1-1-2003

Analysis and design of grade control structures using scrap tires

Sai-Meng Choor
Iowa State University

Recommended Citation

Choor, Sai-Meng, "Analysis and design of grade control structures using scrap tires" (2003). *Retrospective Theses and Dissertations*. 19933.
https://lib.dr.iastate.edu/rtd/19933

Analysis and design of grade control structures using scrap tires

by

Sai-Meng Choor

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Civil Engineering (Environmental Engineering)

Program of Study Committee:
Roy R. Gu, Co-major Professor
Robert Lohnes, Co-major Professor
U. Sunday Tim

Iowa State University

Ames, Iowa

2003

Graduate College
Iowa State University

This is to certify that the master's thesis of

Sai-Meng Choor

has met the thesis requirements of Iowa State University

Signatures have been redacted for privacy

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks and appreciation to my major professors, Dr. Roy Gu and Dr. Robert Lohnes, for their patience and guidance all through my graduate study. I also thank Dr. U. Sunday Tim for serving on my graduate committee.

A separate part of this research involves the measurement of unit weight of various potential filling materials and the resultant hydraulic resistance of these materials in tires, which are performed by my fellow graduate student Chee-Siong Cheong. I also thank him for his support throughout the research.

Finally, many thanks go out to my parents, Sow-Hee and Tuck-Wah Choor. Without their support and encouragement completion of this these would not have been possible.

CHAPTER 1. INTRODUCTION

1.1 THE PROBLEM

Channel degradation is a natural event that can occur in any stream or river. The phenomenon is especially acute in western Iowa because of the easily erodible deep loess in that region. As streams erode deeper, the structural safety of numerous roads, bridges and other infrastructures in western Iowa is being affected. As degradation proceeds, channel banks also become unstable and landslides occur, causing channel width to increase. The widening of the channel causes the loss of valuable farmland, which can never be reclaimed once it occurs. The unaffected farmland becomes increasingly difficult to farm because of the dissected landscape (Levich, 1994).

In the attempts to slow down the channel degradation process, low drop (less than 4 feet high) grade stabilization structures have been constructed at many locations on the rivers and streams in western Iowa. They have also become a well-accepted solution to stream erosion. Grade control structures are overfalls that raise the stream bed elevation which decreases the slope of the flow line for a given reach upstream, and thus reduces the streamflow velocity within that reach. This decreases erosion and allows sediment to settle upstream of the structures. The lower velocity also reduces the stream power, thereby, decreasing the rate of channel degradation (Boyken, 1998). Grade control structures will perform satisfactorily when they are designed and constructed properly.

The cost of these structures has increased considerably over the years. Early structures cost approximately $3500 to construct (Goehring, 1981). Current practice for controlling stream degradation and erosion at bridges employs steel sheetpile, concrete or riprap grade control structures that can cost $5,000 to $200,000. One concrete flume in Taylor County, Iowa had a final construction cost of $1,120,000 (Magner, 1994).

1.2 SCRAP TIRE REUTILIZATION

It has been estimated that more than 200 million scrap tires are added annually to the 2 billion waste tires that have already been stockpiled in the United States (Zimmerman, 1997). Also, Iowa legislatures have banned non-shredded scrap tires from Iowa landfills. The increasing scrap tire stockpiles pose serious environmental and health problems to the public. The waste tire problem is one that will continue to grow until more tires find environmentally sound and cost effective markets. The use of scrap tires in civil engineering projects is one of the ways to reduce the number of scrap tires that are stockpiled.

More economical methods of stream channel stabilization utilizing lower cost alternatives to concrete and natural rock are needed. Whole scrap tires are currently utilized for surface erosion control and stream bank stabilization. Design guidelines for these structures are available from Natural Resources Conservation Service (NRCS). In the NRCS structures, tires are laced together and used as a protective layer or mat over stream banks or soil embankments. The top, toe, upstream and downstream ends of the mattress are tied into the banks, and if scour is anticipated, riprap is placed at the toe for additional protection.

Scrap tires are also being used in other civil engineering projects. Poh et al. (1995) presented the use of scrap tires and geotextiles in hill slope protection. Scrap tires of a single size were stacked and filled with granite aggregate and quarry dust to form a retaining wall. Geotextiles were used as the reinforcement to stabilize the structure, and to improve the drainage within the structure. The proposed rubber tire wall has been found to be inexpensive and simple to construct. Scrap tires and shredded tires have also been used as: lightweight fill in embankments, retaining structures, and subsurface drainage (Gebhardt, 1997).

1.3 OBJECTIVES AND METHODOLOGY

It is proposed that the entire mass of the stream grade structures will be composed of whole scrap tires that are tied together and filled with materials to provide adequate erosion resistance to the flowing water. It is the purpose of this research to evaluate the technical feasibility of the simpler, less costly stream grade control structures constructed of whole scrap tires. If these structures are found to be feasible, it is likely that more engineers will use scrap tires in grade control and bank stabilization structures. This will recycle a material that has become problematic and at the same time provide a relatively inexpensive method of stabilizing degrading streams.

The objectives of this study are to analyze and evaluate the structure as a whole, the fastening system, and filling materials and to provide design recommendations. Theoretical studies are first performed to analyze soil and water forces acting on the grade control structures. The calculated forces (horizontal and vertical reactions) are then used to determine the required stability, i.e. minimum requirement for the pullout strength of a binding system or tires and the required unit weight of the structure (tires and fill materials) to resist uplift and sliding. The safety factor for overcoming uplift is computed from the vertical reactions and the weight of the structure, i.e. the ratio of the weight to the vertical forces. The factor of safety against sliding is determined by comparing the river bottom shear stress and the horizontal reaction. The safety factor of the fastening system is the ratio of the strength of the system to the hydraulic force applied to the structure. Fastening systems evaluation and system strength measurements are carried out in laboratory tests. The analysis results and design information for the structures are programmed into a computer model named Grade Control Structure Design Model (GCSDM). Design charts are developed based on the outputs from GCSDM. Laboratory tests and measurements are also conducted in a separate study to evaluate fill material and their strength and hydraulic resistance. The study is performed by Chee-Siong Cheong, and will be included in his Master's thesis.

1.4 ORGANIZATION OF THESIS

This thesis is divided into the following five areas:

1) Chapter 2 is the discussion of methods to perform river flow analysis, force analysis, riverbed shear analysis and stability analysis of the whole structure.
2) Chapter 3 describes the characteristics of scrap tires grade control structures.
3) In Chapter 4, evaluation of fastening systems is discussed.
4) Chapter 5 is the description of the Grade Control Structures Design Model (GCSDM) program.
5) Chapter 6 describes two illustrative examples of GCSDM and shows the results.
6) Chapter 7 summarizes the information assimilated and states conclusions for the less expensive scrap tire grade control structures.

CHAPTER 2. THEORETICAL ANALYSIS

2.1 RIVER FLOW ANALYSIS

The knowledge of the magnitude and frequency of floods is essential for the effective planning and safe design of grade control structures. On gaged streams long-term flood data are available to calculate design-flood flow rate. However, flow rates for most streams in western Iowa are not available, as they are small streams and not gaged. For these streams, drainage area, which can be easily obtained, is used as the primary variable to estimate design-flood discharge. The flow rate at any cross section is estimated by using the following equation:

$$Q = 422.58 \times (LF) \times (RI)^{.301} \times (D_A)^{0.504} \tag{2.1}$$

where Q is the flow rate (cfs), LF is the land use factor for the area (Table 2.1), RI is the design recurrence interval and D_A is the drainage area (mi^2).

Table 2.1 Land Use Factor (LF)

Land Use	Land Description				
	Very Hilly	Hilly	Rolling	Flat	Very Flat (no ponds)
Mixed Cover	1.0	0.8	0.6	0.4	0.2
Permanent Pasture	0.6	0.5	0.4	0.2	0.1
Permanent Woods	0.3	0.25	0.2	0.1	0.05

(Source: Iowa Runoff Chart, IDOT)

Assuming uniform flow, the flow rate determined from Equation 2.1 is used to calculate the depth of flow before the structure is in place from Manning's formula. Manning's formula in English units is written as:

$$Q = \frac{1.486}{n} A R^{2/3} S^{1/2} \tag{2.2}$$

where n is the Manning's roughness coefficient, A is the wetted surface area (sq. ft), R is the hydraulic radius (ft) and S is the channel bottom slope.

Since $A = W \times D$ and $P = 2D + W$, hydraulic radius, R, for prismatic rectangular channel can then be expressed as:

$$R = \frac{A}{P} = \frac{W \times D}{2D + W} \tag{2.3}$$

where P is the wetted perimeter (ft), W is the section width (ft), and D is the depth of flow section (ft). If the channel section is not rectangular, water surface width is used as the section width. By substituting Equation 2.3 into 2.2, Manning's formula can be modified into:

$$Q = \frac{1.486}{n} (W \times D) (\frac{W \times D}{2D + W})^{2/3} S^{1/2} \tag{2.4}$$

or

$$\frac{(W \times D)^{5/3}}{(2D + W)^{2/3}} = \frac{Q \times n}{1.486 \times S^{1/2}} \tag{2.5}$$

or

$$\frac{D^{5/3}}{(2D + W)^{2/3}} = \frac{Q \times n}{1.486 \times W^{5/3} \times S^{1/2}} \tag{2.6}$$

The depth of flow section, *D*, which can be determined by the method of successive substitution, is the depth of water under a given flow rate before the dam is constructed. To simplify the problem, the tailwater depth after the construction is assumed to be the pre-construction water depth.

Bernoulli's equation is then used to determine the depth of water overtopping the dam. To determine the flow depth, two points of close proximity are selected: points 2 and 3 are located on the upstream and downstream slopes of dam respectively. Since the profile of flow depth from section 2 to 3 is not constant, the depth of flow overtopping the dam is estimated as the average of the depths at section 2 and 3. According to Bernoulli's equation, which assumes a negligible loss of internal energy, the total energy head at a point in the upstream section 2 in Figure 2.1 is equal to the total energy head at a point in the downstream section 3; or

$$y_2 + \frac{V_2^{\ 2}}{2g} = y_c + \frac{V_c^{\ 2}}{2g} \tag{2.7}$$

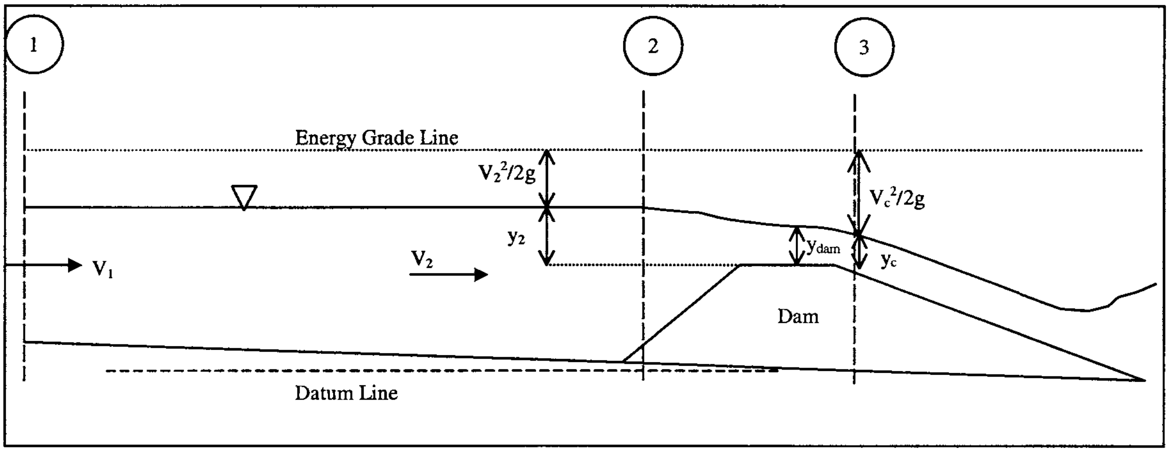


Figure 2.1 A schematic of various forms of energy in uniform flow (not to scale)

However, under critical flow condition,

$$V_c = \sqrt{g y_c} \tag{2.8}$$

and

$$y_c = \left(\frac{Q^2}{W^2 g} \right)^{\frac{1}{3}} \tag{2.9}$$

Thus, Equation 2.7 becomes

$$y_2 + \frac{V_2^{\,2}}{2g} = y_c + \frac{g y_c}{2g} \tag{2.10}$$

or

$$y_2 + \frac{V_2^{\,2}}{2g} = \frac{3}{2} y_c \tag{2.11}$$

or

$$y_2 + \frac{V_2^{\,2}}{2g} = \frac{3}{2} \left(\frac{Q^2}{W^2 g} \right)^{\frac{1}{3}} \tag{2.12}$$

Solving for y_2, Equation 2.12 becomes

$$y_2 = \frac{3}{2} \left(\frac{Q^2}{W^2 g} \right)^{\frac{1}{3}} - \frac{V_2^{\,2}}{2g} \tag{2.13}$$

However, since V_2 is a function of y_2 and can be approximated by continuity equation, or

$$V_2 = \frac{Q}{(y_2 + h_{dam})W} \tag{2.14}$$

y_2 has to be solved using the method of successive substitution also.

After getting y_2, the depth of flow overtopping the dam, y_{dam}, can be estimated by the following equation:

$$y_{dam} = \frac{y_2 + y_c}{2} \tag{2.15}$$

2.2 FORCE ANALYSIS

There are three ways in which a low drop grade control structure of whole scrap tires may fail: by sliding on a plane within the tires above the foundation or along the contact between the tires and soil foundation, by uplift due to overflowing water and by internally breaking the fastening system or tearing the tire rubber itself, as shown in Figure 2.2. Sliding along the foundation of the structure may be resisted by the shear resistance between the soil surface and the bottom layer of tires. Because the damage to fastening systems and the pullout of the connecting materials from tires may also cause structural failure, the fastening systems and connection to the tires should be strong enough to resist all potential hydraulic forces. Uplift due to overflowing water can be overcome by properly designed weight of the structure, i.e. the required unit weight.

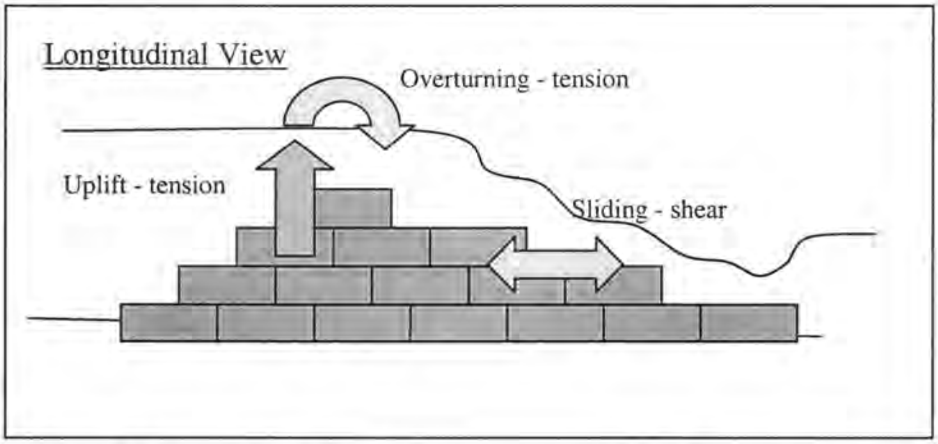


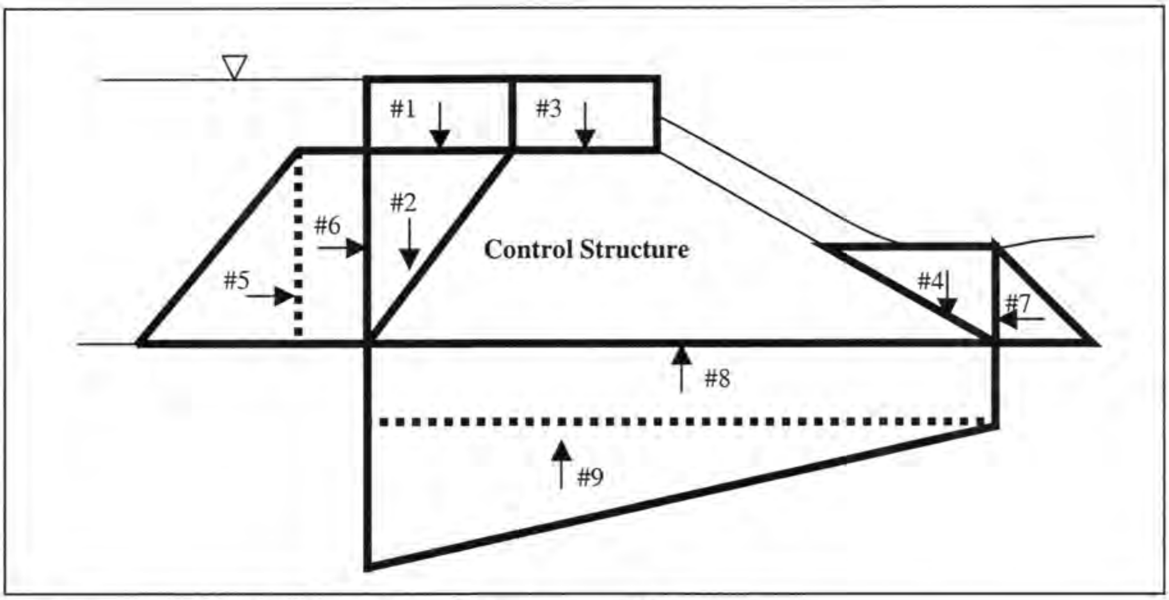
Figure 2.2 Tensile and shear stresses created by the flowing water

A control structure segment of unit width is considered in the force analysis. The forces considered in this analysis are 1) hydrostatic and uplift forces, 2) weight of the column of water resting on the surface of the structure, 3) forces due to sediment deposits, 4) drag and lift forces due to fluid flow and 5) the weight of the structure itself. Figure 2.3(a) shows the 9 components of hydrostatic and uplift forces, and weight of water acting on the structure. Figure 2.3(b) shows 2 components of sediment forces. Figure 2.3(c) shows the drag and lift forces due to fluid flow, and the weight of the control structure itself. The centroid and direction of these forces are shown in the figures also.

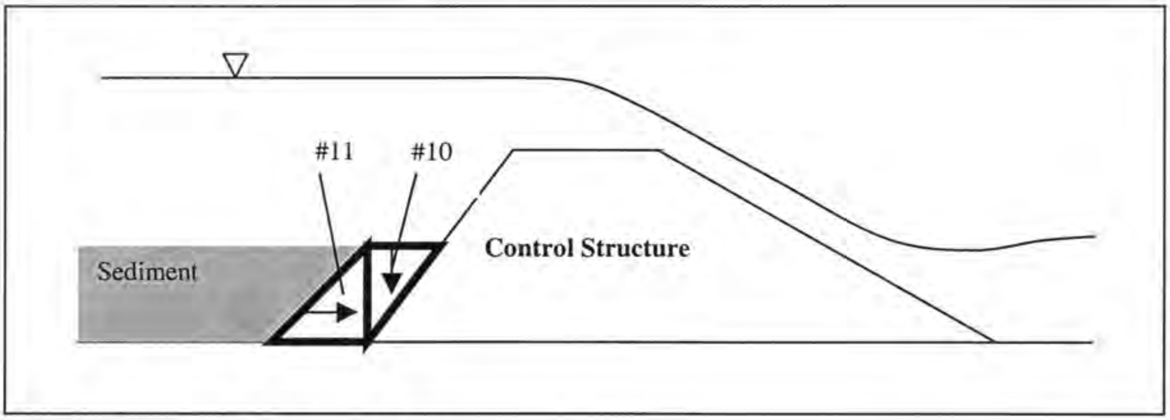
The force acting through the centroid of each component is the volume of each component multiplied by its unit weight. The hydrostatic pressure on any vertical wall of the submerged structure is a triangle with a base width of y, where y is the water depth. The total hydrostatic force on that face is then the volume of the triangle, which is $1/2y^2$, multiplied by the unit weight of water. The force acts horizontally at a height of $y/3$ from the base.

Uplift force on the structure is due to the water pressure acting upward from the pores beneath the structure. “Uplift pressure equals the water head at the upstream and downstream faces and with a linear variation between in the form of a trapezoid.” (Gupta, 1989) The total uplift force that acts through the centroid is given by the volume of the trapezoid multiplied by the unit weight of water.

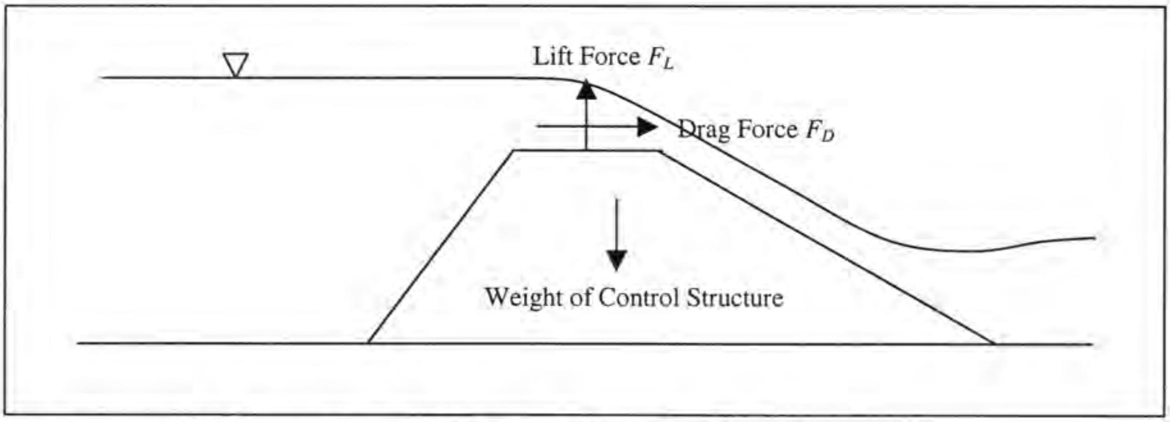
Weight of water resting on the surface of the structure is included as a resisting force. The volume of the column of water can be determined based on its shape. To calculate the forces due to sediment deposits, the depth of sediment deposits has to be estimated. This depth is taken as the height of the control structure to generate the worst-case loading under



(a) Components of hydrostatic, uplift and water forces



(b) Components of sediment forces



(c) Drag and lift forces due to fluid flow and weight of structure

Figure 2.3 Forces on a control structure complete sedimentation. Weight of sediment resting on the upstream face of the structure is also included as a force. Rankine's theory of active earth pressure is used to determine the horizontal force due to sediment deposits. The ratio of Rankine's active earth pressure (σ_a) to vertical stress (σ_v) is called the coefficient of active earth pressure (K_a), or

$$K_a = \frac{\sigma_a}{\sigma_v} = \tan^2(45 - \frac{\phi}{2}) \qquad (2.16)$$

where ϕ is the friction angle of sediment.

