NATIONAL TECHNICAL UNIVERSITY OF ATHENS

SCHOOL OF CIVIL ENGINEERING

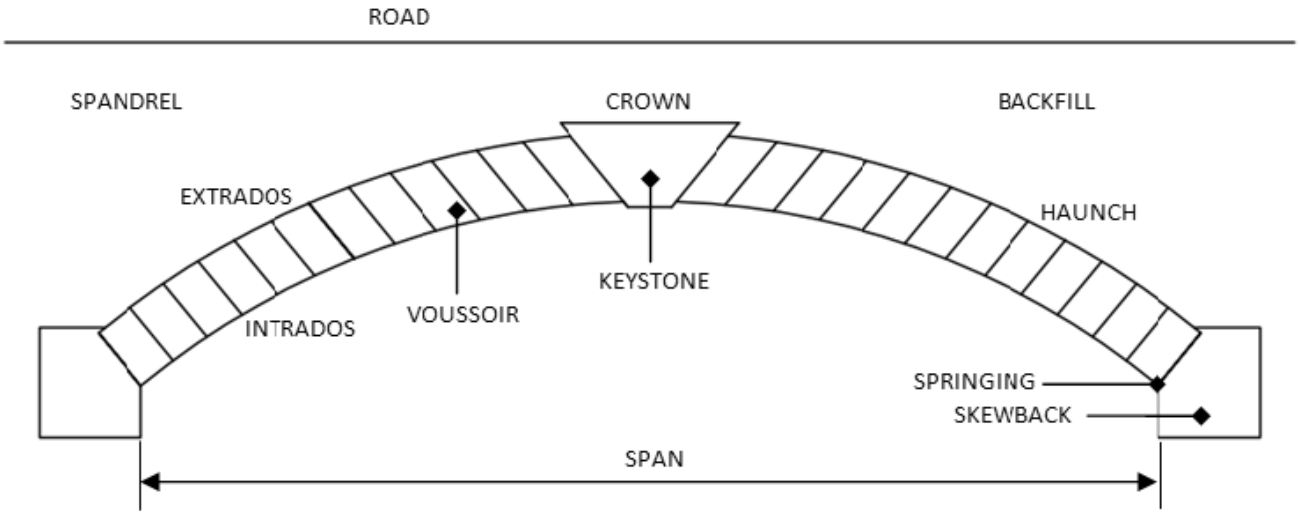
DEPARTMENT OF STRUCTURAL ENGINEERING

POSTGRADUATE COURSE OF STUDIES

ANALYSIS AND DESIGN OF STRUCTURES

POSTGRADUATE THESIS

PARAMETRIC LIMIT ANALYSIS OF RIGID BODY ASSEMBLIES



POSTGRADUATE STUDENT: ANTONIADES LAURENCE–KIMON

THESIS SUPERVISOR: PROFESSOR KOUMOUSIS VLASIOS

ATHENS, OCTOBER 2018

Cover Figure: Diagram of the Structural Elements of a Masonry Arch Bridge.
Source: "https://sites.google.com/site/cmtengineer1/masonry-arch-structure".

ABSTRACT

The primary scope of this thesis was the investigation of limit analysis of rigid body assemblies, following Livesley's 'Limit Analysis of Structures Formed from Rigid Blocks' (1978).

The investigation has been carried out through the development of the 'Arch Stability' MATLAB program, using the methodology proposed by the original author, following the provided formulation.

The geometry of a circular arch was selected to be further investigated, since it is a common geometry that has also been investigated in Heyman's 'The Stone Skeleton' (1966), as well as in Milankovitch's 'Beitrag zur Theorie der Druckkurven' (1904).

A numerical confirmation of the minimum thickness of a semicircular arch under its own weight, as calculated by the Serbian scholar Milutin Milankovitch (1904), has been achieved using the 'Arch Stability' MATLAB program.

The secondary scope of this thesis was the investigation of the stability of embankments, following Spencer's 'A Method of Analysis of the Stability of Embankments Assuming Parallel Inter-Slice Forces' (1967).

The investigation has been carried out through the development of the 'Slope Stability' MATLAB program, using the methodology proposed by the original author, following the provided formulation.

The basic concepts of the original method of analysis (1967) were followed, using the revised formulation (1973), where the moment equilibrium equation for each slice is being formulated by considering the moments of the forces about the middle of the base of each slice, instead of the centre of rotation.

A confirmation of the results of the examples presented in the original investigation has been achieved using the 'Slope Stability' MATLAB program.

TABLE OF CONTENTS

TABLE OF FIGURES

1. Introduction

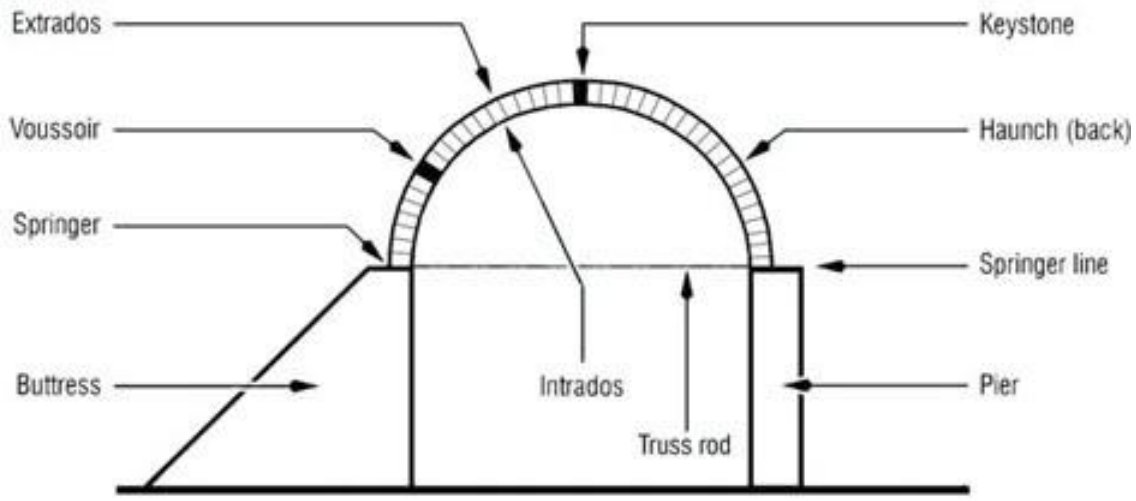


Figure 1-1: *Terminology for an Arch.*
Source: Auroville Earth Institute (AVEI).

1.1 Primary scope of this thesis

The primary scope of this thesis was the investigation of limit analysis of rigid body assemblies, following Livesley's 'Limit Analysis of Structures Formed from Rigid Blocks' (1978).

The investigation should be carried out through the development of a parametric MATLAB program, using the methodology proposed by the original author, following the provided formulation.

The MATLAB program should be able to replicate the functionality of the original FORTRAN program, also having the ability to be parametrically extended, to include alternative formulations.

The next step would be to use the MATLAB program to determine the limit load of a simple structure formed from rigid blocks.

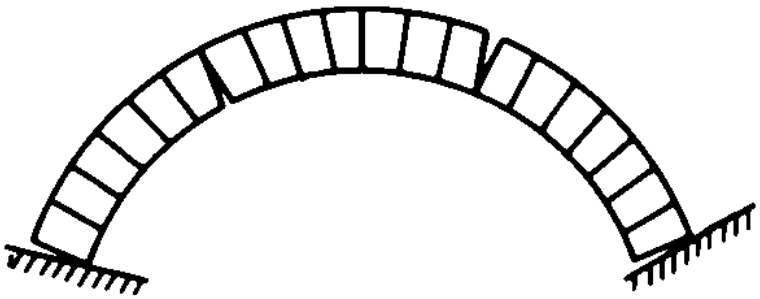


Figure 1-2: *Collapse of a Voussoir Arch (after Pippard and Baker).*
Source: Heyman, J. (1966). The Stone Skeleton.

The geometry of a circular arch was selected to be further investigated, since it is a common geometry that has also been investigated in Heyman's 'The Stone Skeleton' (1966), as well as in Milankovitch's 'Beitrag zur Theorie der Druckkurven' (1904).

Livesley's limit analysis method also takes into account of the Coulomb friction, while Heyman's method assumes that (i) stone has no tensile strength, (ii) the compressive strength of stone is effectively infinite, (iii) sliding of one stone upon another cannot occur.

It is obvious that by using a parametric programming code following Livesley's formulation, Heyman's formulation can also be investigated by assuming zero tensile strength, infinite compressive strength, and infinite Coulomb friction, through the adoption of the appropriate values of the block interface stress resultants limits, namely $0 \le q \le +\infty$, $0 \le s \le +\infty$, $-\infty \le t \le +\infty$.

The MATLAB program should be also able to confirm the minimum thickness of a semicircular arch under its own weight, as calculated by the Serbian scholar Milutin Milankovitch (1904).

According to Milankovitch's theory of the thrust line, the minimum thickness of a semicircular arch under its own weight is $d_{min} = 0.1075 \cdot R$, while the thrust line touches the intrados at the point corresponding to the rupture angle $54°29'$.

1.2 Secondary scope of this thesis

The secondary scope of this thesis was the investigation of the stability of embankments, following Spencer's 'A Method of Analysis of the Stability of Embankments Assuming Parallel Inter-Slice Forces' (1967).

As in the previous case, the investigation should be carried out through the development of a parametric MATLAB program, using the methodology proposed by the original author, following the provided formulation.

The MATLAB program should be able to replicate the functionality of the original ATLAS program, also having the ability to be parametrically extended, to include alternative formulations.

The next step would be to use the MATLAB program to confirm the results of the examples presented in the original investigation.

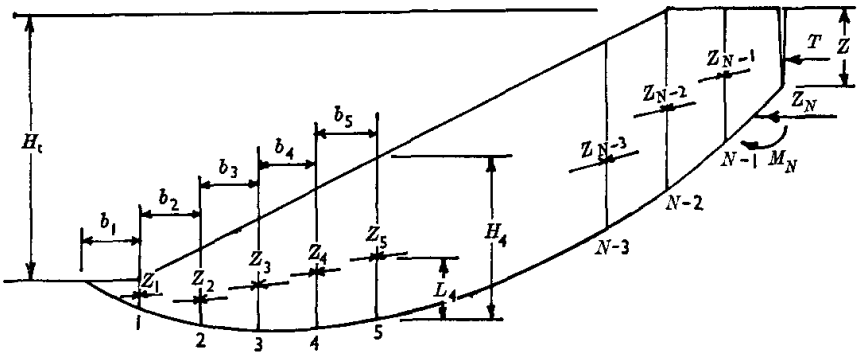


Figure 1-3: *Cross-Section Through Embankment.*
Source: Spencer, E. (1973). Thrust Line Criterion in Embankment Stability Analysis.

The geometry of an embankment discretized into vertical slices, as used in embankment stability analysis problems, roughly resembles the circular arch geometry.

From a technical point of view, these two problems may seem unrelated to each other, but from a programming view, these two problems are highly related as far as the configuration of geometry is concerned, with respect to the equations of equilibrium, thus, common programming techniques can be used to address these two otherwise unrelated problems.

1.3 References

1. *Livesley, R. (1978). Limit Analysis of Structures Formed from Rigid Blocks. International Journal for Numerical Methods in Engineering 12(12):1853-1871 · January 1978.*
2. *Heyman, J. (1966). The Stone Skeleton. International Journal of Solids and Structures 2(2):249-279 · April 1966.*
3. *Milankovitch, M. (1904). Beitrag zur Theorie der Druckkurven. Dissertation zur Erlangung der Doktorwürde, K.K. Technische Hochschule, Vienna.*
4. *Spencer, E. (1967). A Method of Analysis of the Stability of Embankments Assuming Parallel Inter-Slice Forces. Géotechnique Volume 17 Issue 1, March 1967, pp. 11-26.*
5. *Spencer, E. (1973). Thrust Line Criterion in Embankment Stability Analysis. Géotechnique Volume 23 Issue 1, March 1973, pp. 85-100.*

This page is intentionally left blank

2. Limit Analysis of Rigid Block Structures [Livesley (1978)]

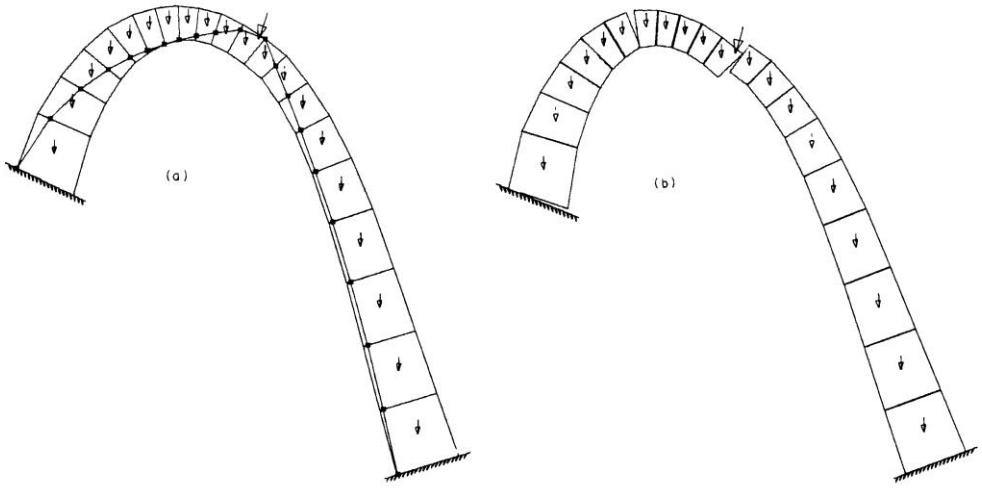


Figure 2-1: *Thrust Line and Collapse Mechanism for a Typical Arch.*
Source: Livesley, R. (1978). Limit Analysis of Structures Formed from Rigid Blocks.

2.1 Synopsis

Livesley (1978) adapted a technique previously developed for the analysis of rigid-plastic structural frames to provide a formal procedure for finding the limit load of any structure formed from rigid blocks.

In the proposed procedure the load factor is maximized subject to the equilibrium equations of the structure and linear constraints imposed by criteria of failure at the block interfaces.

Initially, it was assumed that the limiting shear force associated with sliding at a block interface is independent of the normal component of force across the interface.

This assumption means that the normality rule is satisfied, so that the upper- and lower-bound theorems of classical limit analysis apply.

Then, it was assumed that the limit on the shear force at a block interface is that associated with Coulomb friction.

The computational algorithm that was devised for the previous case was easily extended to deal with this situation.

However, it became evident that the failure mechanism computed by the algorithm will not necessarily satisfy the normality rule.

The corresponding limit load may therefore be an over-estimate of the true failure load, even though it is computed by a lower-bound (equilibrium) approach.

Finally, a criterion was established for testing the validity of a failure load computed in these circumstances.

2.2 A general formulation of linear limit analysis problems

If the 'equilibrium' and 'mechanism' formulations of a limit analysis problem are linearized, they give rise to dual linear programming problems.

The limit analysis problem for proportional loading can be written as 'Maximize the load factor λ, subject to the equilibrium equations $\lambda \mathbf{p} = \mathbf{H}\,\mathbf{r}$ and the yield constraints $-\mathbf{r}^{\mathbf{L}} \le \mathbf{r} \le \mathbf{r}^{\mathbf{U}}$.'

In the above 'equilibrium' formulation, λ is the load factor, $\mathbf{p}$ is the vector of the loads, $\mathbf{H}$ is the matrix of the coefficients of the three scalar equilibrium equations, $\mathbf{r}$ is the vector of the stress resultants, and $\mathbf{r}^{\mathbf{L}}$ and $\mathbf{r}^{\mathbf{U}}$ are the vectors of the stress resultants lower and upper limits respectively.

Details of the above procedure, as it appears when applied to the collapse analysis of rigid-jointed frameworks, have been given elsewhere. The collapse mechanism may be obtained as a by-product of the analysis.

Heyman (1966) showed that the collapse load of a masonry arch or dome could be formulated as a problem in limit analysis. That work led Livesley (1978) to consider the possibility of converting an existing computer program for plane skeletal frame collapse, based on the procedure outlined above, into a program for the collapse analysis of plane masonry arches of arbitrary shape.

2.3 The collapse analysis of masonry arches

At first sight it seems self-evident that the trapezoidal blocks of a typical arch correspond to the elements of rigid-jointed frameworks, while the block interfaces seem to be the element boundaries.

However, this superficial identification ignores the fact that the yield constraints $-\boldsymbol{r}^{L} \le \boldsymbol{r} \le \boldsymbol{r}^{U}$ operate on vectors $\boldsymbol{r}$ which are associated with elements, not element boundaries.

It follows that to apply the general ideas set out in the previous section we must regard the block interfaces as our elements, and treat the blocks simply as extended nodes connecting the elements, at which equilibrium must be satisfied.

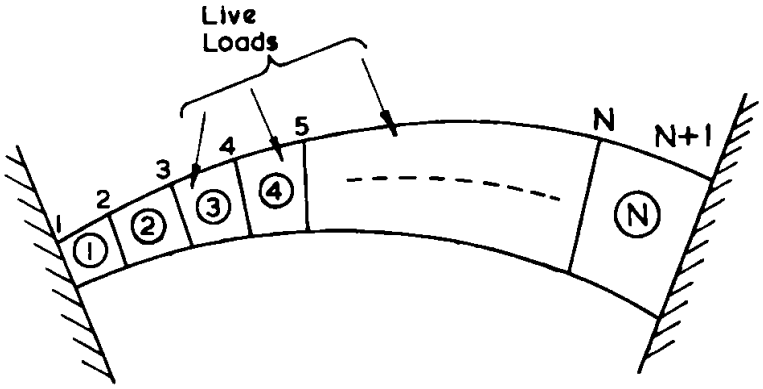


Figure 2-2: *A Plane Arch of N Rigid Blocks.*
Source: Livesley, R. (1978). Limit Analysis of Structures Formed from Rigid Blocks.

The stress distribution over the block contact area may be represented, as far as overall equilibrium is concerned, by the stress-resultants q, s, t. These three self-equilibrating quantities make up the vector $\boldsymbol{r}$ for the element. Associated with $\boldsymbol{r}$ is a deformation vector ε whose components ε_q, ε_s, ε_t 'correspond' to q, s, t in a virtual work sense.

