa1,ldl1;
float gb2,h11w,hw12,ht12,x5,x56,x6,x7,p34,p5,p6,p7,m5,m6,m7;
float l5x,l5y,l5a,l5c,l5d,l56a,l56aa,l56b,l56bb;
float l56x,lev56,mw34,mw56,wearth3,w34,w56;
float h121,xh4,xh5,xh6,w8,w9,w10,lev9,lev10,mw8,mw9,mw10;
int tb,tt,i,j,t1n,no1,no2,no3;

extern double cos();
extern double sin();
extern double sqrt();
extern float fromax();

printf("Enter name of output file:");
scanf("%s",fn_out,"w");
fp_out = fopen(fn_out,"w");
printf("Output in file:%s\n\n\n",fn_out);
printf("        ------------------------------------------------------------------       \n");
printf("         | DESIGN OF CANTILEVER RETAINING STRUCTURE |        \n");
printf("        ------------------------------------------------------------------       \n\n\n");
printf("What is the height (in ft.) of the earth to be supported?\n");
scanf("%f", &h);
printf("What is the angle (in deg.) which the backfill makes with the horizontal?\n");
scanf("%f",&deltab);
printf("What is the height (in ft.) of the surcharge?\n");
scanf("%f", &surcht);
/* unit weight of water = 62.5 pcf  */
```

```
if (surcht == 0)
        goto p1;
printf("What is the density (pcf) of the surcharge material?\n");
scanf("%f", &gsur);
printf("What is the angle (in deg.) of internal friction of the surcharge material?\n");
scanf("%f", &alphas);
p1:printf("What is the coefficient of friction between masonry and soil? (if unknown, you
may answer it as 0.40)\n");
scanf("%f", &miu);
printf("What is the compressive strength, fc (in ksi) of concrete?\n");
scanf("%f", &fc);
printf("What is the yield strength, fy (in ksi) of concrete?\n");
scanf("%f", &fy);
printf("What is the maximum soil pressure (in ksf)?\n");
scanf("%f", &maxspr);
printf("What is the height (in ft.) of the water table in this region?\n");
scanf("%f", &hw);
printf("How many layers of soil are there below the earth's surface(you are advised to
give the value of 1 or 2)?\n");
scanf("%f",&layer);
fprintf(fp_out,"  ----------------------\n");
fprintf(fp_out,"  | INPUT DATA |\n");
fprintf(fp_out,"  ----------------------\n");
fprintf(fp_out,"Height of earth to be supported = %.2f ft.\n",h);
fprintf(fp_out,"Angle which the backfill makes with the horizontal = %.2f
deg.\n",deltab);
fprintf(fp_out,"Height of the surcharge = %.2f ft.\n",surcht);
fprintf(fp_out,"Density of the surcharge material = %.2f pcf.\n",gsur);
fprintf(fp_out,"Angle of internal friction of the surcharge material = %.2f deg.\n",alphas);
fprintf(fp_out,"Coefficient of friction between masonry and soil = %.2f\n",miu);
fprintf(fp_out,"Compressive strength of concrete,fc = %.2f ksi.\n",fc);
fprintf(fp_out,"Yield strength of steel,fy = %.2f ksi.\n",fy);
fprintf(fp_out,"Maximum soil pressure = %.2f ksf.\n",maxspr);
fprintf(fp_out,"Height of water table = %.2f ft.\n",hw);
fprintf(fp_out,"Layers of soil below the earth's surface = %.1f\n",layer);
aalpha = alphas * 3.1415926 / 180.0;
if (deltab == 0)
        goto cab;
ddeltab = deltab * 3.1415926 / 180.0;
csd = cos(ddeltab);
csa = cos(aalpha);
ca1 = pow(((csd * csd) - (csa * csa)),0.5);
ca2 = (csd - ca1)/(csd + ca1);
ca = csd * ca2;
goto cad;
cab:ca = (1 - (sin(aalpha)))/(1 + (sin(aalpha)));
cad:fprintf(fp_out,"  -----------------------------------------------------------------\n");
fprintf(fp_out,"      | DESIGN OF CANTILEVER RETAINING STRUCTURE |\n");
fprintf(fp_out,"      -----------------------------------------------------------------\n\n");
fprintf(fp_out,"Coefficient of active pressure = %.4f\n",ca);
romax = fromax(fc,fy);
ro = 0.5 * romax;
m = fy/((0.85)*(fc));
ru = ro * fy * 1000 * (1 - (0.5 * ro * m));
```

```
fprintf(fp_out,"Reinforcement ratio,Ro = %.4f\n",ro);
fprintf(fp_out,"Strength coefficient of resistance,Ru = %.2f psi.\n",ru);
h1 = (h + 4);
/* h1 = total height of wall taking frost penetration into account */
t = 0.1 * h1;
/* t = thickness of footing */
fprintf(fp_out,"Total height of wall(including frost penetration) = %.2f ft.\n",h1);
fprintf(fp_out,"Thickness of footing = %.2f ft.\n",t);
x1 = ca * gsur * surcht;
if (layer == 1)
        goto layer1;
if (layer == 2)
        goto layer2;
if (layer > 2)
printf("This program is not capable of handling more than 2 soil layers.SORRY!\n");
goto endd;
layer1:printf("What is the density of the soil(in pcf) below the earth's surface?\n");
scanf("%f",&g1);
printf("What is the angle(in deg.) of internal friction of the soil below the earth's
surface?\n");
scanf("%f",&alpha1);
aalpha = alpha1 * 3.1415926 / 180.0;
ca = (1 - (sin(aalpha)))/(1 + (sin(aalpha)));
fprintf(fp_out,"Coefficient of active pressure = %.4f\n",ca);
if (hw != 0)
        goto hw11;
x2 = ca * g1 * h1;
p1 = x1 * h1;
p2 = 0.5 * x2 * h1;
p = (p1 * h1 * 0.5) + (p2 * (h1/3));
wsub1 = (surcht * gsur) + (g1 * h1);
x = sqrt((2 * p) / wsub1);
goto lb;
lbb:x = x + 0.2;
lb:lbase = 1.5 * x;
fprintf(fp_out,"Base length of the footing = %.2f ft.\n\n",lbase);
y = h1 - t;
m1 = x1 * 0.5 * y * y;
m2 = (x2 * (y * y * y))/(6 * h1);
mu = (1.7*(m1 + m2)) / 1000 ;
t1 = sqrt((mu * 12000)/(0.9 * ru * 12.0));
tb = t1 + 5;
if (tb < 12)
        tb = 12.0;
tt = 12.0;
fprintf(fp_out,"STEM DIMENSIONS:\n");
fprintf(fp_out,"----------------------------\n");
fprintf(fp_out,"  a) stem thickness at the bottom = %d in.\n",tb);
fprintf(fp_out,"  b) stem thickness at the top    = %d in.\n\n",tt);
yy1 = .825 * h1;
yy2 = yy1 * yy1;
yy3 = x1 * yy1;
yy4 = x2 / h1;
w1 = (gsur/1000) * surcht * (x - 1);
```

```
w2 = (g1/1000) * y * (x-1);
w3 = (0.15 - (g1/1000)) * y * ((tb - tt)/24);
w4 = 0.15 * lbase * t;
w5 = 0.15 * y;
wtot = w1 + w2 + w3 + w4 + w5;
lev1 = (x - 1) * 0.5;
lev2 = lev1;
lev3 = (2 * lev1) - ((tb - tt) / 36.0);
lev4 = (lbase * 0.5);
lev5 = (lev1 * 2.0) + 0.5;
mw1 = w1 * lev1;
mw2 = w2 * lev2;
mw3 = w3 * lev3;
mw4 = w4 * lev4;
mw5 = w5 * lev5;
mwt = (mw1 + mw2 + mw3 + mw4 + mw5);
ywm = mwt/wtot;
resmom = wtot * (lbase - ywm);
ovemom = ((p1 * h1 * 0.5) + (p2 * (h1/3)))/1000.0;
fsover = resmom/ovemom;
fprintf(fp_out,"Overturning stability:\n");
fprintf(fp_out,"-------------------------\n");
fprintf(fp_out,"Resisting moment = %.2f ft-kips\n",resmom);
fprintf(fp_out,"Overturning moment = %.2f ft-kips\n",ovemom);
if (fsover < 2.0)
        goto lbb;
else
xbar = (mwt + ovemom) / wtot;
xbar1 = lbase/3.0;
xbar2 = (2.0 * lbase) / 3.0;
xbar3 = xbar + .08;
xbar4 = (xbar - xbar2)*12.0;
fprintf(fp_out,"Factor of safety against overturning = %.2f / %.2f = %.2f --OK-- (F.S. >
2.0)\n\n\n",resmom,ovemom,fsover);
if (xbar > xbar1 && xbar < xbar2)
fprintf(fp_out,"The position of the resultant soil pressure due to the service load was
checked and found to be within the middle-third of the base --- OK\n\n");
 else if (xbar > xbar1 && xbar < xbar3)
        fprintf(fp_out,"The resultant soil pressure lies %.1f in. outside the middle-third,
however,it is very close, and the limiting condition of zero stress at the heel is considered
adequate.\n\n\n",xbar4);
 if (xbar > xbar1 && xbar > xbar3)
   goto lbb;
fprintf(fp_out,"Sliding stability:\n");
fprintf(fp_out,"-------------------\n");
fsldg = (p1 + p2)/1000.0;
ffric = miu * wtot;
fssldg = ffric / fsldg;
fprintf(fp_out,"Sliding force = %.2f kips\n",fsldg);
fprintf(fp_out,"Resisting force = %.2f kips\n",ffric);
fprintf(fp_out,"Factor of safety against sliding = %.2f / %.2f =
%.2f\n",ffric,fsldg,fssldg);
if (fssldg > 1.5)
        goto stp;
```

```
else fprintf(fp_out,"Key to be provided as factor of safety against sliding (%.2f) is less
than 1.5\n",fssldg);
ke = 8 * t;
fprintf(fp_out,"Key dimensions(CRSI Handbook):The key has a square section (%.1fin.
x %.1fin.)\n",ke,ke);
fprintf(fp_out,"        Generally it is desirable to place the front face of the key about 5
in. in front of the back face of the stem.This will permit anchoring the stem reinforcement,
if present, in the key.\n\n\n");
goto heel;
stp:fprintf(fp_out,"---- OK (F.S. > 1.5).\n\n\n");
heel:fprintf(fp_out,"-------------------------------------------------------------------\n");
fprintf(fp_out,"                design of heel cantilever \n");
fprintf(fp_out, "  -----------------------------------------------------------------\n\n");
wearth = g1 * y / 1000.0;
wsurch = gsur * surcht / 1000.0;
wfooting = 0.15 * t;
wu = (1.4 * (wearth + wfooting)) + (1.7 * wsurch);
lev6 = x - (tb / 12.0);
muw = 0.5 * wu * lev6 * lev6;
vuw = wu * lev6;
phivcw = 0.0204 * sqrt(fc * 1000.0) * tb;
if (phivcw < vuw)
        dreqd = (tb * vuw / phivcw) + 1.0;
else goto com;
t1n = dreqd + 2.5;
fprintf(fp_out,"Heel thickness = %d\n",t1n);
run = (muw * 12000) / (0.9 * 12.0 * dreqd * dreqd);
ron = (1.0 - sqrt(1.0 - (0.002 * m * run / fy)))/m;
minro = 0.2 / fy;
  if (minro > ron)
        ron = 1.333 * ron;
else ron = ron;
goto comm;
com: ron = ro;
t1n = t * 12.0;
fprintf(fp_out,"Heel thickness = %d\n",t1n);
dreqd = (t1n - 2.5);
comm:reqdas = ron * 12.0 * dreqd;
a11 = 0.62 - reqdas;
a12 = 0.88 - reqdas;
a13 = 1.20 - reqdas;
a14 = 1.58 - reqdas;
a15 = 2.00 - reqdas;
a16 = 2.54 - reqdas;
a17 = 3.12 - reqdas;
if (a11 > 0 && a11 < (a12,a13,a14,a15,a16,a17))
        goto a11;
if (a12 > 0 && a12 < (a11,a13,a14,a15,a16,a17))
        goto a12;
if (a13 > 0 && a13 < (a11,a12,a14,a15,a16,a17))
        goto a13;
if (a14 > 0 && a14 < (a11,a12,a13,a15,a16,a17))
        goto a14;
if (a15 > 0 && a15 < (a11,a12,a13,a14,a16,a17))
```

```
        goto a15;
if (a16 > 0 && a16 < (a11,a12,a13,a14,a15,a17))
        goto a16;
if (a17 > 0 );
        goto a17;
a11:fprintf(fp_out,"Choose #5 @ 6 in.(As = 0.62 sq.in./ft.)\n");
goto b11;
a12:fprintf(fp_out,"Choose #6 @ 6 in.(As = 0.88 sq.in./ft.)\n");
goto b12;
a13:fprintf(fp_out,"Choose #7 @ 6 in.(As = 1.20 sq.in./ft.)\n");
goto b13;
a14:fprintf(fp_out,"Choose #8 @ 6 in.(As = 1.58 sq.in./ft.)\n");
goto b14;
a15:fprintf(fp_out,"Choose #9 @ 6 in.(As = 2.00 sq.in./ft.)\n");
goto b15;
a16:fprintf(fp_out,"Choose #10 @ 6 in.(As = 2.54 sq.in./ft.)\n");
goto b16;
a17:fprintf(fp_out,"Choose #11 @ 6 in.(As = 3.12 sq.in./ft.)\n");
goto b17;
b11:reqdas = 0.62;
goto c1;
b12:reqdas = 0.88;
goto c1;
b13:reqdas = 1.20;
goto c1;
b14:reqdas = 1.58;
goto c1;
b15:reqdas = 2.00;
goto c1;
b16:reqdas = 2.54;
goto c1;
b17:reqdas = 3.12;
c1:df = sqrt(2.0 * reqdas / 3.1415926);
ld1 = (20 * reqdas * fy)/(sqrt(1000.0 * fc));
ld2 = 0.4 * df * fy;
if (ld1 > ld2)
        ldf = ld1;
else ldf = ld2;
ld = ldf * 1.4;
fprintf(fp_out,"The required development length of the bars (at the top) is %.2f in. into
the  toe of the footing measured from the stem reinforcement.",ld);
ld11 = (ld + 5.0)/12.0;
fprintf(fp_out,"Use an embedment length of %.1f ft. from the backface of the
wall.\n\n\n",ld11);
fprintf(fp_out,"-----------------------------------------------------------------------\n");
fprintf(fp_out,"                design of toe cantilever                    \n");
fprintf(fp_out,"-----------------------------------------------------------------------\n\n");
p11 = (2.0 * wtot)/lbase;
x4 = 0.5 * x;
p12 = x * p11 / lbase;
p13 = (lbase - t) * p11 / lbase;
vutoe = 1.7 * (((p11 + p13) * 0.5) - wfooting) * t;
mu11 = 0.5 * p11 * x4 * x4 * 0.6667;
mu12 = 0.5 * p12 * x4 * x4 * 0.3333;
```

```
mu13 = 0.5 * wfooting * x4 * x4;
muw = 1.7 * (mu11 + mu12 - mu13);
d22:d22 = (t * 12.0) - 2.5;
phivc = 0.85 * 2.0 * sqrt(fc * 1000) * 12.0 * d22 / 1000.0;
if (phivc > vutoe)
        goto run;
else d22 = d22 +1.0;
goto d22;
run:fprintf(fp_out,"PhiVc(%.2f k.) > Vu(%.2f k.) ---- OK\n",phivc,vutoe);
run = (muw * 12000) / (0.9 * 12.0 * d22 * d22);
ron = (1.0 - sqrt(1.0 - (0.002 * m * run / fy)))/m;
minro = 0.2 / fy;
if (minro > ron)
        ron = 1.333 * ron;
else ron = ron;
goto comm1;
com1:ron = ro;
comm1:reqdas = ron * 12.0 * d22;
a11 = 0.62 - reqdas;
a12 = 0.88 - reqdas;
a13 = 1.20 - reqdas;
a14 = 1.58 - reqdas;
a15 = 2.00 - reqdas;
a16 = 2.54 - reqdas;
a17 = 3.12 - reqdas;
if (a11 > 0 && a11 < (a12,a13,a14,a15,a16,a17))
        goto a21;
if (a12 > 0 && a12 < (a11,a12,a14,a15,a16,a17))
        goto a22;
if (a13 > 0 && a13 < (a11,a12,a14,a15,a16,a17))
        goto a23;
if (a14 > 0 && a14 < (a11,a12,a13,a15,a16,a17))
        goto a24;
if (a15 > 0 && a15 < (a11,a12,a13,a14,a16,a17))
        goto a25;
if (a16 > 0 && a16 < (a11,a12,a13,a14,a15,a17))
        goto a26;
if (a17 > 0 );
        goto a27;
a21:fprintf(fp_out,"Choose #5 @ 6 in.(As = 0.62 sq.in./ft.)\n");
goto b21;
a22:fprintf(fp_out,"Choose #6 @ 6 in.(As = 0.88 sq.in./ft.)\n");
goto b22;
a23:fprintf(fp_out,"Choose #7 @ 6 in.(As = 1.20 sq.in./ft.)\n");
goto b23;
a24:fprintf(fp_out,"Choose #8 @ 6 in.(As = 1.58 sq.in./ft.)\n");
goto b24;
a25:fprintf(fp_out,"Choose #9 @ 6 in.(As = 2.00 sq.in./ft.)\n");
goto b25;
a26:fprintf(fp_out,"Choose #10 @ 6 in.(As = 2.54 sq.in./ft.)\n");
goto b26;
a27:fprintf(fp_out,"Choose #11 @ 6 in.(As = 3.12 sq.in./ft.)\n");
goto b27;
b21:reqdas = 0.62;
```

```
goto c2;
b22:reqdas = 0.88;
goto c2;
b23:reqdas = 1.20;
goto c2;
b24:reqdas = 1.58;
goto c2;
b25:reqdas = 2.00;
goto c2;
b26:reqdas = 2.54;
goto c2;
b27:reqdas = 3.12;
c2:df = sqrt(2.0 * reqdas / 3.1415926);
ld1 = (20 * reqdas * fy) / (sqrt(1000.0 * fc));
ld2 = 0.4 * df * fy;
if (ld1 > ld2)
        ldf = ld1;
else ldf = ld2;
ld = ldf * 1.4;
fprintf(fp_out,"The required development length of the bottom bars is %.2f in.\n",ld);
ldl = (ld + 5.0)/12.0;
fprintf(fp_out,"Use an embedment length of %.1f ft. from the front face of the
wall\n\n\n",ldl);
fprintf(fp_out,"--------------------------------------------------------------------\n");
fprintf(fp_out,"              design of reinforcement of wall             \n");
fprintf(fp_out,"--------------------------------------------------------------------\n\n");
vutoe = 1.7 * (((p11 + p13) * 0.5) - wfooting) * t;
mu11 = 0.5 * p11 * x4 * x4 * 0.6667;
mu12 = 0.5 * p12 * x4 * x4 * 0.3333;
mu13 = 0.5 * wfooting * x4 * x4;
muw = 1.7 * (mu11 + mu12 - mu13);
ruwall = mu * 12000 / (0.9 * 12 * y * y);
ronwall = (1 - sqrt(1.0 - (0.002 * m * ruwall / fy))) / m;
if (minro > ronwall)
        ronwall = 1.333 * ronwall;
else ronwall = ronwall;
reqdas = ronwall * 12 * y;
a11 = 0.62 - reqdas;
a12 = 0.88 - reqdas;
a13 = 1.20 - reqdas;
a14 = 1.58 - reqdas;
a15 = 2.00 - reqdas;
a16 = 2.54 - reqdas;
a17 = 3.12 - reqdas;
if (a11 > 0 && a11 < (a12,a13,a14,a15,a16,a17))
        goto a31;
if (a12 > 0 && a12 < (a11,a12,a14,a15,a16,a17))
        goto a32;
if (a13 > 0 && a13 < (a11,a12,a14,a15,a16,a17))
        goto a33;
if (a14 > 0 && a14 < (a11,a12,a13,a15,a16,a17))
        goto a34;
if (a15 > 0 && a15 < (a11,a12,a13,a14,a16,a17))
        goto a35;
```

```
if (a16 > 0 && a16 < (a11,a12,a13,a14,a15,a17))
        goto a36;
if (a17 > 0 )
        goto a37;
a31:fprintf(fp_out,"Choose #5 @ 6 in.(As = 0.62 sq.in./ft.)\n");
goto b31;
a32:fprintf(fp_out,"Choose #6 @ 6 in.(As = 0.88 sq.in./ft.)\n");
goto b32;
a33:fprintf(fp_out,"Choose #7 @ 6 in.(As = 1.20 sq.in./ft.)\n");
goto b33;
a34:fprintf(fp_out,"Choose #8 @ 6 in.(As = 1.58 sq.in./ft.)\n");
goto b34;
a35:fprintf(fp_out,"Choose #9 @ 6 in.(As = 2.00 sq.in./ft.)\n");
goto b35;
a36:fprintf(fp_out,"Choose #10 @ 6 in.(As = 2.54 sq.in./ft.)\n");
goto b36;
a37:fprintf(fp_out,"Choose #11 @ 6 in.(As = 3.12 sq.in./ft.)\n");
goto b37;
b31:reqdas = 0.62;
goto c3;
b32:reqdas = 0.88;
goto c3;
b33:reqdas = 1.20;
goto c3;
b34:reqdas = 1.58;
goto c3;
b35:reqdas = 2.00;
goto c3;
b36:reqdas = 2.54;
goto c3;
b37:reqdas = 3.12;
c3:df = sqrt(2.0 * reqdas / 3.1415926);
ld1 = (20 * reqdas * fy) / (sqrt(1000.0 * fc));
ld2 = 0.4 * df * fy;
if (ld1 > ld2)
        ldf = ld1;
else ldf = ld2;
ld = ldf * 1.4;
fprintf(fp_out,"The required development length of the bottom bars is %.2f in.\n",ld);
ldl = (ld + 5.0)/12.0;
fprintf(fp_out,"Use an embedment length of %.1f ft. \n\n\n",ldl);
goto layer50;
hw11:printf("                                              \n");
fprintf(fp_out,"------------------------------------------------------------ \n");
fprintf(fp_out,"      calculations for one layer with water-table        \n");
fprintf(fp_out,"------------------------------------------------------------ \n");
gb1 = g1 - 62.5;
h11 = h1 - hw;
x2 = g1 * ca * h11;
x3 = gb1 * hw * ca;
x4 = gw * hw;
p1 = x1 * h11;
p2 = 0.5 * x2 * h11;
p3 = (x1 + x2) * hw;
```

```
p4 = 0.5 * (x3 + x4) * hw;
p = (p1 * (hw + h11/2)) + (p2 * (h11/3 + hw)) + (p3 * hw/2) + (p4 * hw/3);
wsub1 = (surcht * gsur) + (g1 * h11) + ((gb1+gw) * hw);
x = pow(((2 * p) / wsub1),0.5);
goto lay1b;
lay1bb:printf("                                        \n");
fprintf(fp_out,"Recalculating as F.S. is below requirement!\n");
x = x + 0.2;
lay1b:lbase = 1.5 * x;
fprintf(fp_out,"Base length of the footing = %.2f ft.\n\n",lbase);
m1 = p1 * (h11/2 + hw - t);
m2 = p2 * (h11/3 + hw - t);
m3 = (x1 + x2) * (hw - t) * (hw - t) * 0.5;
m4 = ((x3 + x4) * (pow((hw-t),3))) / (6 * hw);
mu = (1.7 * (m1 + m2 + m3 + m4)) / 1000;
t1 = pow(((mu * 12000)/(0.9 * ru * 12.0)),0.5);
tb = t1 + 5;
if (tb < 12)
        tb = 12.0;
tt = 12.0;
fprintf(fp_out,"STEM DIMENSIONS:\n");
fprintf(fp_out,"----------------------------\n");
fprintf(fp_out,"  a) stem thickness at the bottom = %d in.\n",tb);
fprintf(fp_out,"  b) stem thickness at the top    = %d in.\n\n",tt);
w1 = (gsur / 1000) * surcht * (x - 1);
w2 = (g1 / 1000) * h11 * (x - 1);
w3 = (gb1 / 1000) * (hw - t) * (x - 1);
tbt = tb - tt;
w4 = (0.15 - (g1 / 1000)) * ((tbt * h11 * h11) / (24 * (h1 - t)));
w5 = (0.15 - (gb1 / 1000)) * 0.5 * ((tbt / 12) + (h11 / (12 * (h1 - t)))) * (hw - t);
w6 = 0.15 * lbase * t;
w7 = 0.15 * (h1 - t);
wtot = w1 + w2 + w3 + w4 + w5 + w6 + w7;
lev1 = (x - 1) * 0.5;
lev4 = (2 * lev1) - ((tbt * h11) / ((h1 - t) * 36));
lev5 = (2 * lev1) - ((tbt * h11) / ((h1 - t) * 24)) - ((1 / 3) * ((tbt / 12) - ((tbt * h11) / ((h1 -
t) * 36))));
lev6 = lbase * 0.5;
lev7 = (lev1 * 2.0) + 0.5;
mw1 = w1 * lev1;
mw2 = w2 * lev1;
mw3 = w3 * lev1;
mw4 = w4 * lev4;
mw5 = w5 * lev5;
mw6 = w6 * lev6;
mw7 = w7 * lev7;
mwt = mw1 + mw2 + mw3 + mw4 + mw5 + mw6 + mw7;
ywm = mwt/wtot;
resmom = wtot * (lbase - ywm);
ovemom = ((p1 * (h11/2 + hw)) + (p2 * ((h11/3) + hw)) + (p3 * (hw/2)) + (p4 *
(hw/3)))/1000.0;
fsover = resmom/ovemom;
fprintf(fp_out,"Overturning stability:\n");
fprintf(fp_out,"-------------------------\n");
```

```
fprintf(fp_out,"Resisting moment = %.2f ft-kips\n",resmom);
fprintf(fp_out,"Overturning moment = %.2f ft-kips\n",ovemom);
if (fsover < 2.0)
        goto lay1bb;
else
xbar = (mwt + ovemom) / wtot;
xbar1 = lbase/3.0;
xbar2 = (2.0 * lbase) / 3.0;
xbar3 = xbar + .08;
xbar4 = (xbar - xbar2)*12.0;
fprintf(fp_out,"Factor of safety against overturning = %.2f / %.2f = %.2f --OK-- (F.S. >
2.0)\n\n\n",resmom,ovemom,fsover);
if (xbar > xbar1 && xbar < xbar2)
fprintf(fp_out,"The position of the resultant soil pressure due to the service load was
checked and found to be within the middle-third of the base --- OK\n\n");
 else if (xbar > xbar1 && xbar < xbar3)
  fprintf(fp_out,"The resultant soil pressure lies %.1f in. outside the middle-third,
however,it is very close, and the limiting condition of zero stress at the heel is considered
adequate.\n\n\n",xbar4);
if (xbar > xbar1 && xbar > xbar3)
        goto lay1bb;
fprintf(fp_out,"Sliding stability:\n");
fprintf(fp_out,"--------------------\n");
fsldg = (p1 + p2 + p3 + p4) / 1000;
ffric = miu * wtot;
fssldg = ffric / fsldg;
fprintf(fp_out,"Sliding force = %.2f kips\n",fsldg);
fprintf(fp_out,"Resisting force = %.2f kips\n",ffric);
fprintf(fp_out,"Factor of safety against sliding = %.2f / %.2f =
%.2f\n",ffric,fsldg,fssldg);
if (fssldg > 1.5)
        goto stp1;
else fprintf(fp_out,"Key to be provided as factor of safety against sliding (%.2f) is less
than 1.5\n",fssldg);
ke = 8 * t;
fprintf(fp_out,"Key dimensions(CRSI Handbook):The key has a square section (%.1fin.
x %.1fin.)\n",ke,ke);
fprintf(fp_out,"        Generally it is desirable to place the front face of the key about 5
in. in front of the back face of the stem.This will permit anchoring the stem reinforcement,
if present, in the key.\n\n\n");
goto heel1;
stp1:fprintf(fp_out,"---- OK (F.S. > 1.5).\n\n\n");
heel1:fprintf(fp_out,"-----------------------------------------------------------------\n");
fprintf(fp_out,"                design of heel cantilever \n");
fprintf(fp_out, "       ---------------------------------------------------------\n\n");
wearth1 = g1 * h11 / 1000.0;
wearth2 = gb1 * (hw - t) / 1000.0;
wsurch = gsur * surcht / 1000.0;
wfooting = 0.15 * t;
wu = (1.4 * (wearth1 + wearth2 + wfooting)) + (1.7 * wsurch);
lev8 = x - (tb / 12.0);
muw = 0.5 * wu * lev8 * lev8;
vuw = wu * lev8;
phivcw = 0.0204 * sqrt(fc * 1000.0) * tb;
```