Drag and lift forces on the structure due to fluid flow can be expressed as:

$$F_D = C_D A \frac{\rho V_{dam}^2}{2g} \qquad (2.17)$$

$$F_L = C_L A \frac{\rho V_{dam}^2}{2g} \qquad (2.18)$$

where F_D is the drag force, C_D is the drag coefficient, which is equal to 1.0 for turbulent flow (Morris, 1972), ρ is the density of water, V_{dam} is the flow velocity above the structure, F_L is the lift force, C_L is the lift coefficient, which is equal to 0.25 (Morris, 1972).

2.3 STABILITY ANALYSIS

In order to determine the required unit weight of a grade control structure and to compute the factor of safety against both uplift and sliding, two basic rules of statics are applied:

$$\sum \text{vertical forces} = 0 \qquad (2.19)$$

$$\sum \text{horizontal forces} = 0 \qquad (2.20)$$

From Equation 2.19 above, the required weight of the structure is determined by subtracting the uplift and lift forces from the downward weight forces as shown below:

$$\begin{aligned} weight_{dam} = \; & force\#8 + force\#9 - force\#1 \\ & - force\#2 - force\#3 - force\#4 \\ & - force\#10 + F_L \end{aligned} \tag{2.21}$$

Equation 2.21 gives a factor of safety against uplift of 1.0. This required weight of dam, when divided by its volume gives the saturated unit weight of the dam needed. This saturated unit weight is then be converted to effective unit weight by the equation below:

$$\gamma'_{dam} = \gamma_{dam,sat} - \gamma_w \tag{2.22}$$

where γ'_{dam} is the effective or buoyant unit weight of dam (pcf), $\gamma_{dam,sat}$ is the saturated unit weight of dam (pcf), γ_w is the unit weight of water (62.4 pcf). To determine dry unit weight of dam from the effective unit weight, the equations used in the derivation are as follow:

$$\gamma'_{dam} = \frac{W_s - V_s \gamma_w}{V} \tag{2.23}$$

$$\gamma_{dam,dry} = \frac{W_s}{V} \tag{2.24}$$

where W_s is the weight of solids (lb), V_s is the volume of solids (cu. ft), V is the total volume (cu. ft) and $\gamma_{dam,dry}$ is the dry unit weight of dam (pcf). Since the specific gravity of dam, G can be expressed as:

$$G = \frac{W_s}{V_s \gamma_w} \tag{2.25}$$

Equation 2.23 becomes

$$\gamma'_{dam} = \frac{W_s - \frac{W_s}{G}}{V} \tag{2.26}$$

or

$$\gamma'_{dam} = \frac{GW_s - W_s}{GV} \tag{2.27}$$

or

$$\gamma'_{dam} = \frac{W_s(G-1)}{GV} \tag{2.28}$$

Substituting Equation 2.24 into 2.28 gives

$$\gamma'_{dam} = \gamma_{dam,dry}\frac{(G-1)}{G} \tag{2.29}$$

Solving for $\gamma_{dam,dry}$, Equation 2.29 becomes

$$\gamma_{dam,dry} = \gamma'_{dam}(\frac{G}{G-1}) \tag{2.30}$$

However, Equation 2.30 can be simplified further to eliminate the specific gravity variable in the equation. Different specific gravities within the range is assumed, as shown in Table 2.2, and the average value of *G/(G-1)* is then be used to replace the term *G/(G-1)* in Equation 2.30. Thus, Equation 2.30 becomes

$$\gamma_{dam,dry} = 1.60\gamma'_{dam} \tag{2.31}$$

By referring back to Equation 2.20, the horizontal pushing force, R_H acting on the structure can be determined as follow:

$$\begin{aligned} R_H = \quad & force\#5 + force\#6 - force\#7 \\ & + force\#11 + F_D \end{aligned} \tag{2.32}$$

Table 2.2 Approximation of the ratio G/(G-1)

Assumed G	$G/(G-1)$	Average $G/(G-1)$
2.65	1.606	1.60
2.66	1.602	
2.67	1.598	
2.68	1.595	
2.69	1.591	

The horizontal force is compared with the resisting shear stress that can be provided by the structure and the river bottom discussed in the next section.

2.4 SHEAR ANALYSIS OF CONTROL STRUCTURE AND RIVERBED MATERIAL

The resisting shear stress between the soil and the control structure, τ_f is given as:

$$\tau_f = c + \sigma' \tan \phi \qquad (2.33)$$

where c is the cohesion between the structure and river bottom, σ' is the effective normal stress due to the weight of the structure (pcf), and ϕ is the friction angle of river bottom soil. The effective unit weight of the structure is used to determine σ' as in the following equation:

$$\sigma' = \gamma'_{dam} H_{dam,equivalent} \qquad (2.34)$$

where $H_{dam,equivalent}$ is the equivalent height of the trapezoidal structure as a rectangular structure. This shear stress, when converted to shear force, is the resisting force of the dam and the soil beneath it against sliding. Factor of safety of the control structure against sliding is then computed by comparing these two stresses described above together.

CHAPTER 3. CHARACTERISTICS OF TIRES GRADE CONTROL STRUCTURES

3.1 TIRE CROSS SECTION

The cross section of a passenger car tire has two main components, tread and sidewall. Passenger car tire treads are reinforced with steel belts, whereas only liners support the sidewalls, as shown in Figure 3.1. Therefore, the strength of the tire sidewall is generally weaker than the tread and is the portion most likely to fail.

3.2 TIRE VOLUME ESTIMATION BASED ON SIZE SPECIFICATION

There are a lot of useful information molded on the sidewall of each tire. Not only showing the brand name of the tire, most importantly, the sidewall also shows size specification of the tire. A typical size specification for passenger car tires is *XXX/YY* R*ZZ*, where *XXX* is the cross-section width in millimeters, *YY* is the aspect ratio (ratio of the cross-section height to its width) in percent, R is the abbreviation for radial construction and *ZZ* is the rim diameter in inches.

By utilizing its size specification, the outside diameter and volume of a tire are estimated. These size variables are converted to a standardized unit, i.e. either SI or English units. Since only English units are being used in this research, all the tire dimensions are expressed in English units. Thus, the tire cross-section width, $W_{section}$ (ft) becomes

$$W_{section} = \frac{XXX}{304.8} \tag{3.1}$$

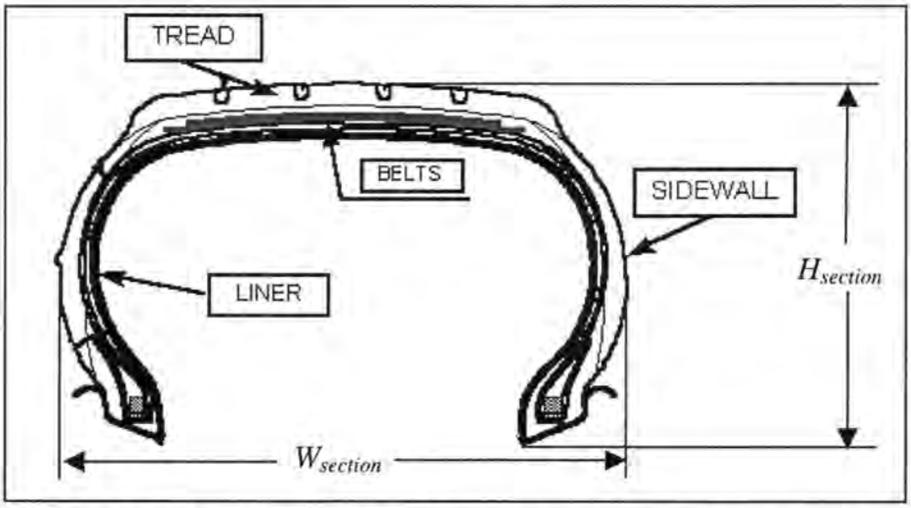


Figure 3.1 The cross section of a typical auto tire (picture taken from www.us.pirelli.com)

To calculate the outside diameter of a tire, the following cross-section height equation is used:

$$H_{section} = W_{section} \times \frac{YY}{100} \tag{3.2}$$

where $H_{section}$ is the tire cross-section height (ft). The outside diameter, D_o (ft) can be expressed as:

$$D_o = 2H_{section} + \frac{ZZ}{12} \tag{3.3}$$

Assuming the tire as a cylindrical object with a diameter of D_o and a height of $W_{section}$, the volume of the tire, V_{tire} (ft^3) can then be estimated by the following volume equation:

$$V_{tire} = \frac{\pi}{4}(D_o)^2 W_{section} \tag{3.4}$$

3.3 VOLUMETRIC ANALYSIS OF TIRE GRADE CONTROL STRUCTURES

The total number of layers of tires and the number of rows in each layer are needed to estimate the total volume of a tire grade control structure. Number of layers can be determined by dividing the proposed structure height by the section width of tires used in the construction, rounded to the nearest integer. Dividing the river section width by the outside diameter of tire gives the number of rows per layer after rounding to the nearest integer. Figure 3.2 shows one possible stacking method of a tire grade control structure. In this structure, tires are stacked following the proposed fore or back slope line. Tires can always be stacked beyond the slope line, but can never be within the line.

In order to determine the number of tires in each layer, the width of each layer is needed and is expressed as:

$$Width_n = TopWidth_{dam} + (n-1)Width_{tire}\left[S_1 + S_2\right] \qquad (3.5)$$

where $Width_n$ is the width of the nth layer of the control structure (ft) (top being the 1st layer), $TopWidth_{dam}$ is the top width of the structure (ft), $Width_{tire}$ is the cross-section width of each tire (ft), S_1 and S_2 are the proposed fore and back slopes, respectively. Dividing the width of each layer by the outside diameter of a tire gives the number of tires in that layer per row. The estimated number of tires required for various grade control structures with different geometry is shown in Table 3.1.

Total volume of the control structure can then be determined by adding tires in each layer and row together multiplied by the volume of a typical tire estimated based on Equation 3.4.

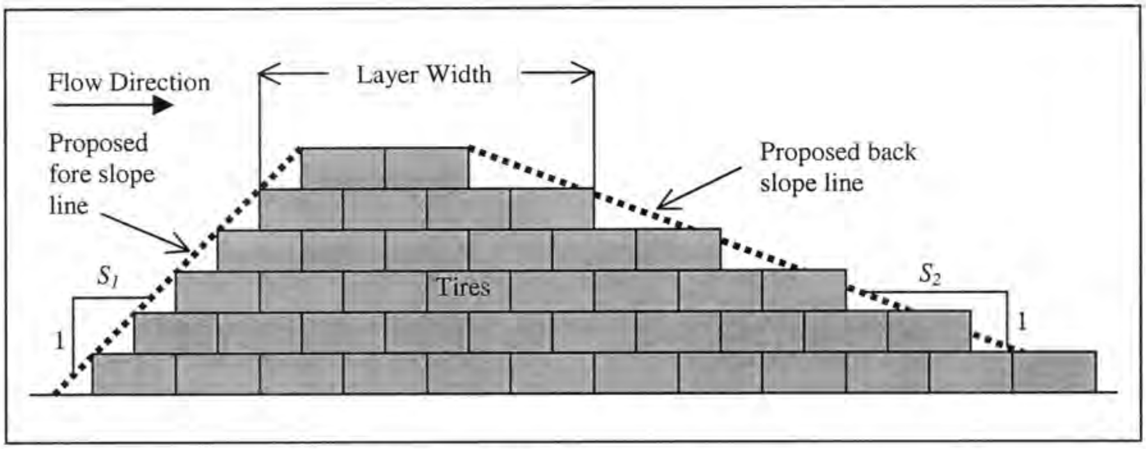


Figure 3.2 One possible stacking method of tire grade control structure

Table 3.1 Estimated number of tires for various structures with different geometry in a 30-foot wide channel

Top Width* (ft)	Height (ft)	Upstream Slope (S_1:1)	Downstream Slope (S_2:1)	Estimated Number of Tires
4.9	4	2	8	1155
4.9	4	1	8	1080
4.9	4	1	4	705
4.9	2	1	8	225
4.9	2	1	4	165

* Approximately equal to the outside diameter of two tires

3.4 LABORATORY MEASUREMENTS OF UNIT WEIGHT OF TIRE RUBBER

Tires used as grade control structure construction materials are filled with filling materials and thus, they are considered as composite materials. Since the unit weight requirement determined from previous chapter is for the whole grade control structure, it is also the requirement for the composite materials. To estimate the resultant unit weight due to rubber and filling materials, the unit weight of rubber only is needed. Results from the unit weight measurement are also used to estimate the percent volume of rubber content in a tire.

3.4.1 Description

The purposes of these laboratory measurements are to determine the average unit weight of tire rubber and to estimate the rubber content of a tire. Water displacement tests were carried out on 6 whole scrap tires of 3 different sizes. For each test, a scrap tire was weighed before submerging into water. The volume of water displaced by each tire was also measured.

3.4.2 Results

Table 3.2 shows the mass of the 6 tires before submergence, the volume of water displaced and their respective unit weights. The average tire unit weight, which is also the tire rubber unit weight, is 98.93 pcf. Referring to the same table, Goodyear tires have also produced the highest unit weight among the tires tested.

Based on its size specification, the total volume of each individual tire is estimated by using Equation 3.4. Having estimated the total volume of each tire, the percent volume of rubber (i.e. the volume of solid rubber) in tire is then determined. Table 3.3 shows the computation process and results. The average tire rubber content determined is 8 percent.

Table 3.2 Mass and volume measurement results of 6 different tires

Tire Brand & Series Name	Size Specification	Measured Weight (lb)	Measured Volume (ml)	Unit Weight (pcf)
Goodyear Integrity	P 205/65 R15	18.21	5200	99.16
Goodyear Recatta 2	P 205/65 R15	18.91	5400	99.16
Michelin XZ4	P 195/75 R14	17.06	5200	92.90
Goodyear Vector	P 195/75 R14	17.80	4500	112.01
Grappler II	P 165/80 R13	13.15	4000	93.09
Grappler II	P 165/80 R13	13.74	4000	97.27
			Average=	98.93

Table 3.3 Percent volume analysis of rubber materials in tires

	Size			(1)	(2)	(3)	(4)	(5)	(6)
Tire	Specifications			Section	D_o	Estimated Volume		Rubber	
Brand & Name	XXX	YY	ZZ	Width (ft)	(ft)	(ft^3)	(ml)	Vol (ml)	Vol (%)
Goodyear Integrity	205	65	15	0.67	2.12	2.38	67503.19	5200	7.7
Goodyear Recatta 2	205	65	15	0.67	2.12	2.38	67503.19	5400	8.0
Michelin XZ4	195	75	14	0.64	2.13	2.27	64329.41	5200	8.1
Goodyear Vector	195	75	14	0.64	2.13	2.27	64329.41	4500	7.0
Grappler II	165	80	13	0.54	1.95	1.62	45755.17	4000	8.7
Grappler II	165	80	13	0.54	1.95	1.62	45755.17	4000	8.7
								Average=	8.0

(1) Equation 3.1

(2) Equation 3.3

(3) Equation 3.4

(4) = (3) * 28316.85

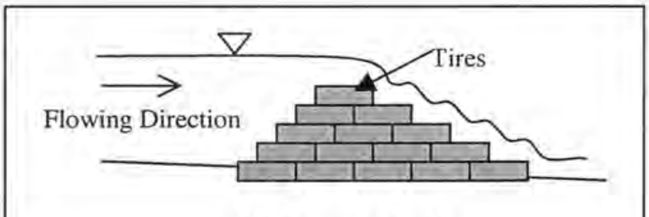
(5) From Table 3.2

(6) = (5) / (4) * 100

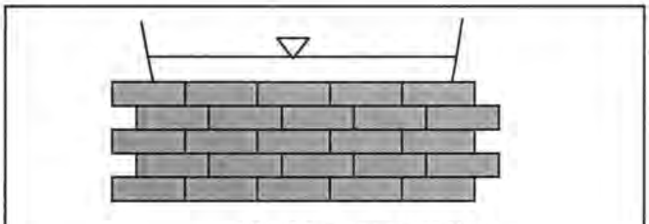
CHAPTER 4. EVALUATION OF FASTENING SYSTEMS

4.1 INTRODUCTION

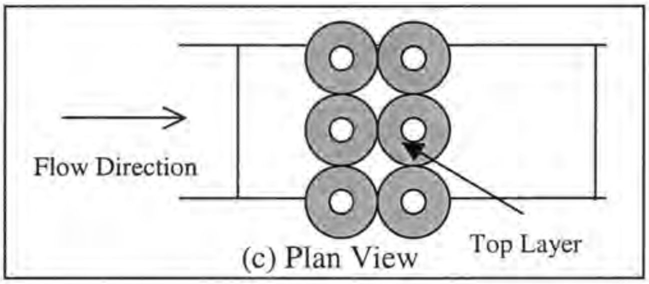
This portion of the research studies stream grade control structures composed of whole scrap tires that are stacked and fastened together. Figure 4.1(a) shows the longitudinal section, Figure 4.1(b) shows the transverse section, and Figure 4.1(c) shows plan view of the type of the proposed structure. Two stacking arrangements of tires are shown in Figure 4.2.



(a) Longitudinal



(b) Transverse View



(c) Plan View

Figure 4.1 Schematics of scrap tire grade control structure

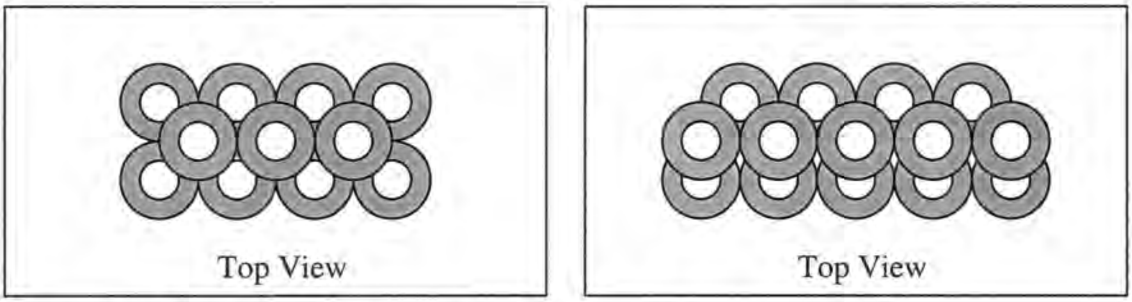


Figure 4.2 Possible alignments of tires

In order to connect the tires into a coherent structure, binding systems between the tires of the same layer and layers above and below are needed. For the tires of the same layer, only one contact will exist at the tread of any two contacting tires; however, several contacts are possible on the sidewall between layers, depending on the alignment of the tires (see Figure 4.2). It is proposed that connections are to be made at all contact locations.

The sliding along the foundation of the structure may be resisted by the shear resistance between the surface of the soil and the foundation. Because the damage to fastening systems and the pullout of the connecting materials from tires may cause structure failure, it is required that the fastening system and connection are strong enough to resist all potential hydraulic forces, including both tensile and shear stresses produced by the flowing water. The safety factor of the fastening system is determined by comparing the strength of the system to the hydraulic force applied to the structure

The condition of the a tire will have great effect on its overall strength, therefore tires selected for testing are required to have few cracks in the rubber and no steel belt showing through the rubber. The same criteria should be applied to tires that will be used in the construction of grade control structures.

4.2 LABORATORY MEASUREMENTS OF TENSILE AND SHEAR STRENGTH

As discussed in Section 3.1, the cross section of a car tire is divided into two main components, which are the steel belts reinforced tread and the liner supported sidewall. Because all contact surfaces among the tires are located either on the sidewall or the tire tread, these surfaces must be strength tested. Due to its lower strength and ease of cutting, tire sidewalls are selected to be strength tested. This will provide conservative strength data.

4.2.1 Objectives

The objectives of this part of the study are to determine the tensile and shear pullout strength of various binding systems for auto tires. Three fastening methods were considered, including lag screws, machine bolts and steel cables. Tests were performed on 3/8" and ½" diameter lag screws, and on 3/8" diameter machine bolts. The strength of steel cables was not tested as they are used to lace around the tires in the control structure. The factor preventing cables from failing is their breaking strength, which is normally provided along with the manufacturer's specifications.

4.2.2 Test Description

Tire coupons cut from the sidewalls of the auto tires selected were used to evaluate the actual tire-to-tire connections. Thickness of each tire coupon, which was also the sidewall thickness of each tire, was measured. At least 3 replicas were made for each type of test on each tire.

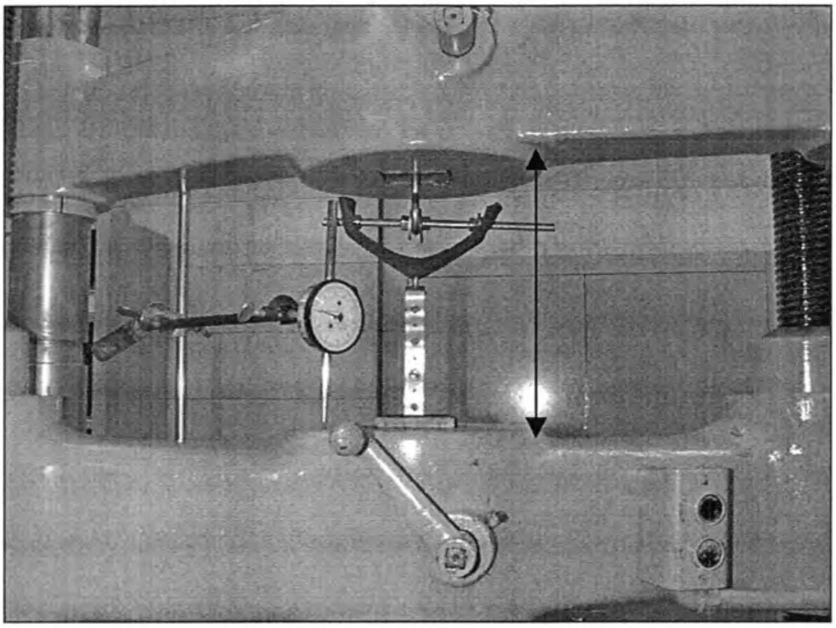
To test for the tensile pullout strength of both lag screw and machine bolt on tire, tire coupons with lag screws or machine bolts drilled through them were prepared. Only machine bolts were fastened with nuts and 7/8-inch outside diameter washers on the threaded end. A threaded rod was used to support the coupons at both ends. Vertical force was applied through the cap of the screws or bolts by a MTS machine, as shown in Figure 4.3 until failure. The shear test of both systems described above used the same MTS machine but a different set up, as shown in Figure 4.4. A bolt or screw that penetrated a tire coupon, was pulled downward by the MTS machine through a steel cable until failure.