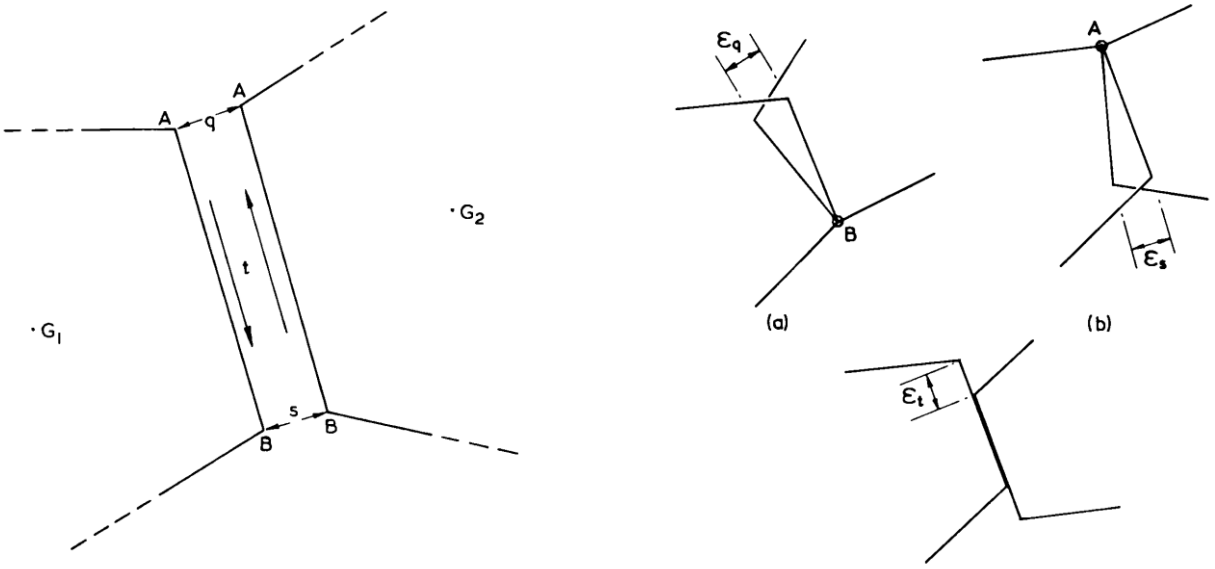


Figure 2-3: *Sign Conventions for the Components of* $\boldsymbol{r}$ *and* $\boldsymbol{\varepsilon}$.
Source: Livesley, R. (1978). Limit Analysis of Structures Formed from Rigid Blocks.

The block has dead-weight W, which makes up the vector $\boldsymbol{w}$ for the element, and may also support one or more live loads p_x, p_y, m_z, which make up the vector $\boldsymbol{p}$ for the element.

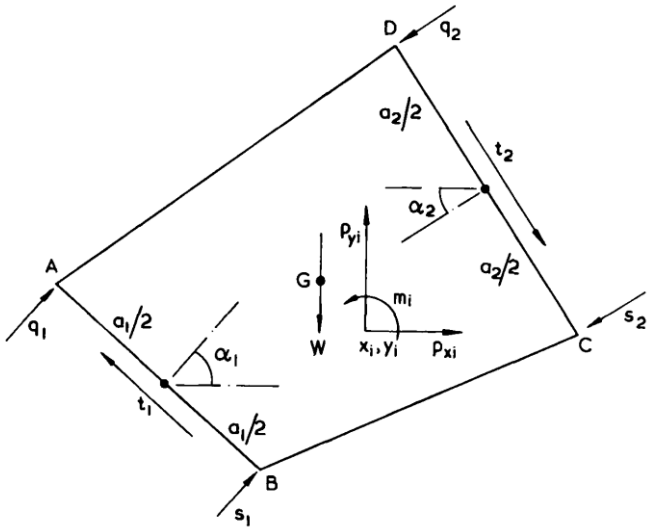


Figure 2-4: *Notation for Block Equilibrium Equations.*
Source: Livesley, R. (1978). Limit Analysis of Structures Formed from Rigid Blocks.

The coefficients of the three scalar equilibrium equations (force equilibrium in the x and y directions and moment equilibrium about the centre of gravity G of the block) make up the matrices $\boldsymbol{G}$ and $\boldsymbol{H}$ for the block, corresponding to the interfaces AB and CD respectively.

$$\begin{bmatrix} 0 \\ -W \\ 0 \end{bmatrix} + \begin{bmatrix} \Sigma\, p_{xi} \\ \Sigma\, p_{yi} \\ \Sigma[\, p_{yi}x_i' - p_{xi}y_i' + m_{zi}\,] \end{bmatrix} =$$

$$\begin{bmatrix} -C_1 & -C_1 & S_1 \\ -S_1 & -S_1 & -C_1 \\ C_1y_A' - S_1x_A' & C_1y_B' - S_1x_B' & -C_1x_1' - S_1y_1' \end{bmatrix} \begin{bmatrix} q_1 \\ s_1 \\ t_1 \end{bmatrix} + \qquad \boldsymbol{w} + \boldsymbol{p} = \boldsymbol{G}\,\boldsymbol{r_1} + \boldsymbol{H}\,\boldsymbol{r_2}$$

$$\begin{bmatrix} C_2 & C_2 & -S_2 \\ S_2 & S_2 & C_2 \\ S_2x_D' - C_2y_D' & S_2x_C' - C_2y_C' & S_2y_2' + C_2x_2' \end{bmatrix} \begin{bmatrix} q_2 \\ s_2 \\ t_2 \end{bmatrix}$$

where:

$$S1 = \sin\alpha_1 \qquad C1 = \cos\alpha_1 \qquad S2 = \sin\alpha_2 \qquad C2 = \cos\alpha_2$$

$$x_A' = x_A - x_G \qquad x_B' = x_B - x_G \qquad x_C' = x_C - x_G \qquad x_D' = x_D - x_G$$

The arch equilibrium equations may be written in matrix form as:

$$\begin{bmatrix} w_1 \\ \cdot \\ \cdot \\ \cdot \\ w_N \end{bmatrix} + \begin{bmatrix} p_1 \\ \cdot \\ \cdot \\ \cdot \\ p_N \end{bmatrix} = \begin{bmatrix} G_1 & H_1 & & & 0 \\ & G_2 & H_2 & & \\ & & & & \\ 0 & & & & \\ & & & G_N & H_N \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ \cdot \\ \cdot \\ \cdot \\ r_{N+1} \end{bmatrix} \qquad \mathbf{w} + \mathbf{p} = \mathbf{H}\,\mathbf{r}$$

Note that there are three more scalar unknowns than scalar equations, corresponding to the three degrees of statical indeterminacy of the arch (the number of variables is $[3\times(N+1)]$ while the number of equations is $[3\times N]$).

The statical indeterminacy of the arch means that the solution of the above formulated system of linear block equilibrium equations is not unique. Thus, by taking also into account of any assumed constraints, a limit-analysis problem may be formulated.

2.4 Shear force independent of the normal component of force

Initially, it was assumed that the limiting shear force associated with sliding at a block interface is independent of the normal component of force across the interface.

The constraints on the individual components of the vector $\boldsymbol{r}$ associated with each element are assumed to be as: $-q^L \le q \le q^U$, $-s^L \le s \le s^U$ and $-t^L \le t \le t^U$. These constraints may be combined and written as: $-\boldsymbol{r}^{\boldsymbol{L}} \le \boldsymbol{r} \le \boldsymbol{r}^{\boldsymbol{U}}$ for each element, and $-\mathbf{r}^{\mathbf{L}} \le \mathbf{r} \le \mathbf{r}^{\mathbf{U}}$ for all the elements.

We can now use the equilibrium equations and the constraints to formulate two limit-analysis problems.

The first is concerned with whether the arch can support its own weight.

Putting $\mathbf{p} = \mathbf{0}$ in the equilibrium equations we look for a solution of the equation $\mathbf{w} = \mathbf{H}\,\mathbf{r}$ which satisfies the constraints on $\mathbf{r}$.

If this solution exists we ask what load factor λ associated with the live loads will cause collapse of the arch.

This involves maximizing λ subject to the equilibrium equations $\lambda\,\mathbf{p} + \mathbf{w} = \mathbf{H}\,\mathbf{r}$ and the same constraints on $\mathbf{r}$ as before.

Both these problems may be reduced to the problem formulated in the previous section.

The dead-load problem is equivalent to the maximization of a parameter λ_w, where $\lambda_w\,\mathbf{w} = \mathbf{H}\,\mathbf{r}$ and $-\mathbf{r}^{\mathbf{L}} \le \mathbf{r} \le \mathbf{r}^{\mathbf{U}}$, the process being terminated as soon as a value $\lambda_w = 1$ is reached.

If the maximum value of λ_w is less than unity then clearly the arch cannot support its own weight.

If a valid solution to the dead-load problem exists and $\mathbf{r}_w$ is the value of $\mathbf{r}$ associated with this solution, the live-load problem can be written in the form 'Maximize λ subject to the equilibrium equations $\lambda \mathbf{p} = \mathbf{H}(\mathbf{r} - \mathbf{r}_w)$ and the yield constraints $-\mathbf{r}^L - \mathbf{r}_w \le \mathbf{r} - \mathbf{r}_w \le \mathbf{r}^U - \mathbf{r}_w$'. This is essentially the same problem as before in the 'shifted' variables. $\mathbf{r} - \mathbf{r}_w$.

The final pivotal row of the modified matrix $\mathbf{H}$ defines the collapse mechanism, giving (to within an arbitrary multiplier) the values of the variables which correspond in a virtual work sense to the various components of $\mathbf{r}$. In the present case these variables are the deformations ε_q, ε_s, ε_t.

2.5 Yield constraints involving Coulomb friction

Drucker (1954) showed that in general the lower-bound (equilibrium) theorem is not true when yield is associated with Coulomb friction. Drucker's disproof of the theorem is by means of a counter-example.

When the limit on the shear force at a block interface is that associated with Coulomb friction, the failure mechanism computed by the algorithm will not necessarily satisfy the normality rule.

The corresponding limit load may therefore be an over-estimate of the true failure load, even though it is computed by a lower-bound (equilibrium) approach.

Upon further investigation, it became evident that while some counter-examples show that limit analysis does not give the correct failure load, other examples show that it does.

That fact led Livesley (1978) to investigate why this division occurs, establishing a criterion for testing the validity of a failure load computed in these circumstances.

It seems that in the cases that limit analysis gives an over-estimate of the true failure load, the maximizing algorithm assumes that the boundaries do positive work, implying some movement of supposedly fixed surfaces.

2.6 A naive approach of simple Coulomb friction with no cohesion

In a linear programming limit analysis problem, the stress limits associated with each element have to be linearized and expressed as constraints on the set of element stress parameters $\boldsymbol{r}$.

By a suitable choice of $\boldsymbol{r}$, followed if necessary by the introduction of additional (auxiliary) variables, it is possible to express these constraints in the form $-\boldsymbol{r}^{\boldsymbol{L}} \le \boldsymbol{r} \le \boldsymbol{r}^{\boldsymbol{U}}$.

If the condition at a block interface is one of simple Coulomb friction with no cohesion, the constraints on the components of $\boldsymbol{r}$ are:

(a) $0 \le q \le q^U$

(b) $0 \le s \le s^U$

(c) $-\mu(q+s) \le t \le +\mu(q+s)$

The constraints can be put in the required simple form by the introduction of two additional variables at each interface:

u: $u = \mu(q+s) - t$

v: $v = \mu(q+s) + t$

The vector $\boldsymbol{r}$ for an interface is now regarded as consisting of the five quantities q, s, t, u, v, the constraints being:

q: $0 \le q \le q^U$

s: $0 \le s \le s^U$

t: *unconstrained*

u: $0 \le u$

v: $0 \le v$

These constraints are in the form $-\boldsymbol{r}^{\boldsymbol{L}} \le \boldsymbol{r} \le \boldsymbol{r}^{\boldsymbol{U}}$.

The equations of the additional variables can be written as:

$$u = \mu(q+s) - t \quad \rightarrow \quad 0 = -\mu \cdot q - \mu \cdot s + 1 \cdot t + 1 \cdot u + 0 \cdot v \quad \rightarrow$$

$$v = \mu(q+s) + t \quad \rightarrow \quad 0 = -\mu \cdot q - \mu \cdot s - 1 \cdot t + 0 \cdot u + 1 \cdot v \quad \rightarrow$$

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -\mu & -\mu & 1 & 1 & 0 \\ -\mu & -\mu & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} q \\ s \\ t \\ u \\ v \end{bmatrix} \quad \text{or} \quad \boldsymbol{0} = \boldsymbol{B}\,\boldsymbol{r} \quad \text{where} \quad \boldsymbol{B} = \begin{bmatrix} -\mu & -\mu & 1 & 1 & 0 \\ -\mu & -\mu & -1 & 0 & 1 \end{bmatrix}$$

Similar equations hold for each interface. These homogeneous equations may be combined with the previously formulated equations of block equilibrium to give:

$$\begin{bmatrix} \boldsymbol{w}_1 \\ \cdot \\ \cdot \\ \cdot \\ \boldsymbol{w}_N \\ \cdots \\ \boldsymbol{0} \\ \cdot \\ \cdot \\ \cdot \\ \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{p}_1 \\ \cdot \\ \cdot \\ \cdot \\ \boldsymbol{p}_N \\ \cdots \\ \boldsymbol{0} \\ \cdot \\ \cdot \\ \cdot \\ \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{G}_1 & \boldsymbol{H}_1 & & & \boldsymbol{0} & \\ & \boldsymbol{G}_2 & \boldsymbol{H}_2 & & & \\ & & & & & \\ \boldsymbol{0} & & & & & \\ & & & & \boldsymbol{G}_N & \boldsymbol{H}_N \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{B}_1 & & & & & \\ & & & & \boldsymbol{0} & \\ & & & & & \\ & \boldsymbol{0} & & & & \\ & & & & & \boldsymbol{B}_{N+1} \end{bmatrix} \begin{bmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \\ \cdot \\ \cdot \\ \cdot \\ \boldsymbol{r}_{N+1} \end{bmatrix}$$

Note that, since each $\boldsymbol{r}$ now has 5 components, columns of zeros must be added to the $\boldsymbol{G}$ and $\boldsymbol{H}$ matrices in the positions associated with the additional variables u and v.

The above formulated system of linear block equilibrium equations can be written in the same form as previously, $\mathbf{w} + \mathbf{p} = \mathbf{H}\,\mathbf{r}$, provided that the quantities $\mathbf{w}$, $\mathbf{p}$, $\mathbf{H}$ and $\mathbf{r}$ are appropriately re-defined.

The dead-weight and live-load limit analysis problems now have exactly the same algebraic form as before.

2.7 Correction of the deformation mechanism

In the above formulation, the three linearly independent variables q, s and t still define the state of stress at an interface, while the variables u and v are introduced simply for computational convenience.

Due to the introduction of these two additional variables, the final pivotal row of the modified matrix $\mathbf{H}$ may not define the collapse mechanism correctly.

The solution algorithm computes the failure mechanism as a set of deformations which 'correspond' in a work sense to those elements of $\mathbf{r}$ which have attained limiting values, assuming that the deformation mechanism obeys the normality rule. However, this rule is not obeyed by a mechanism involving Coulomb friction, a fact originally pointed out by Drucker (1954).

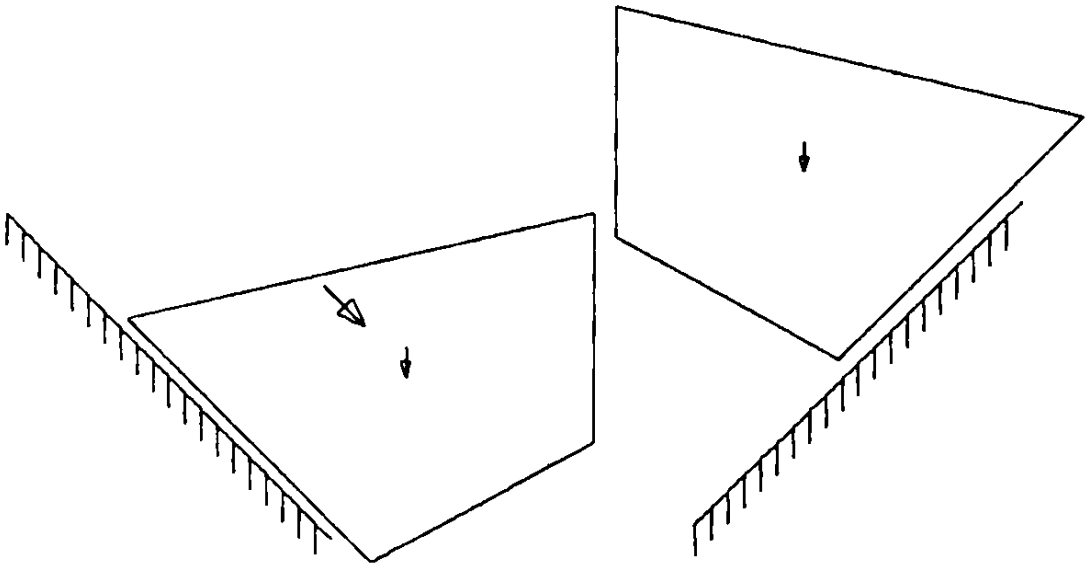


Figure 2-5: *Computer Plot of Incorrect Failure Mechanism.*
Source: Livesley, R. (1978). Limit Analysis of Structures Formed from Rigid Blocks.

If failure occurs by sliding with Coulomb friction, the stress-resultant which reaches its limiting value is either u or v, and the computed non-zero deformation variable is the 'corresponding' ε_u or ε_v. This has displacement components both normal and tangential to the interface, and thus produces an incorrect failure mechanism, since a sliding failure mechanism has only one displacement component tangential to the interface.

The incorrect failure mechanism can be corrected by replacing the Coulomb friction constraint at this interface by a simple constraint $-t_k^L \le t_k$ or $t_k \le t_k^U$, where $-t_k^L$ or t_k^U is the numerical value of t_k which has just been computed.

Making this constraint 'active' in place of the constraint on u_k or v_k involves a Gauss-Jordan transformation of the coefficient matrix **H** which sets the element of the pivotal row of **H** corresponding to t_k to the correct value of the deformation variable $(\varepsilon_t)_k$.

In linear programming terms this transformation involves converting t_k from a 'basic' to a 'non-basic' variable.