```
if (phivcw < vuw)
        dreqd = (tb * vuw / phivcw) + 1.0;
else goto comal;
t1n = dreqd + 2.5;
fprintf(fp_out,"Heel thickness = %d\n",t1n);
run = (muw * 12000) / (0.9 * 12.0 * dreqd * dreqd);
ron = (1.0 - sqrt(1.0 - (0.002 * m * run / fy)))/m;
minro = 0.2 / fy;
  if (minro > ron)
        ron = 1.333 * ron;
else ron = ron;
goto commal;
comal: ron = ro;
commal:reqdas = ron * 12.0 * dreqd;
a11 = 0.62 - reqdas;
a12 = 0.88 - reqdas;
a13 = 1.20 - reqdas;
a14 = 1.58 - reqdas;
a15 = 2.00 - reqdas;
a16 = 2.54 - reqdas;
a17 = 3.12 - reqdas;
if (a11 > 0 && a11 < (a12,a13,a14,a15,a16,a17))
        goto a111;
if (a12 > 0 && a12 < (a11,a13,a14,a15,a16,a17))
        goto a121;
if (a13 > 0 && a13 < (a11,a12,a14,a15,a16,a17))
        goto a131;
if (a14 > 0 && a14 < (a11,a12,a13,a15,a16,a17))
        goto a141;
if (a15 > 0 && a15 < (a11,a12,a13,a14,a16,a17))
        goto a151;
if (a16 > 0 && a16 < (a11,a12,a13,a14,a15,a17))
        goto a161;
if (a17 > 0 );
        goto a171;
a111:fprintf(fp_out,"Choose #5 @ 6 in.(As = 0.62 sq.in./ft.)\n");
goto b111;
a121:fprintf(fp_out,"Choose #6 @ 6 in.(As = 0.88 sq.in./ft.)\n");
goto b121;
a131:fprintf(fp_out,"Choose #7 @ 6 in.(As = 1.20 sq.in./ft.)\n");
goto b131;
a141:fprintf(fp_out,"Choose #8 @ 6 in.(As = 1.58 sq.in./ft.)\n");
goto b141;
a151:fprintf(fp_out,"Choose #9 @ 6 in.(As = 2.00 sq.in./ft.)\n");
goto b151;
a161:fprintf(fp_out,"Choose #10 @ 6 in.(As = 2.54 sq.in./ft.)\n");
goto b161;
a171:fprintf(fp_out,"Choose #11 @ 6 in.(As = 3.12 sq.in./ft.)\n");
goto b171;
b111:reqdas = 0.62;
goto c11;
b121:reqdas = 0.88;
goto c11;
b131:reqdas = 1.20;
```

```
goto c11;
b141:reqdas = 1.58;
goto c11;
b151:reqdas = 2.00;
goto c11;
b161:reqdas = 2.54;
goto c11;
b171:reqdas = 3.12;
c11:df = sqrt(2.0 * reqdas / 3.1415926);
ld1 = (20 * reqdas * fy)/(sqrt(1000.0 * fc));
ld2 = 0.4 * df * fy;
if (ld1 > ld2)
        ldf = ld1;
else ldf = ld2;
ld = ldf * 1.4;
fprintf(fp_out,"The required development length of the bars (at the top) is %.2f in. into
the  toe of the footing measured from the stem reinforcement.",ld);
ld11 = (ld + 5.0)/12.0;
fprintf(fp_out,"Use an embedment length of %.1f ft. from the backface of the
wall.\n\n\n",ld11);
fprintf(fp_out,"----------------------------------------------------------------------\n");
fprintf(fp_out,"              design of toe cantilever                  \n");
fprintf(fp_out,"----------------------------------------------------------------------\n\n");
p11 = (2.0 * wtot)/lbase;
x4 = 0.5 * x;
p12 = x * p11 / lbase;
p13 = (lbase - t) * p11 / lbase;
vutoe = 1.7 * (((p11 + p13) * 0.5) - wfooting) * t;
mu11 = 0.5 * p11 * x4 * x4 * 0.6667;
mu12 = 0.5 * p12 * x4 * x4 * 0.3333;
mu13 = 0.5 * wfooting * x4 * x4;
muw = 1.7 * (mu11 + mu12 - mu13);
d221:d22 = (t * 12.0) - 2.5;
phivc = 0.85 * 2.0 * sqrt(fc * 1000) * 12.0 * d22 / 1000.0;
if (phivc > vutoe)
        goto run1;
else d22 = d22 +1.0;
goto d221;
run1:fprintf(fp_out,"PhiVc(%.2f k.) > Vu(%.2f k.) ---- OK\n",phivc,vutoe);
run = (muw * 12000) / (0.9 * 12.0 * d22 * d22);
ron = (1.0 - sqrt(1.0 - (0.002 * m * run / fy)))/m;
minro = 0.2 / fy;
if (minro > ron)
        ron = 1.333 * ron;
else ron = ron;
goto comm11;
com11:ron = ro;
comm11:reqdas = ron * 12.0 * d22;
a11 = 0.62 - reqdas;
a12 = 0.88 - reqdas;
a13 = 1.20 - reqdas;
a14 = 1.58 - reqdas;
a15 = 2.00 - reqdas;
a16 = 2.54 - reqdas;
```

```
a17 = 3.12 - reqdas;
if (a11 > 0 && a11 < (a12,a13,a14,a15,a16,a17))
        goto a211;
if (a12 > 0 && a12 < (a11,a12,a14,a15,a16,a17))
        goto a221;
if (a13 > 0 && a13 < (a11,a12,a14,a15,a16,a17))
        goto a231;
if (a14 > 0 && a14 < (a11,a12,a13,a15,a16,a17))
        goto a241;
if (a15 > 0 && a15 < (a11,a12,a13,a14,a16,a17))
        goto a251;
if (a16 > 0 && a16 < (a11,a12,a13,a14,a15,a17))
        goto a261;
if (a17 > 0 );
        goto a271;
a211:fprintf(fp_out,"Choose #5 @ 6 in.(As = 0.62 sq.in./ft.)\n");
goto b211;
a221:fprintf(fp_out,"Choose #6 @ 6 in.(As = 0.88 sq.in./ft.)\n");
goto b221;
a231:fprintf(fp_out,"Choose #7 @ 6 in.(As = 1.20 sq.in./ft.)\n");
goto b231;
a241:fprintf(fp_out,"Choose #8 @ 6 in.(As = 1.58 sq.in./ft.)\n");
goto b241;
a251:fprintf(fp_out,"Choose #9 @ 6 in.(As = 2.00 sq.in./ft.)\n");
goto b251;
a261:fprintf(fp_out,"Choose #10 @ 6 in.(As = 2.54 sq.in./ft.)\n");
goto b261;
a271:fprintf(fp_out,"Choose #11 @ 6 in.(As = 3.12 sq.in./ft.)\n");
goto b271;
b211:reqdas = 0.62;
goto c21;
b221:reqdas = 0.88;
goto c21;
b231:reqdas = 1.20;
goto c21;
b241:reqdas = 1.58;
goto c21;
b251:reqdas = 2.00;
goto c21;
b261:reqdas = 2.54;
goto c21;
b271:reqdas = 3.12;
c21:df = sqrt(2.0 * reqdas / 3.1415926);
ld1 = (20 * reqdas * fy) / (sqrt(1000.0 * fc));
ld2 = 0.4 * df * fy;
if (ld1 > ld2)
        ldf = ld1;
else ldf = ld2;
ld = ldf * 1.4;
fprintf(fp_out,"The required development length of the bottom bars is %.2f in.\n",ld);
ldl = (ld + 5.0)/12.0;
fprintf(fp_out,"Use an embedment length of %.1f ft. from the front face of the
wall\n\n\n",ldl);
fprintf(fp_out,"------------------------------------------------------------------------\n");
```

```
fprintf(fp_out,"          design of reinforcement of wall          \n");
fprintf(fp_out,"--------------------------------------------------------------------------\n\n");
vutoe = 1.7 * (((p11 + p13) * 0.5) - wfooting) * t;
mu11 = 0.5 * p11 * x4 * x4 * 0.6667;
mu12 = 0.5 * p12 * x4 * x4 * 0.3333;
mu13 = 0.5 * wfooting * x4 * x4;
muw = 1.7 * (mu11 + mu12 - mu13);
y = h1 - t;
ruwall = mu * 12000 / (0.9 * 12 * y * y);
ronwall = (1 - sqrt(1.0 - (0.002 * m * ruwall / fy))) / m;
if (minro > ronwall)
        ronwall = 1.333 * ronwall;
else ronwall = ronwall;
reqdas = ronwall * 12 * (h1 - t);
a11 = 0.62 - reqdas;
a12 = 0.88 - reqdas;
a13 = 1.20 - reqdas;
a14 = 1.58 - reqdas;
a15 = 2.00 - reqdas;
a16 = 2.54 - reqdas;
a17 = 3.12 - reqdas;
if (a11 > 0 && a11 < (a12,a13,a14,a15,a16,a17))
        goto a311;
if (a12 > 0 && a12 < (a11,a12,a14,a15,a16,a17))
        goto a321;
if (a13 > 0 && a13 < (a11,a12,a14,a15,a16,a17))
        goto a331;
if (a14 > 0 && a14 < (a11,a12,a13,a15,a16,a17))
        goto a341;
if (a15 > 0 && a15 < (a11,a12,a13,a14,a16,a17))
        goto a351;
if (a16 > 0 && a16 < (a11,a12,a13,a14,a15,a17))
        goto a361;
if (a17 > 0 )
        goto a371;
a311:fprintf(fp_out,"Choose #5 @ 6 in.(As = 0.62 sq.in./ft.)\n");
goto b311;
a321:fprintf(fp_out,"Choose #6 @ 6 in.(As = 0.88 sq.in./ft.)\n");
goto b321;
a331:fprintf(fp_out,"Choose #7 @ 6 in.(As = 1.20 sq.in./ft.)\n");
goto b331;
a341:fprintf(fp_out,"Choose #8 @ 6 in.(As = 1.58 sq.in./ft.)\n");
goto b341;
a351:fprintf(fp_out,"Choose #9 @ 6 in.(As = 2.00 sq.in./ft.)\n");
goto b351;
a361:fprintf(fp_out,"Choose #10 @ 6 in.(As = 2.54 sq.in./ft.)\n");
goto b361;
a371:fprintf(fp_out,"Choose #11 @ 6 in.(As = 3.12 sq.in./ft.)\n");
goto b371;
b311:reqdas = 0.62;
goto c31;
b321:reqdas = 0.88;
goto c31;
b331:reqdas = 1.20;
```

```
goto c31;
b341:reqdas = 1.58;
goto c31;
b351:reqdas = 2.00;
goto c31;
b361:reqdas = 2.54;
goto c31;
b371:reqdas = 3.12;
c31:df = sqrt(2.0 * reqdas / 3.1415926);
ld1 = (20 * reqdas * fy) / (sqrt(1000.0 * fc));
ld2 = 0.4 * df * fy;
if (ld1 > ld2)
        ldf = ld1;
else ldf = ld2;
ld = ldf * 1.4;
fprintf(fp_out,"The required development length of the bottom bars is %.2f in.\n",ld);
ldl = (ld + 5.0)/12.0;
fprintf(fp_out,"Use an embedment length of %.1f ft. \n\n\n",ldl);
goto layer50;

/*      ----------------------------------------------------    */
/*      calculations for two layers without water-table         */
/*      ----------------------------------------------------    */

layer2:printf("What is the depth of the first soil layer?\n");
scanf("%f",&h11);
printf("What is the density of the soil in the first layer?\n");
scanf("%f",&g1);
printf("What is the angle of internal friction of the soil in the first layer?\n");
scanf("%f",&alpha1);
printf("What is the density of the soil in the second layer?\n");
scanf("%f",&g2);
printf("What is the angle of internal friction of the soil in the second layer?\n");
scanf("%f",&alpha2);
h12 = h1 - h11;
aalpha = alpha1 * 3.1415926 / 180.0;
ca = (1 - (sin(aalpha)))/(1 + (sin(aalpha)));
fprintf(fp_out,"Coefficient of active pressure = %.4f\n",ca);
if (hw != 0)
        goto hw22;
x2 = ca * g1 * h11;
aalpha = alpha2 * 3.1415926 / 180.0;
ca = (1 - (sin(aalpha)))/(1 + (sin(aalpha)));
fprintf(fp_out,"------------------------------------------------------------ \n");
fprintf(fp_out,"    calculations for two layers without water-table         \n");
fprintf(fp_out,"------------------------------------------------------------ \n");
fprintf(fp_out,"Coefficient of active pressure = %.4f\n",ca);
x3 = ca * g2 * h12;
p1 = x1 * h11;
p2 = 0.5 * x2 * h11;
p3 = (x1 + x2) * h12;
p4 = 0.5 * x3 * h12;
p = (p1 * (h12 + h11/2)) + (p2 * (h11/3 + h12)) + (p3 * h12/2) + (p4 * h12/3);
wsub1 = (surcht * gsur) + (g1 * h11) + (g2 * h12);
```

```
x = pow(((2 * p) / wsub1),0.5);
goto lay2b;
lay2bb:fprintf(fp_out,"Recalculating as F.S. is below requirement!\n");
x = x + 0.2;
lay2b:lbase = 1.5 * x;
fprintf(fp_out,"Base length of the footing = %.2f ft.\n\n",lbase);
m1 = x1 * h11 * (h11/2 + h12 - t);
m2 = 0.5 * x2 * h11 * (h11/3 + h12 - t);
m3 = (x1 + x2) * (h12 - t) * (h12 - t) * 0.5;
m4 = 0.5 * ((h12-t)*x3/h12) *(h12 - t) * (h12 - t) * 0.3333;
mu = (1.7 * (m1 + m2 + m3 + m4)) / 1000;
t1 = sqrt((mu * 12000)/(0.9 * ru * 12.0));
tb = t1 + 5;
if (tb < 12)
        tb = 12.0;
tt = 12.0;
fprintf(fp_out,"STEM DIMENSIONS:\n");
fprintf(fp_out,"----------------------------\n");
fprintf(fp_out," a) stem thickness at the bottom = %d in.\n",tb);
fprintf(fp_out," b) stem thickness at the top    = %d in.\n\n",tt);
w1 = (gsur / 1000) * surcht * (x - 1);
w2 = (g1 / 1000) * h11 * (x - 1);
w3 = (g2 / 1000) * (h12 - t) * (x - 1);
tbt = tb - tt;
w4 = (0.15 - (g1 / 1000)) * ((tbt * h11 * h11) / (24 * (h1 - t)));
w5 = (0.15 - (g2 / 1000)) * 0.5 * ((tbt / 12) + (h11 / (12 * (h1 - t)))) * (h12 - t);
w6 = 0.15 * lbase * t;
w7 = 0.15 * (h1 - t);
wtot = w1 + w2 + w3 + w4 + w5 + w6 + w7;
lev1 = (x - 1) * 0.5;
lev4 = (2 * lev1) - ((tbt * h11) / ((h1 - t) * 36));
lev5 = (2 * lev1) - ((tbt * h11) / ((h1 - t) * 24)) - ((1 / 3) * ((tbt / 12) - ((tbt * h11) / ((h1 -
t) * 36))));
lev6 = lbase * 0.5;
lev7 = (lev1 * 2.0) + 0.5;
mw1 = w1 * lev1;
mw2 = w2 * lev1;
mw3 = w3 * lev1;
mw4 = w4 * lev4;
mw5 = w5 * lev5;
mw6 = w6 * lev6;
mw7 = w7 * lev7;
mwt = mw1 + mw2 + mw3 + mw4 + mw5 + mw6 + mw7;
ywm = mwt/wtot;
resmom = wtot * (lbase - ywm);
ovemom = ((p1 * (h11 / 2 + h12)) + (p2 * ((h11 / 3) + h12)) + (p3 * (h12 / 2)) + (p4 *
(h12 / 3)))/1000.0;
fsover = resmom/ovemom;
fprintf(fp_out,"Overturning stability:\n");
fprintf(fp_out,"--------------------------\n");
fprintf(fp_out,"Resisting moment = %.2f ft-kips\n",resmom);
fprintf(fp_out,"Overturning moment = %.2f ft-kips\n",ovemom);
if (fsover < 2.0)
        goto lay2bb;
```

```
else
xbar = (mwt + ovemom) / wtot;
xbar1 = lbase/3.0;
xbar2 = (2.0 * lbase) / 3.0;
xbar3 = xbar + .08;
xbar4 = (xbar - xbar2)*12.0;
fprintf(fp_out,"Factor of safety against overturning = %.2f / %.2f = %.2f --OK-- (F.S. >
2.0)\n\n\n",resmom,ovemom,fsover);
if (xbar > xbar1 && xbar < xbar2)
fprintf(fp_out,"The position of the resultant soil pressure due to the service load was
checked and found to be within the middle-third of the base --- OK\n\n");
 else if (xbar > xbar1 && xbar < xbar3)
  fprintf(fp_out,"The resultant soil pressure lies %.1f in. outside the middle-third,
however,it is very close, and the limiting condition of zero stress at the heel is considered
adequate.\n\n\n",xbar4);
if (xbar > xbar1 && xbar > xbar3)
        goto lay2bb;
fprintf(fp_out,"Sliding stability:\n");
fprintf(fp_out,"--------------------\n");
fsldg = (p1 + p2 + p3 + p4) / 1000;
ffric = miu * wtot;
fssldg = ffric / fsldg;
fprintf(fp_out,"Sliding force = %.2f kips\n",fsldg);
fprintf(fp_out,"Resisting force = %.2f kips\n",ffric);
fprintf(fp_out,"Factor of safety against sliding = %.2f / %.2f =
%.2f\n",ffric,fsldg,fssldg);
if (fssldg > 1.5)
        goto stp2;
else fprintf(fp_out,"Key to be provided as factor of safety against sliding (%.2f) is less
than 1.5\n",fssldg);
ke = 8 * t;
fprintf(fp_out,"Key dimensions(CRSI Handbook):The key has a square section (%.1fin.
x %.1fin.)\n",ke,ke);
fprintf(fp_out,"        Generally it is desirable to place the front face of the key about 5
in. in front of the back face of the stem.This will permit anchoring the stem reinforcement,
if present, in the key.\n\n\n");
goto heel2;
stp2:fprintf(fp_out,"---- OK (F.S. > 1.5).\n\n\n");
heel2:fprintf(fp_out,"-----------------------------------------------------------------\n");
fprintf(fp_out,"              design of heel cantilever \n");
fprintf(fp_out, "   ----------------------------------------------------------------\n\n");
wearth1 = g1 * h11 / 1000.0;
wearth2 = g2 * (h12 - t) / 1000.0;
wsurch = gsur * surcht / 1000.0;
wfooting = 0.15 * t;
wu = (1.4 * (wearth1 + wearth2 + wfooting)) + (1.7 * wsurch);
lev8 = x - (tb / 12.0);
muw = 0.5 * wu * lev8 * lev8;
vuw = wu * lev8;
phivcw = 0.0204 * sqrt(fc * 1000.0) * tb;
if (phivcw < vuw)
        dreqd = (tb * vuw / phivcw) + 1.0;
else goto coma2;
t1n = dreqd + 2.5;
```

```
fprintf(fp_out,"Heel thickness = %d\n",t1n);
run = (muw * 12000) / (0.9 * 12.0 * dreqd * dreqd);
ron = (1.0 - sqrt(1.0 - (0.002 * m * run / fy)))/m;
minro = 0.2 / fy;
  if (minro > ron)
        ron = 1.333 * ron;
else ron = ron;
goto comma2;
coma2: ron = ro;
comma2:reqdas = ron * 12.0 * dreqd;
a11 = 0.62 - reqdas;
a12 = 0.88 - reqdas;
a13 = 1.20 - reqdas;
a14 = 1.58 - reqdas;
a15 = 2.00 - reqdas;
a16 = 2.54 - reqdas;
a17 = 3.12 - reqdas;
if (a11 > 0 && a11 < (a12,a13,a14,a15,a16,a17))
        goto a112;
if (a12 > 0 && a12 < (a11,a13,a14,a15,a16,a17))
        goto a122;
if (a13 > 0 && a13 < (a11,a12,a14,a15,a16,a17))
        goto a132;
if (a14 > 0 && a14 < (a11,a12,a13,a15,a16,a17))
        goto a142;
if (a15 > 0 && a15 < (a11,a12,a13,a14,a16,a17))
        goto a152;
if (a16 > 0 && a16 < (a11,a12,a13,a14,a15,a17))
        goto a162;
if (a17 > 0 );
        goto a172;
a112:fprintf(fp_out,"Choose #5 @ 6 in.(As = 0.62 sq.in./ft.)\n");
goto b112;
a122:fprintf(fp_out,"Choose #6 @ 6 in.(As = 0.88 sq.in./ft.)\n");
goto b122;
a132:fprintf(fp_out,"Choose #7 @ 6 in.(As = 1.20 sq.in./ft.)\n");
goto b132;
a142:fprintf(fp_out,"Choose #8 @ 6 in.(As = 1.58 sq.in./ft.)\n");
goto b142;
a152:fprintf(fp_out,"Choose #9 @ 6 in.(As = 2.00 sq.in./ft.)\n");
goto b152;
a162:fprintf(fp_out,"Choose #10 @ 6 in.(As = 2.54 sq.in./ft.)\n");
goto b162;
a172:fprintf(fp_out,"Choose #11 @ 6 in.(As = 3.12 sq.in./ft.)\n");
goto b172;
b112:reqdas = 0.62;
goto c12;
b122:reqdas = 0.88;
goto c12;
b132:reqdas = 1.20;
goto c12;
b142:reqdas = 1.58;
goto c12;
b152:reqdas = 2.00;
```

```
goto c12;
b162:reqdas = 2.54;
goto c12;
b172:reqdas = 3.12;
c12:df = sqrt(2.0 * reqdas / 3.1415926);
ld1 = (20 * reqdas * fy)/(sqrt(1000.0 * fc));
ld2 = 0.4 * df * fy;
if (ld1 > ld2)
        ldf = ld1;
else ldf = ld2;
ld = ldf * 1.4;
fprintf(fp_out,"The required development length of the bars (at the top) is %.2f in. into
the  toe of the footing measured from the stem reinforcement.",ld);
ldl1 = (ld + 5.0)/12.0;
fprintf(fp_out,"Use an embedment length of %.1f ft. from the backface of the
wall.\n\n\n",ldl1);
fprintf(fp_out,"----------------------------------------------------------------------\n");
fprintf(fp_out,"              design of toe cantilever              \n");
fprintf(fp_out,"----------------------------------------------------------------------\n\n");
p11 = (2.0 * wtot)/lbase;
x4 = 0.5 * x;
p12 = x * p11 / lbase;
p13 = (lbase - t) * p11 / lbase;
vutoe = 1.7 * (((p11 + p13) * 0.5) - wfooting) * t;
mu11 = 0.5 * p11 * x4 * x4 * 0.6667;
mu12 = 0.5 * p12 * x4 * x4 * 0.3333;
mu13 = 0.5 * wfooting * x4 * x4;
muw = 1.7 * (mu11 + mu12 - mu13);
d222:d22 = (t * 12.0) - 2.5;
phivc = 0.85 * 2.0 * sqrt(fc * 1000) * 12.0 * d22 / 1000.0;
if (phivc > vutoe)
        goto run1;
else d22 = d22 +1.0;
goto d222;
run2:fprintf(fp_out,"PhiVc(%.2f k.) > Vu(%.2f k.) ---- OK\n",phivc,vutoe);
run = (muw * 12000) / (0.9 * 12.0 * d22 * d22);
ron = (1.0 - sqrt(1.0 - (0.002 * m * run / fy)))/m;
minro = 0.2 / fy;
if (minro > ron)
        ron = 1.333 * ron;
else ron = ron;
goto comm11;
com12:ron = ro;
comm12:reqdas = ron * 12.0 * d22;
a11 = 0.62 - reqdas;
a12 = 0.88 - reqdas;
a13 = 1.20 - reqdas;
a14 = 1.58 - reqdas;
a15 = 2.00 - reqdas;
a16 = 2.54 - reqdas;
a17 = 3.12 - reqdas;
if (a11 > 0 && a11 < (a12,a13,a14,a15,a16,a17))
        goto a212;
if (a12 > 0 && a12 < (a11,a12,a14,a15,a16,a17))
```

```
        goto a222;
if (a13 > 0 && a13 < (a11,a12,a14,a15,a16,a17))
        goto a232;
if (a14 > 0 && a14 < (a11,a12,a13,a15,a16,a17))
        goto a242;
if (a15 > 0 && a15 < (a11,a12,a13,a14,a16,a17))
        goto a252;
if (a16 > 0 && a16 < (a11,a12,a13,a14,a15,a17))
        goto a262;
if (a17 > 0 );
        goto a272;
a212:fprintf(fp_out,"Choose #5 @ 6 in.(As = 0.62 sq.in./ft.)\n");
goto b212;
a222:fprintf(fp_out,"Choose #6 @ 6 in.(As = 0.88 sq.in./ft.)\n");
goto b222;
a232:fprintf(fp_out,"Choose #7 @ 6 in.(As = 1.20 sq.in./ft.)\n");
goto b232;
a242:fprintf(fp_out,"Choose #8 @ 6 in.(As = 1.58 sq.in./ft.)\n");
goto b242;
a252:fprintf(fp_out,"Choose #9 @ 6 in.(As = 2.00 sq.in./ft.)\n");
goto b252;
a262:fprintf(fp_out,"Choose #10 @ 6 in.(As = 2.54 sq.in./ft.)\n");
goto b262;
a272:fprintf(fp_out,"Choose #11 @ 6 in.(As = 3.12 sq.in./ft.)\n");
goto b272;
b212:reqdas = 0.62;
goto c22;
b222:reqdas = 0.88;
goto c22;
b232:reqdas = 1.20;
goto c22;
b242:reqdas = 1.58;
goto c22;
b252:reqdas = 2.00;
goto c22;
b262:reqdas = 2.54;
goto c22;
b272:reqdas = 3.12;
c22:df = sqrt(2.0 * reqdas / 3.1415926);
ld1 = (20 * reqdas * fy) / (sqrt(1000.0 * fc));
ld2 = 0.4 * df * fy;
if (ld1 > ld2)
        ldf = ld1;
else ldf = ld2;
ld = ldf * 1.4;
fprintf(fp_out,"The required development length of the bottom bars is %.2f in.\n",ld);
ldl = (ld + 5.0)/12.0;
fprintf(fp_out,"Use an embedment length of %.1f ft. from the front face of the
wall\n\n\n",ldl);
fprintf(fp_out,"--------------------------------------------------------------------\n");
fprintf(fp_out,"            design of reinforcement of wall            \n");
fprintf(fp_out,"--------------------------------------------------------------------\n\n");
vutoe = 1.7 * (((p11 + p13) * 0.5) - wfooting) * t;
mu11 = 0.5 * p11 * x4 * x4 * 0.6667;
```

```
mu12 = 0.5 * p12 * x4 * x4 * 0.3333;
mu13 = 0.5 * wfooting * x4 * x4;
muw = 1.7 * (mu11 + mu12 - mu13);
y = h1 - t;
ruwall = mu * 12000 / (0.9 * 12 * y * y);
ronwall = (1 - sqrt(1.0 - (0.002 * m * ruwall / fy))) / m;
if (minro > ronwall)
        ronwall = 1.333 * ronwall;
else ronwall = ronwall;
reqdas = ronwall * 12 * (h1 - t);
a11 = 0.62 - reqdas;
a12 = 0.88 - reqdas;
a13 = 1.20 - reqdas;
a14 = 1.58 - reqdas;
a15 = 2.00 - reqdas;
a16 = 2.54 - reqdas;
a17 = 3.12 - reqdas;
if (a11 > 0 && a11 < (a12,a13,a14,a15,a16,a17))
        goto a312;
if (a12 > 0 && a12 < (a11,a12,a14,a15,a16,a17))
        goto a322;
if (a13 > 0 && a13 < (a11,a12,a14,a15,a16,a17))
        goto a332;
if (a14 > 0 && a14 < (a11,a12,a13,a15,a16,a17))
        goto a342;
if (a15 > 0 && a15 < (a11,a12,a13,a14,a16,a17))
        goto a352;
if (a16 > 0 && a16 < (a11,a12,a13,a14,a15,a17))
        goto a362;
if (a17 > 0 )
        goto a372;
a312:fprintf(fp_out,"Choose #5 @ 6 in.(As = 0.62 sq.in./ft.)\n");
goto b312;
a322:fprintf(fp_out,"Choose #6 @ 6 in.(As = 0.88 sq.in./ft.)\n");
goto b322;
a332:fprintf(fp_out,"Choose #7 @ 6 in.(As = 1.20 sq.in./ft.)\n");
goto b332;
a342:fprintf(fp_out,"Choose #8 @ 6 in.(As = 1.58 sq.in./ft.)\n");
goto b342;
a352:fprintf(fp_out,"Choose #9 @ 6 in.(As = 2.00 sq.in./ft.)\n");
goto b352;
a362:fprintf(fp_out,"Choose #10 @ 6 in.(As = 2.54 sq.in./ft.)\n");
goto b362;
a372:fprintf(fp_out,"Choose #11 @ 6 in.(As = 3.12 sq.in./ft.)\n");
goto b372;
b312:reqdas = 0.62;
goto c32;
b322:reqdas = 0.88;
goto c32;
b332:reqdas = 1.20;
goto c32;
b342:reqdas = 1.58;
goto c32;
b352:reqdas = 2.00;
```

```
goto c32;
b362:reqdas = 2.54;
goto c32;
b372:reqdas = 3.12;
c32:df = sqrt(2.0 * reqdas / 3.1415926);
ld1 = (20 * reqdas * fy) / (sqrt(1000.0 * fc));
ld2 = 0.4 * df * fy;
if (ld1 > ld2)
        ldf = ld1;
else ldf = ld2;
ld = ldf * 1.4;
fprintf(fp_out,"The required development length of the bottom bars is %.2f in.\n",ld);
ldl = (ld + 5.0)/12.0;
fprintf(fp_out,"Use an embedment length of %.1f ft. \n\n\n",ldl);
goto layer50;
*