4.2.3 Results

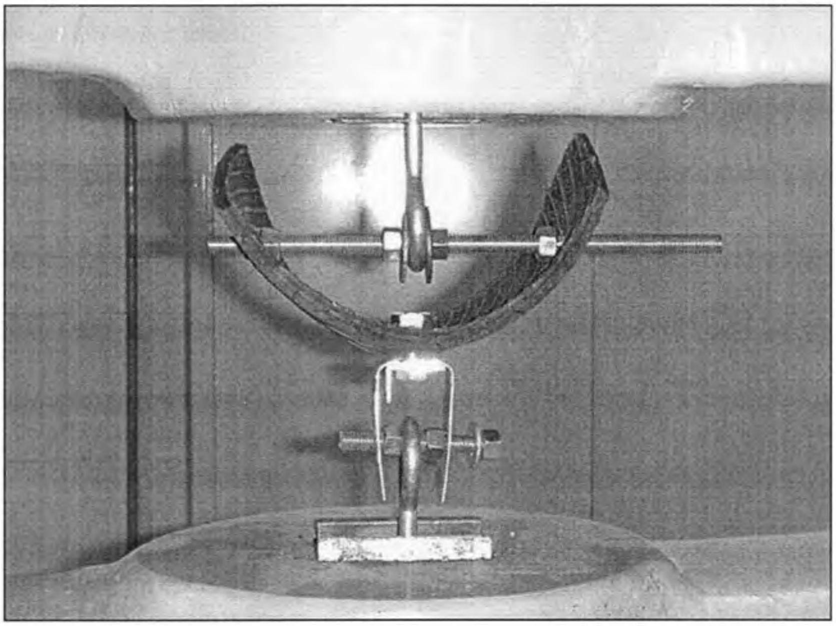
By referring to Table 4.1 and 4.2, the average tensile and shear pullout strength of 3/8" diameter lag screws tested on tire coupons were 45.4 and 55.6 pounds respectively, whereas the ½" diameter lag screws produced 54.4 and 82.8 pounds respectively. On the other hand, 3/8" diameter machine bolts produced almost 180 pounds of tensile pullout strength and 160 pounds of shear pullout strength, which were more than two times the strength of lag screws.

The pullout strength of lag screws is low in these screw-tire connections as tire sidewalls are generally quite soft. Pullout is resisted only by friction between threads and the rubber materials. However, there was already a 9-pound of tensile strength increase resulted from the 1/8" increase in diameter of the screws. Therefore, larger size lag screws are expected to generate greater pullout strength in tire.

The main factor contributed to the higher pullout strength of the machine bolts is the size of washers used on the machine bolts. Washers produce a larger contact surface area on the tire rubber, thus reducing the loading pressure on that area. Higher pullout strength can even be achieved if larger washers were being used.

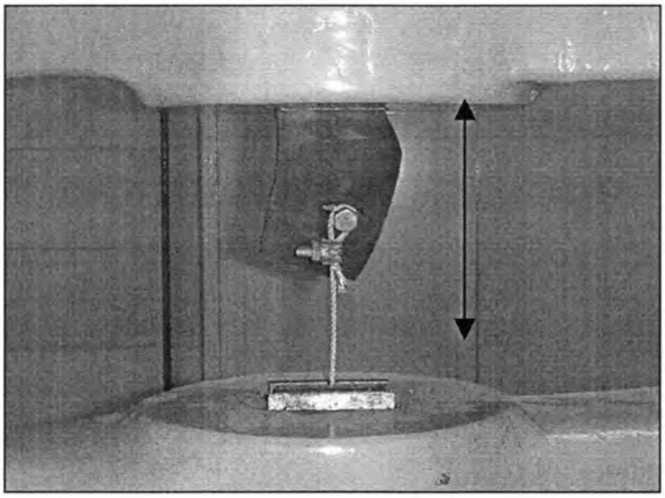


(a) Tensile pullout strength of a lag screw in a tire coupon in testing (Arrow shows the direction of force)

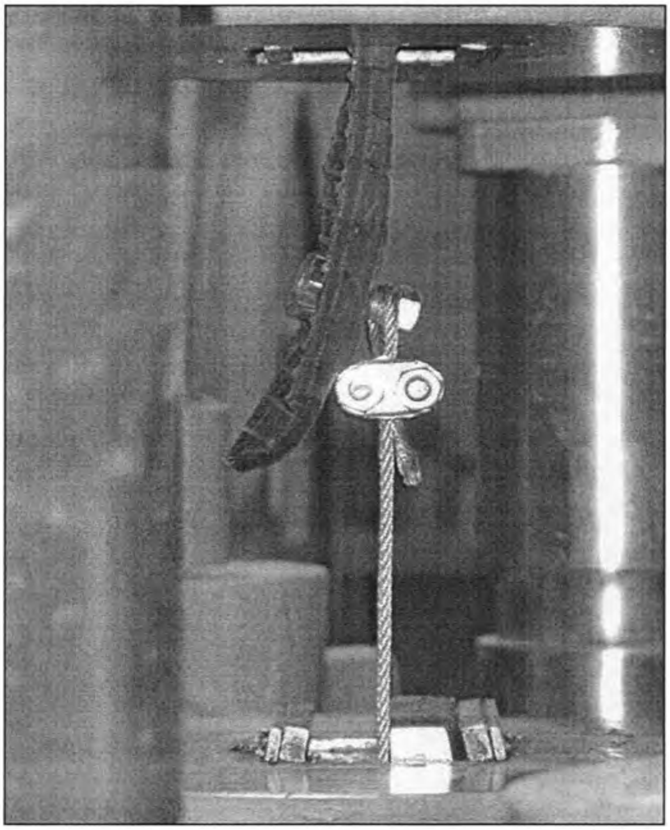


(b) Close up of a machine bolt in a tire coupon in testing

Figure 4.3 Laboratory setup for the tensile pullout strength measurement



(a) Shear pullout strength of a machine bolt in testing



(b) Close up of the machine bolt in testing

Figure 4.4 Laboratory setup for the shear pullout strength measurement

Table 4.1 Tensile Pullout Strength Testing Results of Lag Screws and Machine Bolts

Tire Description	Sidewall Thickness (in.)	Tensile Pullout Strength (lb)		
		Lag Screw		Machine Bolt
		3/8" Diameter	1/2" Diameter	3/8" Diameter
GY P185/75R14	1/4	45.0	55.0	170
	1/4	47.5	60.0	180
	1/4	46.0	52.5	170
GY P205/75R14	1/4	62.5	70.0	200
	1/4	45.0	62.5	200
	1/4	45.0	60.0	180
Liberator II P195/75R14	1/4	35.0	45.0	170
	1/4	40.0	45.0	170
	1/4	42.5	40.0	170
Average =	1/4	45.4	54.4	178.9

Table 4.2 Shear Pullout Strength Testing Results of Lag Screws and Machine Bolts

Tire Description	Sidewall Thickness (in.)	Shear Pullout Strength (lb)		
		Lag Screw		Machine Bolt
		3/8" Diameter	1/2" Diameter	3/8" Diameter
GY P185/75R14	1/4	55	70	150
	1/4	67.5	95	160
	1/4	65	85	150
GY P205/75R14	1/4	55	75	190
	1/4	67.5	90	170
	1/4	50	105	175
Liberator II P195/75R14	1/4	50	75	140
	1/4	45	80	140
	1/4	45	70	135
Average =	1/4	55.6	82.8	156.7

CHAPTER 5. GRADE CONTROL STRUCTURES DESIGN MODEL

5.1 MODEL DESCRIPTION

Analysis and design of stream grade control structures are programmed into a computer model named Grade Control Structures Design Model (GCSDM). GCSDM is a comprehensive, yet simple to use, computer program that utilizes input parameters provided by the user to assist in grade control structure design. The calculation results may be displayed on the computer screen, printed on paper, or output to a computer disk file. Input parameters used in one execution or run of the program may be stored under a name specified by the user and reused for subsequent trials. For ease of use, the program is designed with menus and selection buttons to guide the user through the design process. Detailed information on navigating through the program is presented in Section 5.2 of this chapter.

5.1.1 Model Structure

GCSDM was developed using Microsoft Visual Basic. As shown in Figure 5.1, the model consists of four modules: the data input module, data pre-processing module, stability analysis module, and the data post-processing module. The first module in GCSDM, the data input module, is the most important module in the model as different input will alter the model results. The data pre-processing module is developed primarily to assist in the preparation of the data required in the stability analysis module. Results from the stability analysis module are used in the data post-processing module for presentation of results.

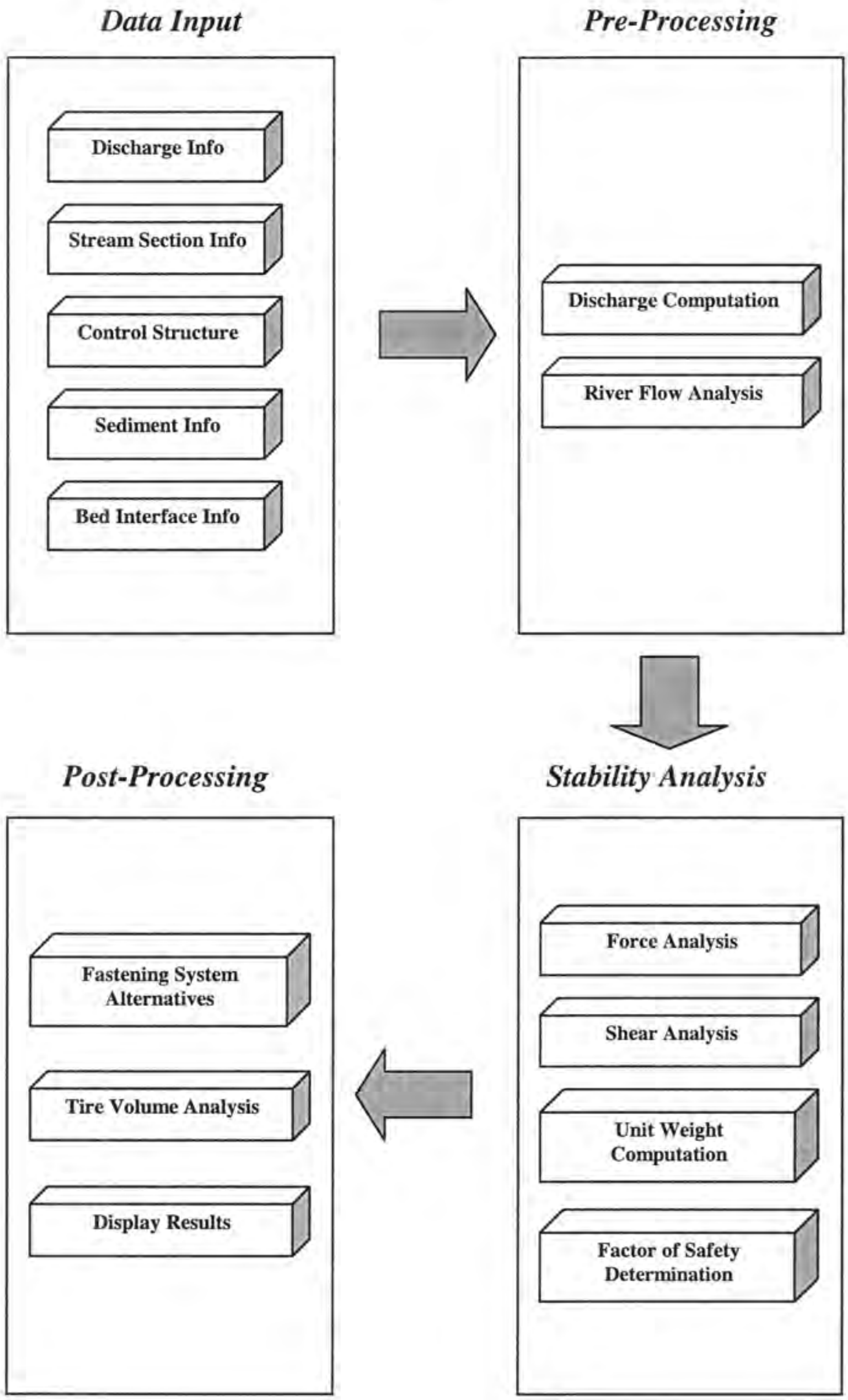


Figure 5.1 A Schematic of GCSDM Model Structure

5.1.2 Data Input Module

Data input module is divided into five different fields: discharge, stream section, control structure dimension, sediment, and bed interface information field. Each information field includes a number of input parameters. GCSDM cannot produce a valid run unless acceptable values are entered for every input parameter on the input screen. If an entered value is out of the acceptable range of that parameter, a warning message will be displayed.

5.1.3 Pre-processing Module

The functionalities of the pre-processing module are to estimate the flow rate at the stream cross section of a future construction site and to run river flow analysis. Discharge equation (Equation 2.1) widely used by the Iowa Department of Transportation (IDOT) is used to estimate the flow rate. This module also uses the open channel flow theory discussed in Section 2.1 to perform river and hydrologic analysis. Equations taken from the theory are Manning's, energy and continuity equations. The model also assumes uniform channel flow to determine the pre-construction water depth.

5.1.4 Stability Analysis Module

Force analysis is performed by simple multiplication of the volume of components of forces and their unit weights, as presented in Section 2.2. For the analysis of shear capacity of bed materials, the model uses Mohr-Coulomb failure criteria. In order to calculate the sediment forces on the dam, the model also uses Rankine's theory of active earth pressure. The user can also use this model to determine the required unit weight of the tire structure to resist both uplift and sliding. Safety factors to overcome the uplift and sliding are computed.

5.1.5 Post-processing Module

There two main purposes of this post-processing module are to perform tire volume analysis to determine the required unit weight of filing materials and to select and recommend the appropriate fastening system for the tire grade control structure based on a factor of safety of 1.5. The present fastening systems being considered in this module are lag screws, machine bolts and steel cable connections. The user can use the selected fastening system as guideline to complete the design process. The model results can be presented in tabular form displayed on screen, printed, or stored in a file for later use.

5.2 PROGRAM IMPLEMENTATION

This section provides a detailed step-by-step approach for implementing the Grade Control Structure Design Model (GCSDM). In general, users can explore the interface by making appropriate selections in each menu, button and dialog options. Table 5.1 summarizes the major steps to accomplish a model simulation using the interface.

The GCSDM operates under the Windows 95 or later environment. This program comes in a floppy disk. Before using the model, the user must create a directory named ***user*** under the c: drive, and copy the a: drive's ***tire model*** directory to the ***user*** directory. Next, the user needs to go to the ***user\tire model*** directory on c: drive and click on the ***GCSDM.exe*** file. In the opening interface, a number of pulldown menus and buttons are displayed (Fig. 5.2). The entire GCSDM environment and Graphic-User Interface (GUI) are contained in the main application window. All the user interactions take place in this window. The output of the GCSDM is also displayed in this window.

Table 5.1 Steps in the implementation of GCSDM

Start Up

1. Create a new directory named **user** under c: drive.
2. Copy a: drive's **tire model** directory to c:\user
3. Go to **c:\user\tire model** directory and click on the **GCSDM.exe** file.

   A new window will appear on the computer screen.

Data Entry

1. Click on the Input-Start button or pull down the Input menu and select an option.

   A new dialog window of each input parameters will appear on the screen.
2. Enter all input fields.
3. Click "Done" when done.

   Entered data will be displayed under the Stability Analysis section.

Open Previously Saved Data

1. Go to File menu and select Open Previous Data, or press Ctrl+O.

   Previously entered data will be displayed under the Inputted Data section.

Save Data

1. Go to File menu and select Save, or press Ctrl+S.

   All entered data will be saved in c:\user\tire model\inputteddatafile.txt.

Save Data As

1. Go to File menu and select Save As, or press Ctrl+A.

   A File Save dialog box will appear on the screen. A file name has to be either selected or entered. All entered data will be saved in that selected or entered file.

Run GCSDM

1. Click on Compute-Stability Analysis button or pull down the Compute menu and select Run. Output is displayed under the Stability Analysis section.

Table 5.1 (Continued)

Save Output to File

1. Click on Output-Save to File button or pull down the Output menu and select Save to File.
   A File Save dialog box will appear on the screen. A file name has to be either selected or entered. All output would be saved in that selected or entered file.

Print Output

1. Click on Output-Print button or pull down the Output menu and select Print Output.
   Output will be sent to a default printer.

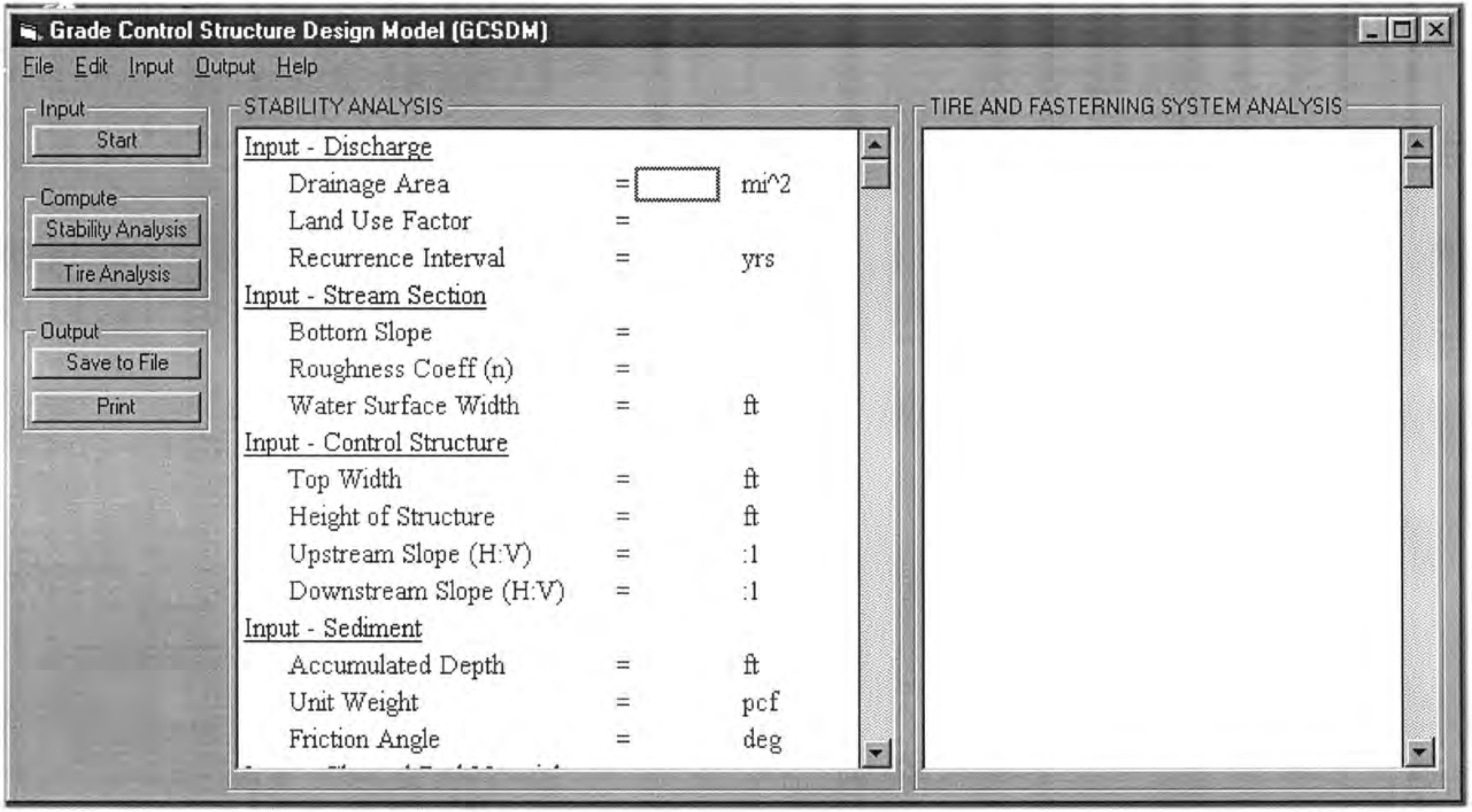
Help

1. Go to Help menu and select Help. Help dialog menu will appear on the screen.

Exit GCSDM

1. Go to File menu and select Exit.

There are several ways to interact with GCSDM. Pulldown menus are the most conventional way. Choosing menu items communicates commands to GCSDM by executing embedded scripts (programs). Menus can be pulled down by pressing a menu bar option's hot key (for example, Alt+F displays the File menu), or with one mouse click. Many of the menu options have shortcut keys such as Ctrl+S for the Save option under the File menu. When the user presses the shortcut key, the user does not need to display the menu first to access the option. Actions are also conveyed with buttons. The button section on the left hand side of the window provides one-button access to many common menu commands. Instead of selecting Discharge under the Input menu, the user can click the Input-Start button.



Input - Discharge
Drainage Area = mi^2
Land Use Factor =
Recurrence Interval = yrs
Input - Stream Section
Bottom Slope =
Roughness Coeff (n) =
Water Surface Width = ft
Input - Control Structure
Top Width = ft
Height of Structure = ft
Upstream Slope (H:V) = :1
Downstream Slope (H:V) = :1
Input - Sediment
Accumulated Depth = ft
Unit Weight = pcf
Friction Angle = deg

Figure 5.2 Opening interface of GCSDM

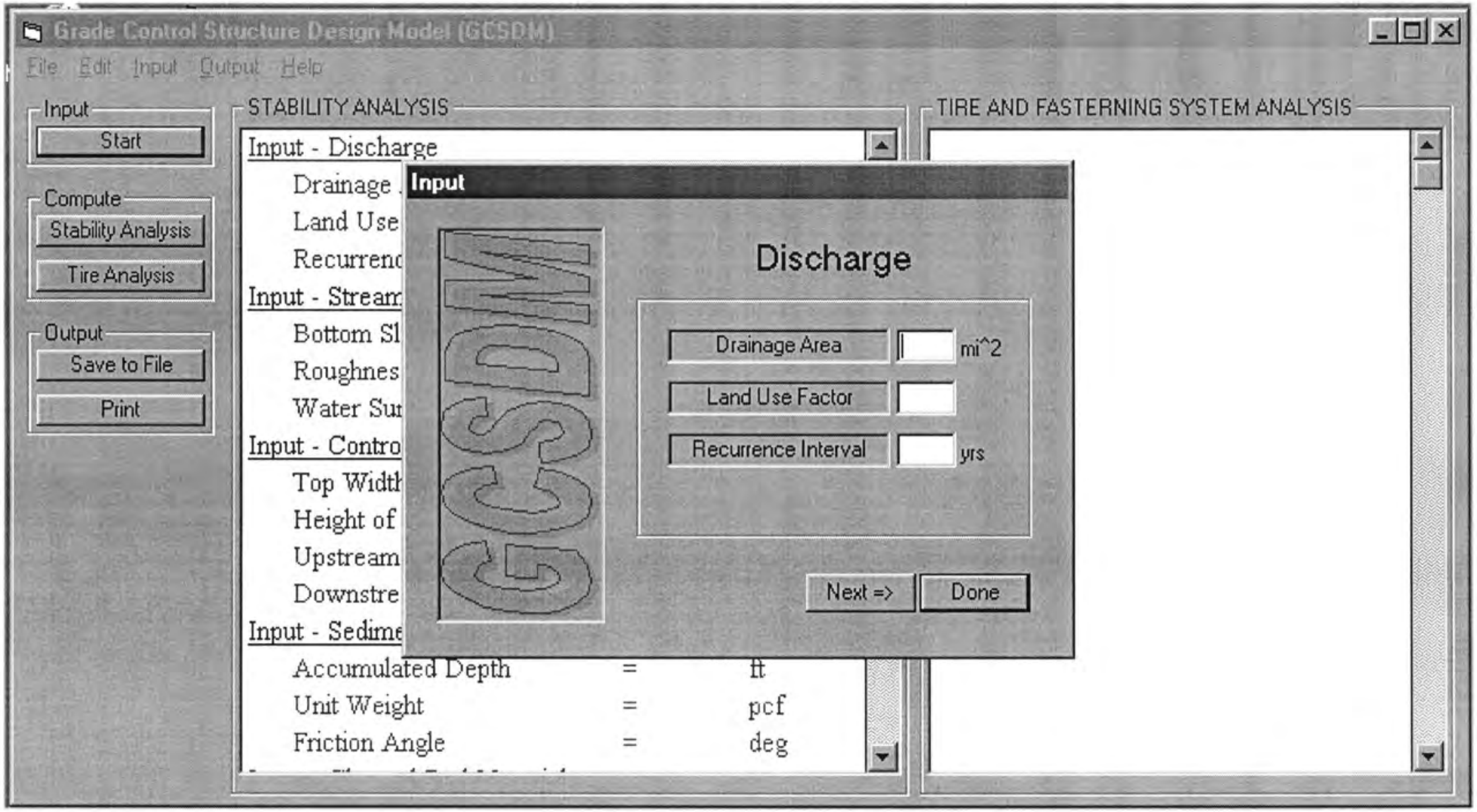
Again, a single mouse click activates the button. Clicking on a button may produce an immediate action or bring up an additional window, depending on the tool.