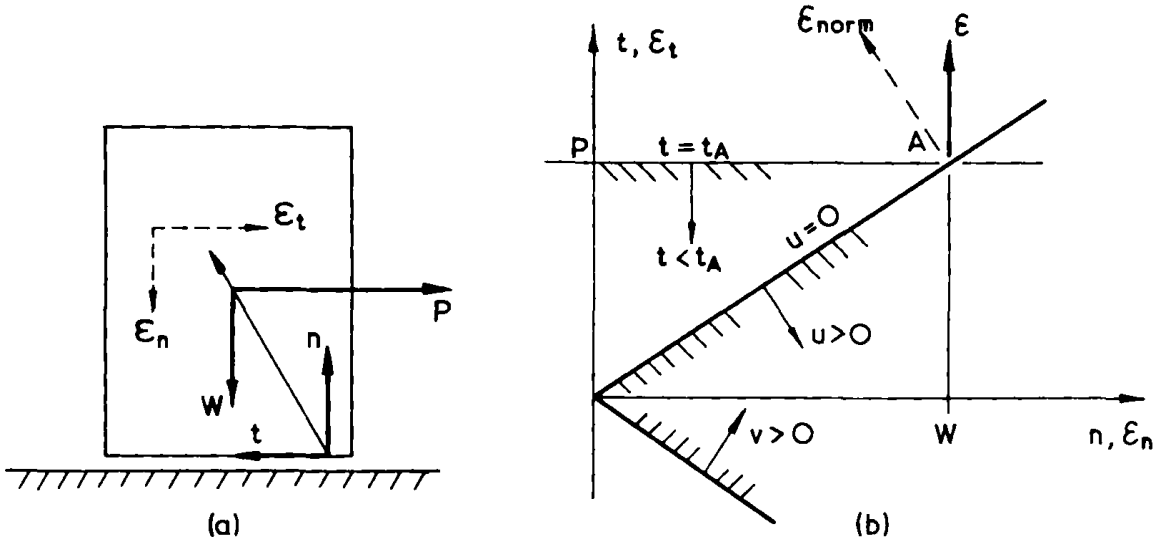


Figure 2-6: *Correct Failure Mechanism* ε *and Incorrect Mechanism* ε_{norm}.
Source: Livesley, R. (1978). Limit Analysis of Structures Formed from Rigid Blocks.

2.8 The validity of limit analysis in the presence of Coulomb friction

Using the procedure described in the previous section, the corrected failure mechanism corresponding to the computed failure load can be easily obtained.

However, from various counter-examples, it is evident that sometimes the maximizing algorithm gives an over-estimate of the true failure load, although, from a purely algebraic perspective, the computational procedure seems to be correct.

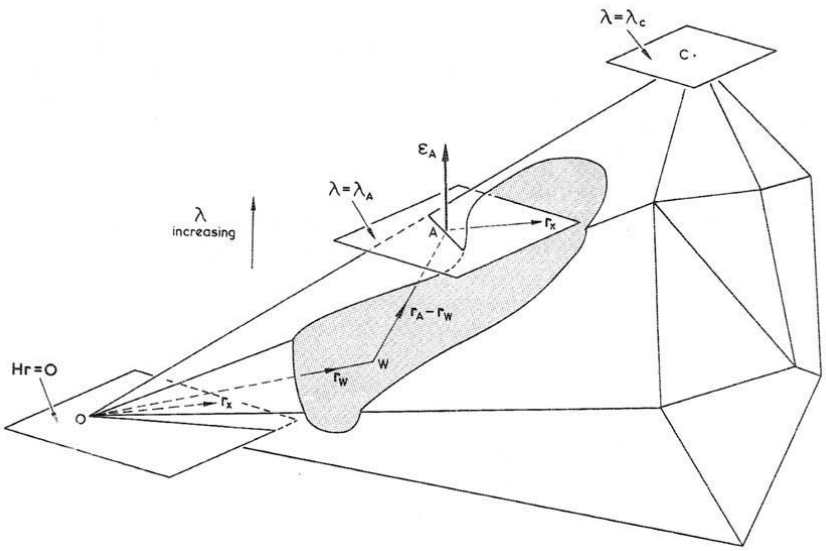


Figure 2-7: *The Yield Surface for an Assembly of Blocks with Coulomb Friction.*
Source: Livesley, R. (1978). Limit Analysis of Structures Formed from Rigid Blocks.

When the limit on the shear force at a block interface is that associated with Coulomb friction, the failure mechanism computed by the algorithm will not necessarily satisfy the normality rule.

Sometimes, a failure mechanism, defined by a set of compatible deformations, exists at some $\lambda_A < \lambda_C$, thus the computed failure load λ_C will not, in general, be attained. Since the yield surface is convex, such a failure would be impossible if the normality rule was obeyed.

In the case where a failure occurs while the normality rule is not obeyed, the computational algorithm may continue the maximization process if a self-equilibrating $\mathbf{r}_x$ which satisfies all the active yield constraints can be found.

In linear programming terms this means that although the yield surface has been reached, the maximizing algorithm will continue the maximization process if a way to re-enter inside the feasible region can be found.

When a new yield constraint is encountered, further progress will depend on the existence of a self-equilibrating $\mathbf{r}_x$ which satisfies this new constraint, and the process will terminate when no $\mathbf{r}_x$ can be found.

Note that the existence of a self-equilibrating $\mathbf{r}_x$ which does not violate the yield conditions is not, of itself, sufficient to ensure failure at a load lower than that obtained by limit analysis. It is the fact that $\mathbf{r}_x$ does positive work in the computed failure mechanism which is the essential feature.

2.9 Conclusion

The results deduced in the previous section show that limit analysis fails to give a ‘correct’ answer only in those friction problems where the failure load is affected by boundary movements.

In these cases, the maximizing algorithm assumes that the boundaries do positive work, implying some movement of supposedly fixed surfaces.

A lower bound algorithm will over-estimate the failure load in such cases, but the existence of the over-estimate can be detected immediately by inspecting the corrected failure mechanism for positive boundary work. The computed failure state is a ‘passive Rankine state’ in soil mechanics terminology.

The information provided by the corrected failure mechanism indicates whether the boundaries are assumed to do positive work or not.

Thus, a criterion for testing the validity of a failure load computed in these circumstances can be established.

The correctness of the limit analysis approach will be indicated by the fact that in the computed mechanism the external boundaries do not move inward, doing positive work.

The result may be stated as follows: ‘If the maximizing algorithm produces a corrected failure mechanism ε_C in which the boundaries do positive work then a failure mechanism exists at some $\lambda_A < \lambda_C$ and λ_C will not, in general, be attained.’

The converse is: ‘If the maximizing algorithm produces a corrected failure mechanism ε_C in which the boundaries do no work then the load factor λ_C is correct, even if the mechanism does not satisfy the normality rule.’

These two statements separate problems involving Coulomb friction into the two classes mentioned at the start of this section.

2.10 References

1. *Livesley, R. (1978). Limit Analysis of Structures Formed from Rigid Blocks. International Journal for Numerical Methods in Engineering 12(12):1853-1871 · January 1978.*

3. ‘Arch Stability’ MATLAB Program

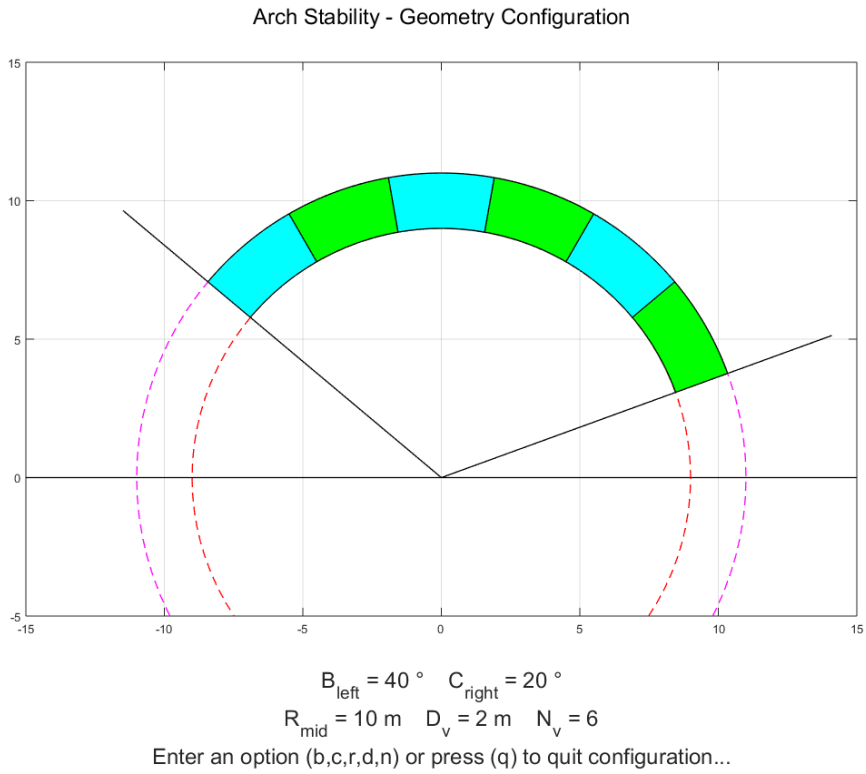


Figure 3-1: *‘Arch Stability’ MATLAB Program.*
Source: MATLAB Program Developed for the Thesis.

3.1 Synopsis

A MATLAB program has been developed, that determines the limit load of any circular arch, constructed with constant width voussoirs.

The ‘Arch Stability’ MATLAB program consists of various scripts, each one of them having a discrete functionality.

The scripts can be used separately, requiring manual interaction, but they can also be invoked from other scripts, providing their functionality without the need of manual interaction.

Each script includes all relative local functions at the end of the file, a syntax supported in MATLAB R2016b or later. Local functions are the most common way to break up programmatic tasks. This approach can be used as the base of code parameterization, facilitating automatic interaction between independent scripts.

In the following sections, the basic scripts of the ‘Arch Stability’ MATLAB program and their respective functionality are being described, complemented by figures depicting the graphical user interface of the program.

3.2 ‘Geometry Configuration’ MATLAB script

A MATLAB script has been developed, facilitating the automatic configuration of the geometry, based on the provided values of the following design variables:

$B(°)$ = Left Base Angle

$C(°)$ = Right Base Angle

$R(m)$ = Arch Middle Radius

$D(m)$ = Voussoir Thickness

N = Number of Voussoirs

The arch angle can be different than straight, being adjusted through the left and right base angle design variables.

Since the provided design variables are the middle radius and the voussoir thickness, the intrados radius can be calculated as $R_{\text{int}} = R_{mid} - D/2$ and the extrados radius as $R_{ext} = R_{mid} + D/2$.

The voussoir width is assumed to be constant throughout the arch, thus all voussoirs are identical.

The geometry is calculated for two voussoir types, quadrilateral and curved, the inner and outer sides being straight line segments or circular arcs respectively, while the interfaces between voussoirs being always straight line segments.

While the coordinates of the four corner points remain the same for both voussoir types, the coordinates of the centre of gravity are different, so the equilibrium equations formulation is slightly different for each type.

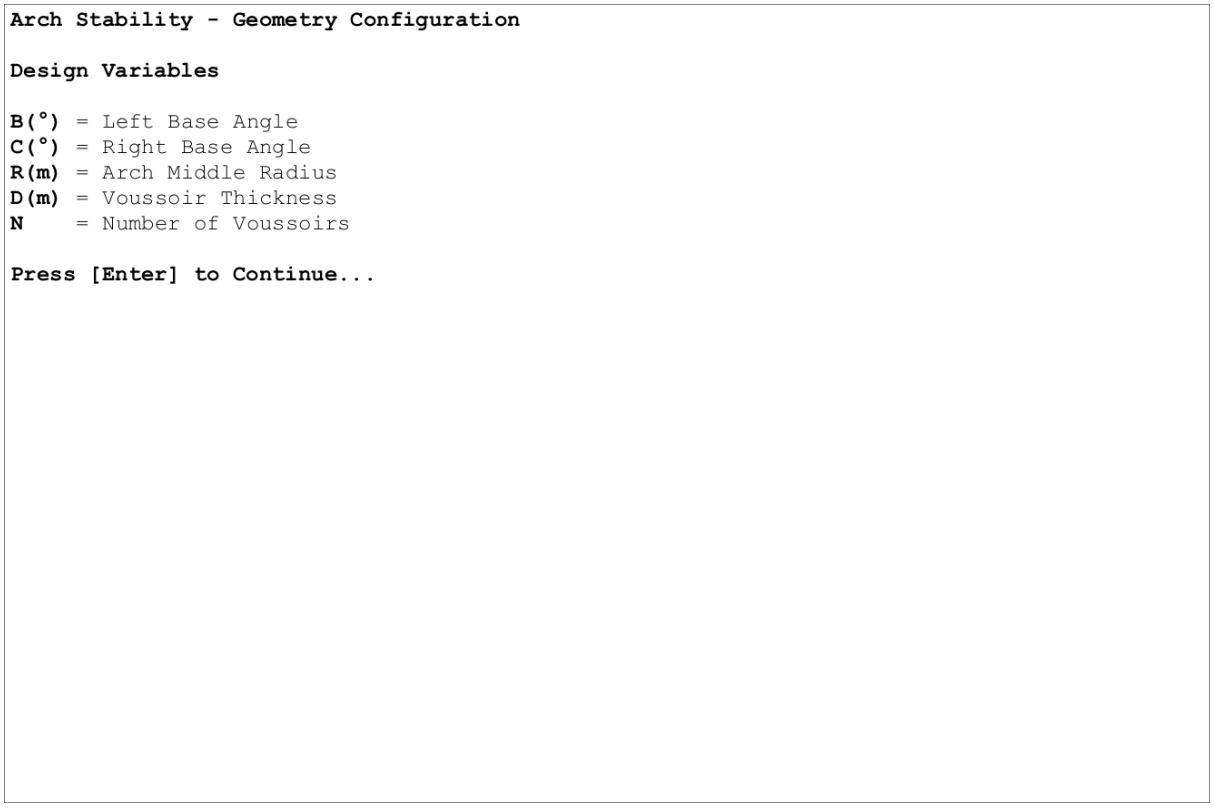


Figure 3-2: *'Arch Stability' MATLAB Program - Initial Screen.*
Source: MATLAB Program Developed for the Thesis.

The initial screen of the ‘Arch Stability’ MATLAB program is depicted in Figure 3-2. In this screen, the design variables are stated, along with their respective description.

The graphical user interface facilitates the configuration of the geometry, visually representing the effect of the design variable values over the total geometry.

By manually assigning the desired values to the design variables, the basic geometry is being presented to the user, as depicted in Figure 3-3.

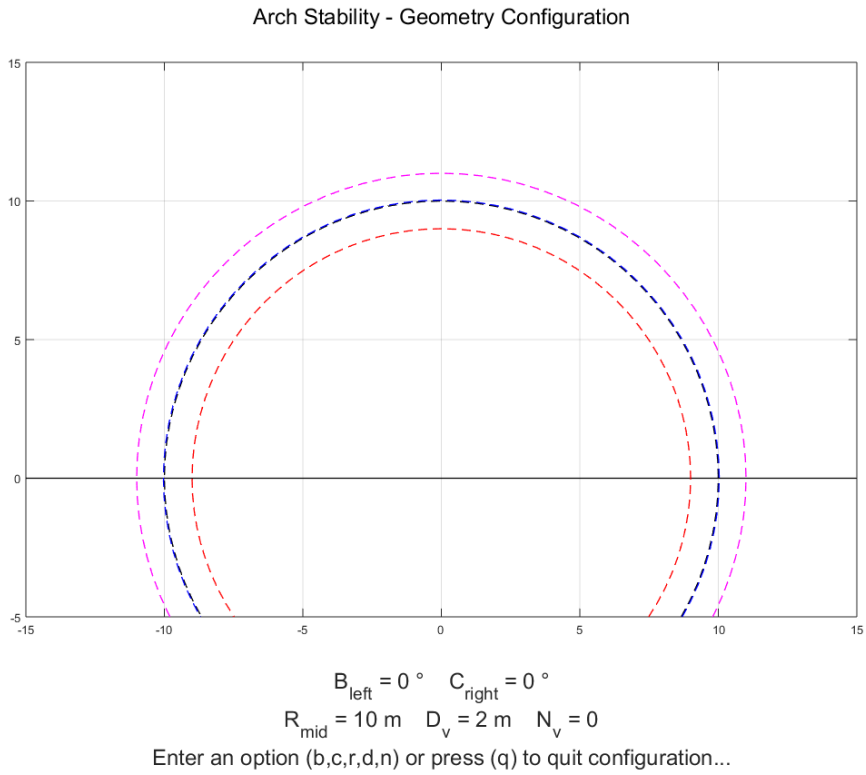


Figure 3-3: *‘Arch Stability’ MATLAB Program - Basic Geometry.*
Source: MATLAB Program Developed for the Thesis.

In the script, there are also included some drawing related functions which are disabled by default, being easily enabled by manually editing the script code and removing the relative percent symbol.

These extra functions can be used to perform a visual verification of the intermediate calculations, based on which the final geometry is being calculated, as depicted in Figure 3-4 and Figure 3-5.

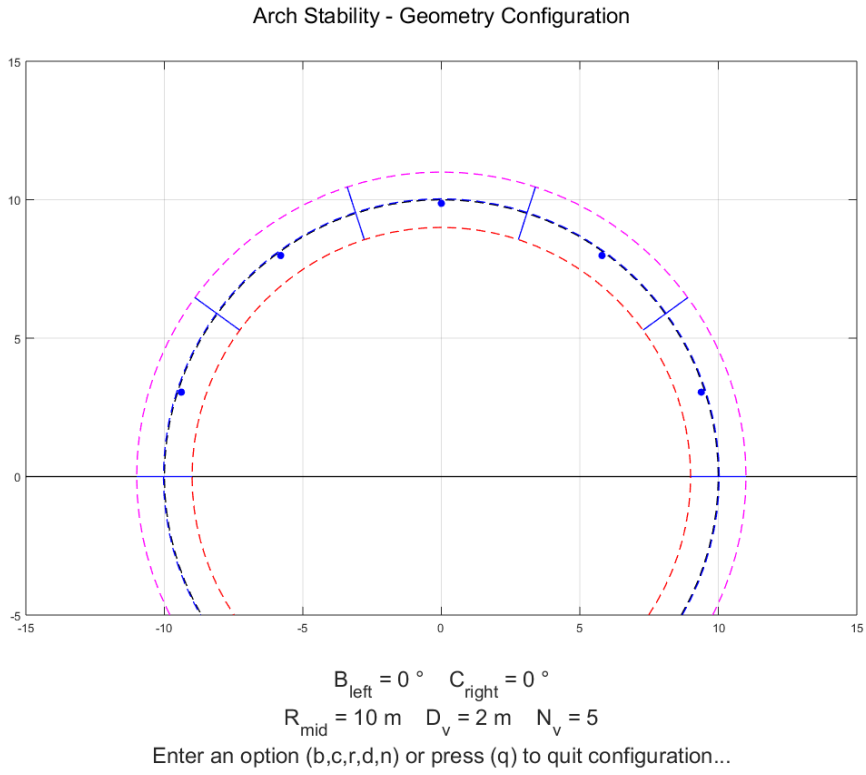


Figure 3-4: *'Arch Stability' MATLAB Program - Auxiliary Geometry.*
Source: MATLAB Program Developed for the Thesis.