/* ________(when the water level is in the first layer)_________ */
w 2 2 : f p r i n t f ( f p _ o u t , "

if (hw <= h12)
        goto hw222;
gb1 = g1 - 62.5;
gb2 = g2 - 62.5;
h11w = h1 - hw;
hw12 = hw - h12;
x2 = ca * g1 * h11w;
x3 = ca * hw12 * gb1;
x4 = gw * hw12;
aalpha = alpha2 * 3.1415926 / 180.0;
ca = (1 - (sin(aalpha)))/(1 + (sin(aalpha)));
 fprintf(fp_out,"------------------------------------------------------------ \n");
 fprintf(fp_out,"      calculations for two layers with water-table        \n");
 fprintf(fp_out,"      when the water level is in the first layer \n");
 fprintf(fp_out,"------------------------------------------------------------ \n");
fprintf(fp_out,"Coefficient of active pressure = %.4f\n",ca);
x5 = ca * gb2 * h12;
x6 = gw * h12;
p1 = x1 * h11w;
p2 = 0.5 * x2 * h11w;
p3 = (x1 + x2) * hw12;
p4 = 0.5 * (x3 + x4) * hw12;
p5 = (x1 + x2 + x3 + x4) * h12;
p6 = 0.5 * h12 * (x5 + x6);
p = (p1 * (h12 + hw12 + h11w/2)) + (p2 * (h11w/3 + hw12 + h12)) + (p3 * (hw12/2 +
h12)) + (p4 * (hw12/3 + h12)) + (p5 * h12 * 0.5) + (p6 * h12/3);
wsub1 = (surcht * gsur) + (g1 * h11w)  + ((gb1+gw) * hw12) + ((gb2+gw) * h12);
x = pow(((2 * p) / wsub1),0.5);
goto lay21b;
lay21bb:printf("                    \n");
fprintf(fp_out,"Recalculating as F.S. is below requirement!\n");
x = x + 0.2;
lay21b:lbase = 1.5 * x;
fprintf(fp_out,"Base length of the footing = %.2f ft.\n\n",lbase);
```

```
m1 = p1 * (h11w/2 + hw12 + h12 - t);
m2 = p2 * (h11w/3 + hw12 + h12 - t);
m3 = p3 * (hw12/2 + h12 - t);
m4 = (p4 * (hw12/3 + h12 - t));
m5 = p5 * (h12/2 - t);
m6 = p6 * (h12/3 - t);
mu = (1.7 * (m1 + m2 + m3 + m4 + m5 + m6)) / 1000;
t1 = pow(((mu * 12000)/(0.9 * ru * 12.0)),0.5);
tb = t1 + 5;
if (tb < 12)
        tb = 12.0;
tt = 12.0;
fprintf(fp_out,"STEM DIMENSIONS:\n");
fprintf(fp_out,"---------------\n");
fprintf(fp_out," a) stem thickness at the bottom = %d in.\n",tb);
fprintf(fp_out," b) stem thickness at the top    = %d in.\n\n",tt);
w1 = (gsur / 1000) * surcht * (x - 1);
w2 = (g1 / 1000) * h11w * (x - 1);
w3 = (gb1 / 1000) * hw12 * (x - 1);
tbt = tb - tt;
w34 = (gb2 / 1000) * (h12 - t) * (x - 1);
w4 = (0.15 - (g1 / 1000)) * ((tbt * h11w * h11w) / (24 * (h1 - t)));
l5x = tbt * h11w / (h1 - t);
l5y = tbt * h11 / (h1 - t);
l5a = (l5x + l5y) * 0.5 * hw12;
l5c = l5x * l5x * hw12 * 0.5;
l5d = 0.5 * hw12 * (l5y - l5x) * (((l5y - l5x)/3) + l5x);
w5 = (0.15 - (gb1 / 1000)) * l5a;
w56 = (0.15 - (gb2 / 1000)) * ((l5y + tbt) * 0.5 * (h12 - t));
w6 = 0.15 * lbase * t;
w7 = 0.15 * (h1 - t);
wtot = w1 + w2 + w3 + w34 + w4 + w5 + w56 + w6 + w7;
lev1 = (x - 1) * 0.5;
lev4 = (2 * lev1) - ((tbt * h11w) / ((h1-t) * 36));
lev5 = (2 * lev1) - ((l5c + l5d) / l5a);
l56a = l5y * h12;
l56aa = l5y * 0.5;
l56b = (tbt - l5y) * h12 * 0.5;
l56bb = l5y + ((tbt - l5y)/3);
l56x = ((l56a * l56aa) + (l56b * l56bb)) / (l56a + l56b);
lev56 = (2 * lev1) - l56x;
lev6 = lbase * 0.5;
lev7 = (lev1 * 2.0) + 0.5;
mw1 = w1 * lev1;
mw2 = w2 * lev1;
mw3 = w3 * lev1;
mw34 = w34 * lev1;
mw4 = w4 * lev4;
mw5 = w5 * lev5;
mw56 = w56 * lev56;
mw6 = w6 * lev6;
mw7 = w7 * lev7;
mwt = mw1 + mw2 + mw3 + mw34 + mw4 + mw5 + mw56 + mw6 + mw7;
ywm = mwt/wtot;
```

```
resmom = wtot * (lbase - ywm);
overnom = ((p1 * (h11w/2 + hw12 + h12)) + (p2 * ((h11w/3) + hw12 + h12)) + (p3 *
(hw12/2 + h12)) + (p4 * (hw12/3 + h12)) + (p5 * h12 * 0.5) + (p6 * h12/3))/1000.0;
fsover = resmom/overnom;
fprintf(fp_out,"Overturning stability:\n");
fprintf(fp_out,"----------------------\n");
fprintf(fp_out,"Resisting moment = %.2f ft-kips\n",resmom);
fprintf(fp_out,"Overturning moment = %.2f ft-kips\n",overnom);
if (fsover < 2.0)
        goto lay21bb;
else
xbar = (mwt + overnom) / wtot;
xbar1 = lbase/3.0;
xbar2 = (2.0 * lbase) / 3.0;
xbar3 = xbar + .08;
xbar4 = (xbar - xbar2)*12.0;
fprintf(fp_out,"Factor of safety against overturning = %.2f / %.2f = %.2f --OK-- (F.S. >
2.0)\n\n\n",resmom,overnom,fsover);
if (xbar > xbar1 && xbar < xbar2)
fprintf(fp_out,"The position of the resultant soil pressure due to the service load was
checked and found to be within the middle-third of the base --- OK\n\n");
 else if (xbar > xbar1 && xbar < xbar3)
  fprintf(fp_out,"The resultant soil pressure lies %.1f in. outside the middle-third,
however,it is very close, and the limiting condition of zero stress at the heel is considered
adequate.\n\n\n",xbar4);
if (xbar > xbar1 && xbar > xbar3)
        goto lay21bb;
fprintf(fp_out,"Sliding stability:\n");
fprintf(fp_out,"--------------------\n");
fsldg = (p1 + p2 + p3 + p4 + p5 + p6) / 1000;
ffric = miu * wtot;
fssldg = ffric / fsldg;
fprintf(fp_out,"Sliding force = %.2f kips\n",fsldg);
fprintf(fp_out,"Resisting force = %.2f kips\n",ffric);
fprintf(fp_out,"Factor of safety against sliding = %.2f / %.2f =
%.2f\n",ffric,fsldg,fssldg);
if (fssldg > 1.5)
        goto stp21;
else fprintf(fp_out,"Key to be provided as factor of safety against sliding (%.2f) is less
than 1.5\n",fssldg);
ke = 8 * t;
fprintf(fp_out,"Key dimensions(CRSI Handbook):The key has a square section (%.1fin.
x %.1fin.)\n",ke,ke);
fprintf(fp_out,"        Generally it is desirable to place the front face of the key about 5
in. in front of the back face of the stem.This will permit anchoring the stem reinforcement,
if present, in the key.\n\n\n");
goto heel21;
stp21:fprintf(fp_out,"---- OK (F.S. > 1.5).\n\n\n");
heel21:fprintf(fp_out,"---------------------------------------------------------------------\n");
fprintf(fp_out,"              design of heel cantilever \n");
fprintf(fp_out, "     -------------------------------------------------------------------\n\n");
wearth1 = g1 * h11w / 1000.0;
wearth2 = gb1 * hw12 / 1000.0;
wearth3 = gb2 * (h12 - t) / 1000.0;
```

```
wsurch = gsur * surcht / 1000.0;
wfooting = 0.15 * t;
wu = (1.4 * (wearth1 + wearth2 + wearth3 + wfooting)) + (1.7 * wsurch);
lev8 = x - (tb / 12.0);
muw = 0.5 * wu * lev8 * lev8;
vuw = wu * lev8;
phivcw = 0.0204 * pow((fc * 1000.0),0.5) * tb;
if (phivcw < vuw)
        dreqd = (tb * vuw / phivcw) + 1.0;
else goto coma21;
t1n = dreqd + 2.5;
fprintf(fp_out,"Heel thickness = %d\n",t1n);
run = (muw * 12000) / (0.9 * 12.0 * dreqd * dreqd);
ron = (1.0 - pow((1.0 - (0.002 * m * run / fy)),0.5))/m;
minro = 0.2 / fy;
  if (minro > ron)
        ron = 1.333 * ron;
else ron = ron;
goto comma21;
coma21: ron = ro;
comma21:reqdas = ron * 12.0 * dreqd;
a11 = 0.62 - reqdas;
a12 = 0.88 - reqdas;
a13 = 1.20 - reqdas;
a14 = 1.58 - reqdas;
a15 = 2.00 - reqdas;
a16 = 2.54 - reqdas;
a17 = 3.12 - reqdas;
if (a11 > 0 && a11 < (a12,a13,a14,a15,a16,a17))
        goto ad1;
if (a12 > 0 && a12 < (a11,a13,a14,a15,a16,a17))
        goto ad2;
if (a13 > 0 && a13 < (a11,a12,a14,a15,a16,a17))
        goto ad3;
if (a14 > 0 && a14 < (a11,a12,a13,a15,a16,a17))
        goto ad4;
if (a15 > 0 && a15 < (a11,a12,a13,a14,a16,a17))
        goto ad5;
if (a16 > 0 && a16 < (a11,a12,a13,a14,a15,a17))
        goto ad6;
if (a17 > 0 );
        goto ad7;
ad1:fprintf(fp_out,"Choose #5 @ 6 in.(As = 0.62 sq.in./ft.)\n");
goto bd1;
ad2:fprintf(fp_out,"Choose #6 @ 6 in.(As = 0.88 sq.in./ft.)\n");
goto bd2;
ad3:fprintf(fp_out,"Choose #7 @ 6 in.(As = 1.20 sq.in./ft.)\n");
goto bd3;
ad4:fprintf(fp_out,"Choose #8 @ 6 in.(As = 1.58 sq.in./ft.)\n");
goto bd4;
ad5:fprintf(fp_out,"Choose #9 @ 6 in.(As = 2.00 sq.in./ft.)\n");
goto bd5;
ad6:fprintf(fp_out,"Choose #10 @ 6 in.(As = 2.54 sq.in./ft.)\n");
goto bd6;
```

```
ad7:fprintf(fp_out,"Choose #11 @ 6 in.(As = 3.12 sq.in./ft.)\n");
goto bd7;
bd1:reqdas = 0.62;
goto c121;
bd2:reqdas = 0.88;
goto c121;
bd3:reqdas = 1.20;
goto c121;
bd4:reqdas = 1.58;
goto c121;
bd5:reqdas = 2.00;
goto c121;
bd6:reqdas = 2.54;
goto c121;
bd7:reqdas = 3.12;
c121:df = pow((2.0 * reqdas / 3.1415926),0.5);
ld1 = (20 * reqdas * fy)/(pow((1000.0 * fc),0.5));
ld2 = 0.4 * df * fy;
if (ld1 > ld2)
        ldf = ld1;
else ldf = ld2;
ld = ldf * 1.4;
fprintf(fp_out,"The required development length of the bars (at the top) is %.2f in. into
the  toe of the footing measured from the stem reinforcement.",ld);
ldl1 = (ld + 5.0)/12.0;
fprintf(fp_out,"Use an embedment length of %.1f ft. from the backface of the
wall.\n\n\n",ldl1);
fprintf(fp_out,"--------------------------------------------------------------------\n");
fprintf(fp_out,"            design of toe cantilever                  \n");
fprintf(fp_out,"--------------------------------------------------------------------\n\n");
p11 = (2.0 * wtot)/lbase;
x4 = 0.5 * x;
p12 = x * p11 / lbase;
p13 = (lbase - t) * p11 / lbase;
vutoe = 1.7 * (((p11 + p13) * 0.5) - wfooting) * t;
mu11 = 0.5 * p11 * x4 * x4 * 0.6667;
mu12 = 0.5 * p12 * x4 * x4 * 0.3333;
mu13 = 0.5 * wfooting * x4 * x4;
muw = 1.7 * (mu11 + mu12 - mu13);
d2221:d22 = (t * 12.0) - 2.5;
phivc = 0.85 * 2.0 * sqrt(fc * 1000) * 12.0 * d22 / 1000.0;
if (phivc > vutoe)
        goto run1;
else d22 = d22 +1.0;
goto d2221;
run21:fprintf(fp_out,"PhiVc(%.2f k.) > Vu(%.2f k.) ---- OK\n",phivc,vutoe);
run = (muw * 12000) / (0.9 * 12.0 * d22 * d22);
ron = (1.0 - sqrt(1.0 - (0.002 * m * run / fy)))/m;
minro = 0.2 / fy;
if (minro > ron)
        ron = 1.333 * ron;
else ron = ron;
goto comm121;
com121:ron = ro;
```

```
comm121:reqdas = ron * 12.0 * d22;
a11 = 0.62 - reqdas;
a12 = 0.88 - reqdas;
a13 = 1.20 - reqdas;
a14 = 1.58 - reqdas;
a15 = 2.00 - reqdas;
a16 = 2.54 - reqdas;
a17 = 3.12 - reqdas;
if (a11 > 0 && a11 < (a12,a13,a14,a15,a16,a17))
        goto ae1;
if (a12 > 0 && a12 < (a11,a12,a14,a15,a16,a17))
        goto ae2;
if (a13 > 0 && a13 < (a11,a12,a14,a15,a16,a17))
        goto ae3;
if (a14 > 0 && a14 < (a11,a12,a13,a15,a16,a17))
        goto ae4;
if (a15 > 0 && a15 < (a11,a12,a13,a14,a16,a17))
        goto ae5;
if (a16 > 0 && a16 < (a11,a12,a13,a14,a15,a17))
        goto ae6;
if (a17 > 0 );
        goto ae7;
ae1:fprintf(fp_out,"Choose #5 @ 6 in.(As = 0.62 sq.in./ft.)\n");
goto be1;
ae2:fprintf(fp_out,"Choose #6 @ 6 in.(As = 0.88 sq.in./ft.)\n");
goto be2;
ae3:fprintf(fp_out,"Choose #7 @ 6 in.(As = 1.20 sq.in./ft.)\n");
goto be3;
ae4:fprintf(fp_out,"Choose #8 @ 6 in.(As = 1.58 sq.in./ft.)\n");
goto be4;
ae5:fprintf(fp_out,"Choose #9 @ 6 in.(As = 2.00 sq.in./ft.)\n");
goto be5;
ae6:fprintf(fp_out,"Choose #10 @ 6 in.(As = 2.54 sq.in./ft.)\n");
goto be6;
ae7:fprintf(fp_out,"Choose #11 @ 6 in.(As = 3.12 sq.in./ft.)\n");
goto be7;
be1:reqdas = 0.62;
goto c221;
be2:reqdas = 0.88;
goto c221;
be3:reqdas = 1.20;
goto c221;
be4:reqdas = 1.58;
goto c221;
be5:reqdas = 2.00;
goto c221;
be6:reqdas = 2.54;
goto c221;
be7:reqdas = 3.12;
c221:df = sqrt(2.0 * reqdas / 3.1415926);
ld1 = (20 * reqdas * fy) / (sqrt(1000.0 * fc));
ld2 = 0.4 * df * fy;
if (ld1 > ld2)
        ldf = ld1;
```

```
else ldf = ld2;
ld = ldf * 1.4;
fprintf(fp_out,"The required development length of the bottom bars is %.2f in.\n",ld);
ldl = (ld + 5.0)/12.0;
fprintf(fp_out,"Use an embedment length of %.1f ft. from the front face of the
wall\n\n\n",ldl);
fprintf(fp_out,"-----------------------------------------------------------------------
\n");fprintf(fp_out,"           design of reinforcement of wall\n");
fprintf(fp_out,"-----------------------------------------------------------------------\n\n");
vutoe = 1.7 * (((p11 + p13) * 0.5) - wfooting) * t;
mu11 = 0.5 * p11 * x4 * x4 * 0.6667;
mu12 = 0.5 * p12 * x4 * x4 * 0.3333;
mu13 = 0.5 * wfooting * x4 * x4;
muw = 1.7 * (mu11 + mu12 - mu13);
y = h1 - t;
ruwall = mu * 12000 / (0.9 * 12 * y * y);
ronwall = (1 - sqrt(1.0 - (0.002 * m * ruwall / fy))) / m;
if (minro > ronwall)
        ronwall = 1.333 * ronwall;
else ronwall = ronwall;
reqdas = ronwall * 12 * (h1 - t);
a11 = 0.62 - reqdas;
a12 = 0.88 - reqdas;
a13 = 1.20 - reqdas;
a14 = 1.58 - reqdas;
a15 = 2.00 - reqdas;
a16 = 2.54 - reqdas;
a17 = 3.12 - reqdas;
if (a11 > 0 && a11 < (a12,a13,a14,a15,a16,a17))
        goto af1;
if (a12 > 0 && a12 < (a11,a12,a14,a15,a16,a17))
        goto af2;
if (a13 > 0 && a13 < (a11,a12,a14,a15,a16,a17))
        goto af3;
if (a14 > 0 && a14 < (a11,a12,a13,a15,a16,a17))
        goto af4;
if (a15 > 0 && a15 < (a11,a12,a13,a14,a16,a17))
        goto af5;
if (a16 > 0 && a16 < (a11,a12,a13,a14,a15,a17))
        goto af6;
if (a17 > 0 )
        goto af7;
af1:fprintf(fp_out,"Choose #5 @ 6 in.(As = 0.62 sq.in./ft.)\n");
goto bf1;
af2:fprintf(fp_out,"Choose #6 @ 6 in.(As = 0.88 sq.in./ft.)\n");
goto bf2;
af3:fprintf(fp_out,"Choose #7 @ 6 in.(As = 1.20 sq.in./ft.)\n");
goto bf3;
af4:fprintf(fp_out,"Choose #8 @ 6 in.(As = 1.58 sq.in./ft.)\n");
goto bf4;
af5:fprintf(fp_out,"Choose #9 @ 6 in.(As = 2.00 sq.in./ft.)\n");
goto bf5;
af6:fprintf(fp_out,"Choose #10 @ 6 in.(As = 2.54 sq.in./ft.)\n");
goto bf6;
```

```
af7:fprintf(fp_out,"Choose #11 @ 6 in.(As = 3.12 sq.in./ft.)\n");
goto bf7;
bf1:reqdas = 0.62;
goto c321;
bf2:reqdas = 0.88;
goto c321;
bf3:reqdas = 1.20;
goto c321;
bf4:reqdas = 1.58;
goto c321;
bf5:reqdas = 2.00;
goto c321;
bf6:reqdas = 2.54;
goto c321;
bf7:reqdas = 3.12;
c321:df = sqrt(2.0 * reqdas / 3.1415926);
ld1 = (20 * reqdas * fy) / (sqrt(1000.0 * fc));
ld2 = 0.4 * df * fy;
if (ld1 > ld2)
        ldf = ld1;
else ldf = ld2;
ld = ldf * 1.4;
fprintf(fp_out,"The required development length of the bottom bars is %.2f in.\n",ld);
ldl = (ld + 5.0)/12.0;
fprintf(fp_out,"Use an embedment length of %.1f ft. \n\n\n",ldl);
goto layer50;


/*              calculations for two layers with water-table          */
/* ________(when the water level is in the second layer)_________ */

hw222:printf("                        \n");
fprintf(fp_out,"----------------------------------------------------------- \n");
fprintf(fp_out,"     calculations for two layers with water-table        \n");
fprintf(fp_out,"     when the water level is in the second layer \n");
fprintf(fp_out,"----------------------------------------------------------- \n");
gb2 = g2 - 62.5;
h121 = h12 - hw;
x2 = ca * g1 * h11;
aalpha = alpha2 * 3.1415926 / 180.0;
ca = (1 - (sin(aalpha)))/(1 + (sin(aalpha)));
fprintf(fp_out,"Coefficient of active pressure = %.4f\n",ca);
x3 = ca * h121 * g2;
x4 = ca * gb2 * hw;
x5 = gw * hw;
x6 = x4 + x5;
p1 = x1 * h11;
p2 = 0.5 * x2 * h11;
p3 = (x1 + x2) * h121;
p4 = 0.5 * x3 * h121;
p5 = (x1 + x2 + x3) * hw;
p6 = 0.5 * hw * x6;
p = (p1 * (h12 + h11/2)) + (p2 * (h11/3 + h12)) + (p3 * (h121/2 + hw)) + (p4 * (h121/3
+ hw)) + (p5 * hw * 0.5) + (p6 * hw/3);
```

```
wsub1 = (surcht * gsur) + (g1 * h11)  + (g2 * h121) + (gb2 * hw) + (gw * hw);
x = pow((2 * p / wsub1),0.5);
goto lay221b;
lay221bb:printf("                              \n");
fprintf(fp_out,"Recalculating as F.S. is below requirement!\n");
x = x + 0.2;
lay221b:lbase = 1.5 * x;
fprintf(fp_out,"Base length of the footing = %.2f ft.\n\n",lbase);
m1 = p1 * (h11/2 + h12 - t);
m2 = p2 * (h11/3 + h12 - t);
m3 = p3 * (h121/2 + hw - t);
m4 = (p4 * (h121/3 + hw - t));
m5 = p5 * (hw/2 - t);
m6 = p6 * (hw/3 - t);
mu = (1.7 * (m1 + m2 + m3 + m4 + m5 + m6)) / 1000;
t1 = pow(((mu * 12000)/(0.9 * ru * 12.0)),0.5);
tb = t1 + 5;
if (tb < 12)
        tb = 12.0;
tt = 12.0;
fprintf(fp_out,"STEM DIMENSIONS:\n");
fprintf(fp_out,"----------------------------\n");
fprintf(fp_out,"  a) stem thickness at the bottom = %d in.\n",tb);
fprintf(fp_out,"  b) stem thickness at the top    = %d in.\n\n",tt);
w1 = (gsur / 1000) * surcht * (x - 1);
w2 = (g1 / 1000) * h11 * (x - 1);
w3 = (g2 / 1000) * h121 * (x - 1);
tbt = tb - tt;
w34 = (gb2 / 1000) * (hw - t) * (x - 1);
xh4 = (tbt * h11 * h11)/ (24 * (h1 - t));
w4 = (0.15 - (g1 / 1000)) * xh4 * 0.5;
w5 = (0.15 - (g2/1000)) * xh4;
xh6 = tbt * (hw - t) / (h1 - t);
xh5 = tbt - xh6 - xh4;
w6 = 0.5 * h121 * xh5 * (0.15 - (g2/1000));
w7 = (tbt - xh6) * (0.15 - gb2/1000) * (hw - t);
w8 = xh6 * (0.15 - gb2/1000) * (hw - t) * 0.5;
w9 = 0.15 * (h1 - t);
w10  = 0.15 * t * lbase;
wtot = w1 + w2 + w3 + w34 + w4 + w5 + w6 + w7 + w8 + w9 + w10;
lev1 = (x - 1) * 0.5;
lev4 = (2 * lev1) - (xh4 / 3);
lev5 = (2 * lev1) - (xh4 / 2);
lev6 = (2 * lev1) - xh4 - (xh5 / 3);
lev7 = (2 * lev1) - ((xh4 + xh5) * 0.5);
lev8 = (2 * lev1) - (xh4 + xh5) - (xh6/3);
lev9 = (2 * lev1) + 0.5;
lev10 = lbase * 0.5;
mw1 = w1 * lev1;
mw2 = w2 * lev1;
mw3 = w3 * lev1;
mw34 = w34 * lev1;
mw4 = w4 * lev4;
mw5 = w5 * lev5;
```

```
mw6 = w6 * lev6;
mw7 = w7 * lev7;
mw8 = w8 * lev8;
mw9 = w9 * lev9;
mw10 = w10 * lev10;
mwt = mw1 + mw2 + mw3 + mw34 + mw4 + mw5 + mw6 + mw7 + mw8 + mw9 +
mw10;
ywm = mwt/wtot;
resmom = wtot * (lbase - ywm);
ovemom = ((p1 * (h11/2 + hw)) + (p2 * ((h11/3) + hw)) + (p3 * (h121/2 + hw)) + (p4 *
(h121/3 + hw)) + (p5 * hw * 0.5) + (p6 * hw/3))/1000.0;
fsover = resmom/ovemom;
fprintf(fp_out,"Overturning stability:\n");
fprintf(fp_out,"-------------------------\n");
fprintf(fp_out,"Resisting moment = %.2f ft-kips\n",resmom);
fprintf(fp_out,"Overturning moment = %.2f ft-kips\n",ovemom);
if (fsover < 2.0)
        goto lay221bb;
else xbar = (mwt + ovemom) / wtot;
xbar1 = lbase/3.0;
xbar2 = (2.0 * lbase) / 3.0;
xbar3 = xbar + .08;
xbar4 = (xbar - xbar2)*12.0;
fprintf(fp_out,"Factor of safety against overturning = %.2f / %.2f = %.2f --OK-- (F.S. >
2.0)\n\n\n",resmom,ovemom,fsover);
if (xbar > xbar1 && xbar < xbar2)
fprintf(fp_out,"The position of the resultant soil pressure due to the service load was
checked and found to be within the middle-third of the base --- OK\n\n");
 else if (xbar > xbar1 && xbar < xbar3)
  fprintf(fp_out,"The resultant soil pressure lies %.1f in. outside the middle-third,
however,it is very close, and the limiting condition of zero stress at the heel is considered
adequate.\n\n\n",xbar4);
if (xbar > xbar1 && xbar > xbar3)
        goto lay221bb;
fprintf(fp_out,"Sliding stability:\n");
fprintf(fp_out,"--------------------\n");
fsldg = (p1 + p2 + p3 + p4 + p5 + p6) / 1000;
ffric = miu * wtot;
fssldg = ffric / fsldg;
fprintf(fp_out,"Sliding force = %.2f kips\n",fsldg);
fprintf(fp_out,"Resisting force = %.2f kips\n",ffric);
fprintf(fp_out,"Factor of safety against sliding = %.2f / %.2f =
%.2f\n",ffric,fsldg,fssldg);
if (fssldg > 1.5)
        goto stp221;
else fprintf(fp_out,"Key to be provided as factor of safety against sliding (%.2f) is less
than 1.5\n",fssldg);
ke = 8 * t;
fprintf(fp_out,"Key dimensions(CRSI Handbook):The key has a square section (%.1fin.
x %.1fin.)\n",ke,ke);
fprintf(fp_out,"        Generally it is desirable to place the front face of the key about 5
in. in front of the back face of the stem.This will permit anchoring the stem reinforcement,
if present, in the key.\n\n\n");
goto hee2121;
```

```
stp221:printf("-----------------------------\n");
fprintf(fp_out,"---- OK (F.S. > 1.5).\n\n\n");
hee2l21:printf("----------------------------\n");
fprintf(fp_out,"---------------------------------------------------------------------\n");
fprintf(fp_out,"                    design of heel cantilever \n");
fprintf(fp_out,"---------------------------------------------------------------------\n\n");
wearth1 = g1 * h11 / 1000.0;
wearth2 = g2 * h121 / 1000.0;
wearth3 = gb2 * (hw - t) / 1000.0;
wsurch = gsur * surcht / 1000.0;
wfooting = 0.15 * t;
wu = (1.4 * (wearth1 + wearth2 + wearth3 + wfooting)) + (1.7 * wsurch);
lev8 = x - (tb / 12.0);
muw = 0.5 * wu * lev8 * lev8;
vuw = wu * lev8;
phivcw = 0.0204 * sqrt(fc * 1000.0) * tb;
if (phivcw < vuw)
        dreqd = (tb * vuw / phivcw) + 1.0;
else goto coma221;
t1n = dreqd + 2.5;
fprintf(fp_out,"Heel thickness = %d\n",t1n);
run = (muw * 12000) / (0.9 * 12.0 * dreqd * dreqd);
ron = (1.0 - sqrt(1.0 - (0.002 * m * run / fy)))/m;
minro = 0.2 / fy;
  if (minro > ron)
        ron = 1.333 * ron;
else ron = ron;
goto comma221;
coma221: ron = ro;
comma221:reqdas = ron * 12.0 * dreqd;
a11 = 0.62 - reqdas;
a12 = 0.88 - reqdas;
a13 = 1.20 - reqdas;
a14 = 1.58 - reqdas;
a15 = 2.00 - reqdas;
a16 = 2.54 - reqdas;
a17 = 3.12 - reqdas;
if (a11 > 0 && a11 < (a12,a13,a14,a15,a16,a17))
        goto ag1;
if (a12 > 0 && a12 < (a11,a13,a14,a15,a16,a17))
        goto ag2;
if (a13 > 0 && a13 < (a11,a12,a14,a15,a16,a17))
        goto ag3;
if (a14 > 0 && a14 < (a11,a12,a13,a15,a16,a17))
        goto ag4;
if (a15 > 0 && a15 < (a11,a12,a13,a14,a16,a17))
        goto ag5;
if (a16 > 0 && a16 < (a11,a12,a13,a14,a15,a17))
        goto ag6;
if (a17 > 0 );
        goto ag7;
ag1:fprintf(fp_out,"Choose #5 @ 6 in.(As = 0.62 sq.in./ft.)\n");
goto bg1;
ag2:fprintf(fp_out,"Choose #6 @ 6 in.(As = 0.88 sq.in./ft.)\n");
```