When GCSDM is first loaded, all the input parameters will contain no data. To begin using the model, the user needs to enter a number of input parameters, by either clicking the Input-Start button or pulling down the Input menu and select. Either way, an additional input window, which is also known as the dialog window, will pop up (Fig. 5.3). Values have to be entered into those specified boxes of each dialog window. After the data are placed into the appropriate boxes, the data will be stored and displayed by clicking the "Done" button (Fig. 5.4).

After the parameters have been entered, the program will run after the user clicks the Compute-Stability Analysis button. River flow, force and stability analysis are performed. Horizontal reaction, uplift force, required dry unit weight of the control structure and the factor of safety against sliding are computed in the force analysis. Computed parameters are displayed on the interface automatically, as shown in Figure 5.5. Input parameters used in this execution of the program may be stored under a name specified by the user and reused for subsequent runs.

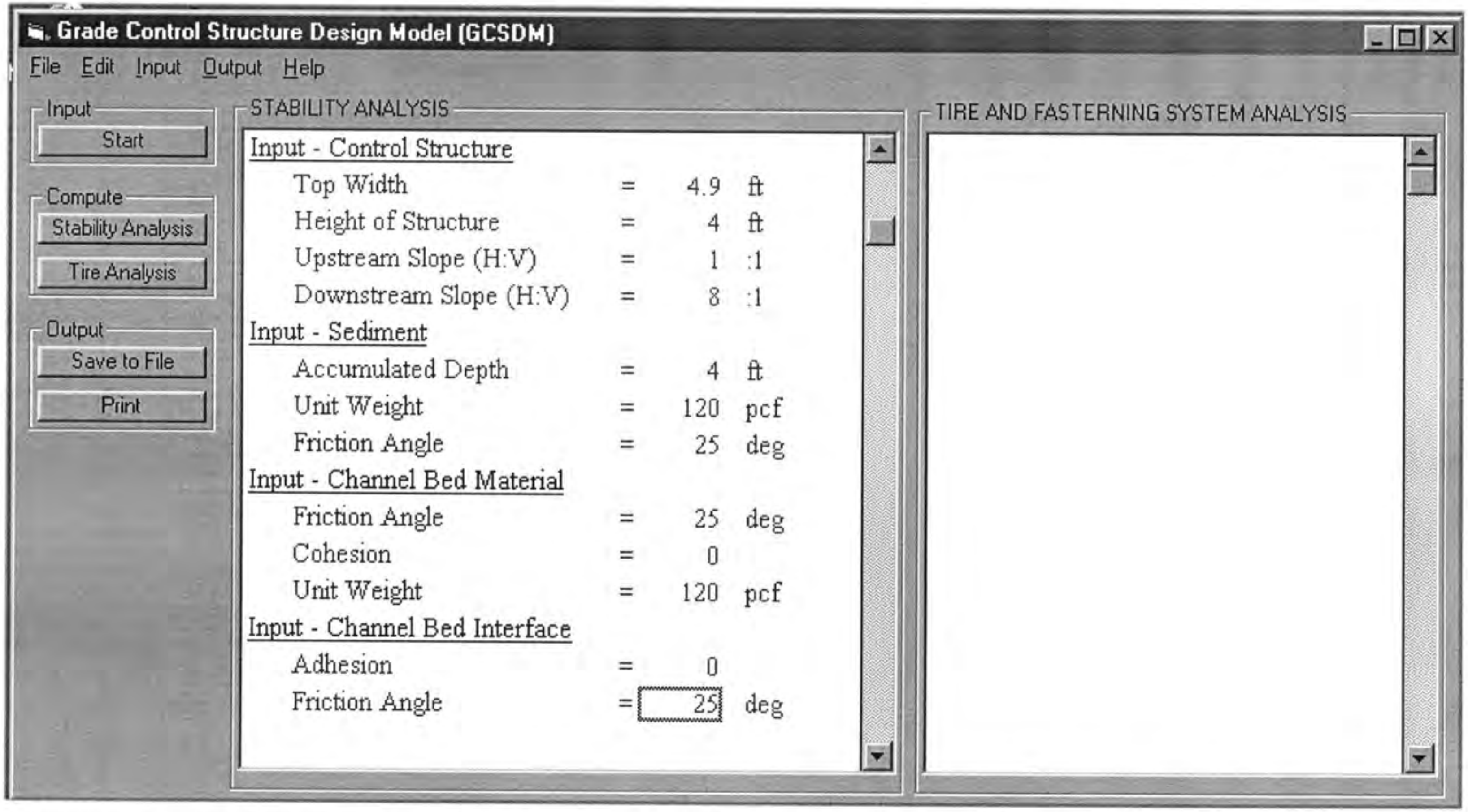
The input parameters can also be edited either through the Input-Start buttons or the Input pulldown menu. If these parameters are changed, GCSDM must run again to do the new computations. The output information from the model can also be sent to the printer or saved as a file.

This version of the GCSDM program also comes with tire and fastening system analysis model. To perform this analysis, the user needs to click the Compute-Tire Analysis button. An optional input window will pop up, as shown in Figure 5.6. Default answers are



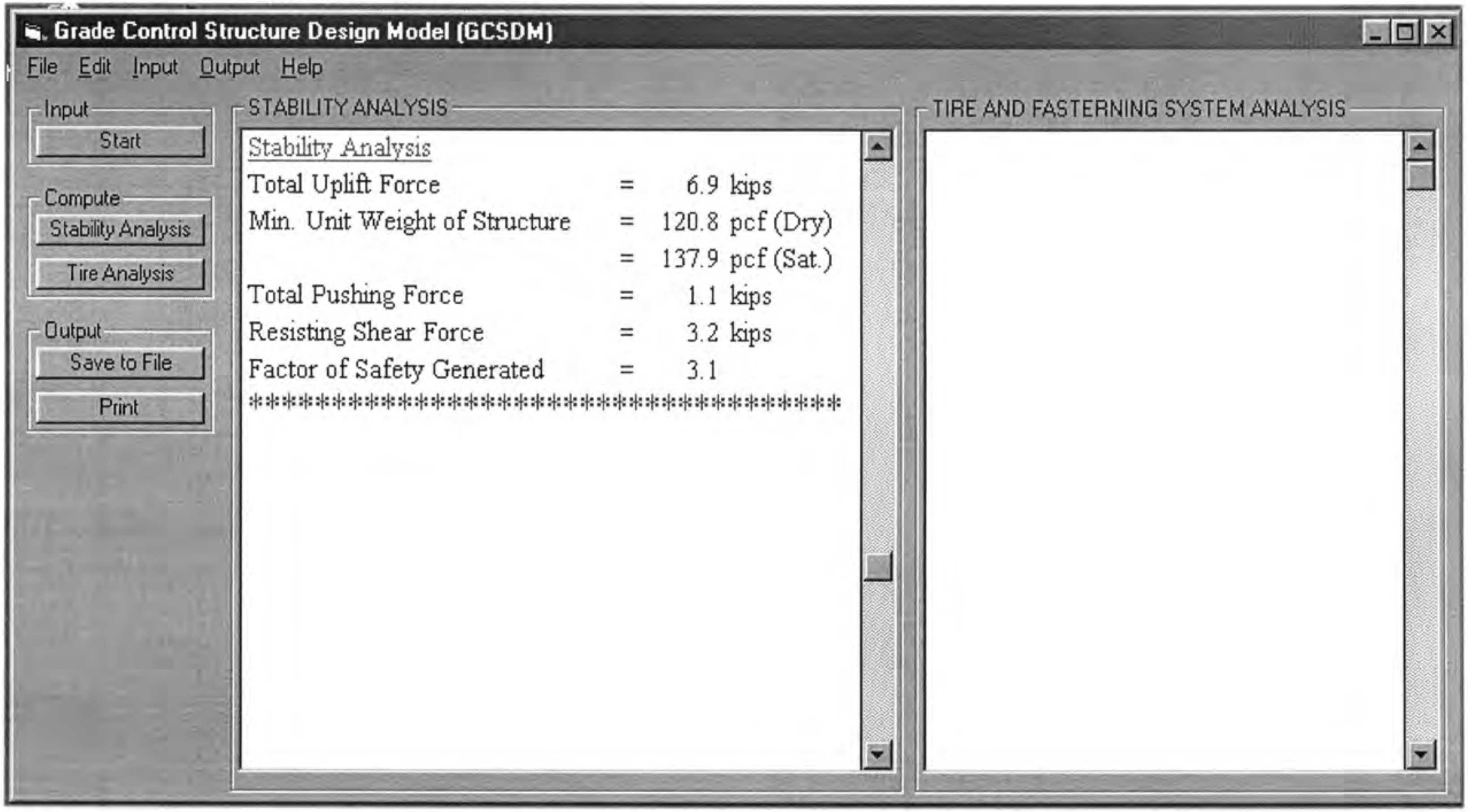
```
Input - Discharge
    Drainage
    Land Use
    Recurrenc
Input - Stream
    Bottom Sl
    Roughnes
    Water Su
Input - Contro
    Top Width
    Height of
    Upstream
    Downstre
Input - Sedime
    Accumulated Depth         =        ft
    Unit Weight               =        pcf
    Friction Angle            =        deg
```

Figure 5.3 Clicking on the Input-Start button or pulling down the Input menu and select Discharge brings up an additional input dialog window.



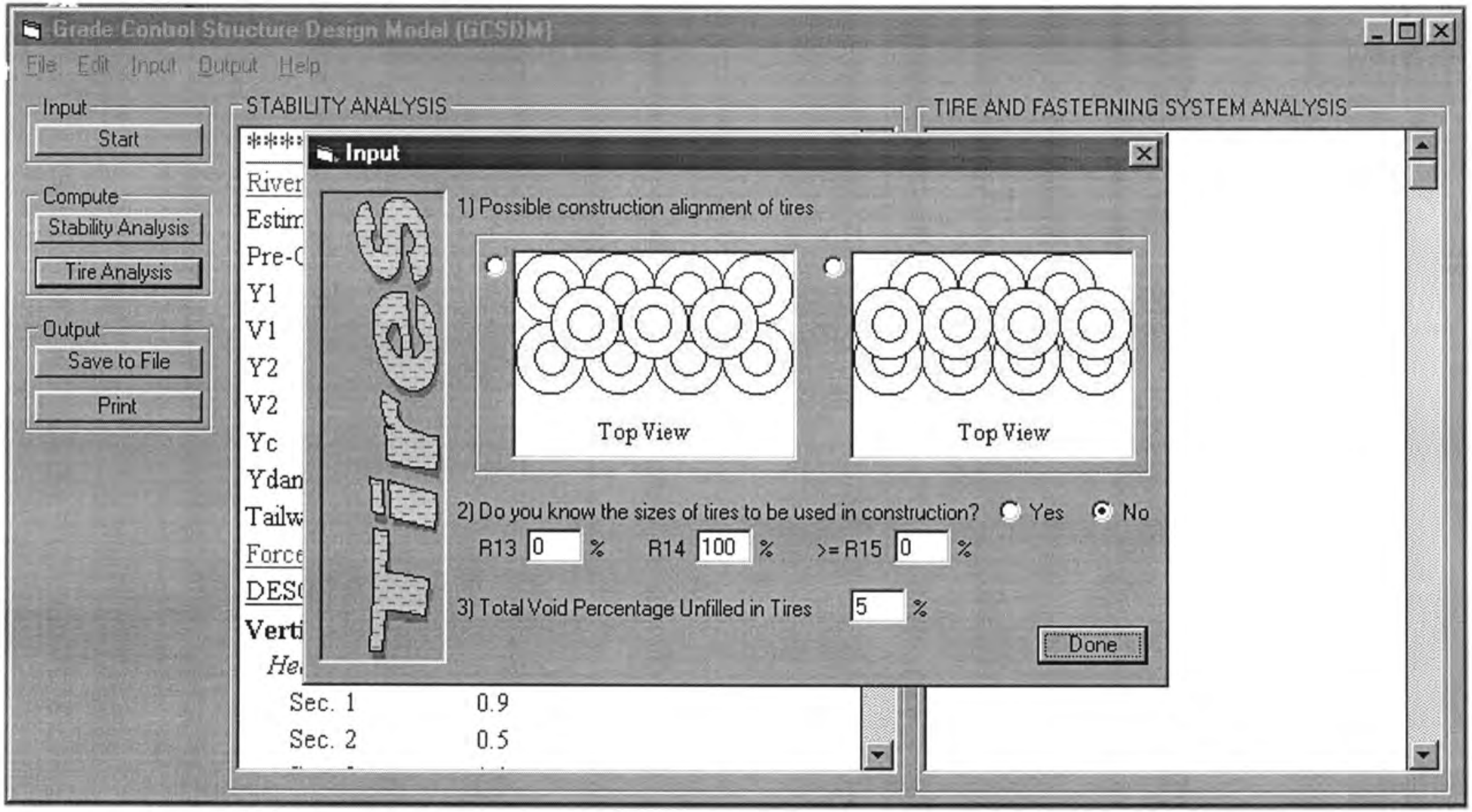
Input - Control Structure
Top Width = 4.9 ft
Height of Structure = 4 ft
Upstream Slope (H:V) = 1 :1
Downstream Slope (H:V) = 8 :1
Input - Sediment
Accumulated Depth = 4 ft
Unit Weight = 120 pcf
Friction Angle = 25 deg
Input - Channel Bed Material
Friction Angle = 25 deg
Cohesion = 0
Unit Weight = 120 pcf
Input - Channel Bed Interface
Adhesion = 0
Friction Angle = 25 deg

Figure 5.4 Entered data are displayed in the Stability Analysis section



```
Stability Analysis
Total Uplift Force                =    6.9 kips
Min. Unit Weight of Structure     =  120.8 pcf (Dry)
                                  =  137.9 pcf (Sat.)
Total Pushing Force               =    1.1 kips
Resisting Shear Force             =    3.2 kips
Factor of Safety Generated        =    3.1
************************************
```

Figure 5.5 Stability analysis outputs are displayed in the Stability Analysis section



1) Possible construction alignment of tires

2) Do you know the sizes of tires to be used in construction? ○ Yes ● No

R13 0 % R14 100 % >= R15 0 %

3) Total Void Percentage Unfilled in Tires 5 %

Sec. 1 0.9

Sec. 2 0.5

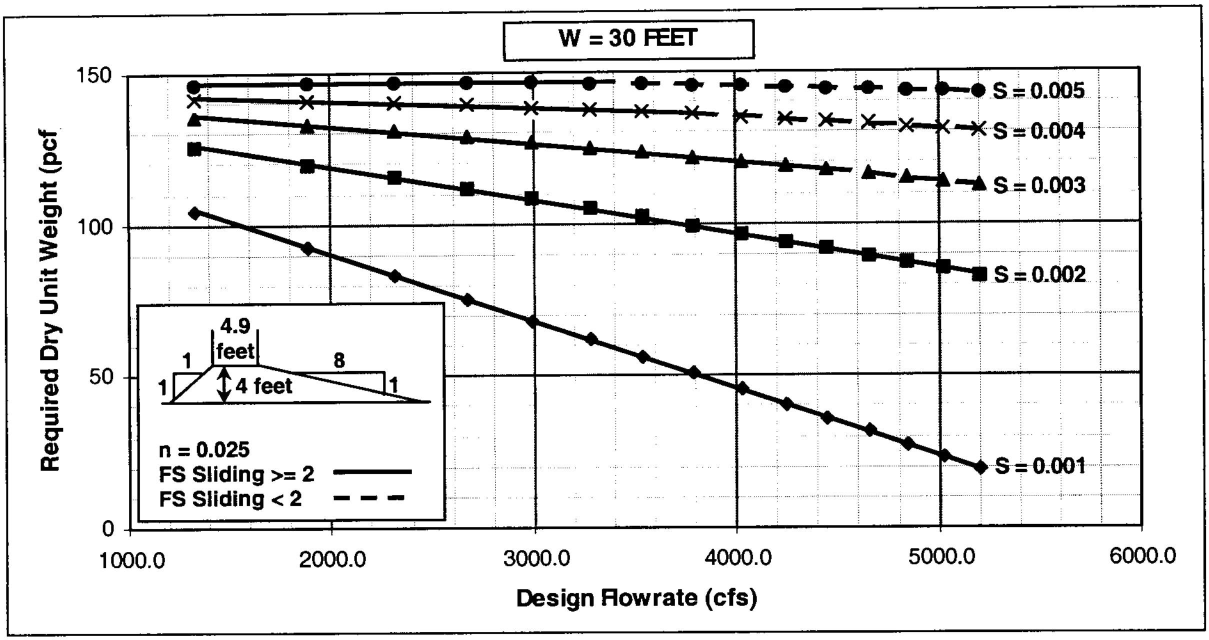
Figure 5.6 The input interface of tire and fastening system analysis model given to the three questions. If the user does not wish to change them or does not know the answers to those questions, he may just click “Done” to perform the analysis. The analysis program will run once the user clicks the “Done” button. Again, outputs from this analysis will be displayed on screen.

To enhance use and minimize misuse of the modeling environment, a Help menu is available to provide information about the interface and its various modeling options. GCSDM also displays “tool tips.” These text descriptions of a button’s function appear inside small yellow boxes, which pop up when the user move the mouse cursor over a button and pause. Upon completing the model simulations, the user can exit GCSDM by simply clicking the Exit option under the File menu.

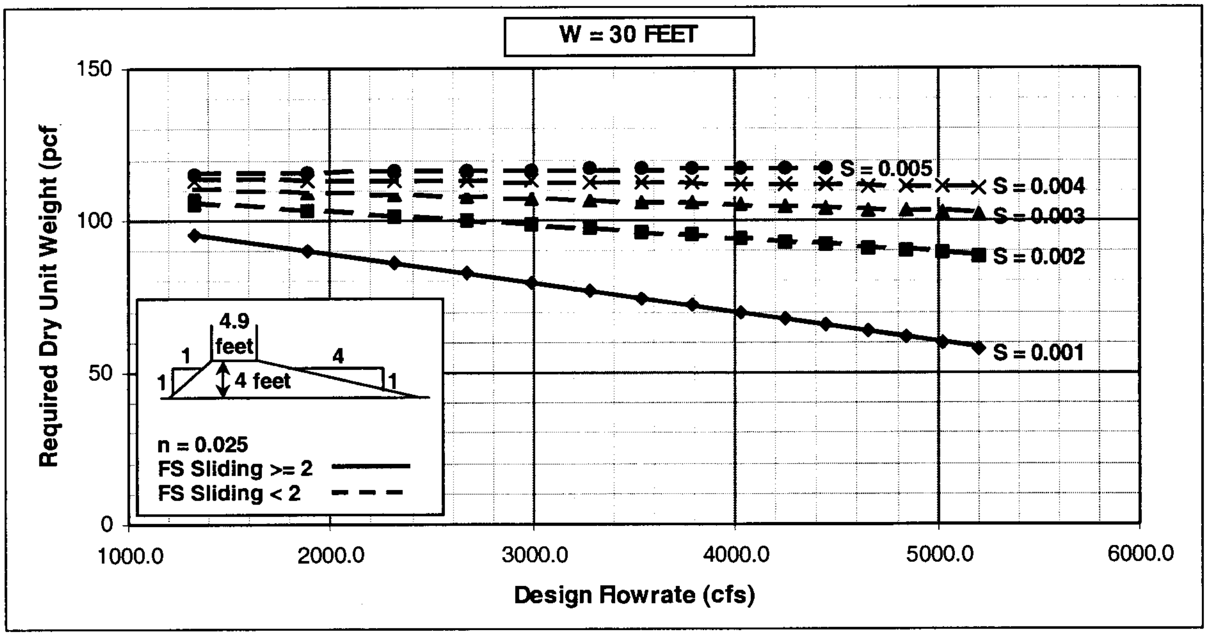
5.3 DESIGN CHARTS

Various design charts can be plotted by compiling the outputs from GCSDM model. A few of the charts are shown in Figure 5.7, 5.8 and 5.9. Users can use those charts to determine the minimum required unit weight of a trapezoidal grade control structure under different conditions quickly and easily.