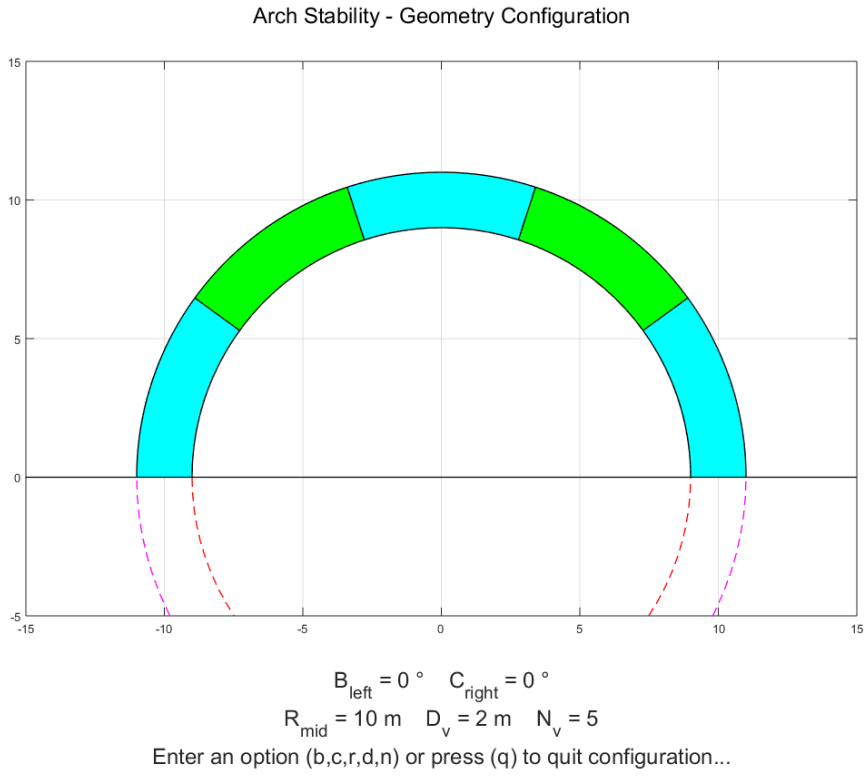


Figure 3-5: *'Arch Stability' MATLAB Program - Final Geometry.*
Source: MATLAB Program Developed for the Thesis.

3.3 ‘Block Equilibrium Equations’ MATLAB script

A MATLAB script has been developed, for the purpose of the formulation of the block equilibrium equations, for either the quadrilateral or the curved voussoir type.

The script facilitates the construction of the matrices $\boldsymbol{G}$ and $\boldsymbol{H}$ for each block, corresponding to the coefficients of the three scalar equilibrium equations (force equilibrium in the x and y directions and moment equilibrium about the centre of gravity G of the block), as well as the composition of the combined matrix $\mathbf{H}$ for all the blocks.

For the quadrilateral voussoir type, depicted in Figure 3-6, the formulation adopted for the derivation of the block equilibrium equations is the one proposed by Livesley.

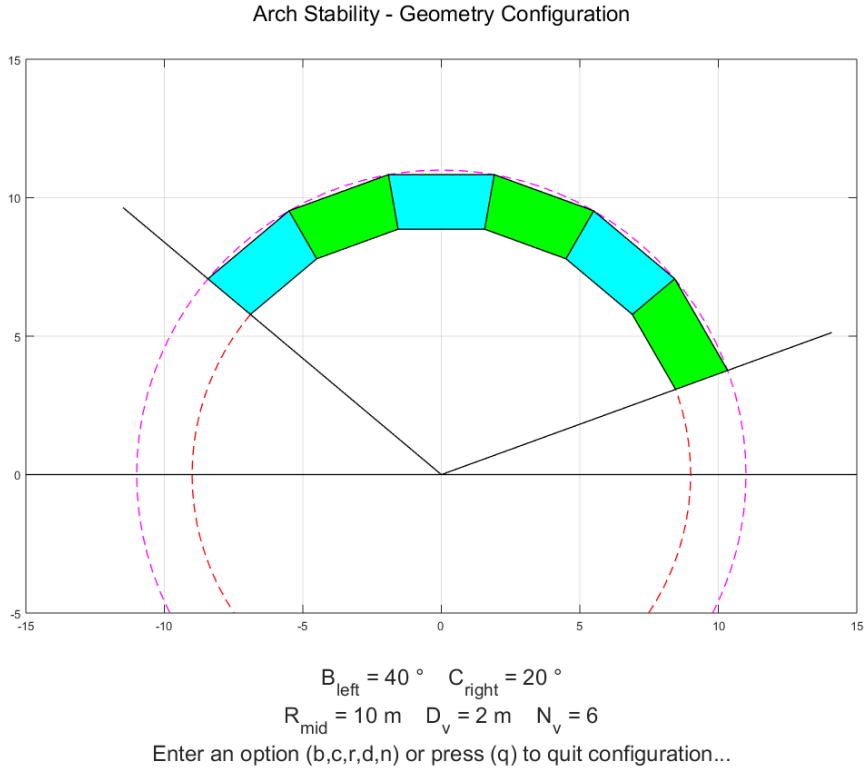


Figure 3-6: *‘Arch Stability’ MATLAB Program - Quadrilateral Voussoir Type.*
Source: MATLAB Program Developed for the Thesis.

For the curved voussoir type, depicted in Figure 3-7, the formulation adopted for the derivation of the block equilibrium equations is slightly different than the one proposed by Livesley, since the coordinates of the centre of gravity are different.

Furthermore, a different formulation based on the circular geometry has been tested, for both voussoir types, where the moment equilibrium equation for each voussoir is formulated by considering the moments of the forces about the centre of the circle. As it was expected, although this alternative formulation uses different expressions, it leads to the same final results as previously.

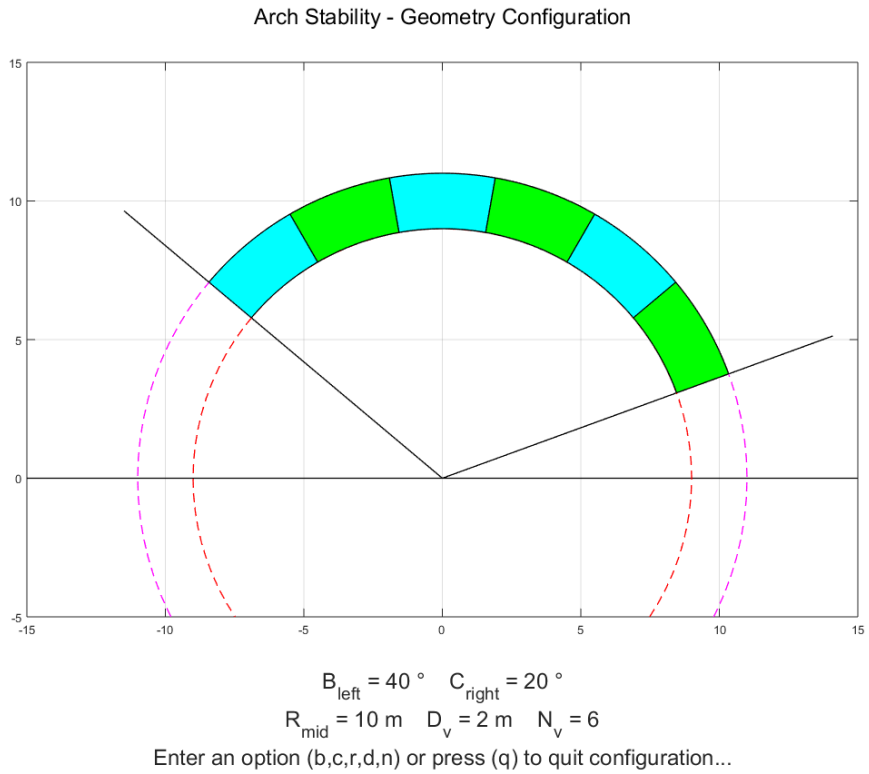


Figure 3-7: *'Arch Stability' MATLAB Program - Curved Voussoir Type.*
Source: MATLAB Program Developed for the Thesis.

By using the Livesley proposed block equilibrium equations MATLAB script, any non-circular arch shape with quadrilateral voussoirs can also be examined, as long as the coordinates of the four corners of the blocks are being provided.

3.4 'Block Dead Weight' MATLAB script

A MATLAB script has been developed, for the purpose of the initialization of the block material specific weight design variable *gamma*.

The area of each block is multiplied by the block material specific weight, providing the block dead weight W, which can be used to formulate the vector $\boldsymbol{w}$ for each block, and the vector $\mathbf{w}$ for all the blocks.

3.5 'Block Live Loads' MATLAB script

A MATLAB script has been developed, for the purpose of the initialization of the block live loads design variables p_x, p_y, m.

These design variables correspond to the block live loads applied at given co-ordinates x_i, y_i, and they can be combined to the vector $\boldsymbol{p}$ for each block, and the vector $\mathbf{p}$ for all the blocks.

3.6 'Block Interface Stress Resultants Limits' MATLAB script

A MATLAB script has been developed, for the purpose of the initialization of the block interface stress resultants limits design variables q^L, q^U, s^L, s^U, t^L, t^U.

These design variables correspond to the lower and upper limits of the stress resultants q, s, t, which make up the vector $\boldsymbol{r}$ for each element, and they can be combined to the vectors $\boldsymbol{r}^L$, $\boldsymbol{r}^U$ for each element, and the vectors $\mathbf{r}^L$, $\mathbf{r}^U$ for all the elements.

3.7 'Coulomb Friction' MATLAB script

A MATLAB script has been developed, for the purpose of the application of yield constraints involving Coulomb friction.

If the condition at a block interface is one of simple Coulomb friction with no cohesion, the constraints on the three interface stress resultants can be put in the required simple form by the introduction of two additional variables.

The three linearly independent variables q, s and t still define the state of stress at an interface, while the additional variables u and v are introduced simply for computational convenience.

The previously expressed vectors $\mathbf{w}$ and $\mathbf{p}$, as well as the matrix $\mathbf{H}$, have to be expanded appropriately to accommodate for the two additional variables.

Also, the block interface stress resultants lower and upper limits vectors $\mathbf{r}^{\mathbf{L}}$, $\mathbf{r}^{\mathbf{U}}$ have to be appropriately expressed, since the simplified formulation requires that the variable t is being unconstrained, while the variables u and v are being positive.

3.8 'Dead-Load Problem' MATLAB script

A MATLAB script has been developed, for the purpose of the definition and the solution of the dead-load problem.

The dead-load problem is equivalent to the maximization of a parameter λ_w, where $\lambda_w \mathbf{w} = \mathbf{H}\,\mathbf{r}$ and $-\mathbf{r}^{\mathbf{L}} \le \mathbf{r} \le \mathbf{r}^{\mathbf{U}}$, the process being terminated as soon as a value $\lambda_w = 1$ is reached.

If the maximum value of λ_w is less than unity then clearly the arch cannot support its own weight.

The MATLAB linear programming solver *linprog* is being used to determine the maximum value of the dead-load factor λ_w, through the minimization of the parameter $-\lambda_w$.

The Lagrange multipliers can be obtained as an output of the solver, indicating the possible kinematic collapse mechanism that may be formed.

3.9 'Live-Load Problem' MATLAB script

A MATLAB script has been developed, for the purpose of the definition and the solution of the live-load problem.

If a valid solution to the dead-load problem exists and $\mathbf{r}_w$ is the value of $\mathbf{r}$ associated with this solution, the live-load problem can be written in the form 'Maximize λ subject to the equilibrium equations $\lambda \mathbf{p} = \mathbf{H}(\mathbf{r} - \mathbf{r}_w)$ and the yield constraints $-\mathbf{r}^L - \mathbf{r}_w \le \mathbf{r} - \mathbf{r}_w \le \mathbf{r}^U - \mathbf{r}_w$ '. This is essentially the same problem as before in the 'shifted' variables. $\mathbf{r} - \mathbf{r}_w$.

The MATLAB linear programming solver *linprog* is being used to determine the maximum value of the live-load factor λ, through the minimization of the parameter $-\lambda$.

The Lagrange multipliers can be obtained as an output of the solver, indicating the possible kinematic collapse mechanism that may be formed.

3.10 'Display Final Results' MATLAB script

A MATLAB script has been developed, for the purpose of the presentation of the final results in the Command Window and the graphical representation of the thrust line and the active constraints over the arch geometry.

The final results presented in the Command Window include the names of the active constraints, which are indicated by the presence of the respective non-zero Lagrange multipliers, as well as the maximum value of the load factor λ.

The thrust line and the active constraints are also graphically represented over the arch geometry in a figure window, providing a visual representation of the collapse mechanism.

3.11 Results obtained with ‘Arch Stability’ MATLAB program

A numerical confirmation of the minimum thickness of a semicircular arch under its own weight, as calculated by the Serbian scholar Milutin Milankovitch (1904), has been achieved using the ‘Arch Stability’ MATLAB program.

According to Milankovitch’s theory of the thrust line, the minimum thickness of a semicircular arch under its own weight is $d_{min} = 0.1075 \cdot R$, while the thrust line touches the intrados at the point corresponding to the rupture angle $54°29'$.

The ‘Dead-Load Problem’ MATLAB script was used, with a geometry of a semicircular arch of middle radius $R = 10m$.

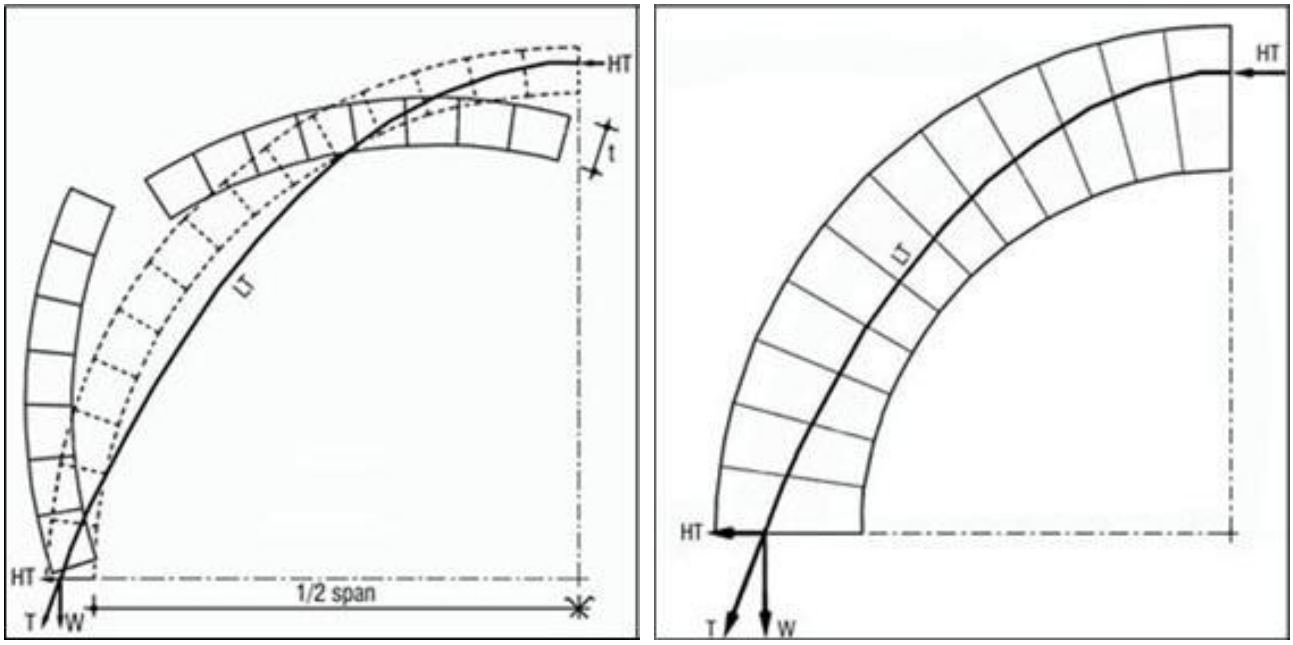


Figure 3-8: *Arch Instability and Stability.*
Source: Auroville Earth Institute (AVEI).

It was expected that for arch thickness values of $d > 0.1075 \cdot R$ the program would return a maximum dead-load factor equal to one, while for arch thickness values of $d < 0.1075 \cdot R$ the program would return a maximum dead-load factor equal to zero, since an arch having a thickness less than the minimum required to contain the thrust line would collapse under its own weight, forming a compatible kinematic collapse mechanism.