```
goto bg2;
ag3:fprintf(fp_out,"Choose #7 @ 6 in.(As = 1.20 sq.in./ft.)\n");
goto bg3;
ag4:fprintf(fp_out,"Choose #8 @ 6 in.(As = 1.58 sq.in./ft.)\n");
goto bg4;
ag5:fprintf(fp_out,"Choose #9 @ 6 in.(As = 2.00 sq.in./ft.)\n");
goto bg5;
ag6:fprintf(fp_out,"Choose #10 @ 6 in.(As = 2.54 sq.in./ft.)\n");
goto bg6;
ag7:fprintf(fp_out,"Choose #11 @ 6 in.(As = 3.12 sq.in./ft.)\n");
goto bg7;
bg1:reqdas = 0.62;
goto c2221;
bg2:reqdas = 0.88;
goto c2221;
bg3:reqdas = 1.20;
goto c2221;
bg4:reqdas = 1.58;
goto c2221;
bg5:reqdas = 2.00;
goto c2221;
bg6:reqdas = 2.54;
goto c2221;
bg7:reqdas = 3.12;
c2221:df = sqrt(2.0 * reqdas / 3.1415926);
ld1 = (20 * reqdas * fy)/(sqrt(1000.0 * fc));
ld2 = 0.4 * df * fy;
if (ld1 > ld2)
        ldf = ld1;
else ldf = ld2;
ld = ldf * 1.4;
fprintf(fp_out,"The required development length of the bars (at the top) is %.2f in. into
the  toe of the footing measured from the stem reinforcement.",ld);
ldl1 = (ld + 5.0)/12.0;
fprintf(fp_out,"Use an embedment length of %.1f ft. from the backface of the
wall.\n\n\n",ldl1);
fprintf(fp_out,"----------------------------------------------------------------------\n");
fprintf(fp_out,"            design of toe cantilever              \n");
fprintf(fp_out,"----------------------------------------------------------------------\n\n");
p11 = (2.0 * wtot)/lbase;
x4 = 0.5 * x;
p12 = x * p11 / lbase;
p13 = (lbase - t) * p11 / lbase;
vutoe = 1.7 * (((p11 + p13) * 0.5) - wfooting) * t;
mu11 = 0.5 * p11 * x4 * x4 * 0.6667;
mu12 = 0.5 * p12 * x4 * x4 * 0.3333;
mu13 = 0.5 * wfooting * x4 * x4;
muw = 1.7 * (mu11 + mu12 - mu13);
d2222:d22 = (t * 12.0) - 2.5;
phivc = 0.85 * 2.0 * sqrt(fc * 1000) * 12.0 * d22 / 1000.0;
if (phivc > vutoe)
        goto run2;
else d22 = d22 +1.0;
goto d2222;
```

```
run22:fprintf(fp_out,"PhiVc(%.2f k.) > Vu(%.2f k.) ---- OK\n",phivc,vutoe);
run = (muw * 12000) / (0.9 * 12.0 * d22 * d22);
ron = (1.0 - sqrt(1.0 - (0.002 * m * run / fy)))/m;
minro = 0.2 / fy;
if (minro > ron)
        ron = 1.333 * ron;
else ron = ron;
goto comm122;
com122:ron = ro;
comm122:reqdas = ron * 12.0 * d22;
a11 = 0.62 - reqdas;
a12 = 0.88 - reqdas;
a13 = 1.20 - reqdas;
a14 = 1.58 - reqdas;
a15 = 2.00 - reqdas;
a16 = 2.54 - reqdas;
a17 = 3.12 - reqdas;
if (a11 > 0 && a11 < (a12,a13,a14,a15,a16,a17))
        goto ah1;
if (a12 > 0 && a12 < (a11,a12,a14,a15,a16,a17))
        goto ah2;
if (a13 > 0 && a13 < (a11,a12,a14,a15,a16,a17))
        goto ah3;
if (a14 > 0 && a14 < (a11,a12,a13,a15,a16,a17))
        goto ah4;
if (a15 > 0 && a15 < (a11,a12,a13,a14,a16,a17))
        goto ah5;
if (a16 > 0 && a16 < (a11,a12,a13,a14,a15,a17))
        goto ah6;
if (a17 > 0 );
        goto ah7;
ah1:fprintf(fp_out,"Choose #5 @ 6 in.(As = 0.62 sq.in./ft.)\n");
goto bh1;
ah2:fprintf(fp_out,"Choose #6 @ 6 in.(As = 0.88 sq.in./ft.)\n");
goto bh2;
ah3:fprintf(fp_out,"Choose #7 @ 6 in.(As = 1.20 sq.in./ft.)\n");
goto bh3;
ah4:fprintf(fp_out,"Choose #8 @ 6 in.(As = 1.58 sq.in./ft.)\n");
goto bh4;
ah5:fprintf(fp_out,"Choose #9 @ 6 in.(As = 2.00 sq.in./ft.)\n");
goto bh5;
ah6:fprintf(fp_out,"Choose #10 @ 6 in.(As = 2.54 sq.in./ft.)\n");
goto bh6;
ah7:fprintf(fp_out,"Choose #11 @ 6 in.(As = 3.12 sq.in./ft.)\n");
goto bh7;
bh1:reqdas = 0.62;
goto c222;
bh2:reqdas = 0.88;
goto c222;
bh3:reqdas = 1.20;
goto c222;
bh4:reqdas = 1.58;
goto c222;
bh5:reqdas = 2.00;
```

```
goto c222;
bh6:reqdas = 2.54;
goto c222;
bh7:reqdas = 3.12;
c222:df = sqrt(2.0 * reqdas / 3.1415926);
ld1 = (20 * reqdas * fy) / (sqrt(1000.0 * fc));
ld2 = 0.4 * df * fy;
if (ld1 > ld2)
        ldf = ld1;
else ldf = ld2;
ld = ldf * 1.4;
fprintf(fp_out,"The required development length of the bottom bars is %.2f in.\n",ld);
ldl = (ld + 5.0)/12.0;
fprintf(fp_out,"Use an embedment length of %.1f ft. from the front face of the
wall\n\n\n",ldl);
fprintf(fp_out,"------------------------------------------------------------------------\n");
fprintf(fp_out,"              design of reinforcement of wall              \n");
fprintf(fp_out,"------------------------------------------------------------------------\n\n");
vutoe = 1.7 * (((p11 + p13) * 0.5) - wfooting) * t;
mu11 = 0.5 * p11 * x4 * x4 * 0.6667;
mu12 = 0.5 * p12 * x4 * x4 * 0.3333;
mu13 = 0.5 * wfooting * x4 * x4;
muw = 1.7 * (mu11 + mu12 - mu13);
y = h1 - t;
ruwall = mu * 12000 / (0.9 * 12 * y * y);
ronwall = (1 - sqrt(1.0 - (0.002 * m * ruwall / fy))) / m;
if (minro > ronwall)
        ronwall = 1.333 * ronwall;
else ronwall = ronwall;
reqdas = ronwall * 12 * (h1 - t);
a11 = 0.62 - reqdas;
a12 = 0.88 - reqdas;
a13 = 1.20 - reqdas;
a14 = 1.58 - reqdas;
a15 = 2.00 - reqdas;
a16 = 2.54 - reqdas;
a17 = 3.12 - reqdas;
if (a11 > 0 && a11 < (a12,a13,a14,a15,a16,a17))
        goto ai1;
if (a12 > 0 && a12 < (a11,a12,a14,a15,a16,a17))
        goto ai2;
if (a13 > 0 && a13 < (a11,a12,a14,a15,a16,a17))
        goto ai3;
if (a14 > 0 && a14 < (a11,a12,a13,a15,a16,a17))
        goto ai4;
if (a15 > 0 && a15 < (a11,a12,a13,a14,a16,a17))
        goto ai5;
if (a16 > 0 && a16 < (a11,a12,a13,a14,a15,a17))
        goto ai6;
if (a17 > 0 )
        goto ai7;
ai1:fprintf(fp_out,"Choose #5 @ 6 in.(As = 0.62 sq.in./ft.)\n");
goto bi1;
ai2:fprintf(fp_out,"Choose #6 @ 6 in.(As = 0.88 sq.in./ft.)\n");
```

```
goto bi2;
ai3:fprintf(fp_out,"Choose #7 @ 6 in.(As = 1.20 sq.in./ft.)\n");
goto bi3;
ai4:fprintf(fp_out,"Choose #8 @ 6 in.(As = 1.58 sq.in./ft.)\n");
goto bi4;
ai5:fprintf(fp_out,"Choose #9 @ 6 in.(As = 2.00 sq.in./ft.)\n");
goto bi5;
ai6:fprintf(fp_out,"Choose #10 @ 6 in.(As = 2.54 sq.in./ft.)\n");
goto bi6;
ai7:fprintf(fp_out,"Choose #11 @ 6 in.(As = 3.12 sq.in./ft.)\n");
goto bi7;
bi1:reqdas = 0.62;
goto c322;
bi2:reqdas = 0.88;
goto c322;
bi3:reqdas = 1.20;
goto c322;
bi4:reqdas = 1.58;
goto c322;
bi5:reqdas = 2.00;
goto c322;
bi6:reqdas = 2.54;
goto c322;
bi7:reqdas = 3.12;
c322:df = sqrt(2.0 * reqdas / 3.1415926);
ld1 = (20 * reqdas * fy) / (sqrt(1000.0 * fc));
ld2 = 0.4 * df * fy;
if (ld1 > ld2)
        ldf = ld1;
else ldf = ld2;
ld = ldf * 1.4;
fprintf(fp_out,"The required development length of the bottom bars is %.2f in.\n",ld);
ldl = (ld + 5.0)/12.0;
fprintf(fp_out,"Use an embedment length of %.1f ft. \n\n\n",ldl);
layer50:printf("                    \n");
fprintf(fp_out,"------------------------------------------------------\n");
fprintf(fp_out,"   temperature and shrinkage reinforcement  \n");
fprintf(fp_out,"------------------------------------------------------\n");
ast = 0.0025 * 6 * (tb + tt);
asfront = 0.6667 * ast;
asback = 0.3333 * ast;
fprintf(fp_out,"Amount of horizontal temperature and shrinkage steel required on the front
face = %.2f\n",asfront);
fprintf(fp_out,"Amount of horizontal temperature and shrinkage steel required on the back
face = %.2f\n",asback);
aa1 = 0.11 - asfront;
a01 = 0.20 - asfront;
a11 = 0.31 - asfront;
a12 = 0.44 - asfront;
a13 = 0.60 - asfront;
a14 = 0.79 - asfront;
a15 = 1.00 - asfront;
a16 = 1.27 - asfront;
a17 = 1.56 - asfront;
```

```
fprintf(fp_out,"For the face side of the wall:\n");
fprintf(fp_out,"----------------------------------\n");
if (aa1 > 0 && aa1 < (a01,a11,a12,a13,a14,a15,a16,a17))
        goto ab1;
if (a01 > 0 && a01 < (aa1,a11,a12,a13,a14,a15,a16,a17))
        goto ab2;
if (a11 > 0 && a11 < (aa1,a01,a12,a13,a14,a15,a16,a17))
        goto ab3;
if (a12 > 0 && a12 < (aa1,a01,a11,a12,a14,a15,a16,a17))
        goto ab4;
if (a13 > 0 && a13 < (aa1,a01,a11,a12,a14,a15,a16,a17))
        goto ab5;
if (a14 > 0 && a14 < (aa1,a01,a11,a12,a13,a15,a16,a17))
        goto ab6;
if (a15 > 0 && a15 < (aa1,a01,a11,a12,a13,a14,a16,a17))
        goto ab7;
if (a16 > 0 && a16 < (aa1,a01,a11,a12,a13,a14,a15,a17))
        goto ab8;
if (a17 > 0 )
        goto ab9;
ab1:fprintf(fp_out,"Choose #3 @ 12 in.(As = 0.11 sq.in./ft.)\n");
goto c13;
ab2:fprintf(fp_out,"Choose #4 @ 12 in.(As = 0.20 sq.in./ft.)\n");
goto c13;
ab3:fprintf(fp_out,"Choose #5 @ 12 in.(As = 0.31 sq.in./ft.)\n");
goto c13;
ab4:fprintf(fp_out,"Choose #6 @ 12 in.(As = 0.44 sq.in./ft.)\n");
goto c13;
ab5:fprintf(fp_out,"Choose #7 @ 12 in.(As = 0.60 sq.in./ft.)\n");
goto c13;
ab6:fprintf(fp_out,"Choose #8 @ 12 in.(As = 0.79 sq.in./ft.)\n");
goto c13;
ab7:fprintf(fp_out,"Choose #9 @ 12 in.(As = 1.00 sq.in./ft.)\n");
goto c13;
ab8:fprintf(fp_out,"Choose #10 @ 12 in.(As = 1.27 sq.in./ft.)\n");
goto c13;
ab9:fprintf(fp_out,"Choose #11 @ 12 in.(As = 1.56 sq.in./ft.)\n");
c13:aa1 = 0.11 - asback;
a01 = 0.20 - asback;
a11 = 0.31 - asback;
a12 = 0.44 - asback;
a13 = 0.60 - asback;
a14 = 0.79 - asback;
a15 = 1.00 - asback;
a16 = 1.27 - asback;
a17 = 1.56 - asback;
fprintf(fp_out,"\nFor the back side of the wall:\n");
fprintf(fp_out,"------------------------------------\n");
if (aa1 > 0 && aa1 < (a01,a11,a12,a13,a14,a15,a16,a17))
        goto ac1;
if (a01 > 0 && a01 < (aa1,a11,a12,a13,a14,a15,a16,a17))
        goto ac2;
if (a11 > 0 && a11 < (aa1,a01,a12,a13,a14,a15,a16,a17))
        goto ac3;
```

```
if (a12 > 0 && a12 < (aa1,a01,a11,a12,a14,a15,a16,a17))
        goto ac4;
if (a13 > 0 && a13 < (aa1,a01,a11,a12,a14,a15,a16,a17))
        goto ac5;
if (a14 > 0 && a14 < (aa1,a01,a11,a12,a13,a15,a16,a17))
        goto ac6;
if (a15 > 0 && a15 < (aa1,a01,a11,a12,a13,a14,a16,a17))
        goto ac7;
if (a16 > 0 && a16 < (aa1,a01,a11,a12,a13,a14,a15,a17))
        goto ac8;
if (a17 > 0 )
        goto ac9;
ac1:fprintf(fp_out,"Choose #3 @ 12 in.(As = 0.11 sq.in./ft.)\n");
goto c113;
ac2:fprintf(fp_out,"Choose #4 @ 12 in.(As = 0.20 sq.in./ft.)\n");
goto c113;
ac3:fprintf(fp_out,"Choose #5 @ 12 in.(As = 0.31 sq.in./ft.)\n");
goto c113;
ac4:fprintf(fp_out,"Choose #6 @ 12 in.(As = 0.44 sq.in./ft.)\n");
goto c113;
ac5:fprintf(fp_out,"Choose #7 @ 12 in.(As = 0.60 sq.in./ft.)\n");
goto c113;
ac6:fprintf(fp_out,"Choose #8 @ 12 in.(As = 0.79 sq.in./ft.)\n");
goto c113;
ac7:fprintf(fp_out,"Choose #9 @ 12 in.(As = 1.00 sq.in./ft.)\n");
goto c113;
ac8:fprintf(fp_out,"Choose #10 @ 12 in.(As = 1.27 sq.in./ft.)\n");
goto c113;
ac9:fprintf(fp_out,"Choose #11 @ 12 in.(As = 1.56 sq.in./ft.)\n");
c113:fprintf(fp_out,"\n-----------\n");
fprintf(fp_out,"Drainage:\n");
fprintf(fp_out,"---------\n");
fprintf(fp_out,"        Adequate drainage of backfill must be provided to prevent water
from accumulating in the backfill material; a common minimum provision is to provide
weep holes (4 in. diameter) every 10 to 15 ft. along the wall.\n");
fprintf(fp_out,"------------------------------------------------------------------------\n");
printf("\n\n *************\n");
printf(" I N P U T\n");
printf(" *************\n\n");
printf("Height of earth to be supported = %.2f ft.\n",h);
printf("Angle which the backfill makes with the horizontal = %.2f deg.\n",deltab);
printf("Height of the surcharge = %.2f ft.\n",surcht);
printf("Density of the surcharge material = %.2f pcf.\n",gsur);
printf("Angle of internal friction of the surcharge material = %.2f deg.\n",alphas);
printf("Coefficient of friction between masonry and soil = %.2f\n",miu);
printf("Compressive strength of concrete,fc = %.2f ksi.\n",fc);
printf("Yield strength of steel,fy = %.2f ksi.\n",fy);
printf("Maximum soil bearing pressure = %.2f ksf.\n",maxspr);
printf("Height of water table = %.2f ft.\n",hw);
printf("Layers of soil below the earth's surface = %.1f\n",layer);
if (layer == 1)
        goto calc;
printf("Depth of the top soil layer = %.2f ft.\n",h11);
printf("Angle of internal friction of soil in the top layer = %.2f deg.\n",alpha1);
```

```
printf("Density of soil in the top layer = %.2f pcf.\n",g1);
printf("Angle of internal friction of soil in the lower layer = %.2f deg.\n",alphas);
printf("Density of soil in the lower layer = %.2f pcf.\n",g2);
calc:printf("\n\n\n*************************************************************
*************\n");
printf("THE DETAILED DESIGN OF THE  CANTILEVER WALL IS IN FILE >>  %s
\n",fn_out);
printf("**************************************************************
******\n");
endd:fclose (fp_out);
return(0);
}
float fromax(fc,fy)
float fc,fy;
{
int i,j;
float ttt;
if (fy == 40.0)
        j = 0;
if (fy == 50.0)
        j = 1;
if (fy == 60.0)
        j = 2;
if (fc == 3.0)
        i = 0;
if (fc == 3.5)
        i = 1;
if (fc == 4.0)
        i = 2;
if (fc == 5.0)
        i = 3;
if (fc == 6.0)
        i = 4;
ttt = table[i][j];
return(ttt);
}
```

## APPENDIX E

## LISTING OF PROGRAM GABION

```
/* ---------------------------------------------------------------------------------- */
/* |   DESIGN OF THE GABION RETAINING STRUCTURE        | */
/* ---------------------------------------------------------------------------------- */

#include <stdio.h>
#include <math.h>

#define LINE 50
#define ROW 50

float a[ROW][LINE];

main()
{
FILE *fopen(),*fp_out;
char fn_out[12];

float h,aepsln1,aalpha1,angphi1,angbet1,unifpo,gamag,gamat,cofact,extn;
float aepsln,aalpha,angphi,angbet,aomega,angdel;
float tota,totamx,totamy,xq1,yq1,we,wax,am,xlcgm,ylcgm;
float xm,ym,aa,aga,asoil,at,agamx,agamy,xt1,yt1,xt,yt,xq,yq,xn,yn,xd,yd;
float yf1,xf,yf,xa1,ya1,xa,ya,aqep,mqep,apnd,mpnd,wm,mm,wt,mt,wc,mc,wf,ha;
float xe,aomega1,c1,c2,c3,ka,sa,sh,sv,ma,d1,d2,d3,d31,kp;
float sp1,sph1,spv1,mp1,sp2,sph2,spv2,fi,fs2,fs3,fs,etas;
float mi,ms,etar,n,ec,z1,z2,t,ec111,ec112,ec11,ec22,nq,nc,ng,iq,ig,dq,dc,dg;
float plim,zlim,zmax,qn,xb,yb,ab,etai1,a11,a22,w1,w11,ca1,w2,w3,bc;
float w4,wi,s,yct,rp,etai,minetai,minaomega;
int ngb,ngb1,i;

extern double sin();
extern double cos();
extern double sqrt();

printf("Enter name of output file:");
scanf("%s",fn_out,"w");
fp_out = fopen(fn_out,"w");
printf("Output in file:%s\n\n\n",fn_out);

fprintf(fp_out,"----------------------------------------------------------------------------------\n");
fprintf(fp_out,"|   DESIGN OF THE GABION RETAINING STRUCTURE    |\n");
fprintf(fp_out,"----------------------------------------------------------------------------------\n");

printf("----------------------------------------------------------------------------------\n");
printf("|  DESIGN OF THE GABION RETAINING STRUCTURE   |\n");
printf("----------------------------------------------------------------------------------\n\n\n");

/* -----------------------input----------------------- */

printf("What is the height (in m.) of the earth to be supported?\n>>");
scanf("%f",&h);
printf("What is the slope of the backfill?\n>>");
scanf("%f",&aepsln1);
```

```
printf("What will be the angle (in deg.) of batter of the structure (if not sure give  the
answer as 6.0)?\n>>");
scanf("%f",&aalphal);
printf("What is the angle (in deg.) of internal friction of the soil to be retained?\n>>");
scanf("%f",&angphil);
printf("What is the uniform surcharge load (t/sq.m.) that the structure has to
support?\n>>");
scanf("%f",&unifpo);
printf("What is the density of the gabion (t/cu.m.)?\n>>");
scanf("%f",&gamag);
printf("What is the density of the backfill and foundation soil(t/cu.m.)?\n>>");
scanf("%f",&gamat);
printf("What is the cohesion factor (t/sq.m.) of the earth for resistance\n against
sliding?\n>>");
scanf("%f",&cofact);
printf("\n\n *********************\n");
printf(" *   I N P U T   *\n");
printf(" *********************\n");
printf("Height of the earth to be supported = %.2f m.\n",h);
printf("Slope of the backfill = %.2f deg.\n",aepsln1);
printf("Angle of batter of the structure = %.2f deg.\n",aalphal);
printf("Angle of internal friction of the soil to be retained = %.2f deg.\n",angphil);
printf("Uniform surcharge load that the structure has to support = %.2f
t/sq.m.\n",unifpo);
printf("Density of the fill of gabion = %.2f t/cu.m.\n",gamag);
printf("Density of the backfill and foundation soil = %.2f t/cu.m.\n",gamat);
printf("Cohesion factor of the earth for resistance against sliding = %.2f
t/sq.m.\n\n\n",cofact);
printf(" C A L C U L A T I N G . . . . . . . . ......\n\n");
extn = 0.5 ;
angbet1 = 90.0;
zer:tota = 0;
totamx = 0;
totamy = 0;
aepsln = (3.1415926 / 180.0) * aepsln1;
aalpha = (3.1415926 / 180.0) * aalphal;
angphi = (3.1415926 / 180.0) * angphil;
angbet = (3.1415926 / 180.0) * angbetl;
/* ------------------------------------------------ */
/*       width of gabion structure             */
/* ------------------------------------------------ */
ngb = h;
if ( (h - ngb) > 0 )
       ngb = ngb + 1;
else ngb = h;
fprintf(fp_out,"\n\nNumber of gabion layers = %d.\n\n",ngb);
fprintf(fp_out,"---------------------------------\n");
fprintf(fp_out,"  GABION #      WIDTH (m.)  \n");
fprintf(fp_out,"---------------------------------\n");
ngbl = ngb - 1;
for ( i=1 ; i<=ngbl ; i=i+2 )
{
if (i != ngbl)
    goto one;
```

```
else goto two;
one:a[i][1] = extn + (i + 1)/4.0;
fprintf(fp_out,"     %d              %.2f\n",i,a[i][1]);
if (i != 1)
        goto twoo;
if (i == 1)
        xq1 = (2 * extn) + a[i][1];
if (i == 1)
        yq1 = h - 1;
twoo:a[i][2] = 1.0;
a[i+1][2] = 1.0;
a[i][3] = a[i][1] * a[i][2];
a[i][4] = a[i][1]/2.0 + 2*extn;
if (i != ngb1)
        goto thr;
two:a[i][1] = extn + (i+1)/4.0;
fprintf(fp_out,"     %d              %.2f\n",i,a[i][1]);
a[i+1][1] = a[i][1] + 4*extn;
fprintf(fp_out,"     %d              %.2f \n",i+1,a[i+1][1]);
a[i][3] = a[i][1];
a[i][4] = (a[i][1] * 0.5) + (2 * extn);
a[i+1][3] = a[i+1][1];
a[i+1][4] = a[i+1][1] / 2.0;
a[i][5] = 0.5;
a[i+1][5] = -0.5;
goto fou;
thr:a[i+1][1] = a[i][1];
fprintf(fp_out,"     %d              %.2f\n",i+1,a[i+1][1]);
a[i+1][3] = a[i][3];
a[i+1][4] = a[i][4];
a[i][5] = yq1 - (i - 0.5);
a[i+1][5] = a[i][5] -1.0;
fou:a[i][6] = a[i][3] * a[i][4];
a[i][7] = a[i][3] * a[i][5];
a[i+1][6] = a[i+1][3] * a[i+1][4];
a[i+1][7] = a[i+1][3] * a[i+1][5];
tota = tota + a[i][3] + a[i+1][3];
totamx = totamx + a[i][6] + a[i+1][6];
totamy = totamy + a[i][7] + a[i+1][7];
if ((i != (ngb-3)) && (i != (ngb-2)))
goto fie;
if (i == (ngb-3) )
        goto fiv;
if (i == (ngb-2) )
        goto six;
fie:we = a[i+1][1];
wax = a[i][1] + 2*extn;
goto fiv;
six:a[i+2][1] = a[i+1][1] + 4*extn;
fprintf(fp_out,"     %d              %.2f\n",i+2,a[i+2][1]);
wax = a[i+1][1] + 2*extn;
we = a[i+2][1];
tota = tota + a[i+2][1];
totamx = totamx + (we * we * 0.5);
```

```
totamy = totamy - (we * 0.5);
if ( i == ngb - 2 )
        goto sev;
fiv:if (i == ngb1)
        goto sev;
}
sev:am = tota;
fprintf(fp_out,"-----------------------------------\n\n");
xlcgm = totamx/tota;
ylcgm = totamy/tota;
fprintf(fp_out,"The position of the C.G. from the leftside of the bottom gabion = %.2f
m.\n",xlcgm);
fprintf(fp_out,"The position of the C.G. from the topside of the bottom gabion = %.2f
m.\n",ylcgm);
fprintf(fp_out,"The height of each gabion = 1.0 m.\n\n");
angdel = angphi;
xm = xlcgm*cos(aalpha) + ylcgm*sin(aalpha);
ym = -xlcgm*sin(aalpha) + ylcgm*cos(aalpha);
aa = (we - 2*extn) * (h-1);
aga = tota - we;
asoil = aa - aga;
at = asoil;
agamx = ((tota*xlcgm)-(we*we*0.5))/aga;
agamy = ((tota*ylcgm)+(we*0.5))/aga;
xt1 = ((aa*(((we-2*extn)*0.5)+2*extn))-(aga*agamx))/asoil;
yt1 = ((aa*(h-1)*0.5)-(aga*agamy))/asoil;
xt = xt1*cos(aalpha)+yt1*sin(aalpha);
yt = -xt1*sin(aalpha)+yt1*cos(aalpha);
xq = xq1*cos(aalpha)+yq1*sin(aalpha);
yq = -xq1*sin(aalpha)+yq1*cos(aalpha);
/* xn1 = we & yn1 = yq1*/
xn = we*cos(aalpha)+yq1*sin(aalpha);
yn = -we*sin(aalpha)+yq1*cos(aalpha);
 /*xd1 = we & yd1 = 0    */
xd = we*cos(aalpha);
yd = -we*sin(aalpha);
xe = xd;
yf1 = -1.0;
xf = yf1*sin(aalpha);
yf = yf1*cos(aalpha);
xa1 = we;
ya1 = -1.0;
xa = xa1*cos(aalpha)+ya1*sin(aalpha);
ya = -xa1*sin(aalpha)+ya1*cos(aalpha);
aqep = ((xe-xq)*(xe-xq))*(tan(aalpha)+tan(aepsln))/2.0;
mqep = aqep*(xe-xf-((xe-xq)/3));
apnd = yq1*yq1*tan(aalpha)/2.0;
mpnd = apnd*(xe-xf+((xn-xe)/3.0));
wm = am*gamag;
mm = wm*(xm-xf);
wt = (at+aqep-apnd)*gamat;
mt = (at*(xt-xf)+mqep-mpnd)*gamat;
wc = unifpo*(xe-xq);
mc = wc*((xq-xf)+((xe-xq)/2));
```

```
wf = ((xa-xf)*(xa-xf))*gamat*tan(aalpha/2);
ha = yq+(xe-xq)*tan(aepsln)-ya;
aomega1 = angphi1+(90-angbet1);
aomega = (3.1415926/180)*aomega1;
c1 = sin(angbet-angdel)*sin(angbet+aepsln);
c2 = sin(angphi+angdel)*sin(angphi-aepsln);
c3 = (sin(angbet)*sin(angbet))*(sin(angbet-
angdel))*((1+(sqrt(c2/c1)))*(1+(sqrt(c2/c1))));
ka = (sin(angbet+angphi))*(sin(angbet+angphi))/c3;
sa = (gamat*ha*ha*ka*0.5) + (unifpo*ha*ka);
fprintf(fp_out,"Active earth pressure = %.2f t/m.\n\n",sa);
sh = sa*cos(aomega);
sv = sa*sin(aomega);
ma = sh*((ha*(ha+3*unifpo/gamat))/(3*(ha+2*unifpo/gamat))-(yf-ya))-sv*(xd-xf);
d1 = sin(angphi+angdel)*sin(angphi+aepsln);
d2 = sin(angbet+angdel)*sin(angbet+aepsln);
d31 = 1-(sqrt(d1/d2));
d3 = (sin(angbet)*sin(angbet))*(sin(angbet+angdel))*d31*d31;
kp = (sin(angbet-angphi))*(sin(angbet-angphi))/d3;
fprintf(fp_out,"Coefficient of active pressure,Ka = %.4f\n",ka);
fprintf(fp_out,"Coefficient of passive pressure,Kp = %.4f\n\n",kp);
sp1 = gamat*yf*yf*kp*0.5;
sph1 = sp1*cos(angphi);
spv1 = sp1*sin(angphi);
mp1 = -sph1*yf/3.0;
sp2 = gamat*ya*ya*kp*0.5;
sph2 = sp2*cos(angphi);
spv2 = sp2*sin(angphi);
fi = sh;
fprintf(fp_out,"  --------------------------------  \n");
fprintf(fp_out,"  factor of safety against sliding  \n");
fprintf(fp_out,"  --------------------------------  \n\n");
fprintf(fp_out,"The sum of the forces causing sliding = %.2f t/m.\n",fi);
fs2 = (wm+wt+wf+wc+sv-spv2)*tan(angphi);
fs3 = cofact*we*cos(aalpha);
fs = (sph2 + fs3 + fs2);
fprintf(fp_out,"The sum of the forces resisting sliding = %.2f t/m.\n\n",fs);
etas = fs/fi;
if (etas > 1.5)
        goto ten;
else extn = extn + 0.5;
fprintf(fp_out,"Recalculating ( as FS < 1.5 ) with base length = %.2f m.\n\n",extn);
goto zer;
ten:fprintf(fp_out,"The factor of safety against sliding = %.2f ----- OK, since FS >
1.5\n\n\n",etas);
mi = ma;
ms = mm+mt+mc+mp1;
fprintf(fp_out,"  ------------------------------------  \n");
fprintf(fp_out,"  factor of safety against overturning  \n");
fprintf(fp_out,"  ------------------------------------  \n\n");
fprintf(fp_out,"Sum of overturning moment = %.2f t-m./m.\n",mi);
fprintf(fp_out,"Sum of restoring moment  = %.2f t-m./m.\n\n",ms);
/*    ------------------------------------    */
/*    factor of safety against overturning    */
```