The two main elements differentiating Figure 5.7, 5.8 and 5.9 are the section width (W) of the stream where the structure is located and the downstream slope (S) of the structure. The three figures are prepared for grade control structures in 30-foot, 40-foot and 50-foot width streams, respectively. Part (a) of each figure is for structures with 8:1 downstream slope only, whereas part (b) is for 4:1 downstream slope. Manning’s roughness of each stream section is assumed to be 0.025. There are also solid and dashed lines in each

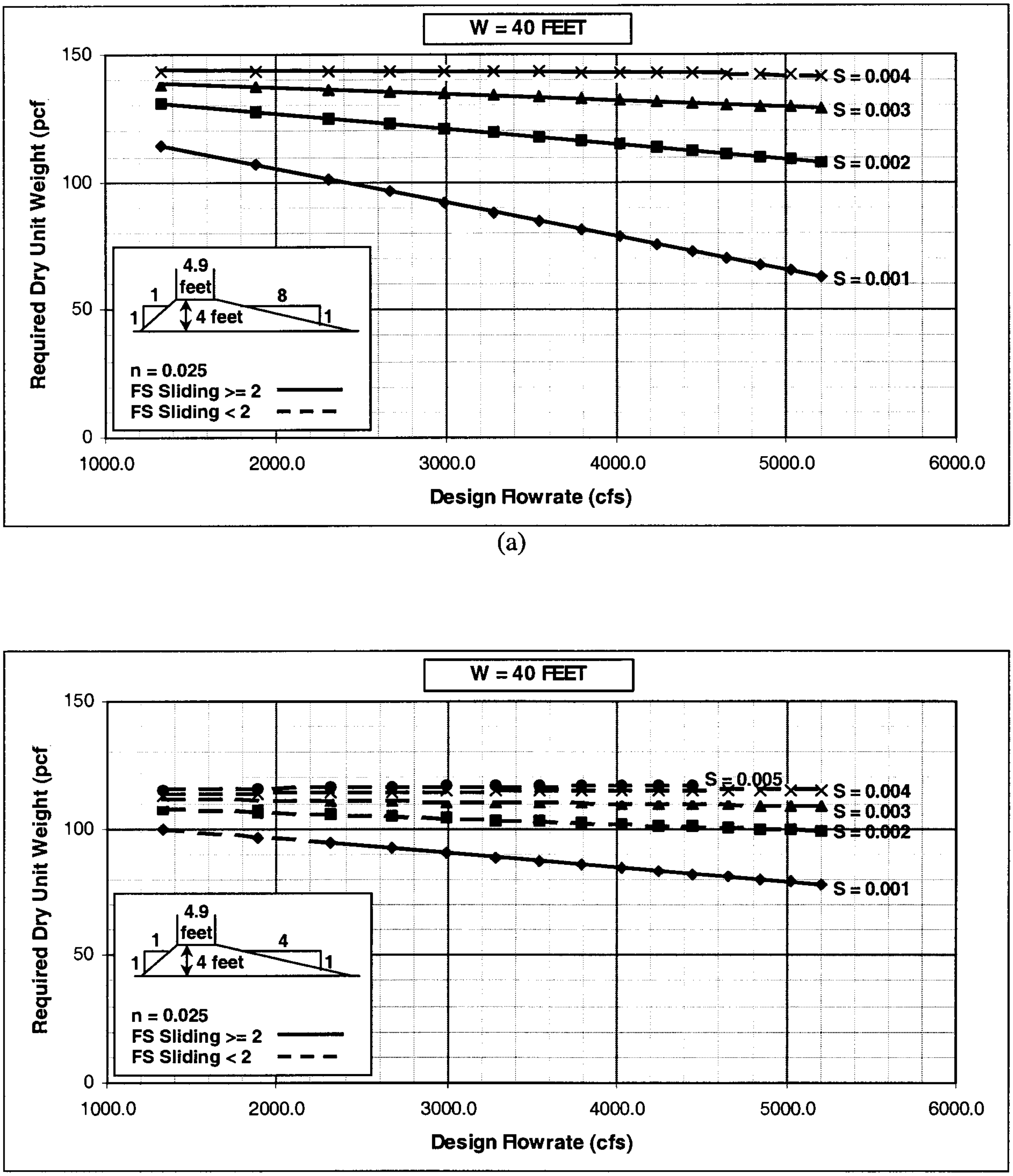


(a)



(b)

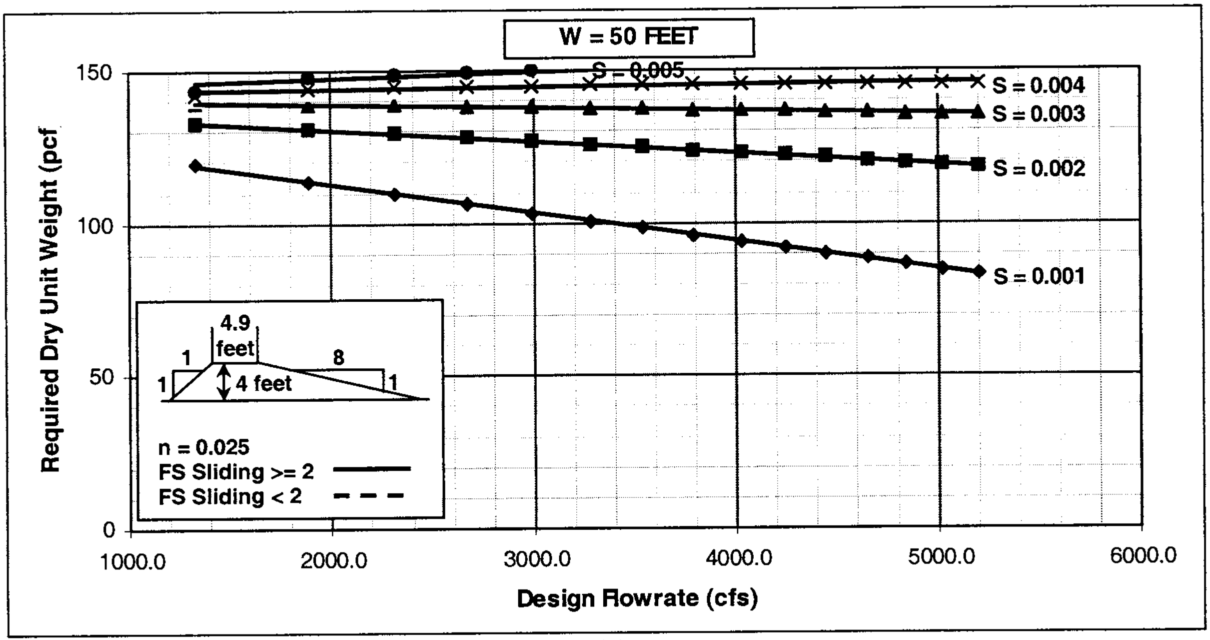
Figure 5.7 Dry unit weight requirement for trapezoidal grade control structure in 30-foot width stream. (a) and (b) are for structures with 8:1 and 4:1 downstream slopes respectively.



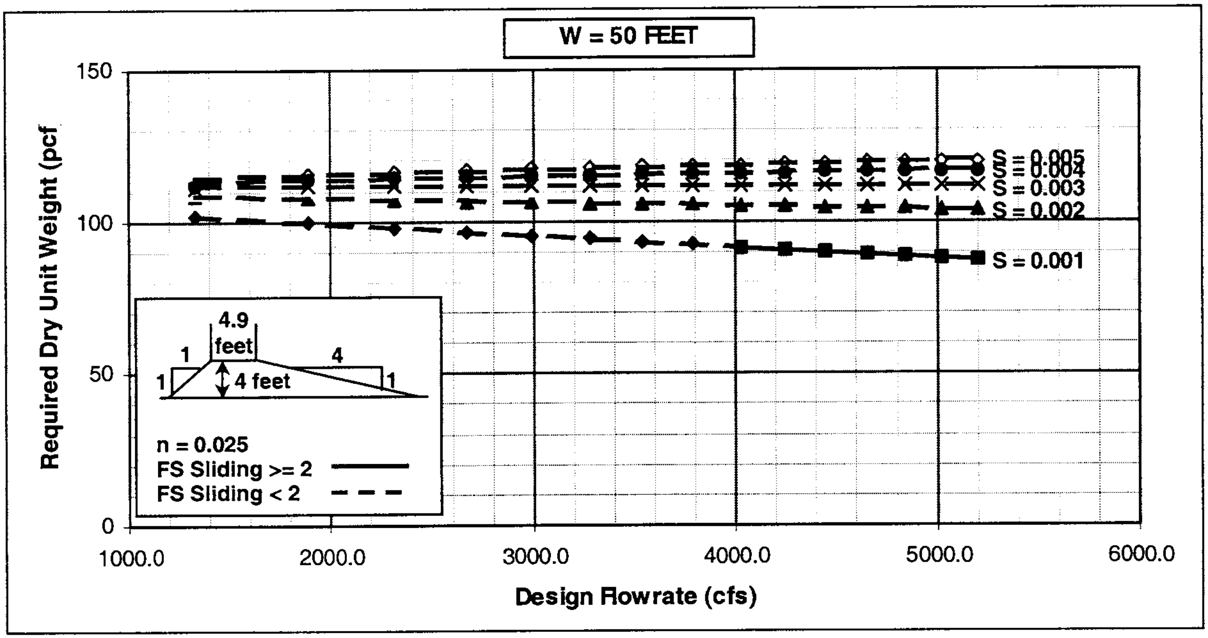
(a)

(b)

Figure 5.8 Dry unit weight requirement for trapezoidal grade control structure in 40-foot width stream. (a) and (b) are for structures with 8:1 and 4:1 downstream slopes respectively.



(a)



(b)

Figure 5.9 Dry unit weight requirement for trapezoidal grade control structure in 50-foot width stream. (a) and (b) are for structures with 8:1 and 4:1 downstream slopes respectively.

design charts. When a dry unit weight is able to produce a factor of safety against sliding of 2 or greater, the section of the line is solid; it is dashed for a factor of safety of 1 and above.

Users need to have both the design flowrate and the channel bottom slope computed before using the design charts. By referring to correct chart, the required dry unit weight of the control structure can then be determined quickly and easily. However, if there are major variations among the chart assumptions and the design conditions, GCSDM model can be used.

Having the design charts set up can definitely help designers in determining the unit weight requirement of each grade control structure under different design conditions more effectively. Generally, a factor of safety against sliding of 1.5 or greater can be achieved by using a construction material with unit weight higher than the required unit weight.

CHAPTER 6. ILLUSTRATIVE EXAMPLES

6.1 INTRODUCTION

A better understanding of the design process of tire grade control structures can be achieved from the two illustrative examples in this chapter. Most of the procedures involved in the design process are explained in detail in the previous chapters and are programmed into the GCSDM model. However, the point cannot be over emphasized that to properly appreciate and utilize the GCSDM model, the users should first become familiar with the long hand methods. Design charts are very easy to use and produce good estimates; however, it is recommended that the design charts should be used only as a preliminary design. GCSDM can always produce more accurate results, as all input variables required by the model are being considered in the analysis. However, GCSDM does require the users to spend more time in data collection. Users should redo the analysis using the GCSDM model, and if possible, verify the analysis results using the long hand methods.

The same drainage basin and channel section are used in the two examples. Example 1 utilizes the design charts presented in Section 5.3 to determine the unit weight requirement for the tire grade control structure. Example 2 shows how to perform the force and stability analysis using the GCSDM model, and is included to demonstrate the use of the model. The unit weight requirement produced by GCSDM is compared with the unit weight determined from the design charts. Safety factors to overcome sliding and uplift determined by the model are analyzed. The strength of the fastening system recommended by the model is checked and factor of safety of the system is computed.

6.2 DESCRIPTION OF TIRE STRUCTURE AND STUDY REACH

A 4 feet high structure with 4.9 feet top width, 1 to 1 fore slope and 8 to 1 back slope is to be constructed. The study reach is a natural rectangular channel. The stream is essentially straight and the cross section is relatively constant in the vicinity of the structure. The following parameters apply to the study reach:

Drainage area upstream = 50 square miles
Average channel bottom slope = 0.002 feet/feet
Roughness coefficient, n = 0.025
Average channel section width = 50 feet

Land in the drainage area is very hilly. Most of the area was originally covered by a continuous mosaic of prairies, forests and wetlands. After the arrival of the Euro-American settlers in the mid – 1800's, the original landscape was gradually converted to agricultural use. Nowadays, most of the area in the drainage basin is pasture and woodland.

6.3 ANALYSIS AND RESULTS

6.3.1 Example 1 – Design Chart

The four main variables needed to use the design charts are the downstream slope of the structure, channel section width, bottom slope, and the design flow rate at the channel section where the structure is located. The first two variables are needed to locate the appropriate chart for the analysis of this grade control structure. With a structure downstream slope of 8:1 and a section width of 50 feet, Figure 5.9(a) is the correct chart to use. Figure 6.1 is a replica of Figure 5.9(a).

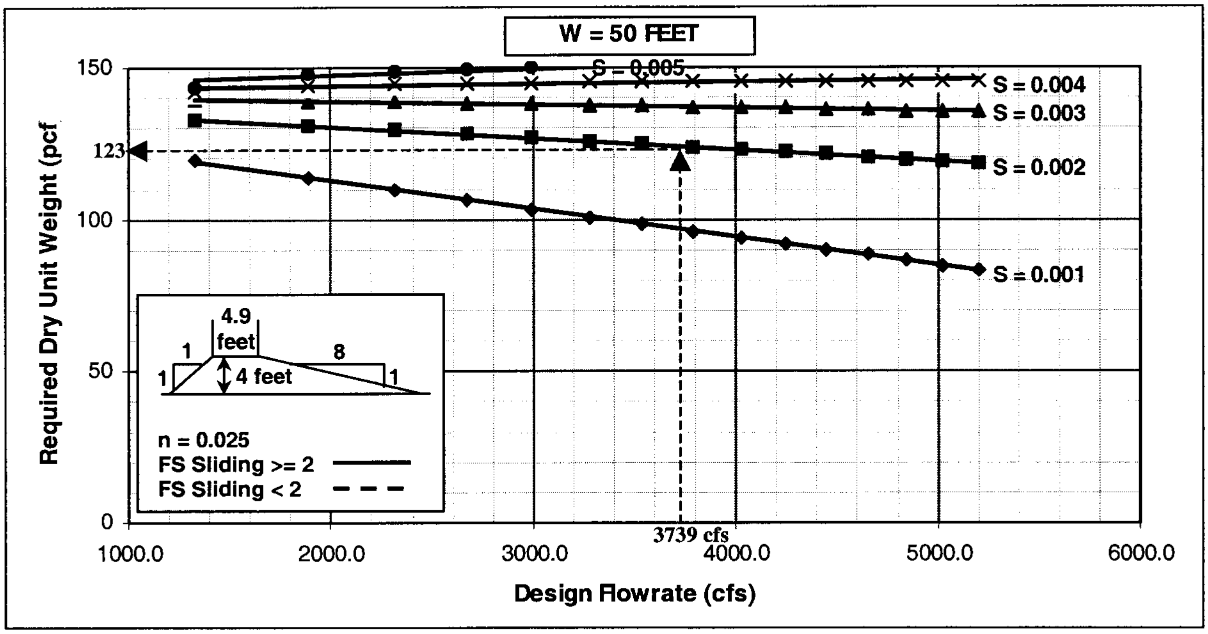


Figure 6.1 Design chart used in the illustrative example 1

Equation 2.1 is used to estimate the design flow rate. Equation 2.1 is written as

$$Q = 422.58 \times (LF) \times (RI)^{.301} \times (D_A)^{0.504} \qquad (6.1)$$

Referring to Table 2.1, the land use factor (*LF*) for the conditions stated above is 1.0. Designing for a 2-year recurrence interval (*RI*), the design flow rate is estimated as follow:

$$Q = 422.58 \times (1) \times (2)^{.301} \times (50)^{0.504}$$

$$Q = 3739 \text{ cfs}$$

Entering Figure 6.1 with $S = 0.001$ and $Q = 3739$ cfs, the required dry unit weight of the structure is estimated as 123 pcf.

6.3.1 Example 2 – GCSDM Model

In addition to providing the required unit weight requirement, GCSDM also calculates the factor of safety of the structure against uplift and sliding. Fastening systems that are able to overcome the hydraulic forces in the structure are also recommended based on a minimum factor of safety of 1.5.

After entering the input variables into the GCSDM, the required dry unit weight computed is 124 pcf, which is just 1 pcf higher that the unit weight determined previously. The stream grade control structure with 4-foot height, 4.9-foot top width, 1:1 upstream slope and 8:1 downstream slope is able to resist all uplift and pushing forces due to flowing water, as long as its dry unit weight is equal or greater than 124 pcf. Factor of safety against uplift and sliding is 1.0 and 2.4, respectively. 1.0 safety factor for uplift is the default number in GCSDM. The safety factor can be increased by increasing in required unit weight of the structure.

If there is a situation where the unit weight requirement of a control structure is below the unit weight of concrete, which is about 150 pcf, the designer can replace concrete as the construction material with other cheaper alternatives, such as riprap and cement stabilized soil, whose unit weight is in the range of 110 to 130 pcf. However, the designer should also consider the hydraulic resistance of those alternatives before selecting them.

Two of the three default fastening systems in GCSDM are able to resist all hydraulic forces within the structure. 3/8" diameter machine bolt with 7/8" outside diameter washer and steel cable are able to produce a factor of safety of at least 1.5. The designer can select either one of them. These systems should be applied among the tires of the same layer and layers above and below it. Connections are to be made at all contact locations between the tires.

CHAPTER 7. CONCLUSIONS AND FUTURE WORK

The use of whole scrap tires as construction materials is of growing interest. Scrap tires can be used in erosion control, stream bank protection and slope stabilization. The feasibility of using scrap tires in grade control structures has been investigated in this study. The stability of the structures as a whole, the fastening system and filling materials were analyzed and evaluated. Theoretical studies were first performed to analyze soil and water forces acting on the grade control structures. The calculated forces were then used to determine the minimum requirement for the pullout strength of a fastening system in tires and the required unit weight of the structure to resist both uplift and sliding. Fastening systems evaluation and the system strength measurements were carried out in laboratories tests. Analysis and design recommendations of tires grade control structures were programmed into a computer model named GCSDM. Design charts were also produced by compiling the outputs from GCSDM developed in this study.

Stream grade control structures with a height of 4 feet, top width of 4.9 feet, 1:1 upstream slope and either 8:1 or 4:1 downstream slope can resist all uplift and pushing forces due to flowing water, as long as their unit weight are equal or higher than the requirement determined by the GCSDM model or design charts. Having the design charts set up can assist designers in determining the unit weight requirement of each grade control structure under different design conditions more effectively.

Generally, a factor of safety against sliding of 1.5 or greater can be achieved by using a construction material with unit weight higher than the required unit weight. Factor of safety against uplift is always 1.0. The required unit weight of the structure can be increased at will to raise the factor of safety against sliding. Referring to Figure 5.7, 5.8 and 5.9, structures with 8:1 downstream slope require higher unit weight to achieve stability compared to structures with just 4:1 slope. However, they can generate higher factor of safety against sliding.

An obvious solution to the unit weight requirement is to fill the tires with concrete; however, this solution will add cost that is likely to be excessive. Several other fill materials are to be considered including, but not limited to, crushed concrete and cement stabilized soil. The unit weight of these materials has to be determined to calculate the resistance forces. The hydraulic resistance of tires containing the fill materials is also to be tested.

The number of car tires required by various structures is estimated based on their structure geometry, and is shown in Table 3.1. The number for other structure geometry can also be approximated in the GCSDM model.

More research is necessary to study the constructability of the tires grade control structures. The comparison of various systems to determine which one has the greatest ease of constructability is recommended. The placement of the filling materials should be studied to reduce the amount of voids in the structure.