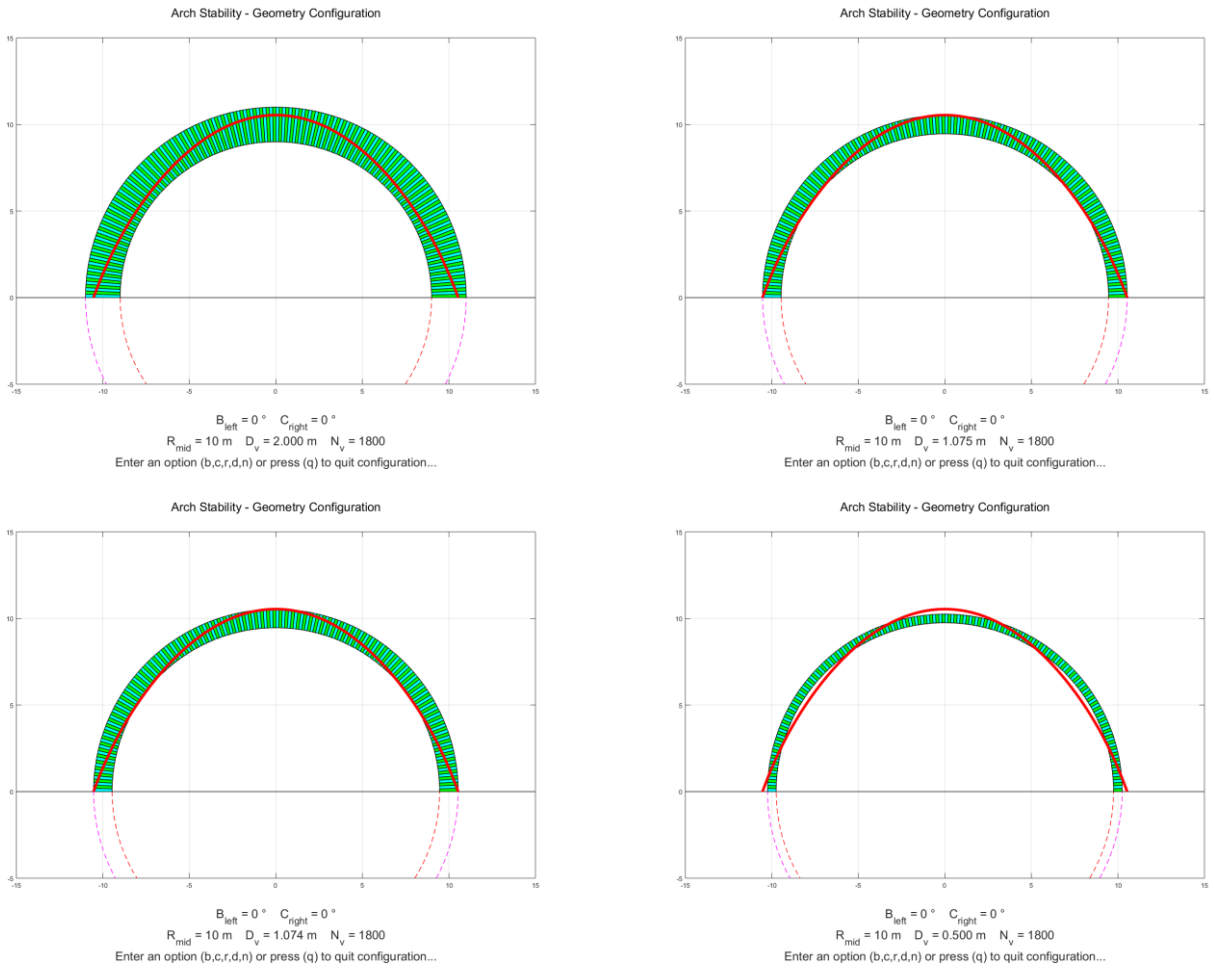


Figure 3-9: *'Arch Stability' MATLAB Program - Arch Thickness and Thrust Line.*
Source: MATLAB Program Developed for the Thesis.

The values of $d = 0.2000 \cdot R$, $d = 0.1075 \cdot R$, $d = 0.1074 \cdot R$ and $d = 0.0500 \cdot R$, depicted in Figure 3-9, were selected to evaluate the validity of the results of the program.

The assumed number of voussoirs has a significant effect on the output of the program. In order to emulate the equilibrium and kinematic conditions of an infinitesimal voussoir, a large number of voussoirs was assumed, equal to 1800, the corresponding angle being equal to 0.1 degrees.

In the first two cases, where the assumed thickness is larger than the minimum required, the program returned a maximum dead-load factor equal to one, as expected.

In the last two cases, where the assumed thickness is smaller than the minimum required, the program returned a maximum dead-load factor equal to zero, as expected.

Using a trial and error procedure, it proved that, according to the program, the values of $d = 0.10747 \cdot R$ and $d = 0.10748 \cdot R$ were across the limit value that triggered the arch collapse.

The value of $d = 0.10748 \cdot R$ is indeed documented as the exact value of the minimum required thickness of a semicircular arch under its own weight, along with a rupture angle of $54.484°$.

For the value of $d = 0.10748 \cdot R$, no active constraints were reported by the program, since the assumed arch thickness is marginally larger than the minimum required to contain the thrust line, thus the stress resultant distribution is statically admissible, being on the safe side, since no Coulomb friction has been assumed for this particular problem.

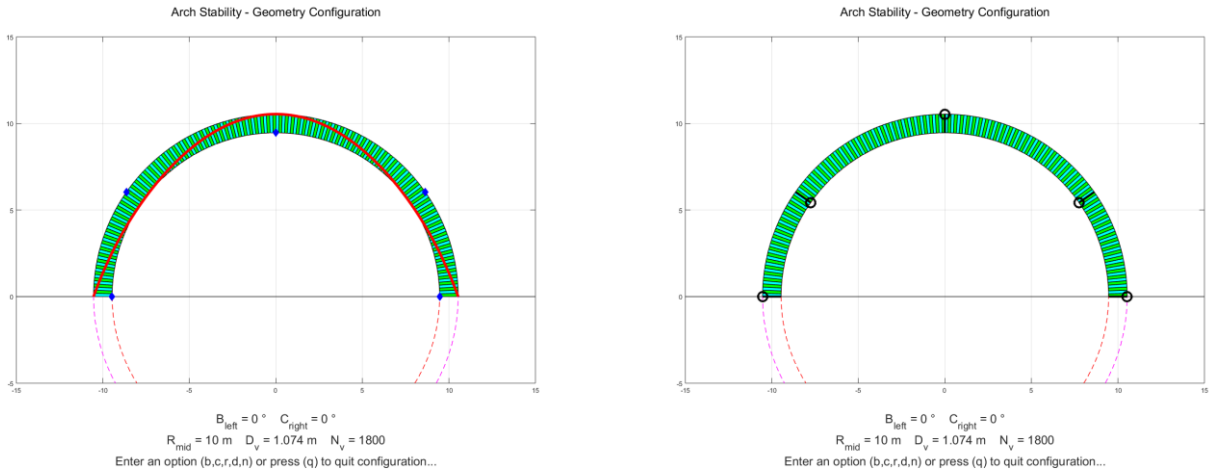


Figure 3-10: *'Arch Stability' MATLAB Program - Active Constraints and Hinges.*
Source: MATLAB Program Developed for the Thesis.

For the value of $d = 0.10747 \cdot R$, the following lower bound active constraints were reported by the program, indicating the corresponding kinematic collapse mechanism:

Element	1	-	Resultant	s	-	Hinge formed at angle	0.0°	-	Extrados
Element	356	-	Resultant	q	-	Hinge formed at angle	35.5°	-	Intrados
Element	901	-	Resultant	s	-	Hinge formed at angle	90.0°	-	Extrados
Element	1446	-	Resultant	q	-	Hinge formed at angle	144.5°	-	Intrados
Element	1801	-	Resultant	s	-	Hinge formed at angle	180.0°	-	Extrados

The reported active constraints are lower bound, meaning that, according to the already assumed stress resultant limits, the corresponding stress resultant has a value of zero, thus the compressive connection is being lost at this particular point. The block interface tends to open at this side, thus a hinge is being formed at the opposite side of the block interface, allowing rigid body rotation.

The corresponding rupture angles, which are being measured from the vertical direction, are $|90.0° - 35.5°| = 54.5°$ and $|90.0° - 144.5°| = 54.5°$.

The minimum required thickness and the corresponding kinematic collapse mechanism indicated by the 'Arch Stability' MATLAB program match exactly the values documented in Milankovitch's theory of the thrust line.

It has to be noted, that, as it was mentioned above, the assumed number of voussoirs has a significant effect on the output of the program. As the assumed number of voussoirs is getting lower, the indicated limit value of the required thickness is affected downwards.

Milankovitch's theory of the thrust line makes an assumption of an infinitesimal voussoir width and, as stated previously, concludes that the thrust line touches the intrados at the point corresponding to the rupture angle $54°29'$.

When the thickness of the arch is assumed to be less than the minimum value, a compatible kinematic mechanism is being formed, and the arch collapses under its own weight.

When the value for the number of voussoirs is assumed to be one, there is only one rigid body block in the arch, so no kinematic mechanism can be formed.

The same holds true also for the values of two and three, where, due to the symmetry of the arch and the dead weight loading, no compatible kinematic mechanism can be formed, thus the arch cannot collapse under its own weight, even when the arch thickness tends to zero.

For larger assumed values, which are not large enough to emulate the equilibrium and kinematic conditions of an infinitesimal voussoir, the arch is considered as being formed from finite rigid body blocks, and the block interfaces are being placed apart by considerable angles.

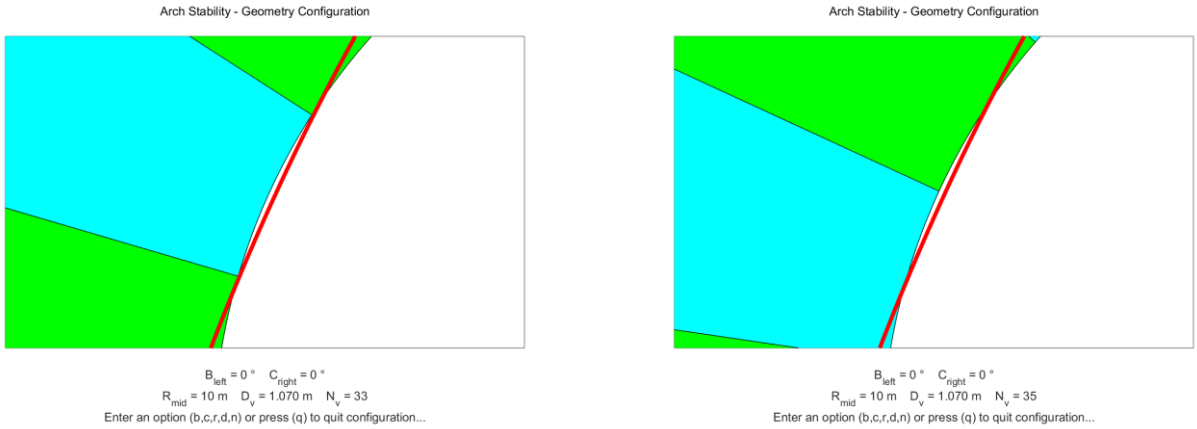


Figure 3-11: *'Arch Stability' MATLAB Program - Assumed Number of Voussoirs Effect.*
Source: MATLAB Program Developed for the Thesis.

As the number of voussoirs becomes lower, the voussoir width becomes larger and the block interfaces are removed from each other, thus there is a possibility that at the point corresponding to the rupture angle 54°29', where the thrust line may marginally pass outside of the body of the arch, there is no block interface, but a part of a rigid body block, as depicted in Figure 3-11.

In that case, there is no problem if the thrust line marginally passes outside of the rigid body block at the region around the rupture angle, as long as it enters the body again before the next block interface, since no hinge can form in the middle of a rigid body block, but only at the interfaces between the blocks.

This is why the assumed value for the number of voussoirs affects the indicated limit value of the required thickness, especially when the number of voussoirs is low.

As the number of voussoirs becomes larger, the voussoir width becomes smaller and the block interfaces are getting closer to each other, resembling the infinitesimal case, thus the indicated limit value of the required thickness matches the theoretically calculated one.

It is obvious that the above observation holds only for the emulation of the equilibrium and kinematic conditions of an infinitesimal voussoir. For all other cases investigated, where the arch is actually formed from finite rigid blocks, either curved or quadrilateral, the actual geometry of the arch has to be replicated through the appropriate initialization of the design variable values.

Other dead-load and live-load problems have also been solved using the ‘Arch Stability’ MATLAB program, providing results identical to their original counterparts.

This fact provides a good indication that the programming code is working exactly as intended, as far as the underlying geometry generation, the formulation of the equilibrium equations, and the application of the minimization algorithm is concerned.

3.12 References

1. *Livesley, R. (1978). Limit Analysis of Structures Formed from Rigid Blocks. International Journal for Numerical Methods in Engineering 12(12):1853-1871 · January 1978.*
2. *Heyman, J. (1966). The Stone Skeleton. International Journal of Solids and Structures 2(2):249-279 · April 1966.*
3. *Milankovitch, M. (1904). Beitrag zur Theorie der Druckkurven. Dissertation zur Erlangung der Doktorwürde, K.K. Technische Hochschule, Vienna.*

This page is intentionally left blank

4. Embankment Stability Analysis [Spencer (1967)]

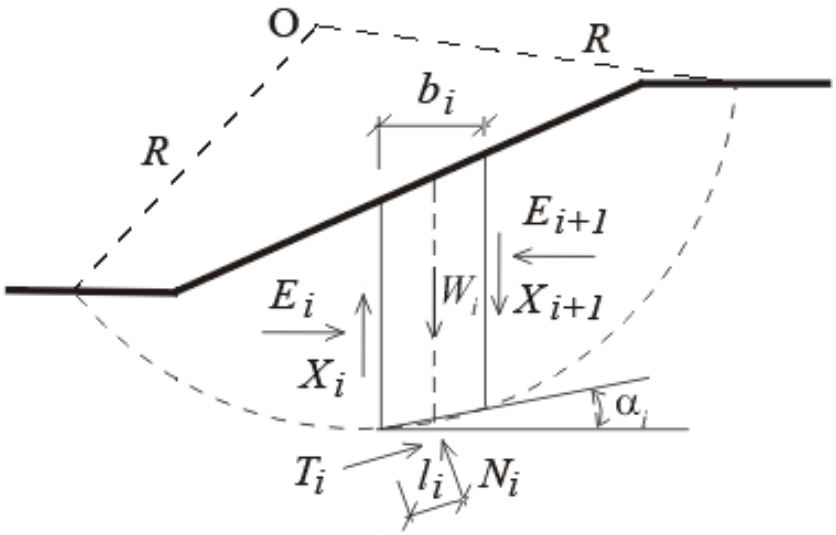


Figure 4-1: *Schematic of the Method of Slices Showing Rotation Centre. Source: Wikimedia Commons.*

4.1 Synopsis

Spencer's method (1967) is a simplified method of analysis of the stability of embankments assuming parallel inter-slice forces.

The method aims at determining the factor of safety of an embankment against failure on a cylindrical slip surface.

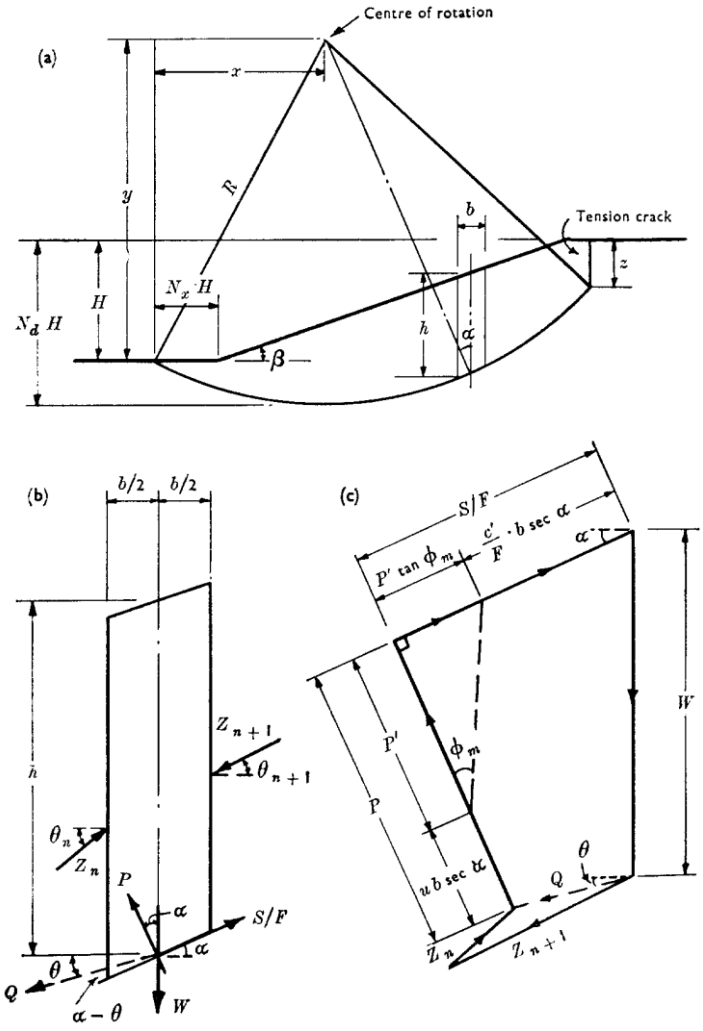


Figure 4-2: *Dimensions of Slip Surface and Forces on a Slice.*
Source: Spencer, E. (1967). A Method of Analysis of the Stability of Embankments.

The analysis is being conducted in terms of effective stress and leads to two equations of equilibrium, the first in respect of forces and the second in respect of moments.

The original purpose of this method was to obtain further information on how Bishop's simplified method (1955) gives accurate results, while it does not satisfy one of the basic conditions of equilibrium.

The next step was to investigate whether the accuracy of Bishop's simplified method is affected either by the height or slope of the embankment or by the soil properties.

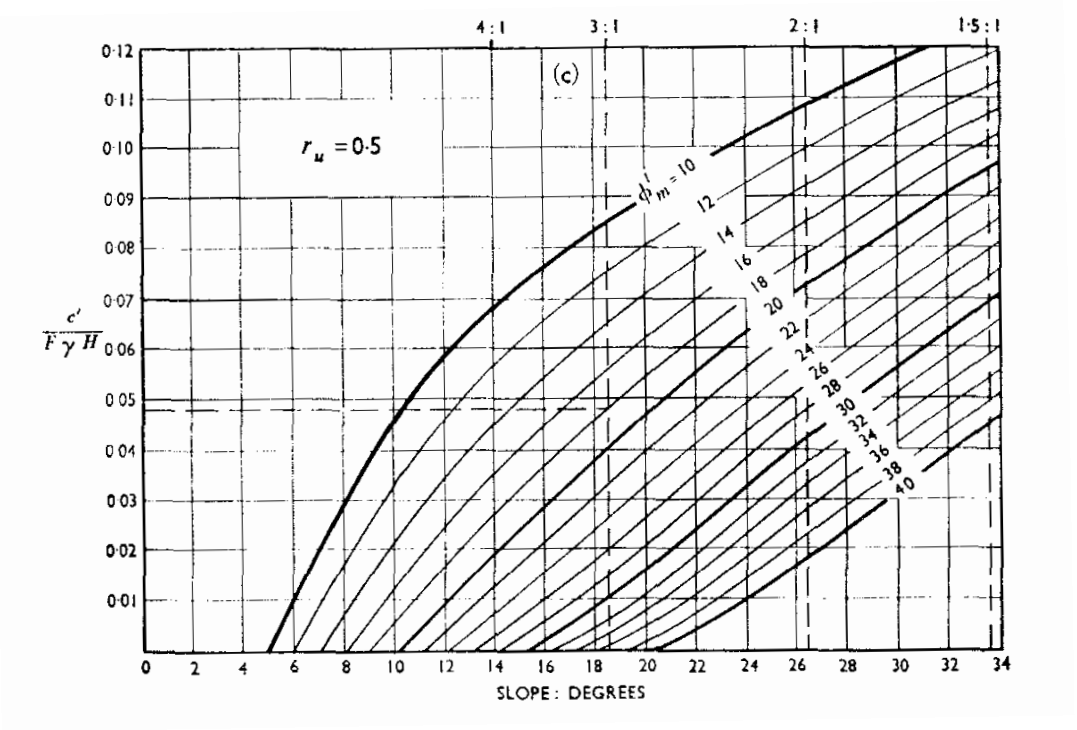


Figure 4-3: *Stability Chart for* $r_u = 0.5$.