```
/*     ------------------------------------     */
if (mi < 0)
        goto ele;
else etar = ms/mi;
goto eln;
ele:fprintf(fp_out,"Factor of safety against overturning is NOT calculated as there is
nooverturning moment.\n\n");
goto elv1;
eln:if (etar > 1.5)
        goto elv;
else extn = extn + 0.5;
fprintf(fp_out,"Recalculating ( as FS < 1.5 ) with base length = %.2f m.\n\n",extn);
goto zer;
/*      ------------------------------------     */
/*      check on foundation bearing pressure     */
/*      ------------------------------------     */
/* n = normal force acting on the base section */
/* ec = eccentricity of the resultant */
/* z1 & z2 are the principal stresses at the toe and heel respectively */
elv:fprintf(fp_out,"Factor of safety against overturning = %.2f --- O.K., since FS >
1.5\n\n\n",etar);
elv1:n = (wm+wc+wt)*cos(aalpha)+sa*sin(aomega+aalpha)-sp1*sin(angdel+aalpha);
ec = (we/2.0)-(ms-mi)/n;
z1 = (1+6*ec/we)*n/we;
z2 = (1-6*ec/we)*n/we;
fprintf(fp_out,"------------------------------------ \n");
fprintf(fp_out,"check on foundation bearing pressure \n");
fprintf(fp_out,"------------------------------------ \n\n");
/* t = shear force acting on base section */
/* plim = limiting pressure */
t = -(wm+wt+wc)*sin(aalpha)+sa*cos(aalpha+aomega)-sp1*cos(angdel+aalpha);
ec111 = 2.7183;
ec112 = 3.14159 * tan(angphi);
ec11 = pow(ec111,ec112);
ec22 = tan(0.7854+angphi/2)*tan(0.7854+angphi/2);
nq = ec11*ec22;
nc = (nq-1)/tan(angphi);
ng = 1.8*(nq-1)*tan(angphi);
iq = 1-0.5*t/n;
ig = iq*iq;
dq = 1-0.35*yf/we;
dc = dq;
dg = 1;
plim = cofact*nc*dc-yf*gamat*nq*dq*iq+0.5*gamat*we*ng*dg*ig;
zlim = plim/3.0;
/* zlim = allowable soil bearing stress */
/* zmax = maximum soil pressure developed */
zmax = z2;
if (zlim > zmax)
        goto twe;
else extn = extn + 0.5;
fprintf(fp_out,"Recalculating with base length = %.2f m. as maximum soil pressure is
greater than the allowable soil bearing pressure.\n\n",extn);
goto zer;
```

```
twe:fprintf(fp_out,"Allowable soil bearing stress = %.2f t/sq.m.\n",zlim);
fprintf(fp_out,"Maximum soil pressure developed = %.2f t/sq.m.\n",zmax);
fprintf(fp_out,"Design safe as the maximum soil pressure is less than the allowable soil
bearing pressure.\n\n");
qn = sqrt(pow((xn-xq),2) + pow((yn-yq),2));
xb = xn + qn*sin(aalpha + aepsln) * sin(aalpha);
yb = yn + qn*sin(aalpha + aepsln) * cos(aalpha);
ab = sqrt(pow((xa-xb),2) + pow((ya-yb),2));
etail = 100.0;
a11 = angphi + 0.0175;
a22 = angphi + 0.2618;
for (aomega = a11; aomega <= a22; aomega = (aomega + 0.0002917) )
{
w1 = qn*qn*tan(aalpha+aepsln)*gamat*0.5;
w11 = 0.5 * gamat * ab * ab;
ca1 = cos(aalpha + aomega) * cos(aalpha + aepsln) / sin(aomega - aepsln);
w2 = w11 * cos(aalpha + aepsln) * cos(aalpha + aomega) / sin(aomega - aepsln);
w3 = pow((xa-xf),2) * (tan(aalpha) + tan(aomega)) * gamat * 0.5;
bc = ab * cos(aalpha + aomega) / sin(aomega - aepsln);
w4 = ((xb-xq) + (bc*cos(aepsln)))*unifpo;
wi = w1+w2+w3+w4+(asoil*gamat)+wm;
s = wi*sin(aomega-angphi)/cos((2*angphi)-aomega);
yct = yf-(xa-xf)*(tan(aalpha)+tan(aomega));
rp = yct*yct*kp*gamat*0.5;
etai = rp/s;
if (etai > etail)
        minetai = etail;
goto ett;
minetai = etai;
etail = etai;
ett:minaomega = aomega;
}
fprintf(fp_out,"   -------------------------- \n");
fprintf(fp_out,"   check for overall stability  \n");
fprintf(fp_out,"   -------------------------- \n\n");
fprintf(fp_out,"Passive resistance developed by toe prism = %.2f t/m.\n",rp);
fprintf(fp_out,"Thrust on the vertical plane of toe prism = %.2f t/m.\n\n",s);
fprintf(fp_out,"Factor of safety for overall stability = %.2f",etai);
if (etai > 1.3)
        goto fif;
else extn = extn + 0.5;
fprintf(fp_out," --- NG ( < 2.0 ). Recalculating with base length = %.2f m.\n\n",extn);
goto zer;
fif:fprintf(fp_out," --- OK (since F.S. > 2.0 )\n\n\n");
fprintf(fp_out,"------------------------------------------------------------------------\n\n");
printf("****************************************************************
************\n");
printf("THE DETAILED DESIGN OUTPUT FOR THE GABION STRUCTURE IS IN
FILE >> %s\n",fn_out);
printf("****************************************************************
************\n");
fclose(fp_out);
return(0);
}
```

# APPENDIX F

# LISTING OF PROGRAM EARWALL

```
/*   ---------------------------------------------------------------------------   */
/*   |  DESIGN OF THE REINFORCED EARTH STRUCTURE  |   */
/*   ---------------------------------------------------------------------------   */


#include <stdio.h>
#include <math.h>
#include <ctype.h>
#define LINE 50
#define ROW 20
float a[ROW][LINE];
main()
{

FILE *fopen(),*fp_out;
char fn_out[15];

float phi11,phi1,phi,ka1,ka,angphi,angphi1,q,gama,h,pe,eall,e,sigmavb;
float sv,sh,spp,kk,kt,k1,acr,as,ared,phi2,h1,hh,ko1,ko2,ar;
float del,angdel,kah,k2,k3,c,b,m1,m2,fso,mc1,mc2,mc,rv,ll6;
float h3,h4,h5,pq,fso1,fso2,e1,fs,gama1,phi3,l,type,size;
float tens;
int ns,i,lb,la,sigvb;
extern double sin();
extern double cos();
extern double tan();
extern double sqrt();

printf("Enter name of output file:");
scanf("%s",fn_out,"w");
fp_out = fopen(fn_out,"w");
printf("Output in file:%s\n\n\n",fn_out);

printf("The R E I N F O R C E D   E A R T H is the most appropriate\n");
printf("type of structure that is suited for this application.\n\n");
printf("The detailed design of the Reinforced Earth Structure follows:\n\n\n\n");


fprintf(fp_out,"   ---------------------------------------------------------------------------   \n");
fprintf(fp_out,"   |  DESIGN OF THE REINFORCED EARTH STRUCTURE  |
\n");
fprintf(fp_out,"   ---------------------------------------------------------------------------   \n\n\n");


printf("ENTER 1 IF CASE IS HORIZONTAL BACKFILL.\nENTER 2 IF CASE IS
SLOPING BACKFILL.\nENTER 3 IF CASE HAS UNIFORM SURCHARGE.\nENTER
4 IF CASE HAS COHESIVE SOIL.\n");
scanf("%f",&type);

if (type == 1) goto one;
if (type == 2) goto two;
if (type == 3) goto three;
if (type == 4) goto four;
```

```
one:printf("What is the height (in ft.) of the earth to be supported?\n>>");
scanf("%f", &h);
printf("What is the angle of internal friction (in deg) of the special backfill soil?\n>>");
scanf("%f", &angphi);
printf("What is the angle of internal friction (in deg) of the retained soil?\n>>");
scanf("%f", &angphi1);
printf("What is the specific weight of the special backfill soil(in pcf.)?\n>>");
scanf("%f",&gama);
printf("What is the specific weight of the retained soil(in pcf.)?\n>>");
scanf("%f",&gama1);
printf("Which one of the two standard reinforcement strip sizes are you using:\nPress 1 if
the size is 40mm x 5mm  OR press 2 if the size is 60mm x 5mm\n>>");
scanf("%f",&size);
printf("What is the horizontal spacing of the steel reinforcement strips (in inches)?\nIf not
available you may give the value as 40 in.\n>>");
scanf("%f",&sh);
printf("What is the vertical spacing of the steel reinforcement strips (in inches)?\nIf not
available you may give the value as 30 in.\n>>");
scanf("%f",&sv);
printf("What is the allowable tensile stress (in psi) of the steel reinforcement strips
?\n\n>>");
scanf("%f",&tens);

printf("\n\n");
printf("        ***************************  \n");
printf("        * I N P U T   D A T A *  \n");
printf("        ***************************  \n\n");

printf("Height of the earth to be supported = %.2f ft.\n",h);
printf("Angle of internal friction of the special backfill soil = %.1f deg.\n",angphi);
printf("Angle of internal friction of the retained soil = %.1f deg.\n",angphi1);
printf("Specific weight of the special backfill soil = %.1f pcf.\n",gama);
printf("Specific weight of the retained soil = %.1f pcf.\n",gama1);
if (size == 1)
        printf("Standard size of reinforcement = 40mm x 5mm.\n");
else printf("Standard size of reinforcement = 60mm x 5mm.\n");
printf("Horizontal spacing of the steel reinforcing strips = %.1f in.\n",sh);
printf("Vertical spacing of the steel reinforcing strips = %.1f in.\n",sv);
printf("Allowable tensile stress of the steel reinforcing strips = %.2f psi.\n",tens);
printf("--------------------------------------------------------------------------\n\n\n");
printf("C A L C U L A T I N G . . . . . . . ..........\n");
if (size == 1)
        as = 200;
else as = 300;
if (size == 1)
        ared = 0.199;
else ared = 0.354;
phi11 = 45 - (angphi1*0.5);
phi1 = phi11*(3.1415926/180.0);
phi3 = angphi1*(3.1415926/180.0);
phi = angphi*(3.1415926/180.0);
phi2 = (45 + angphi*0.5)*(3.1415926/180);
del = angdel*(3.1415926/180.0);
ka = (1-sin(phi1))/(1+sin(phi1));
```

```
fprintf(fp_out,"  ----------------------------------------------\n");
fprintf(fp_out,"  :      WALL WITH HORIZONTAL BACKFILL      :\n");
fprintf(fp_out,"  ----------------------------------------------\n\n");

fprintf(fp_out,"          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^\n");
fprintf(fp_out,"          EXTERNAL STABILITY  \n");
fprintf(fp_out,"          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^\n\n");
fprintf(fp_out," 1.Sliding on Base.\n");
fprintf(fp_out," -------------------\n");

pe = 0.5*ka*gama1*h*h;
l = 1.5*ka*h/(2*tan(phi1));
fprintf(fp_out,"Coefficient of active pressure = %.2f\n",ka);
fprintf(fp_out,"The total horizontal active earth force acting on the back of the wall,Pe =
%.2f lb.\n",pe);
fprintf(fp_out,"For a minimum factor of safety of 1.5 against sliding:\n");
fprintf(fp_out,"The minimum length of reinforcement,L = %.2f ft.\n\n",l);
fprintf(fp_out," 2.Overturning about Toe.\n");
fprintf(fp_out," -------------------------\n");
lb = sqrt(2*ka*h*h/3);
fprintf(fp_out,"For a minimum factor of safety of 2.0 against overturning:\n");
fprintf(fp_out,"The minimum base width of the wall = %d ft.\n\n",lb);
fprintf(fp_out," 3.Bearing Capacity Failure.\n");
fprintf(fp_out," ----------------------------\n");
e2:e=(ka*h*h)/(6*lb);
eall=lb/6.0;
if (e < eall)
        goto e1;
else fprintf(fp_out,"The eccentricity,e = %.2f ft. is greater than the allowable value of
e(=%d/6.0) = %.2f.\n",e,lb,eall);
fprintf(fp_out,"Therefore the base width of the wall has to be increased.\n");
lb = lb + 1;
fprintf(fp_out,"Recomputing the eccentricity with base width of wall,Lb = %d\n",lb);
goto e2;
e1:fprintf(fp_out,"The eccentricity,e = %.2f ft. is less than the allowable value of
e(=Lb/6) = %.2f.----- OK.\n",e,eall);
sigmavb = gama*lb*h/(lb-(2*e));
fprintf(fp_out,"Reqd. soil bearing capacity = %.2f lb./sq.ft.\n\n\n",sigmavb);
pall:fprintf(fp_out,"          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^\n");
fprintf(fp_out,"          INTERNAL STABILITY  \n");
fprintf(fp_out,"          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^\n\n");
ns = h*12/sv;
fprintf(fp_out,"-----------------------------------\n");
fprintf(fp_out,"  Layer #   Depth Z(ft.)   \n");
fprintf(fp_out,"-----------------------------------\n");
a[1][1] = h*0.5/ns;
k1 = 1 - sin(phi);
kt = (1-sin(phi))/(1+sin(phi));
kk = k1 - kt;
fprintf(fp_out,"    1      %5.2f       \n",a[1][1]);
spp = (h - (2*a[1][1]))/(ns-1);
for ( i = 2 ; i <= (ns-1) ; ++i )
{
a[i][1] = a[i-1][1] + spp;
```

```
fprintf(fp_out,"    %2d        %5.2f          \n",i,a[i][1]);
}
a[ns][1] = h - a[1][1];
fprintf(fp_out,"    %2d        %5.2f          \n",ns,a[ns][1]);
fprintf(fp_out,"-----------------------------------\n\n");
e3:fprintf(fp_out,"1.Tensile forces to be resisted by the reinforcement:\n");
fprintf(fp_out,"------------------------------------------------------\n");
fprintf(fp_out,"----------------------------------------------------------------------------------------------
---------------------------------\n");
fprintf(fp_out,"Layer   Depth       Rv          Pe          e          Sigv        K
Sigh        FH\n");
fprintf(fp_out," #     [ft.]      [lb/ft]      [lb]       [ft]       [lb/s.ft]                [lb/s.ft]
[lb]\n");
fprintf(fp_out,"----------------------------------------------------------------------------------------------
---------------------------------\n");
for (i = 1 ; i <= ns ; ++i)
{
a[i][2]=gama*lb*a[i][1];
a[i][3]=gama*ka*a[i][1]*a[i][1]*0.5;
a[i][4]=a[i][3]*a[i][1]/(a[i][2]*3);
a[i][5]=gama*lb*a[i][1]/(lb-(2*a[i][4]));
if (a[i][1] < 20.0)
        a[i][6] = k1 - (a[i][1]*kk/20);
else a[i][6] = (1 - sin(phi))/(1 + sin(phi));
if (a[i][1] > 20.0)
        a[i][7] = tan(phi);
else a[i][7] = 1.5 - ((1.5-tan(phi))*a[i][1]/20);
a[i][8] = a[i][6]*a[i][5];
a[i][9] = a[i][8]*6.05;
fprintf(fp_out," %2d   %5.2f     %8.1f      %8.1f      %5.2f       %8.1f      %6.4f
%8.1f      %8.1f\n",i,a[i][1],a[i][2],a[i][3],a[i][4],a[i][5],a[i][6],a[i][8],a[i][9]);
}
fprintf(fp_out,"----------------------------------------------------------------------------------------------
---------------------------------\n\n");
fprintf(fp_out,"2.Tensile stress in reinforcement:\n");
fprintf(fp_out,"-----------------------------------\n");
acr = as/(25.4*25.4);
fprintf(fp_out,"---------------------------------\n");
fprintf(fp_out,"  Strip #      Tensile Stress \n");
fprintf(fp_out,"                  [psi.]    \n");
fprintf(fp_out,"---------------------------------\n");
for (i = 1 ; i <= ns ; ++i)
{
a[i][10]=a[i][9]/acr;
fprintf(fp_out,"   %2d              %8.1f\n",i,a[i][10]);
}
fprintf(fp_out,"---------------------------------\n");
for (i = 1 ; i <= ns ; ++i)
{
a[i][10]=a[i][9]/acr;
if (a[i][10] < tens)
goto pal;
fprintf(fp_out,"TENSILE STRESS IN STRIP > ALLOWABLE TENSILE
STRESS(%.2f psi.)\n",tens);
```

```
sh = sh - 2;
sv = sv - 2;
fprintf(fp_out,"Recalculate using horizontal strip spacing = %.2f in. and \n
vertical strip spacing = %.2f in.\n",sh,sv);
goto pall;
}
pal:fprintf(fp_out,"TENSILE STRESS IN STRIP < ALLOWABLE TENSILE
STRESS(%.2f psi.) ---- OK.\n\n",tens);
fprintf(fp_out,"3.Tensile stress at connection:\n");
fprintf(fp_out,"-------------------------------\n");
fprintf(fp_out,"--------------------------------\n");
fprintf(fp_out,"  Strip #    Tensile Stress \n");
fprintf(fp_out,"              [psi.]   \n");
fprintf(fp_out,"--------------------------------\n");
for (i = 1 ; i <= ns ; ++i)
{
a[i][11] = 0.85*a[i][8];
a[i][12] = a[i][11]*6.05/ared;
fprintf(fp_out,"   %2d              %8.1f\n",i,a[i][12]);
}
fprintf(fp_out,"--------------------------------\n\n");
for (i = 1 ; i <= ns ; ++i)
{
a[i][11] = 0.85*a[i][8];
a[i][12] = a[i][11]*6.05/ared;
if (a[i][12] < tens)
goto pl;
fprintf(fp_out,"TENSILE STRESS AT CONNECTION > ALLOWABLE TENSILE
STRESS(%.2f psi.)\n",tens);
sh = sh - 2;
sv = sv - 2;
fprintf(fp_out,"Recalculate using horizontal strip spacing = %.2f in. and \n
vertical strip spacing = %.2f in.\n",sh,sv);
goto pall;
}
pl:fprintf(fp_out,"TENSILE STRESS AT CONNECTION < ALLOWABLE TENSILE
STRESS(%.2f psi.) ---- OK.\n\n",tens);
fprintf(fp_out,"4.Pullout of reinforcement:\n");
fprintf(fp_out,"--------------------------\n");
fprintf(fp_out,"------------------------------------------------------------------\n");
fprintf(fp_out,"  Strip #      Le          P          FH        F.S.  \n");
fprintf(fp_out,"              [ft.]      [lb.]      [lb.]        \n");
fprintf(fp_out,"------------------------------------------------------------------\n");
for (i = 1 ; i <= ns ; ++i)
{
if (a[i][1] <= h/2)
a[i][13] = lb - 0.3*h;
else a[i][13] = lb - ((h-a[i][1])/tan(phi2));
a[i][14] = 2*0.1969*gama*a[i][1]*a[i][7]*a[i][13];
a[i][15] = a[i][14]/a[i][9];
fprintf(fp_out,"   %2d      %5.2f      %8.1f      %8.1f
%4.2f\n",i,a[i][13],a[i][14],a[i][9],a[i][15]);
}
fprintf(fp_out,"------------------------------------------------------------------\n");
```

```
for (i = 1 ; i <= ns ; ++i)
{
if (a[i][15] <= 1.5)
        lb = lb + 1;
if (a[i][15] <= 1.5)
        fprintf(fp_out,"RECALCULATING WITH Lb = %d as F.S.(PULLOUT) <
1.5\n",lb);
if (a[i][15] <= 1.5)
        goto e3;
}
goto end;
two:printf("What is the height (in ft.) of the earth to be supported?\n>>");
scanf("%f", &h);
printf("What is the angle of internal friction (in deg) of the special backfill soil?\n>>");
scanf("%f", &angphi);
printf("What is the angle of internal friction (in deg) of the retained soil?\n>>");
scanf("%f", &angphi1);
printf("What is the specific weight of the special backfill soil(in pcf.)?\n>>");
scanf("%f",&gama);
printf("What is the specific weight of the retained soil(in pcf.)?\n>>");
scanf("%f",&gama1);
printf("What is the angle (in deg.) that the backfill soil makes with the horizontal?\n>>");
scanf("%f",&angdel);
printf("Which one of the two standard reinforcement strip sizes are you using:\nPress 1 if
the size is 40mm x 5mm  OR press 2 if the size is 60mm x 5mm\n>>");
scanf("%f",&size);
printf("What is the horizontal spacing of the steel reinforcement strips (in inches)?\nIf not
available you may give the value as 40 in.\n>>");
scanf("%f",&sh);
printf("What is the vertical spacing of the steel reinforcement strips (in inches)?\nIf not
available you may give the value as 30 in.\n>>");
scanf("%f",&sv);
printf("What is the allowable tensile stress (in psi) of the steel reinforcement strips
?\n\n>>");
scanf("%f",&tens);
printf("        ***************************  \n");
printf("        * I N P U T   D A T A *  \n");
printf("        ***************************  \n\n");

printf("Height of the earth to be supported = %.2f ft.\n",h);
printf("Angle that the backfill makes with the horizontal = %.1f deg.\n",angdel);
printf("Angle of internal friction of the special backfill soil = %.1f deg.\n",angphi);
printf("Angle of internal friction of the retained soil = %.1f deg.\n",angphi1);
printf("Specific weight of the special backfill soil = %.1f pcf.\n",gama);
printf("Specific weight of the retained soil = %.1f pcf.\n\n\n",gama1);
if (size == 1)
        printf("Standard size of reinforcement = 40mm x 5mm.\n");
else printf("Standard size of reinforcement = 60mm x 5mm.\n");
printf("Horizontal spacing of the steel reinforcing strips = %.1f in.\n",sh);
printf("Vertical spacing of the steel reinforcing strips = %.1f in.\n",sv);
printf("Allowable tensile stress of the steel reinforcing strips = %.2f psi.\n",tens);
printf("------------------------------------------------------------------------\n\n\n");
printf("C A L C U L A T I N G  . . . . . . ..........\n");
```

```
if (size == 1)
        as = 200;
else as = 300;
if (size == 1)
        ared = 0.199;
else ared = 0.354;
phil1 = 45 - (angphi1*0.5);
phi1 = phil1*(3.1415926/180.0);
phi3 = angphi1*(3.1415926/180.0);
phi = angphi*(3.1415926/180.0);
phi2 = (45 + angphi*0.5)*(3.1415926/180);
del = angdel*(3.1415926/180.0);
ka = (1-sin(phi1))/(1+sin(phi1));
fprintf(fp_out," ---------------------------------------------\n");
fprintf(fp_out," :       WALL WITH SLOPING BACKFILL       :\n");
fprintf(fp_out," ---------------------------------------------\n\n");
fprintf(fp_out,"        ^^^^^^^^^^^^^^^^^^^^^^^^^^^^\n");
fprintf(fp_out,"        EXTERNAL STABILITY   \n");
fprintf(fp_out,"        ^^^^^^^^^^^^^^^^^^^^^^^^^^^^\n\n");
fprintf(fp_out," 1.Sliding on Base.\n");
fprintf(fp_out," -----------------\n");
k1 = cos(del);
k2 = cos(phi1);
k3 = sqrt((k1*k1) - (k2*k2));
kah = k1*k1*((k1-k3)/(k1+k3));
fprintf(fp_out,"Horizontal component of the active earth pressure coeff.,Kah =
%.4f\n",kah);
la = 5;
ll:c = la*gama*(h+(la*0.25))*tan(phi1);
b = 0.75*kah*gama*pow((0.5*la + h),2);
if (c >= b)
        goto la;
else la = la + 1;
goto ll;
la:fprintf(fp_out,"For a minimum factor of safety of 1.5 against sliding:\n");
fprintf(fp_out,"The minimum length of reinforcement,L = %d ft.\n\n",la);
fprintf(fp_out," 2.Overturning about Toe.\n");
fprintf(fp_out," ------------------------\n");
mm1:m1 = (la*la*h*gama*0.5) + (la*la*la*gama/6);
m2 = kah*gama*pow((h + 0.5*la),3)/6;
fso = m1/m2;
fprintf(fp_out,"For the reinforcement length = %d ft.:\n",la);
fprintf(fp_out,"The factor of safety against overturning = %.2f",fso);
if ( fso >= 2.0)
        goto fso;
else la = la + 1;
fprintf(fp_out,"--------- NG (since < 2.0).\n");
fprintf(fp_out,"Recalculating with reinforcement length = %d ft.:\n",la);
goto mm1;
fso:fprintf(fp_out,"-------- OK (since > 2.0)\n\n");
fprintf(fp_out," 3.Bearing Capacity Failure.\n");
fprintf(fp_out," ---------------------------\n");
h1 = h + (la*0.5);
mc1:mc1 = kah*gama*h1*h1*h1/6.0;
```

```
mc2 = gama*(h1-h)*la*la/12.0;
mc = mc1 - mc2;
fprintf(fp_out,"The moment around the centerline at the base = %.2f ft.lb.\n",mc);
rv = (la*h*gama) + (0.25*la*la*gama);
fprintf(fp_out,"Vertical rection = %.2f lb.\n",rv);
l16 = la/6.0;
e = mc/rv;
fprintf(fp_out,"Eccentricity,e = %.2f\n",e);
if (e < l16)
        goto la6;
else fprintf(fp_out,"The length %d is NOT sufficient to maintain the resultant within the
middle third --- OK (since e = %.2f ft. < (L/6.0) = %.2f ft..\n",la,e,l16);
la = la+1;
fprintf(fp_out,"Therefore, RECALCULATE with length, L = %d ft.\n",la);
goto mc1;
la6:fprintf(fp_out,"The length L = %d ft. is sufficient to maintain the resultant within the
middle third ----- OK (since e = %.2f ft. <  L / 6.0 = %.2f ft..\n\n",la,e,l16);
ull:fprintf(fp_out,"          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^\n");
fprintf(fp_out,"          INTERNAL STABILITY  \n");
fprintf(fp_out,"          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^\n\n");
ns = h*12/sv;
a[1][1] = h*0.5/ns;
k1 = 1 - sin(phi);
kt = (1-sin(phi))/(1+sin(phi));
kk = k1 - kt;
spp = (h - (2*a[1][1]))/(ns-1);
for ( i = 2 ; i <= (ns-1) ; ++i )
{
a[i][1] = a[i-1][1] + spp;
}
a[ns][1] = h - a[1][1];
fprintf(fp_out,"1.Rupture of reinforcement:\n");
fprintf(fp_out,"---------------------------\n");
fprintf(fp_out,"----------------------------------------------------------------------------------------------
---------------\n");
fprintf(fp_out," Strip    Depth    Vertical    Pe    Eccentricity    Sigv    Sigh
FH  \n");
fprintf(fp_out,"  #     [ft.]   Load[lb.]   [lb.]    [ft.]     [psf.]  [psf.]
[lb.] \n");
fprintf(fp_out,"----------------------------------------------------------------------------------------------
---------------\n");
hh = la*0.5;
for (i=1 ; i<=ns ; ++i)
{
a[i][16] = hh + a[i][1];
a[i][17] = (a[i][16] + a[i][1]) * 0.5 *la * gama;
a[i][18] = kah * gama * 0.5 * a[i][16] * a[i][16];
a[i][19] = ((a[i][18]*a[i][16]/3) - ((la*0.5*gama*(a[i][16]-a[i][1]))*(la/6)))/a[i][17];
a[i][5] = a[i][17]/(la-(2*a[i][19]));
if (a[i][19] < 0)
        a[i][19] = 0;
if (a[i][19] < 0)
        a[i][5] = a[i][17]/la;
ko1 = 1 - sin(phi);
```

```
ko2 = (1 - sin(phi))/(1 + sin(phi));
a[i][20] = (hh*0.5) + a[i][1];
if (a[i][1] < 20.0)
a[i][6] = ko1 - (a[i][20]*(ko1-ko2)/20);
else a[i][6] = (1 - sin(phi))/(1 + sin(phi));
a[i][8] = a[i][6]*a[i][5];
ar = 6.05;
a[i][9] = ar*a[i][8];
fprintf(fp_out,"  %2d      %5.2f      %8.1f  %8.1f      %5.2f            %8.1f   %8.1f
%8.1f\n",i,a[i][1],a[i][17],a[i][18],a[i][19],a[i][5],a[i][8],a[i][9]);
acr = as/(25.4*25.4);
a[i][10] = a[i][9]/acr;
a[i][21] = 0.85*a[i][9]/0.354;
h3 = hh+h;
h4 = (h3+h)*0.5;
h5 = h4*0.5;
if (a[i][20] <= h5)
        a[i][22] = la - (0.3*h4);
else a[i][22] = la - ((h4-a[i][20])/(tan(phi2)));
if (a[i][1] > 20.0)
        a[i][7] = tan(phi);
else a[i][7] = 1.5 - ((1.5-tan(phi))*a[i][20]/20);
a[i][23] = 2*0.06*3.28*gama*a[i][20]*a[i][22]*a[i][7];
a[i][15] = a[i][23]/a[i][9];
}
fprintf(fp_out,"--------------------------------------------------------------------------------------
----------------\n\n");
fprintf(fp_out,"2.Tensile stress in reinforcement and at the connection & pullout of
reinforcement:\n");
fprintf(fp_out,"------------------------------------------------------------------------------------
\n");
fprintf(fp_out,"--------------------------------------------------------------------------------------
--------------\n");
fprintf(fp_out," Strip    Depth      Stress in     Stress @          Mu*      Le      P
F.S.  \n");
fprintf(fp_out,"  #       [ft.]    Strip [psi.]  Connection[psi]              [ft]    [lb]
\n");
fprintf(fp_out,"--------------------------------------------------------------------------------------
--------------\n");
for (i=1 ; i<=ns ; ++i)
{
fprintf(fp_out,"  %2d      %5.2f     %8.1f         %8.1f         %6.3f    %5.2f    %8.1f
%5.2f\n",i,a[i][1],a[i][10],a[i][21],a[i][7],a[i][22],a[i][23],a[i][15]);
}
fprintf(fp_out,"--------------------------------------------------------------------------------------
-------\n");
for (i=1 ; i<=ns ; ++i)
{
if (a[i][10] < tens)
goto ul;
fprintf(fp_out,"TENSILE STRESS IN STRIP > ALLOWABLE TENSILE
STRESS(%.2f psi.)\n",tens);
sh = sh - 2;
sv = sv - 2;
```