APPENDIX. GCSDM VISUAL BASIC SOURCE CODE

Form1.frm

```
Option Explicit
   Dim n As Long, num As Long, b As Long
   Dim tmptext1() As String, tmpvalue1() As Long
   Dim tmpvalue2() As Double, tmpunit() As String
   Dim tmptexta() As String, tmpvaluea() As Long
   Dim tmpvalueb() As Double, tmpunita() As String

   Dim strFilename As String, CheckingResult As String, result As String

   Dim DA As Double, LF As Double, RI As Double, StreamSlope As Double
   Dim Roughness As Double, StreamWidth As Double, DamTopWidth As Double
   Dim DamHeight As Double, DamUpstreamSlope As Double
   Dim DamDownstreamSlope As Double, SedDepth As Double
   Dim SedSatUnitWeight As Double, SedFrictionAngle As Double
   Dim BedMatFrictionAngle As Double, BedMatCohesion As Double
   Dim BedMatUnitWeight As Double, BedInterfaceFrictionAngle As Double
   Dim BedInterfaceAdhesion As Double, Flowrate As Double

   Dim Y1 As Double, Area1 As Double, V1 As Double
   Dim Yc As Double, Areac As Double, Vc As Double
   Dim Y2 As Double, V2 As Double, Velocity2 As Double
   Dim Y2Test As Double, V2Test As Double, Velocity2Test As Double
   Dim Ydam As Double, tailwater As Double, NoDamDepth As Double

   Dim ka As Double, damlength As Double, HorizontalReaction As Double
   Dim TotalForce1to4 As Double, TotalCG1to4 As Double
   Dim TotalForce5to6 As Double, TotalCG5to6 As Double
   Dim TotalForce7to71 As Double, TotalCG7to71 As Double
   Dim TotalForce8to9 As Double, TotalCG8to9 As Double
   Dim TotalForce10 As Double, TotalCG10 As Double
   Dim TotalForce11 As Double, TotalCG11 As Double
   Dim DamMinSatUnitWeight As Double, DamMinDryUnitWeight As Double

   Dim DamSec1Vol As Double, DamSec2Vol As Double, DamSec3Vol As Double
   Dim DamSec1Force As Double, DamSec2Force As Double, DamSec3Force As Double
   Dim DamSec1CG As Double, DamSec2CG As Double, DamSec3CG As Double
   Dim FlowUplift As Double, FlowDrag As Double
   Dim NetUpliftForce As Double, DamMinEffUnitWeight As Double
   Dim DamTotalWeight As Double, DamTotalCG As Double
   Dim hypo As Double, SinAngle As Double, DamPropulsiveForce As Double
   Dim VerticalReaction As Double
   Dim Vdam As Double, ShearStress As Double
   Dim AreaTrapezium As Double, HeightEquivalent As Double
   Dim ResistingShearStress As Double, ResistingShearForce As Double
   Dim FSSliding As Double, FSUplift As Double, I As Long

   Dim NumLayers As Long, NumRows As Long, NumTires As Long, Layer() As LayerInfo

Private Sub Form_Load()
```

```
    Dim msg

    Form1.Width = 11865
    Form1.Height = 6435
    Form1.Left = (Screen.Width - Form1.Width) / 2
    Form1.Top = (Screen.Height - Form1.Height) / 2

    strFilename = "c:\user\Tire Model\InputedDataFile.txt"
    Call Form1.ReadDataFile
    Call Form1.OpenNewSetup

    'for MSFlexGrid1
    Form1.MSFlexGrid1.ColWidth(0) = 3050
    Form1.MSFlexGrid1.ColWidth(1) = 200
    Form1.MSFlexGrid1.ColWidth(2) = 700
    Form1.MSFlexGrid1.ColWidth(3) = 1125

    Form1.MSFlexGrid2.ColWidth(0) = 2500
    Form1.MSFlexGrid2.ColWidth(1) = 200
    Form1.MSFlexGrid2.ColWidth(2) = 1200

    Form1.Show

    msg = "Do you want to input now?"
    If MsgBox(msg, vbQuestion + vbYesNo, Me.Caption) = vbYes Then
    Form2.Command1.Visible = False
    Form2.Command2.Visible = True
    Call Form1.DischargeCode
    End If

End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)

    Dim msg
    Unload Form2
    Unload Form3
    Unload Form4

End Sub

Private Sub mnuFileNew_Click()
    Call Form1.OpenNewSetup
End Sub

Private Sub mnuFileOpen_Click()
    Call Form1.OpenPreviousSetup
End Sub

Private Sub mnuFileSave_Click()

    strFilename = "c:\user\Tire Model\InputedDataFile.txt"
    Call Form1.SaveDatatoFile
```

```
    strFilename = "c:\user\Tire Model\InputedData.txt"
    Call Form1.SaveDatatoFile

End Sub

Private Sub mnuFileSaveAs_Click()

    Form1.CommonDialog1.FileName = "c:\user\Tire Model\InputedDataFile.txt"
    Form1.CommonDialog1.Flags = cdlOFNOverwritePrompt
    Form1.CommonDialog1.ShowSave
    strFilename = Form1.CommonDialog1.FileName
    Call Form1.SaveDatatoFile

End Sub

Private Sub mnuFileExit_Click()

    Unload Form2
    Unload Form3
    Unload Form4
    Unload Me

End Sub

Private Sub mnuInputDischarge_Click()
    Call Form1.DischargeCode
End Sub

Private Sub mnuInputStreamSection_Click()
    Call Form1.StreamSectionCode
End Sub

Private Sub mnuInputControlStructure_Click()
    Call Form1.ControlStructureCode
End Sub

Private Sub mnuInputSediment_Click()
    Call Form1.SedimentCode
End Sub

Private Sub mnuInputBedInterface_Click()
    Call Form1.StructureBedInterfaceCode
End Sub

Private Sub mnuInputTireInfo_Click()
    Form3.Show
End Sub

Private Sub cmdInputStart_Click()

    Dim msg, i As Long

    If Form1.MSFlexGrid1.TextMatrix(25, 0) <> "" Then
```

```
   msg = "Warning! Previous run results may not be correct anymore if the inputs are changed!
Please redo the analysis."
   If MsgBox(msg, vbCritical + vbOKOnly, Me.Caption) = vbOK Then
   End If
   End If

   Form2.Command1.Visible = False
   Form2.Command2.Visible = True
   Call Form1.DischargeCode

End Sub

Private Sub cmdAnalysisStability_Click()
   Call Form1.RiverAnalysis
End Sub

Private Sub cmdAnalysisTire_Click()

   Dim msg
   If Form1.MSFlexGrid1.TextMatrix(25, 0) = "" Then
   msg = "Warning! Tire analysis can not run as stability analysis has not been performed yet."
   If MsgBox(msg, vbCritical + vbOKOnly, Me.Caption) = vbOK Then
   End If
   Else
   Call Form3.Show
   End If

End Sub

Public Sub DischargeCode()

   Form2.Caption = "Input"
   Form2.lblcounter.Caption = "1"

   Form2.lblTitle.Caption = "Discharge"

   Form2.Label1.Caption = "Drainage Area"
   Form2.Label2.Caption = "Land Use Factor"
   Form2.Label3.Caption = "Recurrence Interval"
   Form2.Label3.Visible = True
   Form2.Label4.Caption = ""
   Form2.Label4.Visible = False

   Form2.txt3.Visible = True
   Form2.txt4.Visible = False

   Form2.lblUnit1.Caption = "mi^2"
   Form2.lblUnit2.Caption = ""
   Form2.lblUnit3.Caption = "yrs"
   Form2.lblUnit4.Caption = ""

   Form2.txt1.Text = Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption, 2)
   Form2.txt2.Text = Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 1, 2)
   Form2.txt3.Text = Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 2, 2)
```

```
    Form2.Show

End Sub

Public Sub StreamSectionCode()

    Form2.Caption = "Input"
    Form2.lblcounter.Caption = "2"

    Form2.lblTitle.Caption = "Stream Section"

    Form2.Label1.Caption = "Bottom Slope"
    Form2.Label2.Caption = "Roughness Coeff, n"
    Form2.Label3.Caption = "Water Surface Width"
    Form2.Label3.Visible = True
    Form2.Label4.Caption = ""
    Form2.Label4.Visible = False

    Form2.txt3.Visible = True
    Form2.txt4.Visible = False

    Form2.lblUnit1.Caption = ""
    Form2.lblUnit2.Caption = ""
    Form2.lblUnit3.Caption = "ft"
    Form2.lblUnit4.Caption = ""

    Form2.txt1.Text = Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 3, 2)
    Form2.txt2.Text = Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 4, 2)
    Form2.txt3.Text = Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 5, 2)

    Form2.Show

End Sub

Public Sub ControlStructureCode()

    Form2.Caption = "Input"
    Form2.lblcounter.Caption = "3"

    Form2.lblTitle.Caption = "Control Structure"

    Form2.Label1.Caption = "Top Width"
    Form2.Label2.Caption = "Height"
    Form2.Label3.Caption = "Upstream Slope"
    Form2.Label3.Visible = True
    Form2.Label4.Caption = "Downstream Slope"
    Form2.Label4.Visible = True

    Form2.txt3.Visible = True
    Form2.txt4.Visible = True

    Form2.lblUnit1.Caption = "ft"
    Form2.lblUnit2.Caption = "ft"
```

```
    Form2.lblUnit3.Caption = ":1(H:V)"
    Form2.lblUnit4.Caption = ":1(H:V)"

    Form2.txt1.Text = Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 6, 2)
    Form2.txt2.Text = Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 7, 2)
    Form2.txt3.Text = Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 8, 2)
    Form2.txt4.Text = Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 9, 2)

    Form2.Show

End Sub

Public Sub SedimentCode()

    Form2.Caption = "Input"
    Form2.lblcounter.Caption = "4"

    Form2.lblTitle.Caption = "Sediment"

    Form2.Label1.Caption = "Accumulated Depth"
    Form2.Label2.Caption = "Unit Weight"
    Form2.Label3.Caption = "Friction Angle"
    Form2.Label3.Visible = True
    Form2.Label4.Caption = ""
    Form2.Label4.Visible = False

    Form2.txt3.Visible = True
    Form2.txt4.Visible = False

    Form2.lblUnit1.Caption = "ft"
    Form2.lblUnit2.Caption = "pcf"
    Form2.lblUnit3.Caption = "deg"
    Form2.lblUnit4.Caption = ""

    Form2.txt1.Text = Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 10, 2)
    Form2.txt2.Text = Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 11, 2)
    Form2.txt3.Text = Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 12, 2)

    Form2.Show

End Sub

Public Sub StructureBedInterfaceCode()

    Form2.Caption = "Input"
    Form2.lblcounter.Caption = "5"

    Form2.lblTitle.Caption = "Channel Bed Interface"

    Form2.Label1.Caption = "Adhesion"
    Form2.Label2.Caption = "Friction Angle"
    Form2.Label3.Caption = ""
    Form2.Label3.Visible = False
    Form2.Label4.Caption = ""
```

```
    Form2.Label4.Visible = False

    Form2.txt3.Visible = False
    Form2.txt4.Visible = False

    Form2.lblUnit1.Caption = ""
    Form2.lblUnit2.Caption = "deg"
    Form2.lblUnit3.Caption = ""
    Form2.lblUnit4.Caption = ""

    Form2.txt1.Text = Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 13, 2)
    Form2.txt2.Text = Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 14, 2)

    Form2.Show

End Sub

Public Sub ReadDataFile()

    Dim i As Long

    Open strFilename For Input As #1

    Input #1, n
    ReDim tmptext1(n), tmpvalue1(n), tmpvalue2(n), tmpunit(n)

    For i = 1 To n
        Input #1, tmpvalue1(i), tmptext1(i), tmpvalue2(i), tmpunit(i)
    Next i

    Close #1

End Sub

Public Sub OpenNewSetup()

    Dim i As Long

    Form1.MSFlexGrid1.TextMatrix(0, 0) = tmptext1(1)

    For i = 1 To (n - 1)
        If tmpvalue1(i + 1) = 2 Then
        Form1.MSFlexGrid1.TextMatrix(i, 1) = "="
        Form1.MSFlexGrid1.TextMatrix(i, 0) = "      " + tmptext1(i + 1)
        Else
        Form1.MSFlexGrid1.TextMatrix(i, 0) = tmptext1(i + 1)
        End If
    Next i

    For i = 0 To (n - 1)
        If tmpvalue1(i + 1) <> "1" Then
        Form1.MSFlexGrid1.TextMatrix(i, 2) = ""
            If tmpunit(i + 1) <> "0" Then
            Form1.MSFlexGrid1.TextMatrix(i, 3) = "  " & tmpunit(i + 1)
```

```
            Else
            Form1.MSFlexGrid1.TextMatrix(i, 3) = ""
            End If
         Else
         Form1.MSFlexGrid1.TextMatrix(i, 2) = ""
         End If
      Next i

      For i = 14 To 19
      If tmpvalue1(i + 1) <> "1" Then
      Form1.MSFlexGrid1.TextMatrix(i, 2) = tmpvalue2(i + 1)
      End If
      Next i

      For i = 1 To n
      If tmpvalue1(i) = 2 Then
      Form1.MSFlexGrid1.Row = i - 1
      Form1.MSFlexGrid1.Col = 0
      Form1.MSFlexGrid1.RowSel = i - 1
      Form1.MSFlexGrid1.ColSel = 3
      Form1.MSFlexGrid1.FillStyle = flexFillRepeat
      Form1.MSFlexGrid1.CellFontUnderline = False
      Form1.MSFlexGrid1.CellForeColor = vbBlue
      Else
      Form1.MSFlexGrid1.Row = i - 1
      Form1.MSFlexGrid1.Col = 0
      Form1.MSFlexGrid1.FillStyle = flexFillSingle
      Form1.MSFlexGrid1.CellFontUnderline = True
      Form1.MSFlexGrid1.CellForeColor = vbBlack
      End If
      Next i

      For i = n + 1 To 66
      Form1.MSFlexGrid1.TextMatrix(i, 0) = ""
      Form1.MSFlexGrid1.TextMatrix(i, 1) = ""
      Form1.MSFlexGrid1.TextMatrix(i, 2) = ""
      Form1.MSFlexGrid1.TextMatrix(i, 3) = ""
      Next i

      For i = 0 To 66
      Form1.MSFlexGrid2.TextMatrix(i, 0) = ""
      Form1.MSFlexGrid2.TextMatrix(i, 1) = ""
      Form1.MSFlexGrid2.TextMatrix(i, 2) = ""
      Next i

      Form1.MSFlexGrid1.Row = 1
      Form1.MSFlexGrid1.Col = 2
      Form1.MSFlexGrid1.TopRow = 0

End Sub

Public Sub OpenPreviousSetup()

   Dim i As Long
```

```
 'for MSFlexGrid1
 For i = 0 To (n - 1)
    If tmpvalue1(i + 1) <> "1" Then
    Form1.MSFlexGrid1.TextMatrix(i, 2) = tmpvalue2(i + 1)
    Else
    Form1.MSFlexGrid1.TextMatrix(i, 2) = ""
    End If
 Next i
 Form1.MSFlexGrid1.Row = 23
 Form1.MSFlexGrid1.Col = 2
 Form1.MSFlexGrid1.TopRow = 8

End Sub

Public Sub SaveDatatoFile()

 Dim i As Long
 Dim tmpvalue1 As String, tmptext As String, tmpvalue2 As Double, tmpunit As String

 Call Form1.SavetoFileErrorChecking

 If (CheckingResult = "True") Then
 Open strFilename For Output As #1

 'for MSFlexGrid1
 Print #1, n

 For i = 0 To (n - 1)
 If (Form1.MSFlexGrid1.TextMatrix(i, 1) = "=") Then
 tmpvalue1 = "2"
 tmptext = Form1.MSFlexGrid1.TextMatrix(i, 0)
 tmpvalue2 = Form1.MSFlexGrid1.TextMatrix(i, 2)
    If Form1.MSFlexGrid1.TextMatrix(i, 3) <> "" Then
    tmpunit = Form1.MSFlexGrid1.TextMatrix(i, 3)
    Else
    tmpunit = "0"
    End If
 Else
 tmpvalue1 = "1"
 tmptext = Form1.MSFlexGrid1.TextMatrix(i, 0)
 tmpvalue2 = "0"
 tmpunit = "0"
 End If

 Print #1, tmpvalue1 & "," & tmptext & "," & tmpvalue2 & "," & tmpunit
 Next i

 Close #1

 Else

 GoTo 100
 End If
```

```
100 End Sub

Public Sub SavetoFileErrorChecking()

   Dim i As Long
   For i = 0 To n - 1
   If tmpvalue1(i + 1) = 2 Then
      If Form1.MSFlexGrid1.TextMatrix(i, 2) = "" Then
      CheckingResult = "False"
      Else
      CheckingResult = "True"
      End If
   End If
   Next i

End Sub

Public Sub RiverAnalysis()

   Dim Const1 As Double, Const2 As Double
   Dim msg, I As Long

   Call Error.ErrorChecking(n, result)

   If result = "True" Then
   DA = Form1.MSFlexGrid1.TextMatrix(1, 2)
   LF = Form1.MSFlexGrid1.TextMatrix(2, 2)
   RI = Form1.MSFlexGrid1.TextMatrix(3, 2)
   StreamSlope = Form1.MSFlexGrid1.TextMatrix(5, 2)
   Roughness = Form1.MSFlexGrid1.TextMatrix(6, 2)
   StreamWidth = Form1.MSFlexGrid1.TextMatrix(7, 2)
   DamTopWidth = Form1.MSFlexGrid1.TextMatrix(9, 2)
   DamHeight = Form1.MSFlexGrid1.TextMatrix(10, 2)
   DamUpstreamSlope = Form1.MSFlexGrid1.TextMatrix(11, 2)
   DamDownstreamSlope = Form1.MSFlexGrid1.TextMatrix(12, 2)
   SedDepth = Form1.MSFlexGrid1.TextMatrix(14, 2)
   SedSatUnitWeight = Form1.MSFlexGrid1.TextMatrix(15, 2)
   SedFrictionAngle = Form1.MSFlexGrid1.TextMatrix(16, 2)

   If SedDepth > DamHeight Then
   MsgBox "Sediment depth entered exceeds the height of control structure. It is now reduced to " _
   & Format(DamHeight, "0.0") & " ft."
   Form1.MSFlexGrid1.TextMatrix(14, 2) = DamHeight
   SedDepth = Form1.MSFlexGrid1.TextMatrix(14, 2)
   End If

   Flowrate = 422.58 * LF * (RI ^ 0.301) * (DA ^ 0.504)

   Call Solver.Polynomial(Roughness, StreamWidth, StreamSlope, Flowrate, NoDamDepth)

   'to check solver
   Const1 = NoDamDepth ^ (5 / 3) / (2 * NoDamDepth + StreamWidth) ^ (2 / 3)
   Const2 = Flowrate * Roughness / 1.486 / StreamWidth ^ (5 / 3) / StreamSlope ^ (1 / 2)
```

```
If (Const1 - Const2) > 0.05 Then
MsgBox "ERROR: Pre-construction depth error!"
End If

tailwater = NoDamDepth

Y1 = NoDamDepth
Area1 = Y1 * StreamWidth
V1 = Flowrate / Area1

Yc = (Flowrate ^ 2 / (StreamWidth ^ 2 * 32.2)) ^ (1 / 3)
Areac = Yc * StreamWidth
Vc = Flowrate / Areac

V2 = 0
Y2 = 3 / 2 * Yc - (V2 ^ 2) / 2 / 32.2
Velocity2 = Flowrate / (Y2 + DamHeight) / StreamWidth
Do Until V2 > Velocity2
   V2 = V2 + 0.01
   Y2 = 3 / 2 * Yc - (V2 ^ 2) / 2 / 32.2
   Velocity2 = Flowrate / (Y2 + DamHeight) / StreamWidth
Loop

'to check velocity 2
Y2Test = 3 / 2 * Yc - 0.01
V2Test = ((3 / 2 * Yc - Y2Test) * 2 * 32.2) ^ 0.5
Velocity2Test = Flowrate / (Y2Test + DamHeight) / StreamWidth
Do Until (Velocity2Test - V2Test) < 0.05
   Y2Test = Y2Test - 0.01
   V2Test = ((3 / 2 * Yc - Y2Test) * 2 * 32.2) ^ 0.5
   Velocity2Test = Flowrate / (Y2Test + DamHeight) / StreamWidth
Loop
If V2 > V2Test Then
   If (V2 - V2Test) > 0.1 Then MsgBox "ERROR: V2 error!"
Else
   If (V2Test - V2) > 0.1 Then MsgBox "ERROR: V2 error!"
End If

Ydam = (Y2 + Yc) / 2

ka = (Tan((45 - SedFrictionAngle / 2) * 3.1416 / 180)) ^ 2

'Output to screen
I = 21
Form1.SetFocus
Form1.MSFlexGrid1.Row = I
Form1.MSFlexGrid1.Col = 0
Form1.MSFlexGrid1.FillStyle = flexFillSingle
Form1.MSFlexGrid1.CellFontUnderline = True
Form1.MSFlexGrid1.CellForeColor = vbRed
Form1.MSFlexGrid1.CellFontSize = 15
Form1.MSFlexGrid1.TextMatrix(I, 0) = "************** OUTPUT **************"
Form1.MSFlexGrid1.TextMatrix(I, 1) = "************** OUTPUT **************"
Form1.MSFlexGrid1.TextMatrix(I, 2) = "************** OUTPUT **************"
```

```
   Form1.MSFlexGrid1.TextMatrix(I, 3) = "*************** OUTPUT ***************"
   Form1.MSFlexGrid1.MergeCells = flexMergeRestrictRows
   Form1.MSFlexGrid1.MergeRow(I) = True

   Form1.MSFlexGrid1.Row = I + 1
   Form1.MSFlexGrid1.Col = 0
   Form1.MSFlexGrid1.FillStyle = flexFillSingle
   Form1.MSFlexGrid1.CellFontUnderline = True
   Form1.MSFlexGrid1.CellForeColor = vbRed
   Form1.MSFlexGrid1.TextMatrix(I + 1, 0) = "River Flow Analysis"

   Form1.MSFlexGrid1.TextMatrix(I + 2, 0) = "Estimated Flowrate =  " & Format(Flowrate, ".0") & "
cfs"
   Form1.MSFlexGrid1.TextMatrix(I + 3, 0) = "Pre-Construction Depth =  " & Format(NoDamDepth,
".0") & " ft"
   Form1.MSFlexGrid1.TextMatrix(I + 4, 0) = "Y1       =  " & Format(Y1, ".0") & " ft"
   Form1.MSFlexGrid1.TextMatrix(I + 5, 0) = "V1       =  " & Format(V1, ".0") & " ft/s"
   Form1.MSFlexGrid1.TextMatrix(I + 6, 0) = "Y2       =  " & Format(Y2, ".0") & " ft"
   Form1.MSFlexGrid1.TextMatrix(I + 7, 0) = "V2       =  " & Format(V2, ".0") & " ft/s"
   Form1.MSFlexGrid1.TextMatrix(I + 8, 0) = "Yc       =  " & Format(Yc, ".0") & " ft"
   Form1.MSFlexGrid1.TextMatrix(I + 9, 0) = "Ydam   =  " & Format(Ydam, ".0") & " ft"
   Form1.MSFlexGrid1.TextMatrix(I + 10, 0) = "Tailwater Depth  =  " & Format(NoDamDepth, ".0") & "
ft"

   Form1.MSFlexGrid1.SetFocus
   Form1.MSFlexGrid1.TopRow = 47
   Form1.MSFlexGrid1.Row = 35
   Form1.MSFlexGrid1.Col = 1
   Call Form1.ForceAnalysis
   Else
   MsgBox "One or more input fields are empty!"
   msg = "Do you want to input now?"
   If MsgBox(msg, vbQuestion + vbYesNo, Me.Caption) = vbYes Then
   Form2.Command1.Visible = False
   Form2.Command2.Visible = True
   Call Form1.DischargeCode
   End If
   End If

End Sub

Private Sub Command1_Click()
   Call Form1.OpenPreviousSetup
   Form1.MSFlexGrid1.SetFocus
End Sub

Public Sub ForceAnalysis()

   Dim Sec1Vol As Double, Sec1Force As Double, Sec1CG As Double
   Dim Sec2Vol As Double, Sec2Force As Double, Sec2CG As Double
   Dim Sec3Vol As Double, Sec3Force As Double, Sec3CG As Double
   Dim Sec4Vol As Double, Sec4Force As Double, Sec4CG As Double
   Dim Sec41Vol As Double, Sec41Force As Double, Sec41CG As Double
   Dim Sec5Vol As Double, Sec5Force As Double, Sec5CG As Double
```

```
   Dim Sec6Vol As Double, Sec6Force As Double, Sec6CG As Double
   Dim Sec7Vol As Double, Sec7Force As Double, Sec7CG As Double
   Dim Sec71Vol As Double, Sec71Force As Double, Sec71CG As Double
   Dim Sec8Vol As Double, Sec8Force As Double, Sec8CG As Double
   Dim Sec9Vol As Double, Sec9Force As Double, Sec9CG As Double
   Dim Sec10Vol As Double, Sec10Force As Double, Sec10CG As Double
   Dim Sec11Vol As Double, Sec11Force As Double, Sec11CG As Double
   Dim I As Long

   damlength = DamTopWidth + DamUpstreamSlope * DamHeight + DamDownstreamSlope *
DamHeight

   'section 1
   Sec1Vol = Ydam * DamUpstreamSlope * DamHeight
   Sec1Force = Sec1Vol * 62.4 / 1000
   Sec1CG = DamUpstreamSlope * DamHeight / 2

   'section 2
   Sec2Vol = DamUpstreamSlope * DamHeight ^ 2 / 2
   Sec2Force = Sec2Vol * 62.4 / 1000
   Sec2CG = DamUpstreamSlope * DamHeight / 3

   'section 3
   Sec3Vol = DamTopWidth * Ydam
   Sec3Force = Sec3Vol * 62.4 / 1000
   Sec3CG = DamUpstreamSlope * DamHeight + DamTopWidth / 2

   If tailwater <= DamHeight Then
   'section 4
   Sec4Vol = (tailwater) ^ 2 * DamDownstreamSlope / 2
   Sec4Force = Sec4Vol * 62.4 / 1000
   Sec4CG = damlength - tailwater * DamDownstreamSlope / 3

   'section 41
   Sec41Vol = 0
   Sec41Force = 0
   Sec41CG = 0
   Else
   'section 4
   Sec4Vol = (DamHeight) ^ 2 * DamDownstreamSlope / 2
   Sec4Force = Sec4Vol * 62.4 / 1000
   Sec4CG = damlength - DamHeight * DamDownstreamSlope / 3

   'section 41
   Sec41Vol = (tailwater - DamHeight) * DamHeight * DamDownstreamSlope
   Sec41Force = Sec41Vol * 62.4 / 1000
   Sec41CG = damlength - DamHeight * DamDownstreamSlope / 2
   End If

   'section 5
   Sec5Vol = (DamHeight) ^ 2 / 2
   Sec5Force = Sec5Vol * 62.4 / 1000
   Sec5CG = DamHeight / 3
```

```
'section 6
Sec6Vol = Ydam * DamHeight
Sec6Force = Sec6Vol * 62.4 / 1000
Sec6CG = DamHeight / 2

If tailwater <= DamHeight Then
'section 7
Sec7Vol = (tailwater) ^ 2 / 2
Sec7Force = Sec7Vol * 62.4 / 1000
Sec7CG = tailwater / 3

'section 71
Sec71Vol = 0
Sec71Force = 0
Sec71CG = 0
Else
'section 7
Sec7Vol = (DamHeight) ^ 2 / 2
Sec7Force = Sec7Vol * 62.4 / 1000
Sec7CG = DamHeight / 3

'section 71
Sec71Vol = (tailwater - DamHeight) * DamHeight
Sec71Force = Sec71Vol * 62.4 / 1000
Sec71CG = DamHeight / 2
End If

'section 8
Sec8Vol = tailwater * damlength
Sec8Force = Sec8Vol * 62.4 / 1000
Sec8CG = damlength / 2

'section 9
Sec9Vol = (DamHeight + Ydam - tailwater) * damlength / 2
Sec9Force = Sec9Vol * 62.4 / 1000
Sec9CG = damlength / 3

'section 10
Sec10Vol = (SedDepth) ^ 2 * DamUpstreamSlope / 2
Sec10Force = Sec10Vol * (SedSatUnitWeight - 62.4) / 1000