Source: Spencer, E. (1967). A Method of Analysis of the Stability of Embankments.

The scope of the investigation was later extended, to obtain charts applicable to analysis in terms of effective stress, which indicate the position of the critical slip circle, and also stability charts similar to those published by Taylor (1948) for total stress analysis.

4.2 Investigation of the accuracy of Bishop's simplified method

In Bishop's paper (1955), a rigorous method of analysis was described which satisfied both force and moment conditions of equilibrium and which took full account of the inter-slice forces.

The rigorous method was extremely lengthy, and it was simplified by assuming that the inter-slice forces were horizontal, satisfying the conditions of equilibrium in respect of moments but not in respect of forces.

Obviously then this expression could not be regarded as rigorous, yet it was found that the value for F given by it was a very close approximation to the final value obtained using the rigorous method.

Spencer's method is more accurate than Bishop's simplified method, taking into full account of the slope θ of the resultant of the pair of the inter-slice forces, along with both force and moment equations.

In fact, if in Spencer's method the value of the slope θ is taken as zero, an equation identical with that of Bishop's simplified method can be obtained.

Notation

θ = assumed value of slope of the resultant of the pair of the inter-slice forces.

θ_1 = precise value of slope of the resultant of the pair of the inter-slice forces.

F_f = value of factor of safety which satisfies force equation, depending on the value of θ.

F_m = value of factor of safety which satisfies moment equation, depending on the value of θ.

F_{f0} = value of factor of safety which satisfies force equation, corresponding to $\theta = 0$.

F_{m0} = value of factor of safety which satisfies moment equation, corresponding to $\theta = 0$.

F_1 = value of factor of safety which satisfies both force and moment equations, corresponding to the precise value of θ.

Using Spencer's method, the effect of the value of the slope θ over the value of factor of safety F in respect of forces and in respect of moments was investigated as follows.

A circular slip surface was chosen arbitrarily, the area inside it divided into vertical strips of equal width and the mean height and base slope of each slice determined graphically.

Several values of θ were chosen and, for each, the values of F_f and F_m were found, which would satisfy the force and the moment equilibrium equations respectively.

A curve was then plotted showing the relationship between F_f and θ, and, on the same graph, a second curve was plotted showing the relationship between F_m and θ.

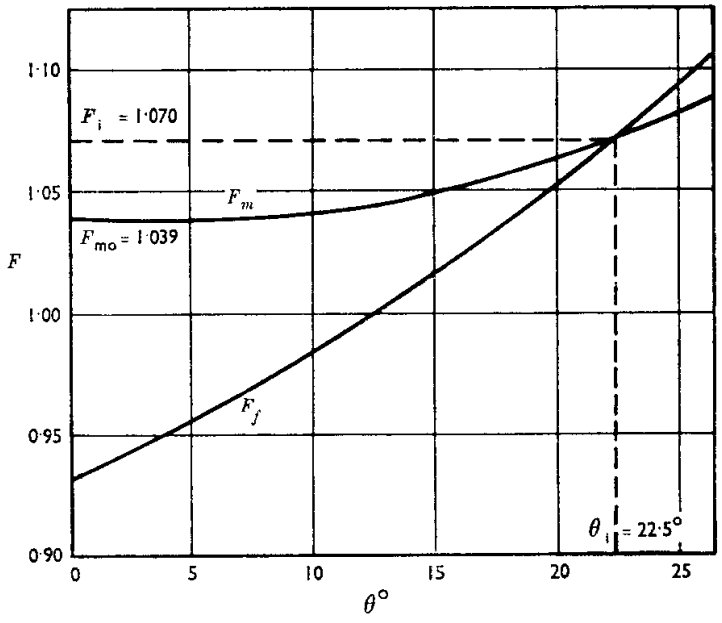


Figure 4-4: *Variation of Fm and Ff with* θ.
Source: Spencer, E. (1967). A Method of Analysis of the Stability of Embankments.

The intersection of the two curves indicated the value of the factor of safety F_1 which would satisfy both equilibrium equations, along with the corresponding slope θ_1 of the inter-slice forces.

Using the values of F_1 and θ_1, the resultants of the inter-slice forces were obtained, from which the values of the inter-slice forces themselves were obtained, along with their points of action.

Several points emerged from this investigation, as follows.

The value of θ_1 was less than the slope of the embankment.

The variation in the value of θ affected the values of F_f to a very much greater extent than those of F_m. In fact, when θ was less than θ_1, the variation in F, was very small indeed. In consequence there was not much difference between the values of F_{m0} and F_1.

The line passing through the points of action of the inter-slice forces passed very close to the lower 'third-point' on each of the inter-slice boundaries. This implied an approximately triangular pressure distribution on these boundaries which was an acceptable result.

Evidently, the basis of the accuracy of Bishop's simplified method lies in the insensitivity of the moment equation to variation in θ when this angle is not greater than θ_1.

4.3 Investigation of the factors that affected the accuracy of Bishop's simplified method

The factors which influence the position of the critical circle in a given case are: the slope of the embankment, the depth to hard stratum, the soil properties ϕ', $c'/\gamma H$ and the pore-pressure, represented in the form of the coefficient r_u.

It has been assumed that the depth to hard stratum was sufficiently great for the position of the critical circle to be determined by the values of the other four factors.

However, the number of factors can be reduced to three since the position of the critical circle is unique for any given value of the parameter $\gamma H \tan\phi'/c'$.

The factors which determine the position of the critical slip circle are therefore: the slope, the pore-pressure coefficient (r_u) and the parameter $\gamma H \tan\phi'/c'$.

With the position of the centre of the critical circle defined, it is therefore a simple matter to devise a series of charts, for different values of r_u, in which the position of the centre of the critical circle can be determined from the values of the slope and $\gamma H \tan\phi'/c'$.

It should be noted that in order to retain the dimensionless form of analysis, the effect of tension cracks at the top of the embankment has been neglected.

4.4 Atlas computer program

The method was programmed for solution on the Atlas computer, installed at Manchester University and officially commissioned in 1962, one of the world's first supercomputers, considered to be the most powerful computer in the world at that time.

Figure 4-5: *The University of Manchester Atlas in January 1963.*
Source: Wikimedia Commons.

The initial program worked in two phases. A slip circle was arbitrarily chosen, and the program calculated the slope θ_1 of the resultant of the pair of the inter-slice forces, and the corresponding value of factor of safety F_1 which satisfies both force and moment equations.

The initial program was used to solve a large number of problems, and the results obtained gave a clear indication of the factors that affected the accuracy of Bishop's simplified method.

In order to identify the position of the critical slip circle, a rectangular grid was set out, one side of the grid coinciding with the perpendicular bisector.

The initial program was modified so that it would deal with more than one slip surface at a time and the coordinates of each intersection were calculated by hand and taken as centres of slip circles.

The program was later modified again so that, when the coordinates of the lower corner of the grid were stated together with the lengths of the sides of the grid, the coordinates of the intersections were calculated by the computer.

The results obtained from this modified program showed that the critical circles for F_{m0} and F_1 were invariably so close as to be almost coincident.

Though some useful information was obtained, it was decided that the program should be modified so as to define the position of the critical circle with greater precision.

The final result was a new pair of programs.

Using the first program, which was of the minimization type, the position of the critical circle for F_{m0} was identified.

Then, using the second program, the value of F_1 was found for this slip circle and also for others whose centres formed a grid pattern around its centre.

These programs have been used to obtain a set of charts giving the position of the critical circles and also a set of stability charts.

4.5 Alternative formulation of the equilibrium equations

In a given problem, there are three equations to be solved: two in respect of forces and one in respect of moments.

If it can be assumed that the inter-slice forces are parallel (i.e. that θ is constant throughout), the two force equilibrium equations become identical.

In this case, there are only two equations to solve, one in respect of forces and one in respect of moments, and the solution is therefore greatly simplified.

In the original method of analysis, the slip surface is assumed to be circular. Also, the effect of tension cracks at the top of the embankment is being neglected.

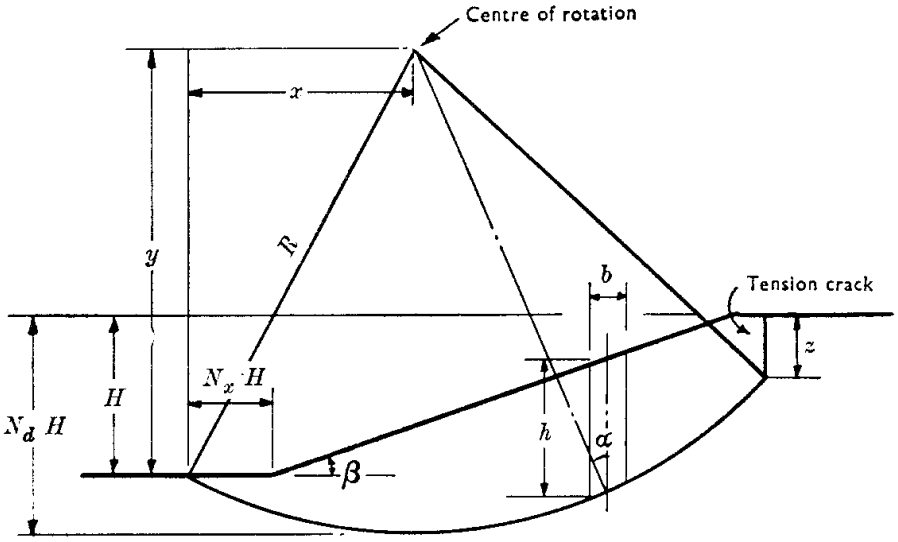


Figure 4-6: *Cross-Section Through a Slip Surface of Constant Curvature.*
Source: Spencer, E. (1967). A Method of Analysis of the Stability of Embankments.

The moment equilibrium equation for each slice is formulated by considering the moments of the forces about the centre of rotation (i.e. the centre of the slip circle).

Then, working again from the first slice to the last, the points of action of the inter-slice forces are found by taking moments about the middle of the base of each slice in turn.

The original method of analysis was later revised by the author, to take into account of slip surfaces of varying curvature, as well as of the effect of tension cracks.

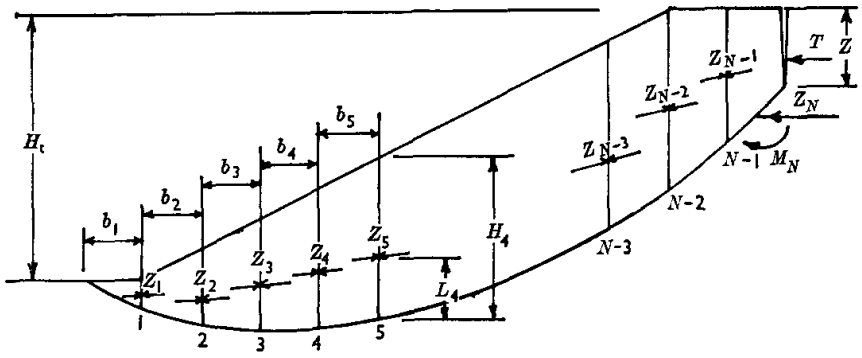


Figure 4-7: *Cross-Section Through a Slip Surface of Varying Curvature.*
Source: Spencer, E. (1973). Thrust Line Criterion in Embankment Stability Analysis.

Since the revised method is applicable to any shape of slip surface, with or without a tension crack, there is no centre of rotation, thus the moment equilibrium equation for each slice is formulated by considering the moments of the forces about the middle of the base of each slice.

Expressions are also being given from which the positions of the lines of thrust for total and effective stress can be obtained.

4.6 References

1. *Spencer, E. (1967). A Method of Analysis of the Stability of Embankments Assuming Parallel Inter-Slice Forces. Géotechnique Volume 17 Issue 1, March 1967, pp. 11-26.*
2. *Spencer, E. (1973). Thrust Line Criterion in Embankment Stability Analysis. Géotechnique Volume 23 Issue 1, March 1973, pp. 85-100.*

5. ‘Slope Stability’ MATLAB Program

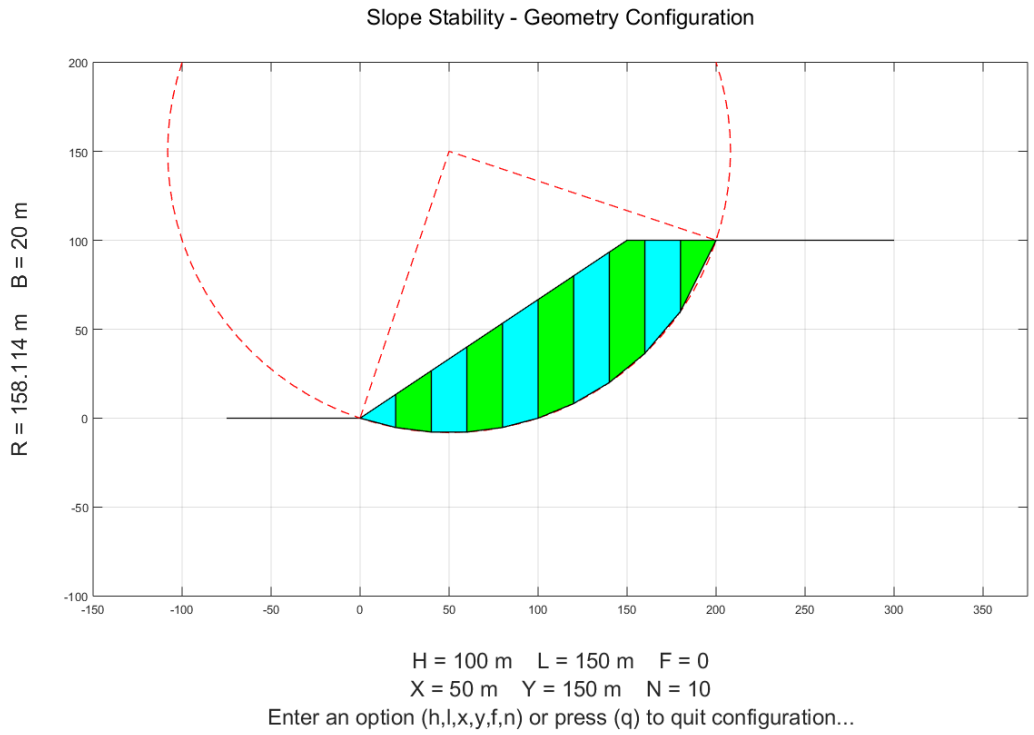


Figure 5-1: *‘Slope Stability’ MATLAB Program.*
Source: MATLAB Program Developed for the Thesis.

5.1 Synopsis

A MATLAB program has been developed, that determines the factor of safety of an embankment against failure on a cylindrical slip surface, using Spencer's method.

The 'Slope Stability' MATLAB program consists of various scripts, each one of them having a discrete functionality.

The scripts can be used separately, requiring manual interaction, but they can also be invoked from other scripts, providing their functionality without the need of manual interaction.

Each script includes all relative local functions at the end of the file, a syntax supported in MATLAB R2016b or later. Local functions are the most common way to break up programmatic tasks. This approach can be used as the base of code parameterization, facilitating automatic interaction between independent scripts.

In the following sections, the basic scripts of the 'Slope Stability' MATLAB program and their respective functionality are being described, complemented by figures depicting the graphical user interface of the program.

5.2 'Geometry Configuration' MATLAB script

A MATLAB script has been developed, facilitating the automatic configuration of the geometry, based on the provided values of the following design variables:

$H(m)$ = Slope Height

$L(m)$ = Slope Length

$X(m)$ = Circle Centre Abscissa

$Y(m)$ = Circle Centre Ordinate

F = Lower End Position Factor

N = Number of Soil Blocks

The critical circle may pass through or beneath the toe of the embankment.

The distance between the toe of the embankment and the lower end of slip surface is expressed as a product of a multiplicative factor F and the height of embankment $H(m)$.

For reasons of simplicity, in the following geometric analysis, the origin $(0,0)$ of the Cartesian coordinate system is assumed to be at the toe of the embankment.

According to the notation of the original method of analysis, the coordinates (x, y) of the centre of the slip circle were measured from the lower end of slip surface.

Thus, for backwards compatibility, the design variables (X, Y) are being considered with respect to the lower end of slip surface, while the private variables (x_c, y_c) are being considered with respect to the toe of the embankment.