```
fprintf(fp_out,"Recalculate using horizontal strip spacing = %.2f in. and \n
vertical strip spacing = %.2f in.\n",sh,sv);
goto ull;
}
ul:fprintf(fp_out,"TENSILE STRESS IN STRIP < ALLOWABLE TENSILE
STRESS(%.2f psi.) ---- OK.\n\n",tens);
for (i=1 ; i<=ns ; ++i)
{
if (a[i][21] < tens)
goto ml;
fprintf(fp_out,"TENSILE STRESS AT CONNECTION > ALLOWABLE TENSILE
STRESS(%.2f psi.)\n",tens);
sh = sh - 2;
sv = sv - 2;
fprintf(fp_out,"Recalculate using horizontal strip spacing = %.2f in. and \n
vertical strip spacing = %.2f in.\n",sh,sv);
goto ull;
}
ml:fprintf(fp_out,"TENSILE STRESS AT CONNECTION < ALLOWABLE TENSILE
STRESS(%.2f psi.) ---- OK.\n\n",tens);
for (i = 1 ; i <= ns ; ++i)
{
if (a[i][15] <= 1.5)
        la = la + 0.5;
if (a[i][15] <= 1.5)
        fprintf(fp_out,"RECALCULATING WITH La = %d as F.S.(PULLOUT) <
1.5\n",la);
if (a[i][15] <= 1.5)
        goto la6;
}
goto end;
three:printf("What is the height (in ft.) of the earth to be supported?\n>>");
scanf("%f", &h);
printf("What is the angle of internal friction (in deg) of the special backfill soil?\n>>");
scanf("%f", &angphi);
printf("What is the angle of internal friction (in deg) of the retained soil?\n>>");
scanf("%f", &angphi1);
printf("What is the specific weight of the special backfill soil(in pcf.)?\n>>");
scanf("%f",&gama);
printf("What is the specific weight of the special backfill soil(in pcf.)?\n>>");
scanf("%f",&gama1);
printf("What is the uniform vertical surcharge loading at the top of the wall(in
psf.)?\n>>");
scanf("%f",&q);
printf("Which one of the two standard reinforcement strip sizes are you using:\nPress 1 if
the size is 40mm x 5mm  OR press 2 if the size is 60mm x 5mm\n>>");
scanf("%f",&size);
printf("What is the horizontal spacing of the steel reinforcement strips (in inches)?\nIf not
available you may give the value as 40 in.\n>>");
scanf("%f",&sh);
printf("What is the vertical spacing of the steel reinforcement strips (in inches)?\nIf not
available you may give the value as 30 in.\n>>");
scanf("%f",&sv);
```

```
printf("What is the allowable tensile stress (in psi) of the steel reinforcement strips
?\n\n>>");
scanf("%f",&tens);
printf("          ***************************  \n");
printf("          * INPUT  DATA *  \n");
printf("          ***************************  \n\n");

printf("Height of the earth to be supported = %.2f ft.\n",h);
printf("Uniform vertical surcharge at the top of the wall = %.1f psf.\n",q);
printf("Angle of internal friction of the special backfill soil = %.1f deg.\n",angphi);
printf("Angle of internal friction of the retained soil = %.1f deg.\n",angphi1);
printf("Specific weight of the special backfill soil = %.1f pcf.\n",gama);
printf("Specific weight of the retained soil = %.1f pcf.\n\n\n",gama1);
if (size == 1)
        printf("Standard size of reinforcement = 40mm x 5mm.\n");
else printf("Standard size of reinforcement = 60mm x 5mm.\n");
printf("Horizontal spacing of the steel reinforcing strips = %.1f in.\n",sh);
printf("Vertical spacing of the steel reinforcing strips = %.1f in.\n",sv);
printf("Allowable tensile stress of the steel reinforcing strips = %.2f psi.\n",tens);
printf("--------------------------------------------------------------------\n\n\n");
printf("C A L C U L A T I N G . . . . . . ..........\n");
if (size == 1)
        as = 200;
else as = 300;
if (size == 1)
        ared = 0.199;
else ared = 0.354;
phi11 = 45 - (angphi1*0.5);
phi1 = phi11*(3.1415926/180.0);
phi3 = angphi1*(3.1415926/180.0);
phi = angphi*(3.1415926/180.0);
phi2 = (45 + angphi*0.5)*(3.1415926/180);
del = angdel*(3.1415926/180.0);
ka = (1-sin(phi1))/(1+sin(phi1));
fprintf(fp_out,"  ---------------------------------------------\n");
fprintf(fp_out,"  :            WALL WITH SURCHARGE          :\n");
fprintf(fp_out,"  ---------------------------------------------\n\n");

fprintf(fp_out,"         ^^^^^^^^^^^^^^^^^^^^^^^^^^^\n");
fprintf(fp_out,"         EXTERNAL STABILITY   \n");
fprintf(fp_out,"         ^^^^^^^^^^^^^^^^^^^^^^^^^^^\n\n");
pe = ka*gama*h*h*0.5;
pq = q*ka*h;
fprintf(fp_out," 1.Sliding on Base.\n");
fprintf(fp_out," ------------------\n");
l = 1.5*(pe+pq)/(((gama*h)+q)*tan(phi1));
fprintf(fp_out,"Coefficient of active pressure = %.2f\n",ka);
fprintf(fp_out,"External horizontal force = %.2f lb./ft.\n",pe);
fprintf(fp_out,"Resultant horizontal force from surcharge loading = %.2f lb./ft.\n",pq);
fprintf(fp_out,"For a minimum factor of safety of 1.5 against sliding:\n");
fprintf(fp_out,"The minimum length of reinforcement,L = %.1f ft.\n\n",l);
fprintf(fp_out," 2.Overturning about Toe.\n");
fprintf(fp_out," -----------------------\n");
lz:fso1 = (q+(gama*h))*l*l*0.5;
```

```
fso2 = (pe*h/3)+(pq*h*0.5);
fso = fso1/fso2;
fprintf(fp_out,"F.S.(overturning) = %.2f ----",fso);
if (fso > 2.0)
      goto fo;
else l = l + 0.25;
if (fso < 2.0)
fprintf(fp_out,"NG (since FS < 2.0). Recalculate with L = %.1f\n",l);
if (fso < 2.0)
      goto lz;
fo:fprintf(fp_out," OK (since FS > 2.0)\n\n");
fprintf(fp_out," 3.Bearing Capacity Failure.\n");
fprintf(fp_out," --------------------------\n");
rv:rv = (q*l) + (h*l*gama);
e = ((pq*h*0.5) + (pe*h/3))/rv;
el = l/6;
fprintf(fp_out,"Vertical reaction,Rv = %.2f lb.\n",rv);
fprintf(fp_out,"Eccentricity = %.2f ft.\n",e);
if (e < el)
      goto x1;
else fprintf(fp_out,"Resultant outside middle third.(since %.2f > (%.1f/6.0 =
%.2f))\n\n",e,l,el);
l = l + 0.25;
fprintf(fp_out,"Recompute with L = %.1f:\n",l);
goto rv;
x1:fprintf(fp_out,"Resultant inside middle third.(since %.2f < (%.1f/6.0 = %.2f))---
OK.\n",e,l,el);
sigvb = 2*rv/(l - (2*e));
fprintf(fp_out,"The ultimate bearing capacity of the foundation = %d psf.\n",sigvb);
cll:fprintf(fp_out,"      ^^^^^^^^^^^^^^^^^^^^^^^^^^^\n");
fprintf(fp_out,"         INTERNAL STABILITY   \n");
fprintf(fp_out,"      ^^^^^^^^^^^^^^^^^^^^^^^^^^^\n\n");
ns = h*12/sv;
a[1][1] = h*0.5/ns;
kl = 1 - sin(phi);
kt = (1-sin(phi))/(1+sin(phi));
kk = kl - kt;
spp = (h - (2*a[1][1]))/(ns-1);
for ( i = 2 ; i <= (ns-1) ; ++i )
{
a[i][1] = a[i-1][1] + spp;
}
a[ns][1] = h - a[1][1];

ff:fprintf(fp_out,"1.Tensile stress in reinforcement and at connection :\n");
fprintf(fp_out,"-----------------------------------------------------\n");
fprintf(fp_out,"FOR RUPTURE:\n");
fprintf(fp_out,"--------------------------------------------------------------------------------
------------\n");
fprintf(fp_out," Reinf.  Depth    e     Sigv.    K    Sigh.    FH      FT      FTc
F.S.   \n");
fprintf(fp_out," Strip                                                              \n");
fprintf(fp_out," #      [ft.]  [ft.]  [psf.]         [psf.]    [lb.]   [lb.]   [lb.]
\n");
```

```
fprintf(fp_out,"-----------------------------------------------------------------------------------------
-------------\n");
for (i=1;i<=ns;++i)
{
a[i][24] = (l*gama*a[i][1])+(q*l);
a[i][25] = ((ka*gama*pow(a[i][1],3)/6.0) + (ka*q*pow(a[i][1],2)*0.5))/(a[i][24]);
a[i][5] = a[i][24]/(1 - (2*a[i][25]));
if (a[i][1] < 20.0)
        a[i][6] = k1 - (a[i][1]*kk/20);
else a[i][6] = (1 - sin(phi))/(1 + sin(phi));
a[i][8] = a[i][6]*a[i][5];
a[i][9] = a[i][8]*6.05;
acr = as / (25.4*25.4);
a[i][10]=a[i][9]/acr;
a[i][26] = 40000/a[i][10];
a[i][21]=0.85*a[i][9];
a[i][27] = a[i][21]/0.354;
a[i][28] = 40000/a[i][27];
if (a[i][26] < a[i][28])
        a[i][29] = a[i][26];
        else a[i][29] = a[i][28];
fprintf(fp_out," %2d    %5.2f    %5.2f    %7.1f    %.3f  %7.1f    %7.1f    %7.1f
%7.1f    %4.2f\n",i,a[i][1],a[i][25],a[i][5],a[i][6],a[i][8],a[i][9],a[i][10],a[i][27],a[i][29]);
}
fprintf(fp_out,"-----------------------------------------------------------------------------------------
-------------\n\n");
for (i=1 ; i<=ns ; ++i)
{
if (a[i][10] < tens)
goto cl;
fprintf(fp_out,"TENSILE STRESS IN STRIP > ALLOWABLE TENSILE
STRESS(%.2f psi.)\n",tens);
sh = sh - 2;
sv = sv - 2;
fprintf(fp_out,"Recalculate using horizontal strip spacing = %.2f in. and \n
vertical strip spacing = %.2f in.\n",sh,sv);
goto cll;
}
cl:fprintf(fp_out,"TENSILE STRESS IN STRIP < ALLOWABLE TENSILE
STRESS(%.2f psi.) ---- OK.\n\n",tens);
for (i=1 ; i<=ns ; ++i)
{
if (a[i][25] < tens)
goto dl;
fprintf(fp_out,"TENSILE STRESS AT CONNECTION > ALLOWABLE TENSILE
STRESS(%.2f psi.)\n",tens);
sh = sh - 2;
sv = sv - 2;
fprintf(fp_out,"Recalculate using horizontal strip spacing = %.2f in. and \n
vertical strip spacing = %.2f in.\n",sh,sv);
goto cll;
}
dl:fprintf(fp_out,"TENSILE STRESS AT CONNECTION < ALLOWABLE TENSILE
STRESS(%.2f psi.) ---- OK.\n\n",tens);
```

```
fprintf(fp_out,"2.Pullout of reinforcement.\n");
fprintf(fp_out,"--------------------------\n");
fprintf(fp_out,"FOR PULLOUT:\n");
fprintf(fp_out,"------------------------------------------------------------------------------------------
-------------\n");
fprintf(fp_out," Reinf.  Depth     e      Sigv.      Sigh.     FH       Mu*    Le      P
F.S.   \n");
fprintf(fp_out," Strip                                                              \n");
fprintf(fp_out," #      [ft.]  [ft.]   [psf.]     [psf.]    [lb.]           [ft.]  [lb.]
\n");
fprintf(fp_out,"------------------------------------------------------------------------------------------
-------------\n");
for (i=1;i<=ns;++i)
{
if (a[i][1] > 20.0)
        a[i][7] = tan(phi);
else a[i][7] = 1.5 - ((1.5-tan(phi))*a[i][1]/20);
if (a[i][1] <= h/2)
a[i][13] = l - 0.3*h;
else a[i][13] = l - ((h-a[i][1])/tan(phi2));
a[i][14] = 2*0.06*3.28*gama*a[i][1]*a[i][13]*a[i][7];
a[i][25] = ((ka*gama*pow(a[i][1],3)/6.0) + (ka*q*pow(a[i][1],2)*0.5))/(l*gama*a[i][1]);
a[i][5] = (l*gama*a[i][1])/(l - (2*a[i][25]));
a[i][8] = a[i][6]*a[i][5];
a[i][9] = a[i][8] * 6.05;
a[i][15] = a[i][14]/a[i][9];
fprintf(fp_out," %2d    %5.2f    %5.2f     %7.1f    %7.1f     %7.1f  %7.1f     %5.2f
%7.1f    %4.2f\n",i,a[i][1],a[i][25],a[i][5],a[i][8],a[i][9],a[i][7],a[i][13],a[i][14],a[i][15]);
}
fprintf(fp_out,"------------------------------------------------------------------------------------------
-------------\n");
for (i=1;i<=ns;++i)
{
if (a[i][15] < 1.5)
        l = l+0.25;
if (a[i][15] < 1.5)
fprintf(fp_out,"Recalculating with L = %.2f ft.as F.S.(pullout) < 1.5:\n",l);
if (a[i][15] < 1.5)
        goto ff;
}
goto end;
four:printf("What is the height (in ft.) of the earth to be supported?\n>>");
scanf("%f", &h);
printf("What is the angle of internal friction (in deg) of the special backfill soil?\n>>");
scanf("%f", &angphi);
printf("What is the angle of internal friction (in deg) of the retained soil?\n>>");
scanf("%f", &angphi1);
printf("What is the specific weight of the special backfill soil(in pcf.)?\n>>");
scanf("%f",&gama);
printf("What is the specific weight of the special backfill soil(in pcf.)?\n>>");
scanf("%f",&gama1);
printf("Which one of the two standard reinforcement strip sizes are you using:\nPress 1 if
the size is 40mm x 5mm  OR press 2 if the size is 60mm x 5mm\n>>");
scanf("%f",&size);
```

```
printf("What is the horizontal spacing of the steel reinforcement strips (in inches)?\nIf not available you may give the value as 40 in.\n>>");
scanf("%f",&sh);
printf("What is the vertical spacing of the steel reinforcement strips (in inches)?\nIf not available you may give the value as 30 in.\n>>");
scanf("%f",&sv);
printf("What is the allowable tensile stress (in psi) of the steel reinforcement strips ?\n\n>>");
scanf("%f",&tens);
printf("          **************************  \n");
printf("          * INPUT  DATA *  \n");
printf("          **************************  \n\n");

printf("Height of the earth to be supported = %.2f ft.\n",h);
printf("Angle of internal friction of the special backfill soil = %.1f deg.\n",angphi);
printf("Angle of internal friction of the retained soil = %.1f deg.\n",angphi1);
printf("Specific weight of the special backfill soil = %.1f pcf.\n",gama);
printf("Specific weight of the retained soil = %.1f pcf.\n\n\n",gama1);
if (size == 1)
        printf("Standard size of reinforcement = 40mm x 5mm.\n");
else printf("Standard size of reinforcement = 60mm x 5mm.\n");
printf("Horizontal spacing of the steel reinforcing strips = %.1f in.\n",sh);
printf("Vertical spacing of the steel reinforcing strips = %.1f in.\n",sv);
printf("Allowable tensile stress of the steel reinforcing strips = %.2f psi.\n",tens);
printf("------------------------------------------------------------------------\n\n\n");
printf("C A L C U L A T I N G . . . . . . .........\n");
if (size == 1)
        as = 200;
else as = 300;
if (size == 1)
        ared = 0.199;
else ared = 0.354;
phi11 = 45 - (angphi1*0.5);
phi1 = phi11*(3.1415926/180.0);
phi3 = angphi1*(3.1415926/180.0);
phi = angphi*(3.1415926/180.0);
phi2 = (45 + angphi*0.5)*(3.1415926/180);
del = angdel*(3.1415926/180.0);
ka = (1-sin(phi1))/(1+sin(phi1));
fprintf(fp_out," ---------------------------------------------\n");
fprintf(fp_out," :  WALL WITH COHESIVE SOIL AND FOUNDATION  :\n");
fprintf(fp_out," ---------------------------------------------\n\n");

fprintf(fp_out,"        ^^^^^^^^^^^^^^^^^^^^^^^^^^^^\n");
fprintf(fp_out,"        EXTERNAL STABILITY  \n");
fprintf(fp_out,"        ^^^^^^^^^^^^^^^^^^^^^^^^^^^^\n\n");
ka = pow(tan(phi1),2);
pe = ka*gama1*h*h*0.5;
fprintf(fp_out," 1.Sliding on Base.\n");
fprintf(fp_out," -----------------\n");
l = 1.5*pe/(gama*h*tan(phi3));
fprintf(fp_out,"Coefficient of active pressure = %.2f\n",ka);
fprintf(fp_out,"External horizontal force = %.2f lb./ft.\n",pe);
fprintf(fp_out,"For a minimum factor of safety of 1.5 against sliding:\n");
```

```
fprintf(fp_out,"The minimum length of reinforcement,L = %.1f ft.\n\n",l);
fprintf(fp_out," 2.Overturning about Toe.\n");
fprintf(fp_out," ------------------------\n");
az:fs = 3*gama*l*l/(2*pe);
fprintf(fp_out,"F.S.(overturning) = %.2f ----",fs);
if (fso > 2.0)
        goto ao;
else l = l + 0.25;
if (fs < 2.0)
fprintf(fp_out,"NG (since FS < 2.0). Recalculate with L = %.1f\n",l);
if (fs < 2.0)
        goto az;
ao:fprintf(fp_out," OK (since FS > 2.0)\n\n");
fprintf(fp_out," 3.Bearing Capacity Failure.\n");
fprintf(fp_out," ---------------------------\n");
av:rv = gama*h*l;
e = (pe*h)/(3*rv);
e1 = l/6;
fprintf(fp_out,"Vertical reaction,Rv = %.2f lb.\n",rv);
fprintf(fp_out,"Eccentricity = %.2f ft.\n",e);
if (e < e1)
        goto ax1;
else fprintf(fp_out,"Resultant outside middle third.(since %.2f > (%.1f/6.0 =
%.2f))\n\n",e,l,e1);
l = l + 0.25;
fprintf(fp_out,"Recompute with L = %.1f:\n",l);
goto av;
ax1:fprintf(fp_out,"Resultant inside middle third.(since %.2f < (%.1f/6.0 = %.2f))---
OK.\n",e,l,e1);
sigvb = 2*rv/(l - (2*e));
fprintf(fp_out,"The ultimate bearing capacity of the foundation = %d psf.\n",sigvb);
ggl:fprintf(fp_out,"         ^^^^^^^^^^^^^^^^^^^^^^^^^^^\n");
fprintf(fp_out,"         INTERNAL STABILITY   \n");
fprintf(fp_out,"         ^^^^^^^^^^^^^^^^^^^^^^^^^^^\n\n");
ns = h*12/sv;
fprintf(fp_out,"----------------------------------\n");
fprintf(fp_out,"  Layer #    Depth Z(ft.)   \n");
fprintf(fp_out,"----------------------------------\n");
a[1][1] = h*0.5/ns;
k1 = 1 - sin(phi);
kt = (1-sin(phi))/(1+sin(phi));
kk = k1 - kt;
fprintf(fp_out,"    1        %4.2f       \n",a[1][1]);
spp = (h - (2*a[1][1]))/(ns-1);
for ( i = 2 ; i <= (ns-1) ; ++i )
{
a[i][1] = a[i-1][1] + spp;
fprintf(fp_out,"   %2d        %4.2f       \n",i,a[i][1]);
}
a[ns][1] = h - a[1][1];
fprintf(fp_out,"   %2d        %4.2f       \n",ns,a[ns][1]);
fprintf(fp_out,"----------------------------------\n\n");
m3:fprintf(fp_out,"1.Tensile forces to be resisted by the reinforcement:\n");
fprintf(fp_out,"-----------------------------------------------------\n");
```

```
fprintf(fp_out,"------------------------------------------------------------------------------------------
--------------------\n");
fprintf(fp_out,"Layer   Depth   Rv   Pe   e   Sigv   K   Sigh
FH\n");
fprintf(fp_out," #   [ft.]   [lb/ft]   [lb]   [ft]   [lb/s.ft]   [lb/s.ft]
[lb]\n");
fprintf(fp_out,"------------------------------------------------------------------------------------------
--------------------\n");
for (i = 1 ; i <= ns ; ++i)
{
a[i][2]=gama*l*a[i][1];
a[i][3]=gama*ka*a[i][1]*a[i][1]*0.5;
a[i][4]=a[i][3]*a[i][1]/(a[i][2]*3);
a[i][5]=gama*l*a[i][1]/(l-(2*a[i][4]));
if (a[i][1] < 20.0)
        a[i][6] = k1 - (a[i][1]*kk/20);
else a[i][6] = (1 - sin(phi))/(1 + sin(phi));
if (a[i][1] > 20.0)
        a[i][7] = tan(phi);
else a[i][7] = 1.5 - ((1.5-tan(phi))*a[i][1]/20);
a[i][8] = a[i][6]*a[i][5];
a[i][9] = a[i][8]*6.05;
fprintf(fp_out," %2d   %5.2f  %7.1f   %7.1f   %4.2f   %7.1f   %4.2f
%7.1f   %7.1f\n",i,a[i][1],a[i][2],a[i][3],a[i][4],a[i][5],a[i][6],a[i][8],a[i][9]);
}
fprintf(fp_out,"------------------------------------------------------------------------------------------
--------------------\n\n");
fprintf(fp_out,"2.Tensile stress in reinforcement:\n");
fprintf(fp_out,"----------------------------------\n");
acr = as/(25.4*25.4);
fprintf(fp_out,"--------------------------------\n");
fprintf(fp_out,"  Strip #    Tensile Stress \n");
fprintf(fp_out,"             [psi.]   \n");
fprintf(fp_out,"--------------------------------\n");
for (i = 1 ; i <= ns ; ++i)
{
a[i][10]=a[i][9]/acr;
fprintf(fp_out,"  %2d         %8.2f\n",i,a[i][10]);
}
fprintf(fp_out,"--------------------------------\n");
for (i=1 ; i<=ns ; ++i)
{
if (a[i][10] < tens)
goto gl;
fprintf(fp_out,"TENSILE STRESS IN STRIP > ALLOWABLE TENSILE
STRESS(%.2f psi.)\n",tens);
sh = sh - 2;
sv = sv - 2;
fprintf(fp_out,"Recalculate using horizontal strip spacing = %.2f in. and \n
vertical strip spacing = %.2f in.\n",sh,sv);
goto ggl;
}
gl:fprintf(fp_out,"TENSILE STRESS IN STRIP < ALLOWABLE TENSILE
STRESS(%.2f psi.) ---- OK.\n\n",tens);
```

```
fprintf(fp_out,"3.Tensile stress at connection:\n");
fprintf(fp_out,"------------------------------\n");
fprintf(fp_out,"--------------------------------\n");
fprintf(fp_out,"  Strip #     Tensile Stress \n");
fprintf(fp_out,"              [psi.]  \n");
fprintf(fp_out,"--------------------------------\n");
for (i = 1 ; i <= ns ; ++i)
{
a[i][11] = 0.85*a[i][8];
a[i][12] = a[i][11]*6.05/ared;
fprintf(fp_out,"  %2d        %8.2f\n",i,a[i][12]);
}
fprintf(fp_out,"--------------------------------\n\n");
for (i=1 ; i<=ns ; ++i)
{
if (a[i][12] < tens)
goto hl;
fprintf(fp_out,"TENSILE STRESS AT CONNECTION > ALLOWABLE TENSILE
STRESS(%.2f psi.)\n",tens);
sh = sh - 2;
sv = sv - 2;
fprintf(fp_out,"Recalculate using horizontal strip spacing = %.2f in. and \n
vertical strip spacing = %.2f in.\n",sh,sv);
goto ggl;
}
hl:fprintf(fp_out,"TENSILE STRESS AT CONNECTION < ALLOWABLE TENSILE
STRESS(%.2f psi.) ---- OK.\n\n",tens);
fprintf(fp_out,"4.Pullout of reinforcement:\n");
fprintf(fp_out,"--------------------------\n");
fprintf(fp_out,"--------------------------------------------------------------\n");
fprintf(fp_out,"  Strip #     Le       P       FH       F.S.  \n");
fprintf(fp_out,"              [ft.]    [lb.]   [lb.]          \n");
fprintf(fp_out,"--------------------------------------------------------------\n");
for (i = 1 ; i <= ns ; ++i)
{
if (a[i][1] <= h/2)
a[i][13] = l - 0.3*h;
else a[i][13] = l - ((h-a[i][1])/tan(phi2));
a[i][14] = 2*0.1969*gama*a[i][1]*a[i][7]*a[i][13];
a[i][15] = a[i][14]/a[i][9];
fprintf(fp_out,"  %2d       %4.2f      %8.2f      %8.2f
%4.2f\n",i,a[i][13],a[i][14],a[i][9],a[i][15]);
}
fprintf(fp_out,"--------------------------------------------------------------\n");

for (i = 1 ; i <= ns ; ++i)
{
if (a[i][15] <= 1.5)
        l = l + 1;
if (a[i][15] <= 1.5)
        fprintf(fp_out,"RECALCULATING WITH L = %.2f ft. as F.S.(PULLOUT) <
1.5\n",l);
if (a[i][15] <= 1.5)
        goto m3;
```

```
}
end:printf("**********************************************************
************\n");
printf("THE DETAILED DESIGN OF THE  REINFORCED  EARTH  IS IN FILE >>
%s \n",fn_out);
printf("*************************************************************
**********\n");
fclose(fp_out);
return(0);
}
```

# APPENDIX G

# LISTING OF PROGRAM SHEETPILE

```
/*-------------------------------------------------------------------------------------------*/
/*                          DESIGN OF SHEET PILES                                            */
/*-------------------------------------------------------------------------------------------*/
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <process.h>
#include <graph.h>
#include <dos.h>
#include <signal.h>
int faiu,fail;/* These are for granular soil */
float cu,z;/* This is for cohesive soil */
float hanchor,hwater,hwall;
int frictionwall;
int aaa,aa1,ch;
float gamau,gamal,shdepth,surcharge;
float c,k,ka,kp,kat1,kat2,kpt1,kpt2,mm2,mpos;
float a,pa,ppa,pp,ppp,pppp,ybar,capty;
float ra,rp,par,p1,c1,c2,c3,c4,c5;
float a1,a2,a3,a4,a5,a6,a7,a8;
float c1,c2,c3,c4,c5,y2,hhwall;
float root4(),root3(),root2(),gx(),gx3();
void check(),input(),output();
int get_mode(),moment();
void clear_screen(),setscreen(int),set_page(int);
int oldmode,screen,page,color;
int ty;
float y;
FILE *fopen(),*fp,*fp1;

float rm[4],rou[4];/*    These are for sheet cross section s  */
int comp();
int comp1();
float linl6(),linl7(),linl8(),lind6(),lind7(),lind8();
float linco1(),linco2(),linco3();

float mater[4][2]={
                  {60.70,490.8},
                  {48.4,360.6},
                  {30.2,184.2},
                  {18.1,84.5}
                 };
int shj,shj1;

float table1[10][2]={
                    {103.0,20.7},
                    {129.0,21.5},
                    {144.0,24.1},
                    {162.0,27.0},
                    {183.0,30.6},
                    {191.0,31.8},
                    {205.0,34.2},
                    {216.0,36.1},
```

```
                    {234.0,39.0},
                    {252.0,42.0}
                  };
int waj,waj1;

int wales();



main()
{
printf("\n\n");
printf(" The following consultation is for the design of sheet \n");
printf(" pile structures.\n");
printf(" Please answer the following questions.\n\n");
printf(" Is this    1.Cantilevered sheet pile;       or\n");
printf("            2. Anchored sheet pile.\n");
printf(" If the height of earth to be retained is  greater  than 10 ft,\n");
printf(" you should choose anchored sheet pile, otherwise the system\n");
printf(" will change it automatically.\n>>");
scanf("%d",&aa1);
printf(" \n");
printf(" 1. Sand;\n");
printf(" 2. Cohesive soil;\n");
printf(" 3. Sand backfill on cohesive soil.\n\n");
printf(" From the given soil types, enter the number corresponding !\n");
printf(" to the type of insitu soil.\n>>");
scanf("%d",&aaa);
if(aaa==1) goto w1;
if(aaa==2) goto w2;
if(aaa==3) goto w3;
   /* The following questions are for sand */
w1:  fp=fopen("heitype","r");
   fscanf(fp,"%f %d",&hhwall,&ty);
   fclose(fp);
   hwall=1*hhwall;
   if(fp==NULL)
   {
   printf(" What is the height of the wall (in ft) ? \n>>");
   scanf("%f",&hwall);
   }

   if(hwall >= 10.0) aa1=2;

   printf(" What is the height of the water table (in ft) ? \n>>");
   scanf("%f",&hwater);
printf(" What is the density of the soil above the water table (in kcf) ?\n>>");
scanf("%f",&gamau);

gamal=gamau-0.0624;
printf(" What is the internal friction angle of the soil (in degree ) ?\n>>");
scanf("%d",&faiu);
printf(" What is the frictional angle between soil and wall(in degree) ?\n");
printf(" This value could be 0.6 -- 0.8 of the internal friction angle.\n>>");
```