Sec10CG = SedDepth * DamUpstreamSlope / 3

'section 11
Sec11Vol = ka * (SedDepth) ^ 2 / 2
Sec11Force = Sec11Vol * (SedSatUnitWeight - 62.4) / 1000
Sec11CG = SedDepth / 3

'total for sections 1 to 4 & 41 (Positive)
TotalForce1to4 = Sec1Force + Sec2Force + Sec3Force + Sec4Force + Sec41Force
TotalCG1to4 = (Sec1Force * Sec1CG + Sec2Force * Sec2CG + Sec3Force * Sec3CG +
Sec4Force * Sec4CG + Sec41Force * Sec41CG) / TotalForce1to4

'total for sections 5 to 6 (Positive)
TotalForce5to6 = Sec5Force + Sec6Force
```

```
TotalCG5to6 = (Sec5Force * Sec5CG + Sec6Force * Sec6CG) / TotalForce5to6

'total for sections 7 & 71 (Negative)
TotalForce7to71 = Sec7Force + Sec71Force
TotalCG7to71 = (Sec7CG * Sec7Force + Sec71Force * Sec71CG) / TotalForce7to71

'total for sections 8 to 9 (Negative)
TotalForce8to9 = Sec8Force + Sec9Force
TotalCG8to9 = (Sec8Force * Sec8CG + Sec9Force * Sec9CG) / TotalForce8to9

'total for section 10 (Positive)
TotalForce10 = Sec10Force
TotalCG10 = Sec10CG

'total for section 11 (Positive)
TotalForce11 = Sec11Force
TotalCG11 = Sec11CG

'Output to screen
I = 32
Form1.MSFlexGrid1.Row = I
Form1.MSFlexGrid1.Col = 0
Form1.MSFlexGrid1.FillStyle = flexFillSingle
Form1.MSFlexGrid1.CellFontUnderline = True
Form1.MSFlexGrid1.CellForeColor = vbRed
Form1.MSFlexGrid1.TextMatrix(I, 0) = "Force Analysis"

Form1.MSFlexGrid1.Row = I + 1
Form1.MSFlexGrid1.Col = 0
Form1.MSFlexGrid1.FillStyle = flexFillSingle
Form1.MSFlexGrid1.CellFontUnderline = True
Form1.MSFlexGrid1.TextMatrix(I + 1, 0) = "DESCRIPTION" & "   FORCE (kips)"

'for vertical hydrostatic forces
Form1.MSFlexGrid1.Row = I + 2
Form1.MSFlexGrid1.Col = 0
Form1.MSFlexGrid1.RowSel = 54
Form1.MSFlexGrid1.ColSel = 0
Form1.MSFlexGrid1.FillStyle = flexFillRepeat
Form1.MSFlexGrid1.CellForeColor = QBColor(1)
Form1.MSFlexGrid1.TextMatrix(I + 2, 0) = "Vertical Hydrostatic Forces"
Form1.MSFlexGrid1.TextMatrix(I + 3, 0) = "  Headwater"
Form1.MSFlexGrid1.TextMatrix(I + 4, 0) = "    Sec. 1          " & Format(Sec1Force, "0.0")
Form1.MSFlexGrid1.TextMatrix(I + 5, 0) = "    Sec. 2          " & Format(Sec2Force, "0.0")
Form1.MSFlexGrid1.TextMatrix(I + 6, 0) = "    Sec. 3          " & Format(Sec3Force, "0.0")

Form1.MSFlexGrid1.TextMatrix(I + 7, 0) = "  Tailwater"
Form1.MSFlexGrid1.TextMatrix(I + 8, 0) = "    Sec. 4          " & Format(Sec4Force, "0.0")
Form1.MSFlexGrid1.TextMatrix(I + 9, 0) = "    Sec. 41         " & Format(Sec41Force, "0.0")

Form1.MSFlexGrid1.Row = I + 2
Form1.MSFlexGrid1.Col = 0
Form1.MSFlexGrid1.CellFontBold = True
```

```
    Form1.MSFlexGrid1.Row = I + 3
    Form1.MSFlexGrid1.Col = 0
    Form1.MSFlexGrid1.CellFontItalic = True

    Form1.MSFlexGrid1.Row = I + 7
    Form1.MSFlexGrid1.Col = 0
    Form1.MSFlexGrid1.CellFontItalic = True

    'for horizontal hydrostatic forces
    Form1.MSFlexGrid1.TextMatrix(I + 10, 0) = "Horizontal Hydrostatic Forces"
    Form1.MSFlexGrid1.TextMatrix(I + 11, 0) = "   Headwater"
    Form1.MSFlexGrid1.TextMatrix(I + 12, 0) = "     Sec. 5              " & Format(Sec5Force, "0.0")
    Form1.MSFlexGrid1.TextMatrix(I + 13, 0) = "     Sec. 6              " & Format(Sec6Force, "0.0")

    Form1.MSFlexGrid1.TextMatrix(I + 14, 0) = "   Tailwater"
    Form1.MSFlexGrid1.TextMatrix(I + 15, 0) = "     Sec. 7              " & Format(Sec7Force, "0.0")
    Form1.MSFlexGrid1.TextMatrix(I + 16, 0) = "     Sec. 71             " & Format(Sec71Force, "0.0")

    Form1.MSFlexGrid1.Row = I + 10
    Form1.MSFlexGrid1.Col = 0
    Form1.MSFlexGrid1.CellFontBold = True

    Form1.MSFlexGrid1.Row = I + 11
    Form1.MSFlexGrid1.Col = 0
    Form1.MSFlexGrid1.CellFontItalic = True

    Form1.MSFlexGrid1.Row = I + 14
    Form1.MSFlexGrid1.Col = 0
    Form1.MSFlexGrid1.CellFontItalic = True

    'for hydrostatic uplift forces
    Form1.MSFlexGrid1.TextMatrix(I + 17, 0) = "Hydrostatic Uplift Forces"
    Form1.MSFlexGrid1.TextMatrix(I + 18, 0) = "     Sec. 8              " & Format(Sec8Force, "0.0")
    Form1.MSFlexGrid1.TextMatrix(I + 19, 0) = "     Sec. 9              " & Format(Sec9Force, "0.0")

    Form1.MSFlexGrid1.Row = I + 17
    Form1.MSFlexGrid1.Col = 0
    Form1.MSFlexGrid1.CellFontBold = True

    'for forces due to sediment
    Form1.MSFlexGrid1.TextMatrix(I + 20, 0) = "Sediment Deposits"
    Form1.MSFlexGrid1.TextMatrix(I + 21, 0) = "     Sec. 10             " & Format(Sec10Force, "0.0")
    Form1.MSFlexGrid1.TextMatrix(I + 22, 0) = "     Sec. 11             " & Format(Sec11Force, "0.0")

    Form1.MSFlexGrid1.Row = I + 20
    Form1.MSFlexGrid1.Col = 0
    Form1.MSFlexGrid1.CellFontBold = True

    Call Form1.StabilityAnalysis

End Sub

Public Sub StabilityAnalysis()
```

```
   BedInterfaceAdhesion = Form1.MSFlexGrid1.TextMatrix(18, 2)
   BedInterfaceFrictionAngle = Form1.MSFlexGrid1.TextMatrix(19, 2)

   DamSec1Vol = (DamHeight * DamUpstreamSlope * DamHeight) / 2
   DamSec2Vol = (DamHeight) * DamTopWidth
   DamSec3Vol = (DamHeight * DamDownstreamSlope * DamHeight) / 2

   Vdam = Flowrate / Ydam / StreamWidth
   'Uplift force due to flow
   FlowUplift = 0.25 * DamTopWidth * 62.4 * (Vdam) ^ 2 / 2 / 32.2 / 1000

   'Drag force due to flow
   FlowDrag = DamTopWidth * 62.4 * (Vdam) ^ 2 / 2 / 32.2 / 1000

   'net uplift force
   NetUpliftForce = FlowUplift + TotalForce8to9 - TotalForce10 - TotalForce1to4

   'min effective unit weight of dam
   DamMinEffUnitWeight = NetUpliftForce * 1000 / ((damlength + DamTopWidth) / 2 * DamHeight)

   'min saturated unit weight of dam
   DamMinSatUnitWeight = DamMinEffUnitWeight + 62.4

   DamSec1Force = DamSec1Vol * (DamMinSatUnitWeight - 62.4) / 1000
   DamSec2Force = DamSec2Vol * (DamMinSatUnitWeight - 62.4) / 1000
   DamSec3Force = DamSec3Vol * (DamMinSatUnitWeight - 62.4) / 1000

   DamSec1CG = DamHeight * DamUpstreamSlope * 2 / 3
   DamSec2CG = DamHeight * DamUpstreamSlope + DamTopWidth / 2
   DamSec3CG = DamHeight * DamUpstreamSlope + DamTopWidth + DamHeight *
DamDownstreamSlope / 3

   'Total of Wt. of dam
   DamTotalWeight = DamSec1Force + DamSec2Force + DamSec3Force
   DamTotalCG = (DamSec1Force * DamSec1CG + DamSec2Force * DamSec2CG +
DamSec3Force * DamSec3CG) / DamTotalWeight

   'V (should equal to zero for now as SF here is 1)
   VerticalReaction = DamTotalWeight - NetUpliftForce

   'H
   HorizontalReaction = TotalForce11 + FlowDrag + TotalForce5to6 - TotalForce7to71

   'to calculate equivalent height
   AreaTrapezium = (damlength + DamTopWidth) / 2 * DamHeight
   HeightEquivalent = AreaTrapezium / damlength

   ResistingShearStress = BedInterfaceAdhesion + (Tan(BedInterfaceFrictionAngle * 3.1416 / 180)) *
(DamMinSatUnitWeight - 62.4) * HeightEquivalent
   ResistingShearForce = ResistingShearStress * damlength / 1000
   ShearStress = ResistingShearStress

   FSUplift = DamTotalWeight / NetUpliftForce
   FSSliding = ResistingShearForce / HorizontalReaction
```

```
   'If FSSliding >= 0 And FSSliding < 2 Then
   'FSSliding = 2
   'ResistingShearForce = HorizontalReaction * FSSliding
   'ResistingShearStress = ResistingShearForce * 1000 / damlength
   'DamMinSatUnitWeight = 62.4 + (ResistingShearStress - BedInterfaceAdhesion) /
(Tan(BedInterfaceFrictionAngle * 3.1416 / 180)) / HeightEquivalent
   'End If

   'FSSliding = ResistingShearForce / HorizontalReaction

   DamMinEffUnitWeight = DamMinSatUnitWeight - 62.4
   DamMinDryUnitWeight = 1.6 * DamMinEffUnitWeight

   I = 55
   Form1.MSFlexGrid1.Row = I
   Form1.MSFlexGrid1.Col = 0
   Form1.MSFlexGrid1.FillStyle = flexFillSingle
   Form1.MSFlexGrid1.CellFontUnderline = True
   Form1.MSFlexGrid1.CellForeColor = vbRed
   Form1.MSFlexGrid1.TextMatrix(I, 0) = "Stability Analysis"
   Form1.MSFlexGrid1.TextMatrix(I + 1, 0) = "Total Uplift Force"
   Form1.MSFlexGrid1.TextMatrix(I + 2, 0) = "Total Pushing Force"
   Form1.MSFlexGrid1.TextMatrix(I + 3, 0) = "Resisting Shear Force"
   Form1.MSFlexGrid1.TextMatrix(I + 4, 0) = "Min. Unit Weight of Structure"
   Form1.MSFlexGrid1.TextMatrix(I + 6, 0) = "Factor of Safety       - Uplift"
   Form1.MSFlexGrid1.TextMatrix(I + 7, 0) = "                              - Sliding"

   Form1.MSFlexGrid1.TextMatrix(I + 1, 1) = "="
   Form1.MSFlexGrid1.TextMatrix(I + 2, 1) = "="
   Form1.MSFlexGrid1.TextMatrix(I + 3, 1) = "="
   Form1.MSFlexGrid1.TextMatrix(I + 4, 1) = "="
   Form1.MSFlexGrid1.TextMatrix(I + 5, 1) = "="
   Form1.MSFlexGrid1.TextMatrix(I + 6, 1) = "="
   Form1.MSFlexGrid1.TextMatrix(I + 7, 1) = "="

   Form1.MSFlexGrid1.TextMatrix(I + 1, 2) = Format(NetUpliftForce, "0.0")
   Form1.MSFlexGrid1.TextMatrix(I + 2, 2) = Format(HorizontalReaction, "0.0")
   Form1.MSFlexGrid1.TextMatrix(I + 3, 2) = Format(ResistingShearForce, "0.0")
   Form1.MSFlexGrid1.TextMatrix(I + 4, 2) = Format(DamMinSatUnitWeight, "0.0")
   Form1.MSFlexGrid1.TextMatrix(I + 5, 2) = Format(DamMinDryUnitWeight, "0.0")
   Form1.MSFlexGrid1.TextMatrix(I + 6, 2) = Format(FSUplift, "0.0")
   Form1.MSFlexGrid1.TextMatrix(I + 7, 2) = Format(FSSliding, "0.0")

   Form1.MSFlexGrid1.TextMatrix(I + 1, 3) = "kips"
   Form1.MSFlexGrid1.TextMatrix(I + 2, 3) = "kips"
   Form1.MSFlexGrid1.TextMatrix(I + 3, 3) = "kips"
   Form1.MSFlexGrid1.TextMatrix(I + 4, 3) = "pcf (Sat.)"
   Form1.MSFlexGrid1.TextMatrix(I + 5, 3) = "pcf (Dry)"
   Form1.MSFlexGrid1.TextMatrix(I + 6, 3) = ""
   Form1.MSFlexGrid1.TextMatrix(I + 7, 3) = ""

   Form1.MSFlexGrid1.Row = I + 8
   Form1.MSFlexGrid1.Col = 0
```

```
    Form1.MSFlexGrid1.FillStyle = flexFillSingle
    Form1.MSFlexGrid1.CellForeColor = vbRed
    Form1.MSFlexGrid1.CellFontSize = 15
    Form1.MSFlexGrid1.TextMatrix(I + 8, 0) = "************************************"
    Form1.MSFlexGrid1.TextMatrix(I + 8, 1) = "************************************"
    Form1.MSFlexGrid1.TextMatrix(I + 8, 2) = "************************************"
    Form1.MSFlexGrid1.TextMatrix(I + 8, 3) = "************************************"
    Form1.MSFlexGrid1.MergeCells = flexMergeRestrictRows
    Form1.MSFlexGrid1.MergeRow(I + 8) = True

    Form1.MSFlexGrid1.Row = 65
    Form1.MSFlexGrid1.Col = 3

End Sub

Public Sub VolumeAnalysis()

    Dim R13TirePercent As Double, R14TirePercent As Double, R15TirePercent As Double
    Dim R13TireXXX As Double, R14TireXXX As Double, R15TireXXX As Double
    Dim R13TireYY As Double, R14TireYY As Double, R15TireYY As Double
    Dim R13TireSectionWidth As Double, R14TireSectionWidth As Double, R15TireSectionWidth As
Double
    Dim R13TireOutsideDiameter As Double, R14TireOutsideDiameter As Double
    Dim R15TireOutsideDiameter As Double
    Dim R13TireVol As Double, R14TireVol As Double, R15TireVol As Double
    Dim TotalUnfillVoids As Double, AvgTireVol As Double
    Dim AvgTireSectionWidth As Double, AvgTireOutsideDiameter As Double
    Dim ActualDamHeight As Double, UnderBedDamHeight As Double

    Dim TotalStructureVol As Double, TotalRubberOnlyVol As Double
    Dim TotalMatAndVoidsVol As Double, TotalMatOnlyVol As Double
    Dim RubberOnlyVolPercent As Double, RubberOnlyUnitWeight As Double
    Dim MatOnlyVolPercent As Double
    Dim TotalMatOnlyUnitWeight As Double

    R13TirePercent = Val(Form1.MSFlexGrid2.TextMatrix(3, 2))
    R14TirePercent = Val(Form1.MSFlexGrid2.TextMatrix(4, 2))
    R15TirePercent = Val(Form1.MSFlexGrid2.TextMatrix(5, 2))
    TotalUnfillVoids = Form1.MSFlexGrid2.TextMatrix(6, 2)

    R13TireXXX = 155
    R14TireXXX = 175
    R15TireXXX = 195

    R13TireYY = 75
    R14TireYY = 70
    R15TireYY = 65

    R13TireSectionWidth = (R13TireXXX * 0.0032808398)
    R14TireSectionWidth = (R14TireXXX * 0.0032808398)
    R15TireSectionWidth = (R15TireXXX * 0.0032808398)

    R13TireOutsideDiameter = R13TireSectionWidth * R13TireYY / 100 * 2 + 13 / 12
    R14TireOutsideDiameter = R14TireSectionWidth * R14TireYY / 100 * 2 + 14 / 12
```

```
R15TireOutsideDiameter = R15TireSectionWidth * R15TireYY / 100 * 2 + 15 / 12

R13TireVol = 3.14159265 / 4 * (R13TireOutsideDiameter) ^ 2 * R13TireSectionWidth
R14TireVol = 3.14159265 / 4 * (R14TireOutsideDiameter) ^ 2 * R14TireSectionWidth
R15TireVol = 3.14159265 / 4 * (R15TireOutsideDiameter) ^ 2 * R15TireSectionWidth

AvgTireSectionWidth = (R13TireSectionWidth * R13TirePercent / 100 + R14TireSectionWidth * R14TirePercent / 100 + R15TireSectionWidth * R15TirePercent / 100)
NumLayers = DamHeight / AvgTireSectionWidth
ActualDamHeight = NumLayers * AvgTireSectionWidth
'UnderBedDamHeight = ActualDamHeight - DamHeight

AvgTireOutsideDiameter = (R13TireOutsideDiameter * R13TirePercent / 100 + R14TireOutsideDiameter * R14TirePercent / 100 + R15TireOutsideDiameter * R15TirePercent / 100)
NumRows = StreamWidth / AvgTireOutsideDiameter

AvgTireVol = (R13TireVol * R13TirePercent / 100 + R14TireVol * R14TirePercent / 100 + R15TireVol * R15TirePercent / 100)

ReDim Layer(NumLayers)
Dim i As Single
For i = 1 To NumLayers
   Layer(i).TotalWidth = DamTopWidth + (i - 1) * AvgTireSectionWidth * DamUpstreamSlope + (i - 1) * AvgTireSectionWidth * DamDownstreamSlope
   Layer(i).NumTiresPerRow = Layer(i).TotalWidth / AvgTireOutsideDiameter
   Layer(i).TotalNumTires = Layer(i).NumTiresPerRow * NumRows
   NumTires = NumTires + Layer(i).TotalNumTires
Next i

TotalStructureVol = NumTires * AvgTireVol
TotalRubberOnlyVol = TotalStructureVol * 8 / 100
TotalMatAndVoidsVol = TotalStructureVol - TotalRubberOnlyVol
TotalMatOnlyVol = TotalMatAndVoidsVol * (100 - TotalUnfillVoids) / 100

'to calculate the required unit weight of materials alone
RubberOnlyVolPercent = TotalRubberOnlyVol / TotalStructureVol
RubberOnlyUnitWeight = 98.93

MatOnlyVolPercent = TotalMatOnlyVol / TotalStructureVol

TotalMatOnlyUnitWeight = (DamMinDryUnitWeight - RubberOnlyUnitWeight * RubberOnlyVolPercent) / MatOnlyVolPercent

'Output to screen
Form1.MSFlexGrid2.Row = 8
Form1.MSFlexGrid2.Col = 0
Form1.MSFlexGrid2.FillStyle = flexFillSingle
Form1.MSFlexGrid2.CellFontUnderline = True
Form1.MSFlexGrid2.CellForeColor = vbRed
Form1.MSFlexGrid2.CellFontSize = 15
Form1.MSFlexGrid2.TextMatrix(8, 0) = "********* OUTPUT *********"
Form1.MSFlexGrid2.TextMatrix(8, 1) = "********* OUTPUT *********"
Form1.MSFlexGrid2.TextMatrix(8, 2) = "********* OUTPUT *********"
Form1.MSFlexGrid2.MergeCells = flexMergeRestrictRows
```

```
  Form1.MSFlexGrid2.MergeRow(8) = True

  Form1.MSFlexGrid2.Row = 9
  Form1.MSFlexGrid2.Col = 0
  Form1.MSFlexGrid2.FillStyle = flexFillSingle
  Form1.MSFlexGrid2.CellFontUnderline = True
  Form1.MSFlexGrid2.CellForeColor = vbRed
  Form1.MSFlexGrid2.TextMatrix(9, 0) = "Tire Analysis"

  Form1.MSFlexGrid2.TextMatrix(10, 0) = "Assumptions"

  Form1.MSFlexGrid2.Row = 11
  Form1.MSFlexGrid2.Col = 0
  Form1.MSFlexGrid2.RowSel = 17
  Form1.MSFlexGrid2.ColSel = 2
  Form1.MSFlexGrid2.FillStyle = flexFillRepeat
  Form1.MSFlexGrid2.CellFontItalic = True
  Form1.MSFlexGrid2.TextMatrix(11, 0) = "   " & "Tire Specs (xxx/yy Rzz)"
  Form1.MSFlexGrid2.TextMatrix(12, 0) = "   - " & R13TireXXX & "/" & R13TireYY & " R13" & " (" &
R13TirePercent & "%)"
  Form1.MSFlexGrid2.TextMatrix(13, 0) = "   - " & R14TireXXX & "/" & R14TireYY & " R14" & " (" &
R14TirePercent & "%)"
  Form1.MSFlexGrid2.TextMatrix(14, 0) = "   - " & R15TireXXX & "/" & R15TireYY & " R15" & " (" &
R15TirePercent & "%)"
  Form1.MSFlexGrid2.TextMatrix(15, 0) = "   " & "Tire Rubber Content"
  Form1.MSFlexGrid2.TextMatrix(16, 0) = "   - Percent Volume"
  Form1.MSFlexGrid2.TextMatrix(16, 1) = "="
  Form1.MSFlexGrid2.TextMatrix(16, 2) = "8.0%"
  Form1.MSFlexGrid2.TextMatrix(17, 0) = "   - Unit Weight"
  Form1.MSFlexGrid2.TextMatrix(17, 1) = "="
  Form1.MSFlexGrid2.TextMatrix(17, 2) = "98.93 pcf"

  Dim a As Single
  a = 18
  Form1.MSFlexGrid2.TextMatrix(a, 0) = "Avg. Tire Section Width"
  Form1.MSFlexGrid2.TextMatrix(a, 1) = "="
  Form1.MSFlexGrid2.TextMatrix(a, 2) = Format(AvgTireSectionWidth, "0.0") & " ft"

  Form1.MSFlexGrid2.TextMatrix(a + 1, 0) = "Avg. Tire Outside Dia."
  Form1.MSFlexGrid2.TextMatrix(a + 1, 1) = "="
  Form1.MSFlexGrid2.TextMatrix(a + 1, 2) = Format(AvgTireOutsideDiameter, "0.0") & " ft"

  Form1.MSFlexGrid2.TextMatrix(a + 2, 0) = "Avg. Tire Volume"
  Form1.MSFlexGrid2.TextMatrix(a + 2, 1) = "="
  Form1.MSFlexGrid2.TextMatrix(a + 2, 2) = Format(AvgTireVol, "0.0") & " ft^3"

  Form1.MSFlexGrid2.TextMatrix(a + 3, 0) = "Total Number of Rows"
  Form1.MSFlexGrid2.TextMatrix(a + 3, 1) = "="
  Form1.MSFlexGrid2.TextMatrix(a + 3, 2) = NumRows

  Form1.MSFlexGrid2.TextMatrix(a + 4, 0) = "Total Number of Layers"
  Form1.MSFlexGrid2.TextMatrix(a + 4, 1) = "="
  Form1.MSFlexGrid2.TextMatrix(a + 4, 2) = NumLayers
```

```
  For i = 1 To NumLayers
  Form1.MSFlexGrid2.TextMatrix(a + 4 + i, 0) = "   - Layer(" & i & ") has " &
Layer(i).NumTiresPerRow & " tires"
  Form1.MSFlexGrid2.TextMatrix(a + 5 + i, 0) = ""
  Form1.MSFlexGrid2.TextMatrix(a + 6 + i, 0) = ""
  Form1.MSFlexGrid2.TextMatrix(a + 5 + i, 1) = ""
  Form1.MSFlexGrid2.TextMatrix(a + 6 + i, 1) = ""
  Form1.MSFlexGrid2.TextMatrix(a + 5 + i, 2) = ""
  Form1.MSFlexGrid2.TextMatrix(a + 6 + i, 2) = ""
  Next i

  Form1.MSFlexGrid2.Row = a + 3 + NumLayers
  Form1.MSFlexGrid2.Col = 0
  Form1.MSFlexGrid2.RowSel = a + 4 + NumLayers
  Form1.MSFlexGrid2.ColSel = 2
  Form1.MSFlexGrid2.FillStyle = flexFillRepeat
  Form1.MSFlexGrid2.CellForeColor = vbBlack

  Form1.MSFlexGrid2.Row = a + 5 + NumLayers
  Form1.MSFlexGrid2.Col = 0
  Form1.MSFlexGrid2.RowSel = a + 11 + NumLayers
  Form1.MSFlexGrid2.ColSel = 2
  Form1.MSFlexGrid2.FillStyle = flexFillRepeat
  Form1.MSFlexGrid2.CellForeColor = QBColor(1)

  Form1.MSFlexGrid2.TextMatrix(a + 5 + NumLayers, 0) = "--------------------------------------------------"
  Form1.MSFlexGrid2.TextMatrix(a + 5 + NumLayers, 1) = "--------------------------------------------------"
  Form1.MSFlexGrid2.TextMatrix(a + 5 + NumLayers, 2) = "--------------------------------------------------"
  Form1.MSFlexGrid2.MergeCells = flexMergeRestrictRows
  Form1.MSFlexGrid2.MergeRow(a + 5 + NumLayers) = True

  Form1.MSFlexGrid2.TextMatrix(a + 6 + NumLayers, 0) = "Total Number of Tires"
  Form1.MSFlexGrid2.TextMatrix(a + 6 + NumLayers, 1) = "="
  Form1.MSFlexGrid2.TextMatrix(a + 6 + NumLayers, 2) = NumTires

  Form1.MSFlexGrid2.TextMatrix(a + 7 + NumLayers, 0) = "Total Dam Volume"
  Form1.MSFlexGrid2.TextMatrix(a + 7 + NumLayers, 1) = "="
  Form1.MSFlexGrid2.TextMatrix(a + 7 + NumLayers, 2) = Format(TotalStructureVol, ".0") & " ft^3"

  Form1.MSFlexGrid2.TextMatrix(a + 8 + NumLayers, 0) = "Estimated Rubber Volume"
  Form1.MSFlexGrid2.TextMatrix(a + 8 + NumLayers, 1) = "="
  Form1.MSFlexGrid2.TextMatrix(a + 8 + NumLayers, 2) = Format(TotalRubberOnlyVol, ".0") & "
ft^3"

  Form1.MSFlexGrid2.TextMatrix(a + 9 + NumLayers, 0) = "Estimated Material Volume"
  Form1.MSFlexGrid2.TextMatrix(a + 9 + NumLayers, 1) = "="
  Form1.MSFlexGrid2.TextMatrix(a + 9 + NumLayers, 2) = Format(TotalMatOnlyVol, ".0") & " ft^3"

  Form1.MSFlexGrid2.TextMatrix(a + 10 + NumLayers, 0) = "Est. Material Unit Weight"
  Form1.MSFlexGrid2.TextMatrix(a + 10 + NumLayers, 1) = "="
  Form1.MSFlexGrid2.TextMatrix(a + 10 + NumLayers, 2) = Format(TotalMatOnlyUnitWeight, ".0") &
" pcf"

  Form1.MSFlexGrid2.TextMatrix(a + 11 + NumLayers, 0) = "--------------------------------------------------"
```

```
    Form1.MSFlexGrid2.TextMatrix(a + 11 + NumLayers, 1) = "----------------------------------------------"
    Form1.MSFlexGrid2.TextMatrix(a + 11 + NumLayers, 2) = "----------------------------------------------"
    Form1.MSFlexGrid2.MergeCells = flexMergeRestrictRows
    Form1.MSFlexGrid2.MergeRow(a + 11 + NumLayers) = True

    Form1.MSFlexGrid2.TextMatrix(a + 12 + NumLayers, 0) = ""
    Form1.MSFlexGrid2.TextMatrix(a + 12 + NumLayers, 1) = ""
    Form1.MSFlexGrid2.TextMatrix(a + 12 + NumLayers, 2) = ""
    Form1.MSFlexGrid2.TextMatrix(a + 13 + NumLayers, 0) = ""
    Form1.MSFlexGrid2.TextMatrix(a + 13 + NumLayers, 1) = ""
    Form1.MSFlexGrid2.TextMatrix(a + 13 + NumLayers, 2) = ""

    b = a + 12 + NumLayers
    Form1.MSFlexGrid2.TopRow = 8

End Sub

Public Sub FasterningAnalysis()

    Dim i As Long
    ReDim Layer(NumLayers)

    For i = 1 To NumLayers
    Layer(i).ShearForce = HorizontalReaction / NumLayers * i
    Next i

    Form1.MSFlexGrid2.Row = b
    Form1.MSFlexGrid2.Col = 0
    Form1.MSFlexGrid2.FillStyle = flexFillSingle
    Form1.MSFlexGrid2.CellFontUnderline = True
    Form1.MSFlexGrid2.CellForeColor = vbRed
    Form1.MSFlexGrid2.TextMatrix(b, 0) = "Fasterning System Analysis"

    Form1.MSFlexGrid2.TextMatrix(b + 1, 0) = "Assumptions"

End Sub

Private Sub Command2_Click()

    Dim ii As Long, jj As Long
    ii = 10
    Form4.txtOutput.Text = "DA" & vbTab & "Y2" & vbTab & "Ydam" & vbTab & "Ytail" & vbTab &
"F11" & vbTab & "Drag" & vbTab & "F5,6" & vbTab & "F7,71" & vbTab & "RH" & vbTab & "Dry U.W."
& vbTab & "FS"
    'Form4.txtOutput.Text = "Dry U.W."
    Do Until ii = 160
    Form1.MSFlexGrid1.TextMatrix(1, 2) = ii
    Call Form1.RiverAnalysis
    'Output to Form4.txtOutput
    Form4.txtOutput.Text = Form4.txtOutput.Text & vbCrLf & Format(DA, "0") & vbTab & Format(Y2,
"0.00") & vbTab & Format(Ydam, "0.00") & vbTab & Format(NoDamDepth, "0.00") & vbTab &
Format(TotalForce11, "0.00") & vbTab & Format(FlowDrag, "0.00") & vbTab &
Format(TotalForce5to6, "0.00") & vbTab & Format(TotalForce7to71, "0.00") & vbTab &
```

```
Format(HorizontalReaction, "0.00") & vbTab & Format(DamMinDryUnitWeight, "0.0") & vbTab &
Format(FSSliding, "0.0")
   Form4.Text1.Text = Form4.Text1.Text & vbCrLf & Format(DamMinDryUnitWeight, "0.0") & vbTab
& Format(FSSliding, "0.0")
   ii = ii + 10
   jj = jj + 1
   Loop

   Form4.Show
End Sub
```