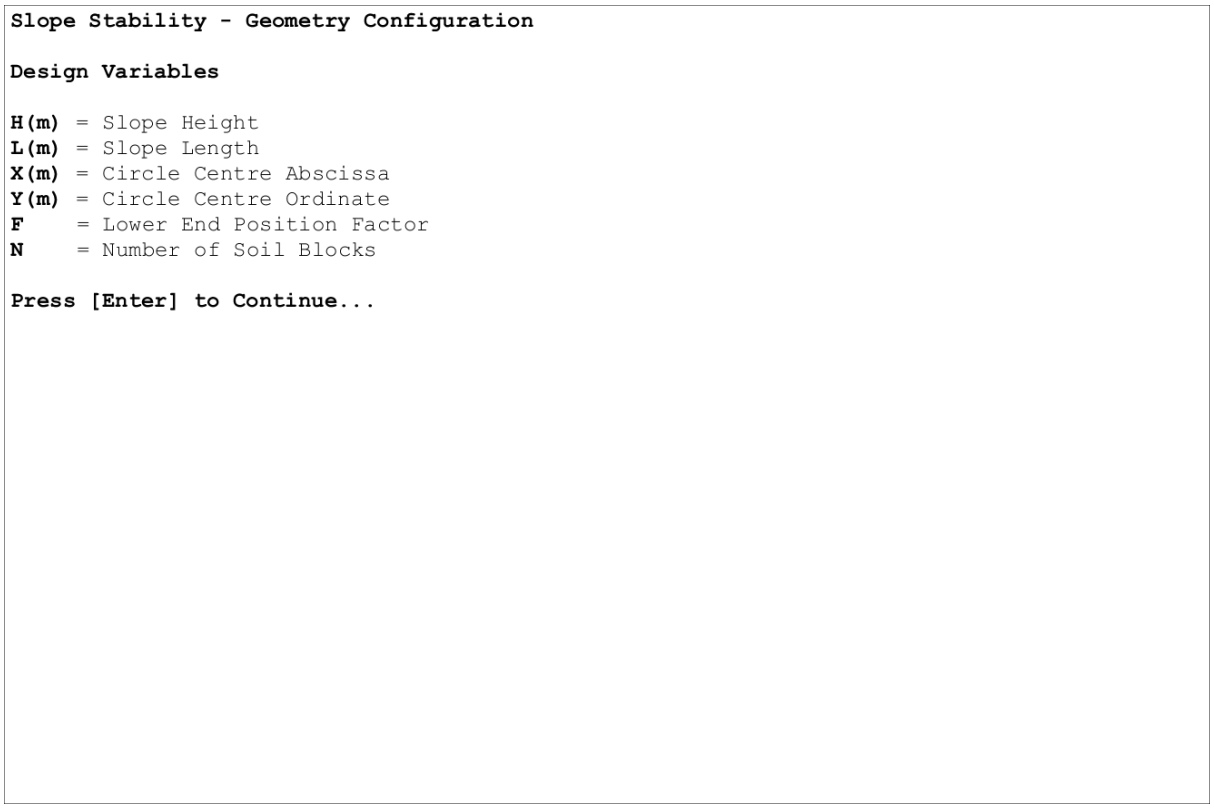


Figure 5-2: *'Slope Stability' MATLAB Program - Initial Screen.*
Source: MATLAB Program Developed for the Thesis.

The initial screen of the ‘Slope Stability’ MATLAB program is depicted in Figure 5-2. In this screen, the design variables are stated, along with their respective description.

The graphical user interface facilitates the configuration of the geometry, visually representing the effect of the design variable values over the total geometry.

By manually assigning the desired values to the design variables, the basic geometry is being presented to the user, as depicted in Figure 5-3.

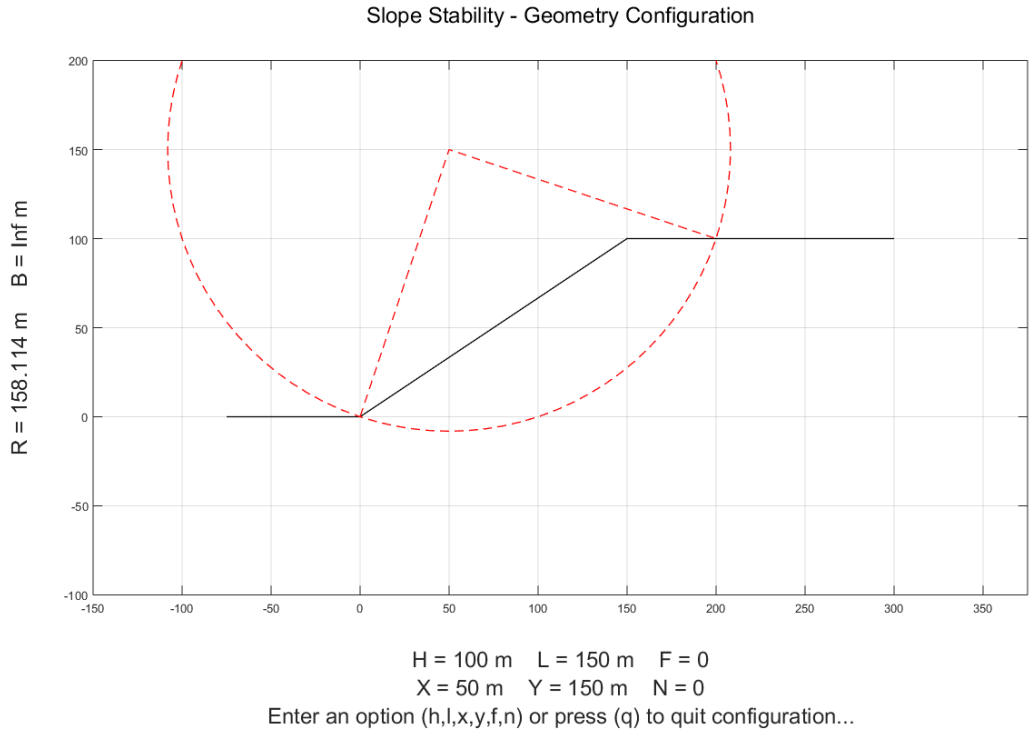


Figure 5-3: *‘Slope Stability’ MATLAB Program - Basic Geometry.*
Source: MATLAB Program Developed for the Thesis.

In the script, there are also included some drawing related functions which are disabled by default, being easily enabled by manually editing the script code and removing the relative percent symbol.

These extra functions can be used to perform a visual verification of the intermediate calculations, based on which the final geometry is being calculated, as depicted in Figure 5-4 and Figure 5-5.

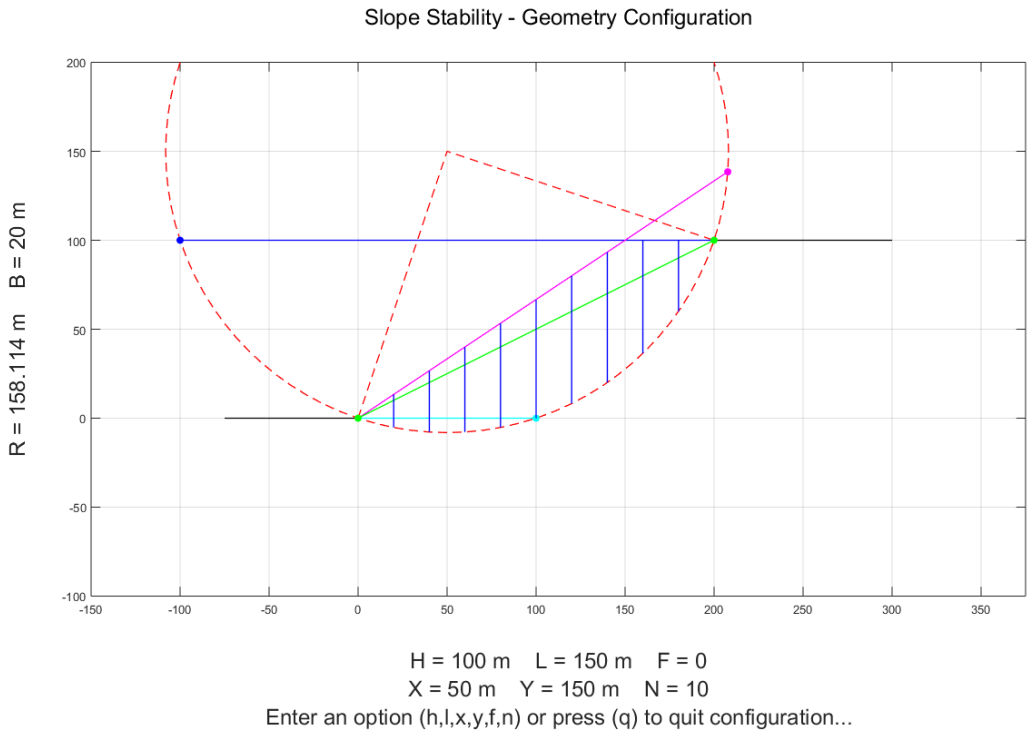


Figure 5-4: *'Slope Stability' MATLAB Program - Auxiliary Geometry.*
Source: MATLAB Program Developed for the Thesis.

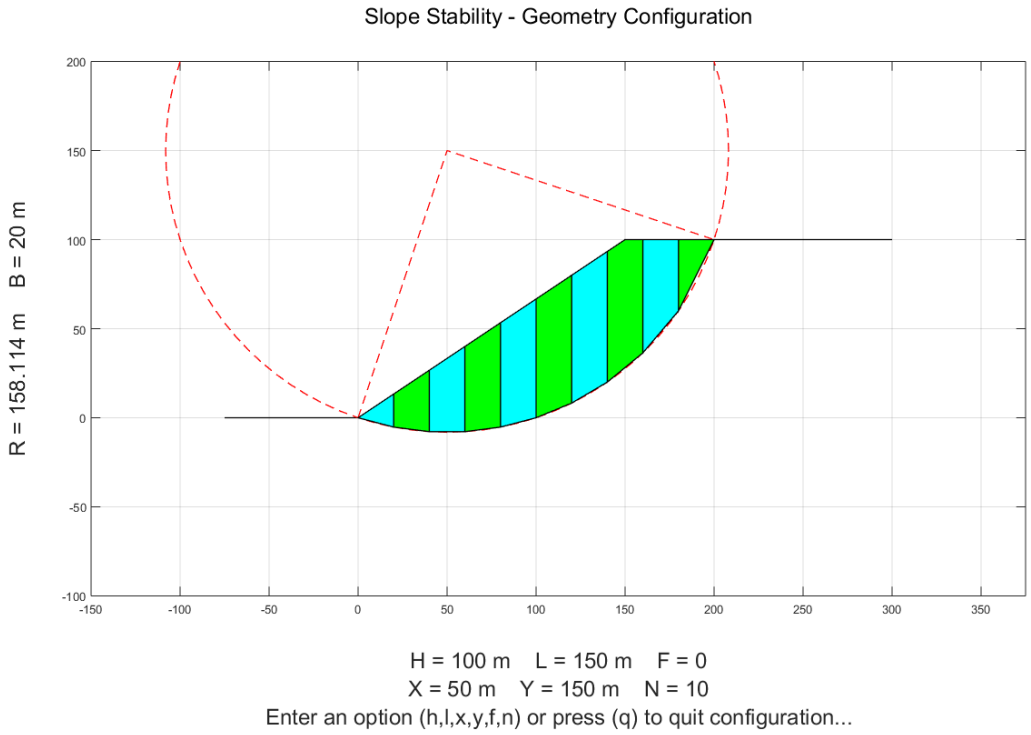


Figure 5-5: *'Slope Stability' MATLAB Program - Final Geometry.*
Source: MATLAB Program Developed for the Thesis.

5.3 ‘Spencer Method’ MATLAB script

A MATLAB script has been developed, for the purpose of the application of Spencer’s method, based on the provided values of the following design variables:

gamma	=	Specific weight of the soil, assumed to be common in all slices.
cohesion	=	Cohesion of soil at the slip surface, assumed to be common at all segments.
friction	=	Angle of internal friction of soil at the slip surface, assumed to be common at all segments.
r_u	=	Pore pressure coefficient at the slip surface, assumed to be common at all segments.

The following design variables were also taken into account, to compensate for any additional external forces acting on the slices:

F_x	=	External horizontal forces acting on each slice.
F_y	=	External vertical forces acting on each slice.
$M1$	=	Moment of forces F_x, F_y rotating about the middle of the base of each slice.

The following design variables were also taken into account, to compensate for the influence of an earthquake:

K_v	=	Factor of vertical acceleration during earthquake.
K_h	=	Factor of horizontal acceleration during earthquake.

Before the execution of the ‘Spencer Method’ MATLAB script, the design variables have to be initialized, and the ‘Geometry Configuration’ MATLAB script has to be executed in order to create the underlying geometry.

5.4 'Initialization' MATLAB script

A MATLAB script has been developed, for the purpose of the initialization of the design variables used in the aforementioned scripts.

The script is structured as follows:

(i) The geometry related design variables are initialized.
(ii) The 'Geometry Configuration' MATLAB script is executed.
(iii) The soil properties related design variables are initialized.
(iv) The 'Spencer Method' MATLAB script is executed.

Using the above script, for a specific problem of embankment stability analysis, by assuming a particular slip circle through the selection of the respective (X,Y) values, the corresponding value of the factor of safety can be easily determined.

However, as it would have been purely coincidental if the slip circle chosen had been the critical one, the determined factor of safety is not the actual factor of safety of the embankment against failure.

Since the position of the critical slip circle is unknown beforehand, the values of the factor of safety corresponding to all possible centres of rotation would have to be determined, the minimum one being the actual factor of safety.

Contours showing the variation in factor of safety at different centres of rotation have been published by Bishop and Morgenstern (1960) and by Sevaldson (1957). The contours are roughly elliptical with the major axis approximately at right angles to the surface of the slope and several times as large as the minor axis.

Using these contours as a guide, the search area for the critical slip circle centre can be narrowed down. A heuristic technique can then be used to determine the critical slip circle centre position.

5.5 ‘Minimization’ MATLAB script

A MATLAB script has been developed, for the purpose of the application of a heuristic technique to determine the critical slip circle centre position.

In the original methodology, in order to identify the position of the critical slip circle, a rectangular grid was set out, one side of the grid coinciding with the perpendicular bisector.

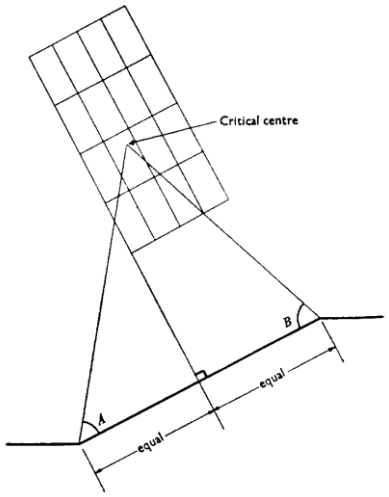


Figure 5-6: *Grid Search Pattern.*
Source: Spencer, E. (1967). A Method of Analysis of the Stability of Embankments.

By using a simple geometrical analysis, the pairs of (X,Y) values corresponding to the grid intersections can be automatically calculated.

Multiple slip surfaces can be examined at a time by creating a *for* loop, providing various pairs of (X,Y) values, executing the previous MATLAB program, and saving the determined factor of safety values in a corresponding vector. The (X,Y) values should be provided as the input argument of the *for* loop, not being initialized inside the loop by the following executed scripts.

It has to be noted that since the search grid is arbitrarily selected, the critical slip circle may correspond to an intermediate point between the examined grid intersections, thus a more refined procedure would be necessary to determine the actual minimum value of the factor of safety of the embankment against failure.

In the original procedure, the position of the critical slip circle was determined by graphical interpolation between the values corresponding to the grid intersections. In order to define the position of the critical circle with greater precision, the original program was modified to a minimization type.

It has to be noted that in the original methodology, two separate problems had to be investigated, one corresponding to the minimum value of F_{m0}, and another corresponding to the minimum value of F_1. The results showed that the critical circles for F_{m0} and F_1 were invariably so close as to be almost coincident. This means that, for simplicity reasons, the position of the critical circle centre could be determined using the examined values of F_{m0}.

The previous MATLAB program can be also easily modified to a minimization type. A way to implement a minimization technique is to create an appropriate function, which, for a given pair of (X,Y) values, returns the value of the factor of safety. This function can be then passed to a minimization algorithm.

In MATLAB, all functions, including local functions, have their own workspaces that are separate from the base workspace. Local functions cannot access variables used by other functions unless you pass them as arguments.

This means that the required function should accept a pair of (X,Y) values as the single input argument and return the corresponding value of the factor of safety as the single output argument.

All other design variables should retain constant values throughout the minimization process. The initialization of these design variables should be performed inside the function, so they would be available to the function workspace. This can be achieved by executing the previous MATLAB program inside the function.

This function can be then passed to a minimization algorithm, to determine the minimum value of the factor of safety, along with the corresponding pair of (X,Y) values, which identify the critical slip circle.

5.6 Results obtained with ‘Slope Stability’ MATLAB program

A confirmation of the results of the examples presented in the original investigation has been achieved using the ‘Slope Stability’ MATLAB program.

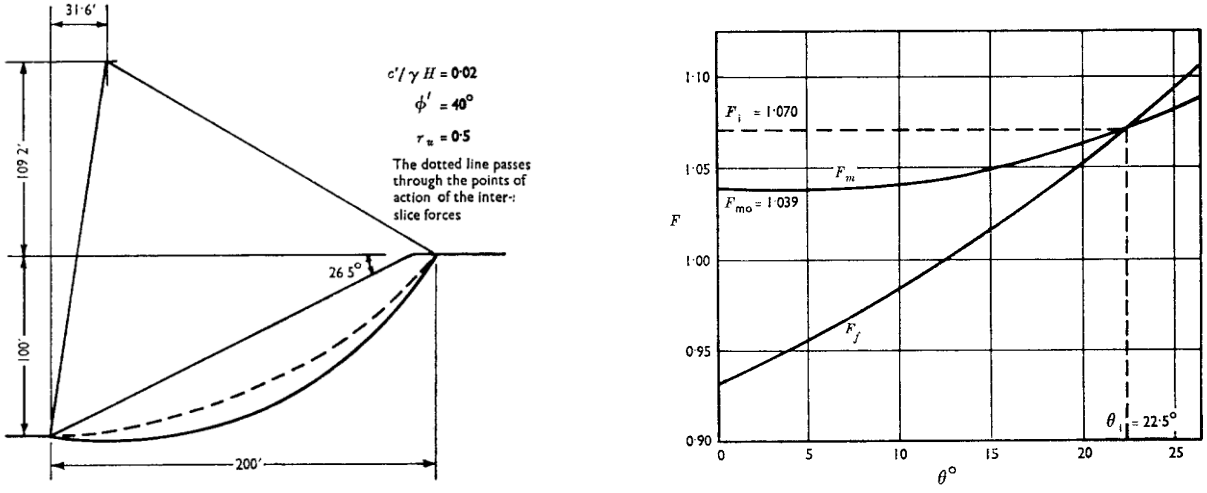


Figure 5-7: *Example Presented in the Original Investigation.*
Source: Spencer, E. (1967). A Method of Analysis of the Stability of Embankments.

The values of the design variables were set as indicated in the above example presented in the original investigation.