```
scanf("%d",&frictionwall);
if(aa1==2)
{
printf(" What is the height of the anchor (in ft) ?\n>>");
scanf("%f",&hanchor);
printf(" What is the surcharge (in ksf) ?\n>>");
scanf("%f",&surcharge);
}
clear_screen();
input();
check();
if(ch==1) goto w1;
/*********************************************/
/** The following is for sand **/
kat1=(sin(3.14159/2+faiu*3.14159/180.0));
kat1=pow(kat1,2.0);
kat2=sin(3.14159*(faiu+frictionwall)/180.0)*sin(faiu*3.14159/180.0);
kat2=kat2/(sin(3.14159/2-frictionwall*3.14159/180.0));
kat2=kat2/sin(3.14159/2);
kat2=sqrt(kat2)+1;
kat2=pow(kat2,2.0);
kat2=kat2*sin(3.14159/2-frictionwall*3.14159/180.0);
ka=kat1/kat2;
kpt1=(sin(3.14159/2-faiu*3.14159/180.0));
kpt1=pow(kpt1,2.0);
kpt2=sin(3.14159*(faiu+frictionwall)/180.0)*sin(3.14159*faiu/180.0);
kpt2=kpt2/sin(3.14159*0.5+frictionwall*3.14159/180.0);
kpt2=sqrt(kpt2);
kpt2=1-kpt2;
kpt2=pow(kpt2,2.0)*sin(3.14159*0.5+3.14159*frictionwall/180.0);
kp=kpt1/kpt2;
k=kp-ka;
pa=((hwall-hwater)*gamau+hwater*gamal)*ka;
if(aa1==2)
pa=((hwall-hwater)*gamau+hwater*gamal)*ka+surcharge*ka;
c=k*gamal;
a=pa/c;
if(aa1==2)
pa=pa-surcharge*ka;
p1=gamau*(hwall-hwater)*ka;
ra=p1*(hwall-hwater)/2.0+(p1+pa)/2*hwater+a/2*pa;
if(aa1==2)
ra=p1*(hwall-hwater)/2.0+(p1+pa)/2.0*hwater+a/2.0*pa+surcharge*hwall*ka;
a1=(hwall-hwater)/3.0+hwater+a;
a2=hwater/3.0*(2*p1+pa)/(p1+pa)+a;
a3=a*2/3;
ybar=(hwall-hwater)*p1/2.0*a1+(p1+pa)/2.0*hwater*a2+pa*a/2.0*a3;
if(aa1==2)
ybar=(hwall-hwater)*p1/2.0*a1+(p1+pa)/2.0*hwater*a2+pa*a/2.0*a3+
surcharge*hwall*hwall/2.0*ka;
ybar=ybar/ra;
if(aa1==2) goto a1;
pp=(gamau*(hwall-hwater)+(hwater+a)*gamal)*kp-gamal*a*ka;
c1=1.0;
```

```
c2=pp/c;
c3=-8*ra/c;
c4=-6*ra/pow(c,2.0)*(2*ybar*c+pp);
c5=-(6*ra*ybar*pp+4*pow(ra,2.0))/pow(c,2.0);
y2=hwall*1.0;
    printf(" Please wait ...\n");
    y=root4(y2);
    shdepth=y+a;
    goto a2;
a1: {;}
    ybar=hanchor-(ybar-a);
    c1=2.0;
c2=3.0*(hanchor+a);
c3=0.0;
c4=-6*ra*ybar/c;
y2=hwall*1.0;
printf(" Please wait ...\n");
y=root3(y2);
shdepth=y+a;
    rp=c*y*y/2.0;
    par=ra-rp;

a2: {;}

    moment();
    output();

    comp1();
    if(aa1==2)
    wales();
goto ed;
/** The above is for sand **/
/** The following is for cohesive soil **/
w2: fp=fopen("heitype","r");
    fscanf(fp,"%f %d",&hwall,&ty);
    fclose(fp);
    if(fp==NULL)
    {
    printf(" What is the height of the wall (in ft)? \n>>");
    scanf("%f",&hwall);
    }
    if(hwall >= 10.0) aa1=2;

    printf(" What is the height of the water table (in ft)? \n>>");
    scanf("%f",&hwater);
    printf(" What is the density of the soil above water table (in kcf) ?\n>>");
    scanf("%f",&gamau);

    gamal=gamau-0.0624;
    printf(" What is the shear strength of the soil (in ksf) ?\n>>");
    scanf("%f",&cu);

if(aa1==2)
{
```

```
printf(" What is the height of the anchor (in ft) ?\n>>");
scanf("%f",&hanchor);
printf(" What is the surcharge (in ksf) ?\n>>");
scanf("%f",&surcharge);
}
clear_screen();
input();
check();
if(ch==1) goto w2;
/*****************************************************/
pa=((hwall-hwater)*gamau+hwater*gamal);
if(aa1==2)
pa=((hwall-hwater)*gamau+hwater*gamal)+surcharge;
ppa=pa;        /*** This is the effective pressure at dredge line ***/
p1=gamau*(hwall-hwater);
ra=p1*(hwall-hwater)/2.0+(p1+pa)/2.0*hwater;
if(aa1==2)
ra=p1*(hwall-hwater)/2.0+(p1+pa-surcharge)/2.0*hwater+surcharge*hwall;
a1=(hwall-hwater)/3.0+hwater;
a2=hwater/3.0*(2*p1+pa)/(p1+pa);
ybar=(hwall-hwater)*p1/2.0*a1+(p1+pa)/2.0*hwater*a2;
if(aa1==2)
ybar=ybar+surcharge*hwall*hwall/2.0;
ybar=ybar/ra;
goto bb1;
   c1=4*cu-pa;
   c2=-2*ra;
   c3=-ra*(12*cu*ybar+ra)/(2*cu+pa);
   y=root2(c1,c2,c3);
   shdepth=y;
   goto bb2;
bb1: c1=1.0;
   c2=2.0*hanchor;
   c3=-2.0*ybar*ra/(4.0*cu-pa);
   y=root2(c1,c2,c3);
   rp=y*(4.0*cu-pa);
   shdepth=y;
   par=ra-rp;

bb2: {;}

   moment();
   printf(" Press any key to continue.\n");
   while(kbhit()==0) {;}
   output();

   comp2();
   if(aa1==2)
   wales();
goto ed;
/* The following is for sandlike backfill and cohesive base */
w3: fp=fopen("heitype","r");
   fscanf(fp,"%f %d",&hwall,&ty);
   fclose(fp);
```

```
    if(fp==NULL)
    {
    printf(" What is the height of the wall (in ft) ?\n>>");
    scanf("%f",&hwall);
    }
    printf(" What is the height of the water table (in ft) ?\n>>");
    scanf("%f",&hwater);
    printf(" What is the density of soil above water table(in kcf) ?\n>>");
    scanf("%f",&gamau);

    gamal=gamau-0.0624;
    printf(" What is the shear strength of the soil below the dredge line (in ksi) ?\n>>");
    scanf("%f",&cu);
    printf(" What is the internal friction angle (in degree) ?\n>>");
    scanf("%d",&faiu);
    printf(" What is the friction angle between wall and backfill (in degree) ?\n");
    printf(" Usually this angle is 0.6 - 0.8 of the internal friction angle.\n>>");
    scanf("%d",&frictionwall);
 if(aa1==2)
  {
 printf(" What is height of the anchor (in ft) ?\n>>");
 scanf("%f",&hanchor);
 printf(" What is the surcharge (in ksf) ?\n>>");
 scanf("%f",&surcharge);
 }
 clear_screen();
 input();
 check();
 if(ch==1) goto w3;
 /*****************************************************/
 kat1=(sin(3.14159/2+faiu*3.14159/180.0));
 kat1=pow(kat1,2.0);
 kat2=sin(3.14159*(faiu+frictionwall)/180.0)*sin(faiu*3.14159/180.0);
 kat2=kat2/(sin(3.14159/2-frictionwall*3.14159/180.0));
 kat2=sqrt(kat2)+1;
 kat2=pow(kat2,2.0);
 kat2=kat2*sin(3.14159/2-frictionwall*3.14159/180.0);
 ka=kat1/kat2;
 kpt1=(sin(3.14159/2-faiu*3.14159/180.0));
 kpt1=pow(kpt1,2.0);
 kpt2=sin(3.14159*(faiu+frictionwall)/180.0)*sin(3.14159*faiu/180.0);
 kpt2=kpt2/sin(3.14159*0.5+frictionwall*3.14159/180.0);
 kpt2=sqrt(kpt2);
 kpt2=1-kpt2;
 kpt2=pow(kpt2,2.0)*sin(3.14159*0.5+3.14159*frictionwall/180.0);
 kp=kpt1/kpt2;
 p1=gamau*(hwall-hwater)*ka;
 pa=gamal*hwater*ka+p1;
 if(aa1==2)
 pa=pa+surcharge*hwall*ka;
 ppa=pa;          /*** This is the effective pressure at dredge line ***/
 ra=p1*(hwall-hwater)/2.0+(p1+pa)/2.0*hwater;
 if(aa1==2)
 {
```

```
pa=pa-surcharge*hwall*ka;
ra=p1*(hwall-hwater)/2.0+(p1+pa)/2.0*hwater+surcharge*hwall*ka;
}
a1=(hwall-hwater)/3.0+hwater;
a2=hwater/3.0*(2*p1+pa)/(p1+pa);
ybar=p1*(hwall-hwater)/2.0*a1+(p1+pa)/2.0*hwater*a2;
if(aa1==2)
ybar=ybar+surcharge*hwall*hwall/2.0*ka;
ybar=ybar/ra;
if(aa1==2) goto cc1;
c1=4*cu-pa/ka;
c2=-2*ra;
c3=-ra*(12*cu*ybar+ra)/(2*cu+pa/ka);
   y=root2(c1,c2,c3);
   shdepth=y;
   goto cc2;
cc1: pa=pa+surcharge*hwall*ka;
   ybar=hanchor-ybar;
   c1=1.0;
   c2=2*hanchor;
   c3=-2*ybar*ra/(4*cu-pa/ka);
   y=root2(c1,c2,c3);
   shdepth=y;
   rp=y*(4*cu-pa/ka);
   par=ra-rp;

cc2: {;}

   moment();
   printf(" Press any key to continue.\n");
   while(kbhit()==0) {;}
   output();

   comp2();
   if(aa1==2)
   wales();

ed: printf("\n The detail is in file sheet.out.\n");
}


/*******************************************************/
float root4(s)
float s;
{
   float a,b1,a0;
   float fa,b,fb1,h=0.01;
   float e1=0.01;
   float f0,rt,bb;
   a=s/4.0;
   bb=s*1.0;
   fa=gx(a);
g10:b=a+h;
   b1=b;
```

```
   fb1=gx(b1);
   if(abs(fa)<=e1) goto g16;
   if(abs(fb1)<=e1) goto g11;
   if(fa*fb1>=0.0) goto g11;
 g12:a0=(a+b)/2.0;
   f0=gx(a0);
   if(abs(f0)<=e1) goto g13;
   if(abs(b-a0)<=e1) goto g13;
   if(fa*f0>=0.0) goto g14;
   b=a0;
   goto g12;
 g14:fa=f0;
   a=a0;
   goto g12;
 g16:a0=a;
 g13:rt=a0;
 g11:if(b1>bb) goto g15;
   a=b1;
   fa=fb1;
   goto g10;
 g15:return(rt);
}
/********************************************************/
float gx(ss)
float ss;
{
float fff;
  fff=c1*pow(ss,4.0)+c2*pow(ss,3.0)+c3*pow(ss,2.0)+c4*ss+c5;
return(fff);
}
/*********************************************************/
float root2(s1,s2,s3)
float s1,s2,s3;
{
float a,b,c,x,x1,x2,x3;
b=sqrt(s2*s2-4.0*s1*s3);
a=s2;
c=2*s1;
x1=(-a-b)/c;
x2=(-a+b)/c;
if(x1*x2<0)
x=max(x1,x2);
else
  {
  x=min(x1,x2);
  printf(" The solution is not unique.\n");
  }
return(x);
}
/**********************************************/
float root3(s)
float s;
{
  float a,b1,a0;
```

```
    float fa,b,fb1,h=0.01;
    float e1=0.01;
    float f0,rt,bb;
    a=s/5.0;
    bb=s*1.0;
    fa=gx3(a);
 g10:b=a+h;
    b1=b;
    fb1=gx3(b1);
    if(abs(fa)<=e1) goto g16;
    if(abs(fb1)<=e1) goto g11;
    if(fa*fb1>=0.0) goto g11;
 g12:a0=(a+b)/2.0;
    f0=gx3(a0);
    if(abs(f0)<=e1) goto g13;
    if(abs(b-a0)<=e1) goto g13;
    if(fa*f0>=0.0) goto g14;
    b=a0;
    goto g12;
 g14:fa=f0;
   a=a0;
   goto g12;
 g16:a0=a;
 g13:rt=a0;
 g11:if(b1>bb) goto g15;
   a=b1;
   fa=fb1;
   goto g10;
 g15:return(rt);
 }
/*****************************************************/
float gx3(ss)
float ss;
{
float fff;
   fff=c1*pow(ss,3.0)+c2*pow(ss,2.0)+c3*ss+c4;
return(fff);
}
/*****************************************************/

void check()
{
ch=0;

if(hwall==0.0 || (hwall<=6.0 || hwall>=35.0))
{
printf(" \n\n INPUT ERROR:\n");
printf(" The height of this type of wall should not be less than 4 ft or\n");
printf(" greater than 35 ft\n");
printf(" Please input again.\n");
ch=1;
}

if(aa1==2)
```

```
{
if(hwater==0.0 || hwater>=hanchor)
{
printf("\n\n INPUT ERROR:\n");
printf(" Usually, the water table is below the anchor to avoid erosion\n");
printf(" Please input again.\n");
ch=1;
}
}

if(aa1==2)
{
if(hanchor==0.0 || hanchor>=hwall)
{
printf("\n\n FATAL ERROR:\n");
printf(" The height of anchor is greater than the height of wall.\n");
printf(" Please input again.\n");
ch=1;
}
}

if(gamau==0 || (gamau<=0.050 || gamau>=0.200))
{
printf("\n\n INPUT ERROR:\n");
printf(" The density of soil is too small or large\n");
printf(" Please input again.\n");
ch=1;
}

if(gamal==0 || (gamal<=0.020 || gamal>gamau))
{
printf("\n\n INPUT ERROR:\n");
printf(" The density of soil below the water table is usually smaller than \n");
printf(" that of soil above the water table.\n");
printf(" Please input again.\n\n");
ch=1;
}

if(aaa!=2)
{
if(faiu==0 || (faiu>=40 || faiu<=18))
{
printf("\n\n INPUT ERROR:\n");
printf(" The internal angle of friction of soil is too small or large.\n");
printf(" Please input again.\n");
ch=1;
}
}

if(aaa!=2)
{
if(frictionwall==0 || ( frictionwall<=10 || frictionwall>=faiu))
{
printf(" \n\n INPUT ERROR:\n");
```

```
printf(" The friction between wall and soil is too small or large\n");
printf(" Please input again.\n");
ch=1;
}
}

if(aa1==2)
{
if(surcharge<0.0 || surcharge>=3.0)
{
printf("\n\n INPUT ERROR:\n");
printf(" The surcharge for design is too large\n");
printf(" Please input again.\n");
ch=1;
}
}
printf("\n");
}

/**********************************************/
int get_mode()
{
union REGS intregs,outregs;
intregs.h.ah=0x0f;
int86(0x10,&intregs,&outregs);
return(outregs.h.al);
}
/**********************************************/
void setscreen(n)
int n;
{
union REGS intregs,outregs;
screen=n;
intregs.h.ah=0;
intregs.h.al=n;
int86(0x10,&intregs,&outregs);
}
/*****************************************/


void set_page(p)
int p;
{
union REGS intregs,outregs;
extern int page;
page=p;
intregs.h.ah=5;
intregs.h.al=p;
int86(0x10,&intregs,&outregs);
}
/*****************************************/

void set_color(f,b)
int f,b;
```

```
{
color=(f & 143)+((b<<4) & 112);
}
/*****************************************/

void clear_screen()
{
union REGS inregs,outregs;
inregs.h.ah=6;
inregs.h.al=0;
inregs.h.bh=7;
inregs.h.ch=0;
inregs.h.cl=0;
inregs.h.dh=24;
inregs.h.dl=79;
int86(0x10,&inregs,&outregs);
}
/*****************************************/
void input()
{
clear_screen();
printf("\n\n");
printf(" *********************\n");
printf(" *   INPUT    *\n");
printf(" *********************\n\n");
printf(" The height of the wall:            %.2f  ft\n",hwall);
printf(" The height of the water table:       %.2f  ft\n",hwater);
printf(" The density of soil above water table:  %.2f  kcf\n",gamau);
/*
printf(" The density of soil below water table:  %.2f  kcf\n",gamal);
*/
if(aaa==1)
  {
printf(" The internal friction angle of soil:   %d   degree\n",faiu);
printf(" The friction angle bewteen wall and soil: %d   degree\n",frictionwall);
  }
if(aaa==2)
  {
printf(" The shear strength of soil:          %.2f  ksf\n",cu);
  }
if(aaa==3)
  {
printf(" The internal friction angle of soil:   %d   degree\n",faiu);
printf(" The friction angle between wall and soil: %d   degree\n",frictionwall);
printf(" The shear strength below dredge line:   %.2f  ksf\n",cu);
  }
if(aa1==2)
  {
printf(" The height of the anchor:            %.2f  ft\n",hanchor);
printf(" The surcharge for design:            %.2f  ksf\n",surcharge);
  }
printf("\n");
}
/**********************************************************/
```

```
int moment()
{
int i,j,k;
float mmm,mm1,xx,yy,dxx;
float mp,m0;
yy=hwall-hwater;
dxx=(hwall-yy)/200.0;
mm2=0.0;
for(i=0;i<=200;i++)
  {
  if(aaa==2)
  {
  ka=1.0;
  kp=1.0;
  }
  xx=yy+i*dxx;
  if(aa1==2)
  {
  mp=-par*(xx-(hwall-hanchor));
  m0=surcharge*xx*xx/2.0*ka;
  }
  else
  {
  mp=0.0;
  m0=0.0;
  }
  mm1=m0+mp+0.5*yy*gamau*yy*ka*(xx-2.0*yy/3.0)+gamau*yy*ka*(xx-yy)*(xx-
yy)/2.0
     +0.5*(xx-yy)*gamal*(xx-yy)*ka*(xx-yy)/3.0;
  mmm=(float)max(fabs(mm1),fabs(mm2));
  if(mmm>mm2)
  {
  mpos=xx;
  mm2=mmm;
  }
  }
  return;
}
/*********************************************************/
void output()
{
FILE *fout,*fopen();
fout=fopen("sheet.out","w");

printf("\n");
printf(" ****************\n");
printf(" * OUTPUT:   *\n");
printf(" ****************\n\n");
printf(" The minimum depth under dredge line:  %.2f  ft\n",shdepth);
printf(" The maximum moment:                   %.2f  kips-ft/ft\n",mm2);
printf(" The maximum moment occures at:        %.2f  ft\n",mpos);
if(aa1==2)
printf(" The pull force at anchored rod:      %.2f  kips/ft\n",par);
if(aaa!=2)
```

```
  {
printf(" Active soil pressure coefficient:    %.2f  \n",ka);
printf(" Passive soil pressure coefficient:   %.2f  \n",kp);
  }
printf(" The effctive pressure at dredge line: %.2f  ksf/ft\n",pa);
printf(" The resultant in active zone:        %.2f  kips/ft\n",ra);
if(aaa==1)
  {
printf(" The pivot is below dredge line:      %.2f  ft\n",a);
printf(" The arm of resultant from pivot:     %.2f  ft\n",ybar);
  }
if(aa1==2)
printf(" The resultant in passive zone:       %.2f  kips/ft\n",rp);

fprintf(fout," ********************\n");
fprintf(fout," *     OUTPUT       *\n");
fprintf(fout," ********************\n");
fprintf(fout," The minimum depth under dredge line: %.2f  ft\n",shdepth);
fprintf(fout," The maximum moment:                  %.2f  kips-ft/ft\n",mm2);
fprintf(fout," The maximum moment occures at:      %.2f  ft\n",mpos);
if(aa1==2)
fprintf(fout," The pull force at anchor rod:       %.2f  kips/ft\n",par);
fprintf(fout," -------------------------------------------------\n\n");
fprintf(fout," The height of wall:                 %.2f  ft\n",hwall);
fprintf(fout," The height of water table:          %.2f  ft\n",hwater);
fprintf(fout," The density of soil:                %.2f  kcf\n",gamau);
if(aaa==1)
  {
fprintf(fout," The internal friction angle of soil: %d   degree\n",faiu);
fprintf(fout," The friction angle between\n");
fprintf(fout," soil and wall:                      %d   degree\n",frictionwall);
  }
if(aaa==2)
fprintf(fout," The shear strength of soil:         %.2f   ksf\n",cu);

if(aaa==3)
  {
fprintf(fout," The internal friction angle:        %d    degree\n",faiu);
fprintf(fout," The friction angle between\n");
fprintf(fout," soil and wall:                      %d    degree\n",frictionwall);
fprintf(fout," The shear strength of soil:         %.2f   ksf\n",cu);
  }
if(aa1==2)
  {
fprintf(fout," The height of anchor:               %.2f   ft\n",hanchor);
fprintf(fout," The surcharge:                      %.2f  ksf\n",surcharge);
  }

fprintf(fout,"\n The following is the cross sections of sheet pile the system has:\n\n");
fprintf(fout," 1. PZ40  Sx=%.2f     Ix=%.2f\n",mater[0][0],mater[0][1]);
fprintf(fout," 2. PZ35  Sx=%.2f     Ix=%.2f\n",mater[1][0],mater[1][1]);
fprintf(fout," 3. PZ27  Sx=%.2f     Ix=%.2f\n",mater[2][0],mater[2][1]);
fprintf(fout," 4. PZ22  Sx=%.2f     Ix=%.2f\n",mater[3][0],mater[3][1]);
if(shj1!=0)
```

```
fprintf(fout,"\n Your choice is No. %d\n\n",shj);
else
fprintf(fout," No cross section is OK for your sheet pile.\n");

if(aa1==2)
  {
fprintf(fout," The following is the list of cross sections of wales in the system:\n\n");
fprintf(fout," 1. C10x30    Sx=%.2f   Ix=%.2f\n",table1[0][1],table1[0][0]);
fprintf(fout," 2. C12x20.7  Sx=%.2f   Ix=%.2f\n",table1[1][1],table1[1][0]);
fprintf(fout," 3. C12x25    Sx=%.2f   Ix=%.2f\n",table1[2][1],table1[2][0]);
fprintf(fout," 4. C12x30    Sx=%.2f   Ix=%.2f\n",table1[3][1],table1[3][0]);
fprintf(fout," 5. MC12x30.9 Sx=%.2f   Ix=%.2f\n",table1[4][1],table1[4][0]);
fprintf(fout," 6. MC12x32.9 Sx=%.2f   Ix=%.2f\n",table1[5][1],table1[5][0]);
fprintf(fout," 7. MC12x37   Sx=%.2f   Ix=%.2f\n",table1[6][1],table1[6][0]);
fprintf(fout," 8. MC12x35   Sx=%.2f   Ix=%.2f\n",table1[7][1],table1[7][0]);
fprintf(fout," 9. MC12x40   Sx=%.2f   Ix=%.2f\n",table1[8][1],table1[8][0]);
fprintf(fout," 10. MC12x45  Sx=%.2f   Ix=%.2f\n",table1[9][1],table1[9][0]);

if(waj1!=0)
fprintf(fout,"\n Your choice is No. %d\n\n",waj);
else
fprintf(fout," No cross section is OK for your wale.\n");
  }
}
/****************************************************************/
int keykb()
{
union REGS inregs,outregs;
inregs.h.ah=o;
return (int86(0x16,&inregs,&outregs));
}
/****************************************************************/
int getkey()
{
char al,ah,c;
int key;
key=keykb();
ah=(key >> 8) & 0xff;
al=key & 0x00ff;
if(al==0)
c=ah;
else
c=al;
return(c);
}
/*******************************************/
int comp1()
{
int i,j=0,k;
int ci;
float rmy,m[4],mcom=100000.0;
/*******************************************/
float alfa,acom=10000000.0;
static float acoma[3]={0.6,0.7,0.8};
```

```
/*******************************************/

float fshcan1,fshcan2,safe;


float eee=29000000.0,sum;
float mp,fy;
/*
static float mater[4][2]={{60.70,490.8},{48.4,360.6},{30.2,184.2},{18.1,84.5}};
*/
printf("\n\n In this system, there are four cross sections for your sheetpile\n\n");
printf(" 1. PZ40  Sx=%.2f      Ix=%.2f\n",mater[0][0],mater[0][1]);
printf(" 2. PZ35  Sx=%.2f      Ix=%.2f\n",mater[1][0],mater[1][1]);
printf(" 3. PZ27  Sx=%.2f      Ix=%.2f\n",mater[2][0],mater[2][1]);
printf(" 4. PZ22  Sx=%.2f      Ix=%.2f\n",mater[3][0],mater[3][1]);


printf(" Please input the yeilding stress!(in ksi)\n>>");
scanf("%f",&fy);

  if(aa1==1)
   {
   for(i=3;i>=0;i--)
        {
        fshcan2=0.65*fy;
        fshcan1=mm2*12.0/mater[i][0];
        if(fshcan1<=fshcan2)
          {
          printf(" Section No. %d could be your choice .\n",i+1);
          fshcan1=fshcan2/(fshcan1);
          printf(" The safty factor is %.2f  .\n",fshcan1);
          shj=i+1;
   j=1;
          goto ca;
          }
        }
   if(j==0)
        {
        shj1=0;
  printf(" No cross  section  listed  is OK  for  your  sheetpile.\n");
        printf(" Probably, you have to increase the yielding stress or\n");
        printf(" use anchored sheetpile.\n");
        goto ca;
        }
   }

sum=shdepth*1.45+hwall;

/******* Safety factor is 1.45 ************************/

for(i=0;i<=3;i++)
{
rou[i]=log10(pow(sum,4.0)/(eee*mater[i][1]));
mp=fy*0.65/12.0*(mater[i][0]);
```

```
rm[i]=mp/mm2;
}
for(i=0;i<=3;i++)
printf(" ratio of moment (in Y axis)=%.2f, log(H/EI)(in X axis)=%.2f\n",rm[i],rou[i]);

/****************************************/
alfa=hwall/(hwall+shdepth);
for(i=0;i<=2;i++)
  {
  if(abs(alfa-acoma[i])<acom)
    {
    acom=abs(alfa-acoma[i]);
    k=i;
    }
  }
acom=acoma[k];
printf(" \n The sand is dense or loose (d/l) ?\n>>");
ci=getkey();

if(ci=='l')
 {
 for(i=0;i<=3;i++)
 {
 if(rou[i]>(-2.0) || rou[i]<=(-3.5))
  {
  m[i]=0.0;
  goto ed1;
  }
  switch(k)
    {
    case 0:
    rmy=linl6(rou[i]);
    break;
    case 1:
    rmy=linl7(rou[i]);
    break;
    case 2:
    rmy=linl8(rou[i]);
    break;
    default:
     {;}
    break;
    }

   if(rmy<=rm[i])
   m[i]=rm[i];
   else
   m[i]=0.0;
  ed1:{;}
  }
  for(i=0;i<=3;i++)
    {
    if(m[i]!=0.0)
```

```
    {
    if(m[i]<=mcom)
     {
     j=i+1;
     mcom=m[i];
     }
    }
  else {;}
  }
 if(j!=0)
  {
  shj=j;
  printf("\n\n The most appropriate cross section is No. %d  with Sx=%.2f
 Ix=%.2f\n",j,mater[j-1][0],mater[j-1][1]);
  safe=mcom/rmy;
  printf(" The safty factor is %.2f\n",safe);
  }
 else
  {
  printf("\n\n No cross section listed is OK for your sheetpile !\n");
  shj1=0;
  }
 }


if(ci=='d')
 {
 for(i=0;i<=3;i++)
  {
if(rou[i]>(-1.75) || rou[i]<=(-4.0))
   {
 m[i]=0.0;
 goto ed2;
   }
 switch(k)
   {
   case 0:
   rmy=lind6(rou[i]);
   break;
   case 1:
   rmy=lind7(rou[i]);
   break;
   case 2:
   rmy=lind8(rou[i]);
   break;
   default:
   {;}
   break;
   }
/*
 rmy=lin2(rou[i]);
*/

 if(rmy<=rm[i])
```

```
 m[i]=rm[i];
 else
 m[i]=0.0;
ed2:{;}
  }
 for(i=0;i<=3;i++)
 {
 if(m[i]!=0.0)
  {
  if(m[i]<=mcom)
    {
   j=i+1;
   mcom=m[i];
    }
  }
  else {;}
 }
 if(j!=0)
 {
 shj=j;
 printf("\n\n The most appropriate cross section is No. %d with Sx=%.2f
Ix=%.2f\n",j,mater[j-1][0],mater[j-1][1]);
 safe=mcom/rmy;
 printf(" The factor of safety  is %.2f\n",safe);
 }
 else
  {
 printf("\n\n No cross section listed is OK for your sheetpile !\n");
 shj1=0;
  }
 }

ca:{;}
return;
}
/***********************************************/
float linl7(s)
float s;