Form2.frm

```
Option Explicit

Private Sub Form_Load()

   Form2.Height = 4020
   Form2.Width = 5475
   Form2.Left = (Screen.Width - Form2.Width) / 2
   Form2.Top = (Screen.Height - Form2.Height) / 2

End Sub

Private Sub Command1_Click()

   Call Form2.ShowInput
   Select Case Form2.lblcounter.Caption
   Case "2"
      Form2.Command1.Visible = False
      Form2.Command2.Visible = True
      Call Form1.DischargeCode
   Case "3"
      Form2.Command1.Visible = True
      Form2.Command2.Visible = True
      Call Form1.StreamSectionCode
   Case "4"
      Form2.Command1.Visible = True
      Form2.Command2.Visible = True
      Call Form1.ControlStructureCode
   Case "5"
      Form2.Command1.Visible = True
      Form2.Command2.Visible = True
      Call Form1.SedimentCode
   End Select
   Form2.txt1.SetFocus

End Sub

Private Sub Command2_Click()

   Call Form2.ShowInput
```

```
  Select Case Form2.lblcounter.Caption
  Case "1"
    Form2.Command1.Visible = True
    Form2.Command2.Visible = True
    Call Form1.StreamSectionCode
  Case "2"
    Form2.Command1.Visible = True
    Form2.Command2.Visible = True
    Call Form1.ControlStructureCode
  Case "3"
    Form2.Command1.Visible = True
    Form2.Command2.Visible = True
    Call Form1.SedimentCode
  Case "4"
    Form2.Command1.Visible = True
    Form2.Command2.Visible = False
    Call Form1.StructureBedInterfaceCode
  End Select
  Form2.txt1.SetFocus

End Sub

Private Sub Command3_Click()

  Call Form2.ShowInput
  Unload Form2
  Form1.SetFocus
  Form1.MSFlexGrid1.SetFocus

End Sub

Public Sub ShowInput()

  Select Case Form2.lblcounter.Caption
  Case "1"
    Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption, 2) = Form2.txt1.Text
    Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 1, 2) = Form2.txt2.Text
    Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 2, 2) = Form2.txt3.Text
    Form1.MSFlexGrid1.TopRow = 0
    Form1.MSFlexGrid1.Row = 3
    Form1.MSFlexGrid1.Col = 2
  Case "2"
    Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 3, 2) = Form2.txt1.Text
    Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 4, 2) = Form2.txt2.Text
    Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 5, 2) = Form2.txt3.Text
    Form1.MSFlexGrid1.TopRow = 0
    Form1.MSFlexGrid1.Row = 7
    Form1.MSFlexGrid1.Col = 2
  Case "3"
    Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 6, 2) = Form2.txt1.Text
    Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 7, 2) = Form2.txt2.Text
    Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 8, 2) = Form2.txt3.Text
    Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 9, 2) = Form2.txt4.Text
    Form1.MSFlexGrid1.TopRow = 0
```

```
      Form1.MSFlexGrid1.Row = 12
      Form1.MSFlexGrid1.Col = 2
   Case "4"
      Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 10, 2) = Form2.txt1.Text
      Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 11, 2) = Form2.txt2.Text
      Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 12, 2) = Form2.txt3.Text
      Form1.MSFlexGrid1.TopRow = 0
      Form1.MSFlexGrid1.Row = 16
      Form1.MSFlexGrid1.Col = 2
   Case "5"
      Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 13, 2) = Form2.txt1.Text
      Form1.MSFlexGrid1.TextMatrix(Form2.lblcounter.Caption + 14, 2) = Form2.txt2.Text
      Form1.MSFlexGrid1.TopRow = 8
      Form1.MSFlexGrid1.Row = 23
      Form1.MSFlexGrid1.Col = 2
   End Select

End Sub
```

Form3.frm

```
Option Explicit

   Dim i As Long

Private Sub Form_Load()

   Form3.Width = 7050
   Form3.Height = 4485
   Form3.Left = (Screen.Width - Form3.Width) / 2
   Form3.Top = (Screen.Height - Form3.Height) / 2

   If Form1.MSFlexGrid2.TextMatrix(0, 0) <> "" Then
      If Form1.MSFlexGrid2.TextMatrix(1, 2) = "Option 1" Then
      Form3.Option1.Value = True
      Else
      Form3.Option2.Value = True
      End If
      If Form1.MSFlexGrid2.TextMatrix(3, 2) = "0" And Form1.MSFlexGrid2.TextMatrix(4, 2) = "100"
Then
      Form3.optSizeNo.Value = True
      Else
      Form3.optSizeYes.Value = True
      End If
      Form3.txtR13Percent = Form1.MSFlexGrid2.TextMatrix(3, 2)
      Form3.txtR14Percent = Form1.MSFlexGrid2.TextMatrix(4, 2)
      Form3.txtR15Percent = Form1.MSFlexGrid2.TextMatrix(5, 2)
      Form3.txtVoids = Form1.MSFlexGrid2.TextMatrix(6, 2)
   End If

End Sub
```

```
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)

   Unload Form3

End Sub

Private Sub Command3_Click()

   Dim TotalPercent As Double

   'to check
   TotalPercent = Val(Form3.txtR13Percent.Text) + Val(Form3.txtR14Percent.Text) +
Val(Form3.txtR15Percent.Text)
   If TotalPercent <> 100 Then
   MsgBox "ERROR: Total tire percentage is incorrect!"
   Form3.Show
   Form3.txtR13Percent.SetFocus
   Else
   If Form3.txtR13Percent.Text = "" Then Form3.txtR13Percent.Text = "0"
   If Form3.txtR14Percent.Text = "" Then Form3.txtR14Percent.Text = "0"
   If Form3.txtR15Percent.Text = "" Then Form3.txtR15Percent.Text = "0"
   For i = 0 To 99
      Form1.MSFlexGrid2.TextMatrix(i, 0) = ""
      Form1.MSFlexGrid2.TextMatrix(i, 1) = ""
      Form1.MSFlexGrid2.TextMatrix(i, 2) = ""
   Next i
   Form1.MSFlexGrid2.Row = 0
   Form1.MSFlexGrid2.Col = 0
   Form1.MSFlexGrid2.RowSel = 99
   Form1.MSFlexGrid2.ColSel = 2
   Form1.MSFlexGrid2.FillStyle = flexFillRepeat
   Form1.MSFlexGrid2.CellFontUnderline = False
   Form1.MSFlexGrid2.CellForeColor = vbBlack
   Call Form3.ShowInput
   Unload Me
   End If

End Sub

Private Sub optSizeYes_GotFocus()

   Form3.optSizeYes.Value = True
   Form3.txtR13Percent.Text = ""
   Form3.txtR14Percent.Text = ""
   Form3.txtR15Percent.Text = ""
   Form3.txtR13Percent.SetFocus

End Sub

Private Sub optSizeNo_GotFocus()

   Form3.optSizeNo.Value = True
   Form3.txtR13Percent.Text = 0
   Form3.txtR14Percent.Text = 100
```

```
    Form3.txtR15Percent.Text = 0

End Sub

Public Sub ShowInput()

    Form1.MSFlexGrid2.Row = 0
    Form1.MSFlexGrid2.Col = 0
    Form1.MSFlexGrid2.FillStyle = flexFillSingle
    'Form1.MSFlexGrid2.CellForeColor = vbBlack
    Form1.MSFlexGrid2.CellFontUnderline = True
    Form1.MSFlexGrid2.TextMatrix(0, 0) = "Input - Tires Information"

    Form1.MSFlexGrid2.Row = 1
    Form1.MSFlexGrid2.Col = 0
    Form1.MSFlexGrid2.RowSel = 6
    Form1.MSFlexGrid2.ColSel = 2
    Form1.MSFlexGrid2.FillStyle = flexFillRepeat
    Form1.MSFlexGrid2.CellForeColor = vbBlue

    Form1.MSFlexGrid2.TextMatrix(1, 0) = "      " + "Stacking Method"
    Form1.MSFlexGrid2.TextMatrix(1, 1) = "="
    If Form3.Option2.Value = True Then
    Form1.MSFlexGrid2.TextMatrix(1, 2) = "Option 2"
    Else
    Form1.MSFlexGrid2.TextMatrix(1, 2) = "Option 1"
    End If

    Form1.MSFlexGrid2.TextMatrix(2, 0) = "      " + "Size Percentage"
    Form1.MSFlexGrid2.TextMatrix(3, 0) = "            " + "R13"
    Form1.MSFlexGrid2.TextMatrix(4, 0) = "            " + "R14"
    Form1.MSFlexGrid2.TextMatrix(5, 0) = "            " + ">= R15"
    Form1.MSFlexGrid2.TextMatrix(3, 1) = "="
    Form1.MSFlexGrid2.TextMatrix(4, 1) = "="
    Form1.MSFlexGrid2.TextMatrix(5, 1) = "="

    Form1.MSFlexGrid2.Row = 2
    Form1.MSFlexGrid2.Col = 2
    Form1.MSFlexGrid2.CellFontItalic = True
    If Form3.txtR13Percent.Text = "0" And Form3.txtR14Percent.Text = "100" Then
    Form1.MSFlexGrid2.TextMatrix(2, 2) = "(unknown)"
    Else
    Form1.MSFlexGrid2.TextMatrix(2, 2) = "(known)"
    End If

    Form1.MSFlexGrid2.Row = 2
    Form1.MSFlexGrid2.Col = 2
    Form1.MSFlexGrid2.RowSel = 99
    Form1.MSFlexGrid2.ColSel = 2
    Form1.MSFlexGrid2.FillStyle = flexFillRepeat
    Form1.MSFlexGrid2.CellAlignment = 0
    Form1.MSFlexGrid2.TextMatrix(3, 2) = Form3.txtR13Percent.Text
    Form1.MSFlexGrid2.TextMatrix(4, 2) = Form3.txtR14Percent.Text
    Form1.MSFlexGrid2.TextMatrix(5, 2) = Form3.txtR15Percent.Text
```

```
    Form1.MSFlexGrid2.TextMatrix(6, 0) = "      " + "Percent Void Unfilled"
    Form1.MSFlexGrid2.TextMatrix(6, 1) = "="
    Form1.MSFlexGrid2.TextMatrix(6, 2) = Form3.txtVoids.Text

    Form1.MSFlexGrid2.TopRow = 0

    Call Form1.VolumeAnalysis
    Call Form1.FasterningAnalysis

    Form1.MSFlexGrid2.Row = 7
    Form1.MSFlexGrid2.Col = 2
    Form1.MSFlexGrid2.SetFocus

End Sub
```

Form4.frm

```
Option Explicit

Private Sub Form_Load()

    Form4.Width = Screen.Width
    Form4.Left = 0
    Form4.txtOutput.Width = Screen.Width

End Sub
```

Error Checking Module

```
Option Explicit

Public Sub ErrorChecking(n As Long, result As String)

    Dim i As Long
    i = 0
    result = "True"
    Do Until i = n - 1
    If Form1.MSFlexGrid1.TextMatrix(i, 1) = "=" And Form1.MSFlexGrid1.TextMatrix(i, 2) = "" Then
    result = "False"
    Exit Do
    End If
    i = i + 1
    Loop

End Sub
```

Module1

```
Option Explicit

Public Type LayerInfo

   TotalWidth As Double
   NumTiresPerRow As Long
   TotalNumTires As Long
   ShearForce As Double

End Type
```

Solver Module

```
Option Explicit

Public Sub Polynomial(n As Double, StreamWidth As Double, StreamSlope As Double, Flowrate As
Double, Depth As Double)

   Dim Const1 As Double, Area As Double, Perimeter As Double
   Dim Q As Double

   Depth = 0.1
   Const1 = 1.486 / n * StreamWidth * StreamSlope ^ 0.5
   Area = StreamWidth * Depth
   Perimeter = StreamWidth + 2 * Depth
   Q = Const1 * (Area / Perimeter) ^ (2 / 3) * Depth
   Do Until Q > Flowrate
   Depth = Depth + 0.01
   Area = StreamWidth * Depth
   Perimeter = StreamWidth + 2 * Depth
   Q = Const1 * (Area / Perimeter) ^ (2 / 3) * Depth
   Loop

End Sub
```

REFERENCES CITED

Boyken, M. S. "Hydrologic and Hydraulic Analyses of Stream Stabilization and Grade Control Structures in Western Iowa." Master's Thesis, Iowa State University, Ames, Iowa, 1998.

Chaudhry, M.H. Open-Channel Flow. Prentice Hall, Inc. Englewood Cliffs, NJ, 1993.

Gebhardt, M. A. "Shear Strength of Shredded Tires as Applied to the Design and Construction of a Shredded Tire Stream Crossing." Master's Thesis, Iowa State University, Ames, Iowa, 1997.

Goehring, R. L. "Preliminary Structural and Hydraulic Analysis of Flume Bridges in Western Iowa." Master's Thesis, Iowa State University, Ames, Iowa, 1981.

Gupta, R. S. Hydrology and Hydraulic Systems. Waveland Press, Inc. Prospect Heights, IL, 1989.

Hanson, G. J., R. A. Lohnes, and F. W. Klaiber. 1985. "Final Report on Phases II and III, Evaluation of Control Structures for Stabilizing Degrading Streams Channels in Western Iowa," ISU-ERI-Ames no. 86050. Iowa State University, Ames, Iowa.

Levich, B. A. "Studies of Tractive Force Models on Degrading Streams." Master's Thesis, Iowa State University, Ames, Iowa, 1994.

Magner, J. L. "Guidelines for Grade Stabilization Structure Placement." Master's Thesis, Iowa State University, Ames, Iowa, 1994.

Morris, H. M., and J. M. Wiggert. Applied Hydraulics in Engineering. John Wiley & Sons. New York, 1972.

Poh, S. H., and B. B. Broms. "Slope Stabilization Using Old Rubber Tires and Geotextiles." Journal of Performance of Constructed Facilities, Vol. 9, No. 1, February 1995.

Voegele, D. M. "The Evaluation of Riprap and Low-Cost Grade Control Structures in Western Iowa." Master's Thesis, Iowa State University, Ames, Iowa, 1997.

Yang, S. "Use of Scrap Tires in Civil Engineering Applications." Ph.D. Dissertation, Iowa State University, Ames, Iowa, 1999.

Zimmerman, P. S. "Compressibility, Hydraulic Conductivity and Soil Infiltration Testing of Tire Shreds and Field Testing of a Shredded Tire Horizontal Drain." Master's Thesis, Iowa State University, Ames, Iowa, 1997.