{
float rmy;
if(s>=(-3.5) && s<=(-3.25))
rmy=(1.0-0.75)/(-3.5-(-3.25))*(s-(-3.25))+0.75;

if(s>-3.25 && s<=-3.0)
rmy=(0.75-0.6)/(-3.25-(-3.0))*(s-(-3.00))+0.60;

if(s>-3.0 && s<=(-2.75))
rmy=(0.60-0.45)/(-3.00-(-2.75))*(s-(-2.75))+0.45;

if(s>-2.75 && s<=(-2.50))
rmy=(0.45-0.36)/(-2.75-(-2.50))*(s-(-2.50))+0.36;

if(s>-2.5 && s<=-2.25)
```

```
rmy=(0.36-0.33)/(-2.50-(-2.25))*(s-(-2.25))+0.33;

if(s>-2.25 && s<=2.0)
rmy=(0.33-0.30)/(-2.25-(-2.00))*(s-(-2.00))+0.30;

return(rmy);
}

/***************************************/
float lind7(s)
float s;
{
float rmy;

if(s>-4.0 && s<=-3.75)
rmy=(0.8-0.68)/(-4.00-(-3.75))*(s-(-3.75))+0.68;

if(s>-3.75 && s<=-3.5)
rmy=(0.68-0.58)/(-3.75-(-3.5))*(s-(-3.50))+0.58;

if(s>-3.5 && s<=-3.25)
rmy=(0.58-0.50)/(-3.5-(-3.25))*(s-(-3.25))+0.50;

if(s>-3.25 && s<=-3.00)
rmy=(0.50-0.43)/(-3.25-(-3.00))*(s-(-3.00))+0.43;

if(s>-3.00 && s<=-2.75)
rmy=(0.43-0.36)/(-3.00-(-2.75))*(s-(-2.75))+0.36;

if(s>-2.75 && s<=-2.50)
rmy=(0.36-0.32)/(-2.75-(-2.50))*(s-(-2.50))+0.32;

if(s>-2.50 && s<=-2.25)
rmy=(0.32-0.28)/(-2.50-(-2.25))*(s-(-2.25))+0.28;

if(s>-2.25 && s<=-2.00)
rmy=(0.28-0.25)/(-2.25-(-2.00))*(s-(-2.00))+0.25;

return(rmy);
}
/**********************************************/
float linl8(s)
float s;

{
float rmy;
if(s>=(-3.5) && s<=(-3.25))
rmy=(1.10-0.85)/(-3.5-(-3.25))*(s-(-3.25))+0.85;

if(s>-3.25 && s<=-3.0)
rmy=(0.85-0.65)/(-3.25-(-3.0))*(s-(-3.00))+0.65;

if(s>-3.0 && s<=(-2.75))
rmy=(0.65-0.50)/(-3.00-(-2.75))*(s-(-2.75))+0.50;
```

```
if(s>-2.75 && s<=(-2.50))
rmy=(0.50-0.41)/(-2.75-(-2.50))*(s-(-2.50))+0.41;

if(s>-2.5 && s<=-2.25)
rmy=(0.41-0.37)/(-2.50-(-2.25))*(s-(-2.25))+0.37;

if(s>-2.25 && s<=2.0)
rmy=(0.37-0.34)/(-2.25-(-2.00))*(s-(-2.00))+0.34;

return(rmy);
}
/************************************************/
float linl6(s)
float s;

{
float rmy;
if(s>=(-3.5) && s<=(-3.25))
rmy=(0.89-0.70)/(-3.5-(-3.25))*(s-(-3.25))+0.70;

if(s>-3.25 && s<=-3.0)
rmy=(0.70-0.54)/(-3.25-(-3.0))*(s-(-3.00))+0.54;

if(s>-3.0 && s<=(-2.75))
rmy=(0.54-0.41)/(-3.00-(-2.75))*(s-(-2.75))+0.41;

if(s>-2.75 && s<=(-2.50))
rmy=(0.41-0.32)/(-2.75-(-2.50))*(s-(-2.50))+0.32;

if(s>-2.5 && s<=-2.25)
rmy=(0.32-0.30)/(-2.50-(-2.25))*(s-(-2.25))+0.30;

if(s>-2.25 && s<=2.0)
rmy=(0.30-0.28)/(-2.25-(-2.00))*(s-(-2.00))+0.28;

return(rmy);
}
/**************************************/
float lind8(s)
float s;
{
float rmy;

if(s>-4.0 && s<=-3.75)
rmy=(0.90-0.74)/(-4.00-(-3.75))*(s-(-3.75))+0.74;

if(s>-3.75 && s<=-3.5)
rmy=(0.74-0.63)/(-3.75-(-3.5))*(s-(-3.50))+0.63;

if(s>-3.5 && s<=-3.25)
rmy=(0.63-0.55)/(-3.5-(-3.25))*(s-(-3.25))+0.55;

if(s>-3.25 && s<=-3.00)
```

```
rmy=(0.55-0.48)/(-3.25-(-3.00))*(s-(-3.00))+0.48;

if(s>-3.00 && s<=-2.75)
rmy=(0.48-0.43)/(-3.00-(-2.75))*(s-(-2.75))+0.43;

if(s>-2.75 && s<=-2.50)
rmy=(0.43-0.37)/(-2.75-(-2.50))*(s-(-2.50))+0.37;

if(s>-2.50 && s<=-2.25)
rmy=(0.37-0.325)/(-2.50-(-2.25))*(s-(-2.25))+0.325;

if(s>-2.25 && s<=-2.00)
rmy=(0.325-0.30)/(-2.25-(-2.00))*(s-(-2.00))+0.30;

return(rmy);
}
/****************************************/
float lind6(s)
float s;
{
float rmy;

if(s>-4.0 && s<=-3.75)
rmy=(0.70-0.60)/(-4.00-(-3.75))*(s-(-3.75))+0.60;

if(s>-3.75 && s<=-3.5)
rmy=(0.60-0.50)/(-3.75-(-3.5))*(s-(-3.50))+0.50;

if(s>-3.5 && s<=-3.25)
rmy=(0.50-0.42)/(-3.5-(-3.25))*(s-(-3.25))+0.42;

if(s>-3.25 && s<=-3.00)
rmy=(0.42-0.35)/(-3.25-(-3.00))*(s-(-3.00))+0.35;

if(s>-3.00 && s<=-2.75)
rmy=(0.35-0.30)/(-3.00-(-2.75))*(s-(-2.75))+0.30;

if(s>-2.75 && s<=-2.50)
rmy=(0.30-0.28)/(-2.75-(-2.50))*(s-(-2.50))+0.28;

if(s>-2.50 && s<=-2.25)
rmy=(0.28-0.25)/(-2.50-(-2.25))*(s-(-2.25))+0.25;

if(s>-2.25 && s<=-2.00)
rmy=(0.25-0.20)/(-2.25-(-2.00))*(s-(-2.00))+0.20;

return(rmy);
}
/*******************************************************/
float linco1(s)
float s;

{
float rmy;
```

```
if(s>=0.3 && s<=0.43)
rmy=(0.86-1.20)/(0.43-0.30)*(s-0.30)+1.20;
if(s>0.43 && s<=0.50)
rmy=(0.78-0.86)/(0.50-0.43)*(s-0.43)+0.86;
if(s>0.50 && s<=0.60)
rmy=(0.74-0.78)/(0.60-0.50)*(s-0.50)+0.78;
if(s>0.60 && s<=0.75)
rmy=(0.70-0.74)/(0.75-0.60)*(s-0.60)+0.74;
if(s>0.75 && s<=1.00)
rmy=(0.68-0.70)/(1.00-0.75)*(s-0.75)+0.70;
if(s>1.00 && s<=1.25)
rmy=(0.66-0.68)/(1.25-1.00)*(s-1.00)+0.68;
if(s>1.25 && s<=1.50)
rmy=(0.64-0.66)/(1.50-1.25)*(s-1.25)+0.66;
if(s>1.50 && s<=1.75)
rmy=(0.62-0.64)/(1.75-1.50)*(s-1.50)+0.64;
if(s>1.75 && s<=2.00)
rmy=(0.60-0.62)/(2.00-1.75)*(s-1.75)+0.62;

return(rmy);
}


float linco2(s)
float s;
{
float rmy;

if(s>=0.35 && s<=0.43)
rmy=(0.80-1.00)/(0.43-0.35)*(s-0.35)+1.0;

if(s>0.43 && s<=0.50)
rmy=(0.66-0.80)/(0.50-0.43)*(s-0.43)+0.80;

if(s>0.50 && s<=0.60)
rmy=(0.60-0.66)/(0.60-0.50)*(s-0.50)+0.66;

if(s>0.60 && s<=0.75)
rmy=(0.56-0.60)/(0.75-0.60)*(s-0.60)+0.60;

if(s>0.75 && s<=1.00)
rmy=(0.54-0.56)/(1.00-0.75)*(s-0.75)+0.56;

if(s>1.00 && s<=1.25)
rmy=(0.52-0.54)/(1.25-1.00)*(s-1.00)+0.54;

if (s>1.25 && s<=1.50)
rmy=(0.50-0.52)/(1.50-1.25)*(s-1.25)+0.52;

if(s>1.50 && s<=1.75)
rmy=(0.48-0.50)/(1.75-1.50)*(s-1.50)+0.50;

if(s>1.75 && s<=2.00)
```

```
rmy=(0.47-0.48)/(2.00-1.75)*(s-1.75)+0.48;

return(rmy);
}


float linco3(s)
float s;
{
float rmy;

if(s>=0.40 && s<=0.50)
rmy=(0.69-0.90)/(0.50-0.40)*(s-0.40)+0.90;

if(s> 0.50 && s<=0.60)
rmy=(0.60-0.69)/(0.60-0.50)*(s-0.50)+0.69;

if(s>0.60 && s<=0.75)
rmy=(0.55-0.60)/(0.75-0.60)*(s-0.60)+0.60;

if(s>0.75 && s<=1.00)
rmy=(0.50-0.55)/(1.00-0.75)*(s-0.75)+0.55;

if(s>1.00 && s<=1.25)
rmy=(0.48-0.50)/(1.25-1.00)*(s-1.00)+0.50;

if(s>1.25 && s<=1.50)
rmy=(0.46-0.48)/(1.50-1.25)*(s-1.25)+0.48;

if(s>1.50 && s<=2.00)
rmy=0.46;

return(rmy);
}
/********************************************/
int comp2()
{
int i,j=0,k;
int ci;
float rmy,rmy1,rmy2,rmy3,m[4],stanum;
/********************************************/
float eee=29000000.0,sum;
float mp,fy;
static float mater[4][2]={{60.70,490.8},{48.4,360.6},{30.2,184.2},{18.1,84.5}};

float fshcan1,fshcan2;


printf("\n\n In this system, there are four cross sections for the sheetpile:\n\n");
printf(" 1. PZ40  Sx=%.2f     Ix=%.2f\n",mater[0][0],mater[0][1]);
printf(" 2. PZ35  Sx=%.2f     Ix=%.2f\n",mater[1][0],mater[1][1]);
printf(" 3. PZ27  Sx=%.2f     Ix=%.2f\n",mater[2][0],mater[2][1]);
printf(" 4. PZ22  Sx=%.2f     Ix=%.2f\n",mater[3][0],mater[3][1]);
```

```
printf(" Please input the yeilding stress!(in ksi)\n>>");
scanf("%f",&fy);


  if(aa1==1)
    {
    for(i=3;i>=0;i--)
         {
         fshcan2=0.65*fy;
         fshcan1=mm2*12.0/mater[i][0];
         if(fshcan1<=fshcan2)
           {
           printf(" Section No. %d could be your choice .\n",i+1);
           fshcan1=fshcan2/(fshcan1);
           shj=i+1;
      printf(" The safty factor is %.2f .\n",fshcan1);
           j=1;
           goto ca;
           }
         }
   if(j==0)
        {
        shj1=0;
   printf(" No cross section listed is OK for your sheetpile.\n");
        printf(" Probally, you have to increase the yielding stress or\n");
        printf(" use anchored sheetpile.\n");
        goto ca;
        }
   }


sum=shdepth*1.45+hwall;
/******* Safety factor is 1.45 ************************/
for(i=0;i<=3;i++)
{
rou[i]=log10(pow(sum,4.0)/(eee*mater[i][1]));
mp=fy*0.65/12.0*(mater[i][0]);
rm[i]=mp/mm2;
}
/*
for (i=0;i<=3;i++)
printf(" ratio of moment (in Y axis)=%.2f, log(H/EI)(in X axis)=%.2f\n",rm[i],rou[i]);
*/

/*****************************************/
stanum=1.25*cu/ppa;

/*
printf(" Sn=%.2f\n",stanum);
*/
rmy1=linco1(stanum);
rmy2=linco2(stanum);
rmy3=linco3(stanum);
```

```
for(i=3;i>=0;i--)
  {
  if(rou[i]<=-3.1 || rou[i]>=-2.1)
    rmy=(rmy1-rmy3)/(-3.10-(-2.00))*(rou[i]-(-2.00))+rmy3;
  if(rou[i]>-3.10 && rou[i]<=-2.60)
    rmy=(rmy1-rmy2)/(-3.10-(-2.60))*(rou[i]-(-2.60))+rmy2;
  if(rou[i]>-2.60 && rou[i]<-2.00)
    rmy=(rmy2-rmy3)/(-2.60-(-2.00))*(rou[i]-(-2.00))+rmy3;

    if(rmy<=rm[i])
    {
    printf(" The most appropriate cross section is No. %d       Sx=%.2f
Ix=%.2f\n",i+1,mater[i][0],mater[i][1]);
    shj=i+1;
    j=1;
    goto ec;

    }
    else
    {;}
  }
ec:{;}

  if(j==0)
   {
   shj1=0;
   printf(" No cross section listed is OK for your sheetpile. \n");
   printf(" You should probably increase the yield strees of steel.\n");
   }
ca:{;}
return;
}
/**********************************************/
int wales()
{
int i,j,k=0;
float qqq,mmid,sxc,sx,sx1;
float deflc,defl;
float fy,fb;
/*
static float table1[10][2]={{103.0,20.7},
                            {129.0,21.5},
                            {144.0,24.1},
                            {162.0,27.0},
                            {183.0,30.6},
                            {191.0,31.8},
                            {205.0,34.2},
                            {216.0,36.1},
                            {234.0,39.0},
                            {252.0,42.0}
                           };
*/
```

```
qqq=par/cos(20.0*3.14159/180.0);

/* The space of anchored rods is 8 ft */

mmid=qqq*8.0*8.0/10.0;

/*  m=qqq*l*l/10  */
printf(" In this system, there are 10 cross sections for your wale .\n\n");

printf(" 1. C10x30    Sx=%.2f  Ix=%.2f\n",table1[0][1],table1[0][0]);
printf(" 2. C12x20.7  Sx=%.2f  Ix=%.2f\n",table1[1][1],table1[1][0]);
printf(" 3. C12x25    Sx=%.2f  Ix=%.2f\n",table1[2][1],table1[2][0]);
printf(" 4. C12x30    Sx=%.2f  Ix=%.2f\n",table1[3][1],table1[3][0]);
printf(" 5. MC12x30.9 Sx=%.2f  Ix=%.2f\n",table1[4][1],table1[4][0]);
printf(" 6. MC12x32.9 Sx=%.2f  Ix=%.2f\n",table1[5][1],table1[5][0]);
printf(" 7. MC12x37   Sx=%.2f  Ix=%.2f\n",table1[6][1],table1[6][0]);
printf(" 8. MC12x35   Sx=%.2f  Ix=%.2f\n",table1[7][1],table1[7][0]);
printf(" 9. MC12x40   Sx=%.2f  Ix=%.2f\n",table1[8][1],table1[8][0]);
printf(" 10. MC12x45  Sx=%.2f  Ix=%.2f\n",table1[9][1],table1[9][0]);

printf("\n Please input the yield stress of the wale.(in ksi)\n>>");
scanf("%f",&fy);
fb=0.75*fy;

/*   The allowable stress is 0.75fy    */

sx1=mmid*12.0/fb;
sxc=sx1/2.0;

for(i=0;i<=9;i++)
{
  if(table1[i][1]>sxc)
    {
    j=i;
    k=1;
    goto p1;
    }
  else {;}
}
p1:{;}
if(k!=0)
  {
printf(" The most appropriate cross section for wale is a pair of \n");
printf(" No. %d    Sx=%.2f    Ix=%.2f\n",j+1,table1[j][1],table1[j][0]);
printf(" channels back-to-back with spacing for the anchor rod.\n");
waj=j+1;
sxc=table1[j][1]/sxc;

printf(" The factor of safety is %.2f\n",sxc);
  }

if(k==0)
{
```

```
waj1=0;
printf(" No cross section listed is OK for your sheet pile.\n");
}
return;
}
```

# WATERWAYS EXPERIMENT STATION REPORTS PUBLISHED UNDER THE COMPUTER-AIDED STRUCTURAL ENGINEERING (CASE) PROJECT

| | Title | Date |
|---|---|---|
| Technical Report K-78-1 | List of Computer Programs for Computer-Aided Structural Engineering | Feb 1978 |
| Instruction Report O-79-2 | User's Guide: Computer Program with Interactive Graphics for Analysis of Plane Frame Structures (CFRAME) | Mar 1979 |
| Technical Report K-80-1 | Survey of Bridge-Oriented Design Software | Jan 1980 |
| Technical Report K-80-2 | Evaluation of Computer Programs for the Design/Analysis of Highway and Railway Bridges | Jan 1980 |
| Instruction Report K-80-1 | User's Guide: Computer Program for Design/Review of Curvilinear Conduits/Culverts (CURCON) | Feb 1980 |
| Instruction Report K-80-3 | A Three-Dimensional Finite Element Data Edit Program | Mar 1980 |
| Instruction Report K-80-4 | A Three-Dimensional Stability Analysis/Design Program (3DSAD) | |
| | Report 1: General Geometry Module | Jun 1980 |
| | Report 3: General Analysis Module (CGAM) | Jun 1982 |
| | Report 4: Special-Purpose Modules for Dams (CDAMS) | Aug 1983 |
| Instruction Report K-80-6 | Basic User's Guide: Computer Program for Design and Analysis of Inverted-T Retaining Walls and Floodwalls (TWDA) | Dec 1980 |
| Instruction Report K-80-7 | User's Reference Manual: Computer Program for Design and Analysis of Inverted-T Retaining Walls and Floodwalls (TWDA) | Dec 1980 |
| Technical Report K-80-4 | Documentation of Finite Element Analyses | |
| | Report 1: Longview Outlet Works Conduit | Dec 1980 |
| | Report 2: Anchored Wall Monolith, Bay Springs Lock | Dec 1980 |
| Technical Report K-80-5 | Basic Pile Group Behavior | Dec 1980 |
| Instruction Report K-81-2 | User's Guide: Computer Program for Design and Analysis of Sheet Pile Walls by Classical Methods (CSHTWAL) | |
| | Report 1: Computational Processes | Feb 1981 |
| | Report 2: Interactive Graphics Options | Mar 1981 |
| Instruction Report K-81-3 | Validation Report: Computer Program for Design and Analysis of Inverted-T Retaining Walls and Floodwalls (TWDA) | Feb 1981 |
| Instruction Report K-81-4 | User's Guide: Computer Program for Design and Analysis of Cast-in-Place Tunnel Linings (NEWTUN) | Mar 1981 |
| Instruction Report K-81-6 | User's Guide: Computer Program for Optimum Nonlinear Dynamic Design of Reinforced Concrete Slabs Under Blast Loading (CBARCS) | Mar 1981 |
| Instruction Report K-81-7 | User's Guide: Computer Program for Design or Investigation of Orthogonal Culverts (CORTCUL) | Mar 1981 |
| Instruction Report K-81-9 | User's Guide: Computer Program for Three-Dimensional Analysis of Building Systems (CTABS80) | Aug 1981 |
| Technical Report K-81-2 | Theoretical Basis for CTABS80: A Computer Program for Three-Dimensional Analysis of Building Systems | Sep 1981 |
| Instruction Report K-82-6 | User's Guide: Computer Program for Analysis of Beam-Column Structures with Nonlinear Supports (CBEAMC) | Jun 1982 |

(Continued)

## WATERWAYS EXPERIMENT STATION REPORTS PUBLISHED UNDER THE COMPUTER-AIDED STRUCTURAL ENGINEERING (CASE) PROJECT

(Continued)

| | Title | Date |
|---|---|---|
| Instruction Report K-82-7 | User's Guide: Computer Program for Bearing Capacity Analysis of Shallow Foundations (CBEAR) | Jun 1982 |
| Instruction Report K-83-1 | User's Guide: Computer Program with Interactive Graphics for Analysis of Plane Frame Structures (CFRAME) | Jan 1983 |
| Instruction Report K-83-2 | User's Guide: Computer Program for Generation of Engineering Geometry (SKETCH) | Jun 1983 |
| Instruction Report K-83-5 | User's Guide: Computer Program to Calculate Shear, Moment, and Thrust (CSMT) from Stress Results of a Two-Dimensional Finite Element Analysis | Jul 1983 |
| Technical Report K-83-1 | Basic Pile Group Behavior | Sep 1983 |
| Technical Report K-83-3 | Reference Manual: Computer Graphics Program for Generation of Engineering Geometry (SKETCH) | Sep 1983 |
| Technical Report K-83-4 | Case Study of Six Major General-Purpose Finite Element Programs | Oct 1983 |
| Instruction Report K-84-2 | User's Guide: Computer Program for Optimum Dynamic Design of Nonlinear Metal Plates Under Blast Loading (CSDOOR) | Jan 1984 |
| Instruction Report K-84-7 | User's Guide: Computer Program for Determining Induced Stresses and Consolidation Settlements (CSETT) | Aug 1984 |
| Instruction Report K-84-8 | Seepage Analysis of Confined Flow Problems by the Method of Fragments (CFRAG) | Sep 1984 |
| Instruction Report K-84-11 | User's Guide for Computer Program CGFAG, Concrete General Flexure Analysis with Graphics | Sep 1984 |
| Technical Report K-84-3 | Computer-Aided Drafting and Design for Corps Structural Engineers | Oct 1984 |
| Technical Report ATC-86-5 | Decision Logic Table Formulation of ACI 318-77, Building Code Requirements for Reinforced Concrete for Automated Constraint Processing, Volumes I and II | Jun 1986 |
| Technical Report ITL-87-2 | A Case Committee Study of Finite Element Analysis of Concrete Flat Slabs | Jan 1987 |
| Instruction Report ITL-87-1 | User's Guide: Computer Program for Two-Dimensional Analysis of U-Frame Structures (CUFRAM) | Apr 1987 |
| Instruction Report ITL-87-2 | User's Guide: For Concrete Strength Investigation and Design (CASTR) in Accordance with ACI 318-83 | May 1987 |
| Technical Report ITL-87-6 | Finite-Element Method Package for Solving Steady-State Seepage Problems | May 1987 |
| Instruction Report ITL-87-3 | User's Guide: A Three Dimensional Stability Analysis/Design Program (3DSAD) Module | Jun 1987 |
| | Report 1: Revision 1: General Geometry | Jun 1987 |
| | Report 2: General Loads Module | Sep 1989 |
| | Report 6: Free-Body Module | Sep 1989 |

(Continued)

## WATERWAYS EXPERIMENT STATION REPORTS PUBLISHED UNDER THE COMPUTER-AIDED STRUCTURAL ENGINEERING (CASE) PROJECT

(Continued)

| | Title | Date |
|---|---|---|
| Instruction Report ITL-87-4 | User's Guide: 2-D Frame Analysis Link Program (LINK2D) | Jun 1987 |
| Technical Report ITL-87-4 | Finite Element Studies of a Horizontally Framed Miter Gate<br>Report 1: Initial and Refined Finite Element Models (Phases A, B, and C), Volumes I and II<br>Report 2: Simplified Frame Model (Phase D)<br>Report 3: Alternate Configuration Miter Gate Finite Element Studies–Open Section<br>Report 4: Alternate Configuration Miter Gate Finite Element Studies–Closed Sections<br>Report 5: Alternate Configuration Miter Gate Finite Element Studies–Additional Closed Sections<br>Report 6: Elastic Buckling of Girders in Horizontally Framed Miter Gates<br>Report 7: Application and Summary | Aug 1987 |
| Instruction Report GL-87-1 | User's Guide: UTEXAS2 Slope-Stability Package; Volume I, User's Manual | Aug 1987 |
| Instruction Report ITL-87-5 | Sliding Stability of Concrete Structures (CSLIDE) | Oct 1987 |
| Instruction Report ITL-87-6 | Criteria Specifications for and Validation of a Computer Program for the Design or Investigation of Horizontally Framed Miter Gates (CMITER) | Dec 1987 |
| Technical Report ITL-87-8 | Procedure for Static Analysis of Gravity Dams Using the Finite Element Method – Phase 1a | Jan 1988 |
| Instruction Report ITL-88-1 | User's Guide: Computer Program for Analysis of Planar Grid Structures (CGRID) | Feb 1988 |
| Technical Report ITL-88-1 | Development of Design Formulas for Ribbed Mat Foundations on Expansive Soils | Apr 1988 |
| Technical Report ITL-88-2 | User's Guide: Pile Group Graphics Display (CPGG) Post-processor to CPGA Program | Apr 1988 |
| Instruction Report ITL-88-2 | User's Guide for Design and Investigation of Horizontally Framed Miter Gates (CMITER) | Jun 1988 |
| Instruction Report ITL-88-4 | User's Guide for Revised Computer Program to Calculate Shear, Moment, and Thrust (CSMT) | Sep 1988 |
| Instruction Report GL-87-1 | User's Guide: UTEXAS2 Slope-Stability Package; Volume II, Theory | Feb 1989 |
| Technical Report ITL-89-3 | User's Guide: Pile Group Analysis (CPGA) Computer Group | Jul 1989 |
| Technical Report ITL-89-4 | CBASIN–Structural Design of Saint Anthony Falls Stilling Basins According to Corps of Engineers Criteria for Hydraulic Structures; Computer Program X0098 | Aug 1989 |

(Continued)

## WATERWAYS EXPERIMENT STATION REPORTS PUBLISHED UNDER THE COMPUTER-AIDED STRUCTURAL ENGINEERING (CASE) PROJECT

(Continued)

| | Title | Date |
|---|---|---|
| Technical Report ITL-89-5 | CCHAN–Structural Design of Rectangular Channels According to Corps of Engineers Criteria for Hydraulic Structures; Computer Program X0097 | Aug 1989 |
| Technical Report ITL-89-6 | The Response-Spectrum Dynamic Analysis of Gravity Dams Using the Finite Element Method; Phase II | Aug 1989 |
| Contract Report ITL-89-1 | State of the Art on Expert Systems Applications in Design, Construction, and Maintenance of Structures | Sep 1989 |
| Instruction Report ITL-90-1 | User's Guide: Computer Program for Design and Analysis of Sheet Pile Walls by Classical Methods (CWALSHT) | Feb 1990 |
| Technical Report ITL-90-3 | Investigation and Design of U-Frame Structures Using Program CUFRBC<br>Volume A: Program Criteria and Documentation<br>Volume B: User's Guide for Basins<br>Volume C: User's Guide for Channels | May 1990 |
| Instruction Report ITL-90-6 | User's Guide: Computer Program for Two-Dimensional Analysis of U-Frame or W-Frame Structures (CWFRAM) | Sep 1990 |
| Instruction Report ITL-90-2 | User's Guide: Pile Group–Concrete Pile Analysis Program (CPGC) Preprocessor to CPGA Program | Jun 1990 |
| Technical Report ITL-91-3 | Application of Finite Element, Grid Generation, and Scientific Visualization Techniques to 2-D and 3-D Seepage and Groundwater Modeling | Sep 1990 |
| Instruction Report ITL-91-1 | User's Guide: Computer Program for Design and Analysis of Sheet-Pile Walls by Classical Methods (CWALSHT) Including Rowe's Moment Reduction | Oct 1991 |
| Instruction Report ITL-87-2 (Revised) | User's Guide for Concrete Strength Investigation and Design (CASTR) in Accordance with ACI 318-89 | Mar 1992 |
| Technical Report ITL-92-2 | Fiinite Element Modeling of Welded Thick Plates for Bonneville Navigation Lock | May 1992 |
| Technical Report ITL-92-4 | Introduction to the Computation of Response Spectrum for Earthquake Loading | Jun 1992 |
| Instruction Report ITL-92-3 | Concept Design Example, Computer Aided Structural Modeling (CASM) | |
| | Report 1: Scheme A | Jun 1992 |
| | Report 2: Scheme B | Jun 1992 |
| | Report 3: Scheme C | Jun 1992 |
| Instruction Report ITL-92-4 | User's Guide: Computer-Aided Structural Modeling (CASM) - Version 3.00 | Apr 1992 |
| Instruction Report ITL-92-5 | Tutorial Guide: Computer-Aided Structural Modeling (CASM) - Version 3.00 | Apr 1992 |

(Continued)

## WATERWAYS EXPERIMENT STATION REPORTS PUBLISHED UNDER THE COMPUTER-AIDED STRUCTURAL ENGINEERING (CASE) PROJECT

(Concluded)

| | Title | Date |
|---|---|---|
| Contract Report ITL-92-1 | Optimization of Steel Pile Foundtions Using Optimality Criteria | Jun 1992 |
| Technical Report ITL-92-7 | Refined Stress Analysis of Melvin Price Locks and Dam | Sep 1992 |
| Instruction Report GL-87-1 | User's Guide: UTEXAS3 Slope-Stability Package; Volume IV, User's Manual | Sep 1992 |
| Contract Report ITL-92-2 | Knowledge-Based Expert System for Selection and Design of Retaining Structures | Sep 1992 |