$H(m)$	=	$100 \cdot 0.3048$	=	30.48
$L(m)$	=	$200 \cdot 0.3048$	=	60.96
$X(m)$	=	$31.6 \cdot 0.3048$	=	9.63168
$Y(m)$	=	$(100+109.2) \cdot 0.3048$	=	63.76416
F	=	0	=	0
N	=	10 – 1000	=	10 – 1000

$c'/\gamma H$	=	0.02	*cohesion*	= $0.02 \cdot gamma \cdot H$
$\phi'(°)$	=	40	*friction*	= $40 \cdot (\pi / 180)$
r_u	=	0.5		

The value of *gamma* was set arbitrarily, since *cohesion* and *gamma* are being linearly dependent through the parameter $\gamma H \tan\phi'/c'$.

The problem was solved using the 'Slope Stability' MATLAB program, by assuming various values for the number of slices, in the range of 10 – 1000.

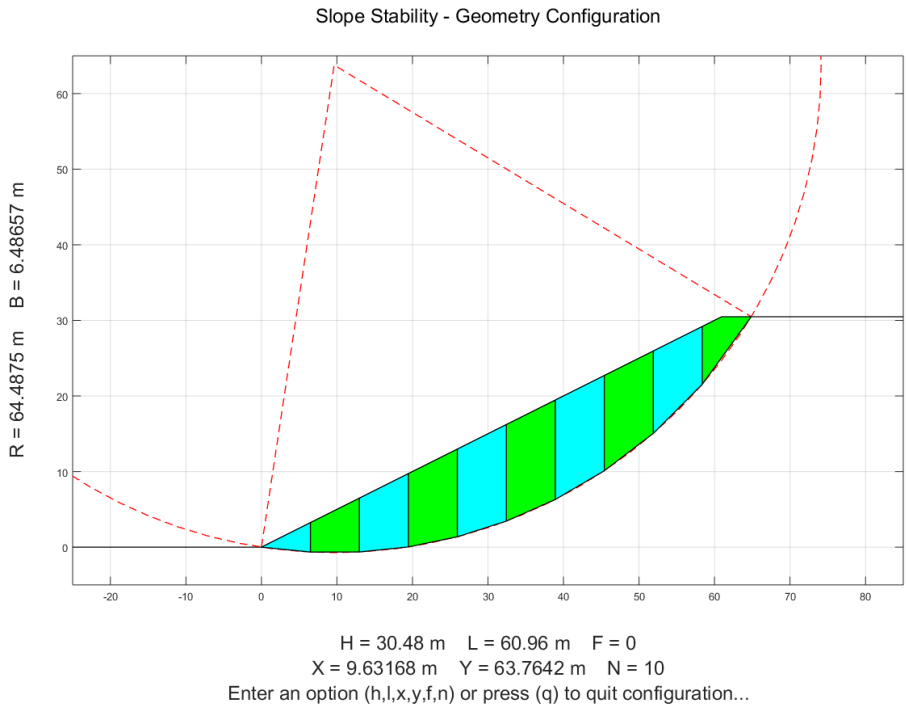


Figure 5-8: *'Slope Stability' MATLAB Program - Example Geometry.*
Source: MATLAB Program Developed for the Thesis.

The effect on SF of varying the number of slices, along with the corresponding value of $\theta(°)$ is being presented below:

N	=	10	SF	=	1.07188	$\theta(°)$	= 22.735
N	=	20	SF	=	1.07068	$\theta(°)$	= 22.491
N	=	50	SF	=	1.06974	$\theta(°)$	= 22.439
N	=	100	SF	=	1.06958	$\theta(°)$	= 22.432
N	=	200	SF	=	1.06954	$\theta(°)$	= 22.431
N	=	500	SF	=	1.06952	$\theta(°)$	= 22.430
N	=	1000	SF	=	1.06952	$\theta(°)$	= 22.430

The solution of the problem, as indicated in the original investigation, is as follows:

N	=	32	SF	=	1.070	$\theta(°)$	= 22.5

However, as it would have been purely coincidental if the slip circle chosen had been the critical one, the determined factor of safety is not the actual factor of safety of the embankment against failure.

Contours showing the variation in factor of safety at different centres of rotation have been published by Bishop and Morgenstern (1960) and by Sevaldson (1957). The contours are roughly elliptical with the major axis approximately at right angles to the surface of the slope and several times as large as the minor axis. In analyses in terms of effective stress, the centre of the critical slip circle is usually located close to the perpendicular bisector of the slope and normally on the 'uphill' side of it.

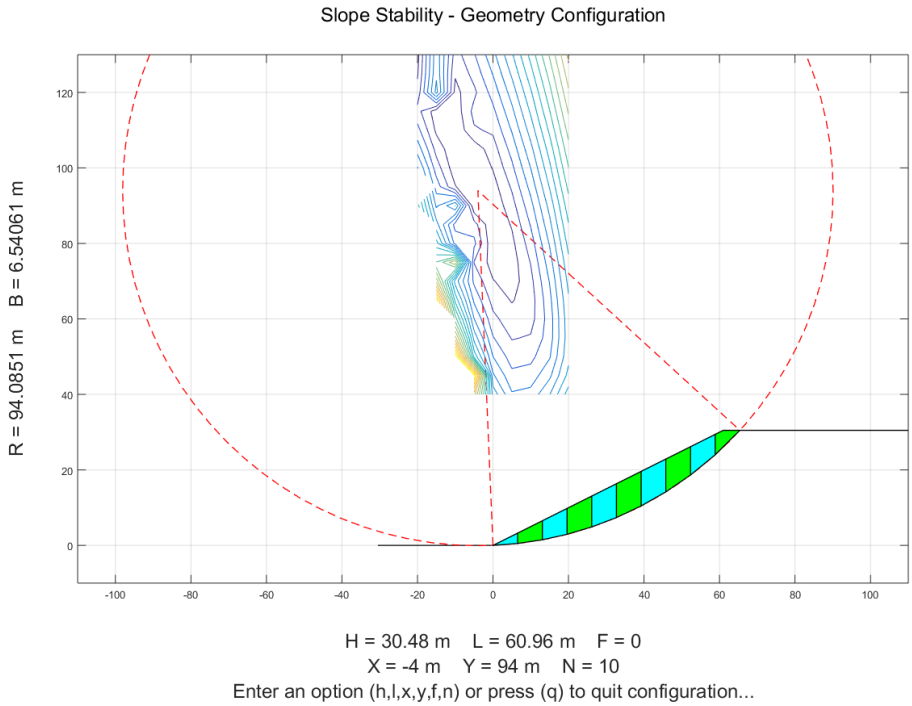


Figure 5-9: *'Slope Stability' MATLAB Program - Factor of Safety Contours.*
Source: MATLAB Program Developed for the Thesis.

A contour plot of the factor of safety values corresponding to various slip circle centres has been superimposed on the embankment geometry plot, providing a comprehensive visual representation. The factor of safety values have been determined using the *for* loop mentioned in the previous section, corresponding to the grid intersections of a search grid with a spacing of $1m$.

It is easily noticed that the shape, position, and orientation of the contours match exactly the description provided in the original investigation.

An approach to determine the actual factor of safety of the embankment against failure and the corresponding critical circle centre, would be the use of the *fmincon* non-linear minimization function, using linear inequality constraints corresponding to the search grid boundaries.

Due to the high non-linearity of the problem, and especially the non-smoothness induced by the methodology followed in the calculation procedure, the *fmincon* function cannot work properly, since the minimization algorithm is being trapped in the countless local minima. The *ga* genetic algorithm minimization function seems to behave better than the *fmincon* function.

A simpler approach to determine the actual factor of safety of the embankment against failure and the corresponding critical circle centre, is the use of the above mentioned *for* loop.

The coordinates of the corners of the grid can be used as index values, along with an appropriate step value, which can be further refined.

Since the general shape, position, and orientation of the contours is already documented in detail, and the search results can be visually inspected and verified, the use of the *for* loop would be the safer approach.

It has to be noted, that in the 'Slope Stability' MATLAB program, the revised formulation has been followed, the moment equilibrium equation for each slice being formulated by considering the moments of the forces about the middle of the base of each slice, instead of the centre of rotation.

While the formulation of an equilibrium equation should not have any effect in the final results, various minor geometry simplification assumptions between different formulations may do so.

Other examples presented in the original investigation have also been solved using the 'Slope Stability' MATLAB program, providing results almost identical to their original counterparts.

This fact provides a good indication that the programming code is working exactly as intended, as far as the underlying geometry generation, the formulation of the equilibrium equations, and the application of the minimization algorithm is concerned.

5.7 References

1. *Spencer, E. (1967). A Method of Analysis of the Stability of Embankments Assuming Parallel Inter-Slice Forces. Géotechnique Volume 17 Issue 1, March 1967, pp. 11-26.*
2. *Spencer, E. (1973). Thrust Line Criterion in Embankment Stability Analysis. Géotechnique Volume 23 Issue 1, March 1973, pp. 85-100.*

6. Conclusion

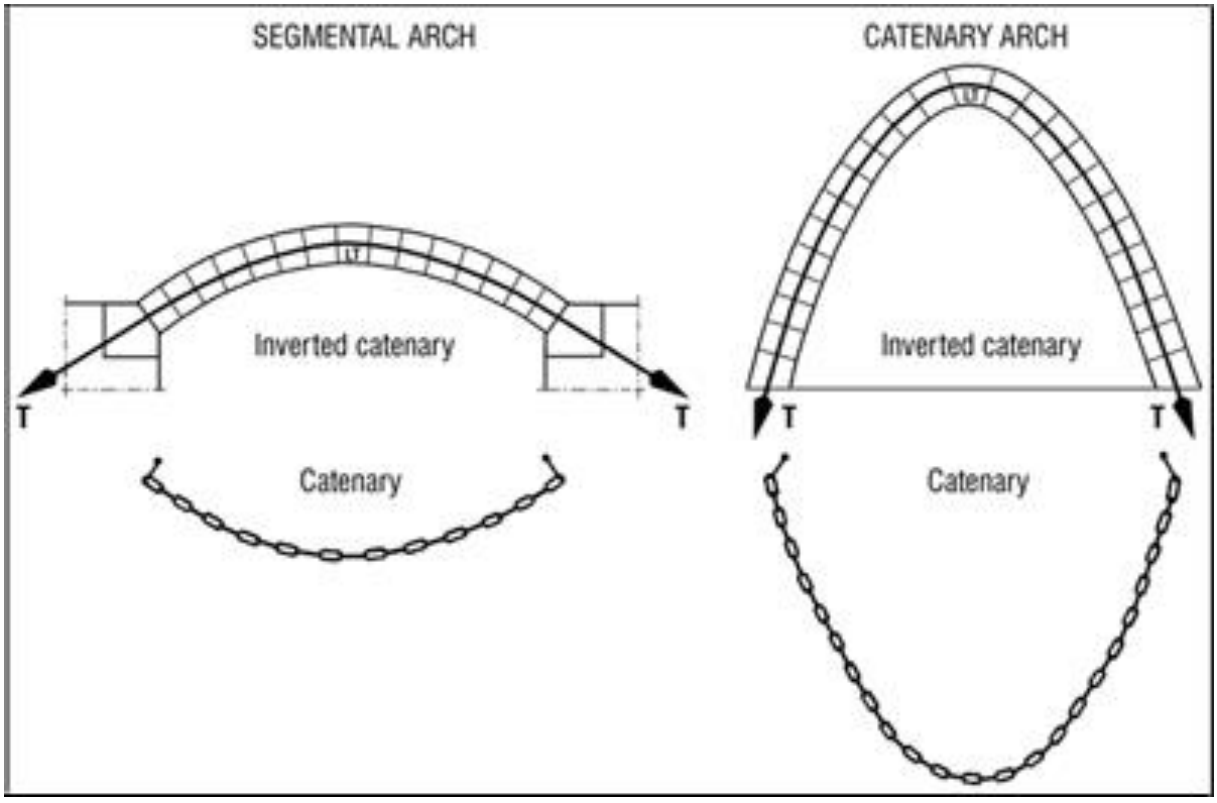


Figure 6-1: *Catenary Curves and Arches.*
Source: Auroville Earth Institute (AVEI).

6.1 Primary scope of this thesis

The primary scope of this thesis was the investigation of limit analysis of rigid body assemblies, following Livesley's 'Limit Analysis of Structures Formed from Rigid Blocks' (1978).

Heyman (1966) showed that the collapse load of a masonry arch or dome could be formulated as a problem in limit analysis.

That work led Livesley (1978) to consider the possibility of converting an existing computer program for plane skeletal frame collapse, into a program for the collapse analysis of plane masonry arches of arbitrary shape.

Drucker (1954) showed that in general the lower-bound (equilibrium) theorem is not true when yield is associated with Coulomb friction. Drucker's disproof of the theorem is by means of a counter-example.

When the limit on the shear force at a block interface is that associated with Coulomb friction, the failure mechanism computed by the algorithm will not necessarily satisfy the normality rule.

The corresponding limit load may therefore be an over-estimate of the true failure load, even though it is computed by a lower-bound (equilibrium) approach.

Upon further investigation, it became evident that while some counter-examples show that limit analysis does not give the correct failure load, other examples show that it does.

That fact led Livesley (1978) to investigate why this division occurs, establishing a criterion for testing the validity of a failure load computed in these circumstances.

It seems that in the cases that limit analysis gives an over-estimate of the true failure load, the maximizing algorithm assumes that the boundaries do positive work, implying some movement of supposedly fixed surfaces.

6.2 Method of investigation

The investigation has been carried out through the development of the 'Arch Stability' MATLAB program, using the methodology proposed by the original author, following the provided formulation.

It was decided that the 'Arch Stability' MATLAB program should be developed using a parametric layout, having the capability to support alternative formulations and easily be extended.

The 'Arch Stability' MATLAB program consists of various scripts, each one of them having a discrete functionality.

The scripts can be used separately, requiring manual interaction, but they can also be invoked from other scripts, providing their functionality without the need of manual interaction.

Each script includes all relative local functions at the end of the file, a syntax supported in MATLAB R2016b or later. Local functions are the most common way to break up programmatic tasks. This approach can be used as the base of code parameterization, facilitating automatic interaction between independent scripts.

The geometry of a circular arch was selected to be further investigated, since it is a common geometry that has also been investigated in Heyman's 'The Stone Skeleton' (1966), as well as in Milankovitch's 'Beitrag zur Theorie der Druckkurven' (1904).

Livesley's limit analysis method also takes into account of the Coulomb friction, while Heyman's method assumes that (i) stone has no tensile strength, (ii) the compressive strength of stone is effectively infinite, (iii) sliding of one stone upon another cannot occur.

It is obvious that by using a parametric programming code following Livesley's formulation, Heyman's formulation can also be investigated by assuming zero tensile strength, infinite compressive strength, and infinite Coulomb friction, through the adoption of the appropriate values of the block interface stress resultants limits, namely $0 \le q \le +\infty$, $0 \le s \le +\infty$, $-\infty \le t \le +\infty$.

6.3 Results of the investigation

A numerical confirmation of the minimum thickness of a semicircular arch under its own weight, as calculated by the Serbian scholar Milutin Milankovitch (1904), has been achieved using the 'Arch Stability' MATLAB program.

According to Milankovitch's theory of the thrust line, the minimum thickness of a semicircular arch under its own weight is $d_{min} = 0.1075 \cdot R$, while the thrust line touches the intrados at the point corresponding to the rupture angle $54°29'$.

The 'Dead-Load Problem' MATLAB script was used, with a geometry of a semicircular arch of middle radius $R = 10m$.

The minimum required thickness and the corresponding kinematic collapse mechanism indicated by the 'Arch Stability' MATLAB program match exactly the values documented in Milankovitch's theory of the thrust line.

Other dead-load and live-load problems have also been solved using the 'Arch Stability' MATLAB program, providing results identical to their original counterparts.

This fact provides a good indication that the programming code is working exactly as intended, as far as the underlying geometry generation, the formulation of the equilibrium equations, and the application of the minimization algorithm is concerned.

6.4 Secondary scope of this thesis

The secondary scope of this thesis was the investigation of the stability of embankments, following Spencer's 'A Method of Analysis of the Stability of Embankments Assuming Parallel Inter-Slice Forces' (1967).

The geometry of an embankment discretized into vertical slices, as used in embankment stability analysis problems, roughly resembles the circular arch geometry.

From a technical point of view, these two problems may seem unrelated to each other, but from a programming view, these two problems are highly related as far as the configuration of geometry is concerned, with respect to the equations of equilibrium, thus, common programming techniques can be used to address these two otherwise unrelated problems.

6.5 Method of investigation

The investigation has been carried out through the development of the 'Slope Stability' MATLAB program, using the methodology proposed by the original author, following the provided formulation.

It was decided that the 'Slope Stability' MATLAB program should be developed using a parametric layout, having the capability to support alternative formulations and easily be extended.

The basic concepts of the original method of analysis (1967) were followed, using the revised formulation (1973), where the moment equilibrium equation for each slice is being formulated by considering the moments of the forces about the middle of the base of each slice, instead of the centre of rotation.

While the formulation of an equilibrium equation should not have any effect in the final results, various minor geometry simplification assumptions between different formulations may do so.

6.6 Results of the investigation

A confirmation of the results of the examples presented in the original investigation has been achieved using the ‘Slope Stability’ MATLAB program.

Contours showing the variation in factor of safety at different centres of rotation have been published by Bishop and Morgenstern (1960) and by Sevaldson (1957). The contours are roughly elliptical with the major axis approximately at right angles to the surface of the slope and several times as large as the minor axis. In analyses in terms of effective stress, the centre of the critical slip circle is usually located close to the perpendicular bisector of the slope and normally on the ‘uphill’ side of it.

The values of the design variables were set as indicated in the main example presented in the original investigation.

The determined factor of safety value is almost identical to the one indicated in the original investigation example.

A contour plot of the factor of safety values corresponding to various slip circle centres has been superimposed on the embankment geometry plot, providing a comprehensive visual representation.

It is easily noticed that the shape, position, and orientation of the contours match exactly the description provided in the original investigation.

Other examples presented in the original investigation have also been solved using the ‘Slope Stability’ MATLAB program, providing results almost identical to their original counterparts.

This fact provides a good indication that the programming code is working exactly as intended, as far as the underlying geometry generation, the formulation of the equilibrium equations, and the application of the minimization algorithm is concerned.

6.7 References

1. *Livesley, R. (1978). Limit Analysis of Structures Formed from Rigid Blocks. International Journal for Numerical Methods in Engineering 12(12):1853-1871 · January 1978.*
2. *Heyman, J. (1966). The Stone Skeleton. International Journal of Solids and Structures 2(2):249-279 · April 1966.*
3. *Milankovitch, M. (1904). Beitrag zur Theorie der Druckkurven. Dissertation zur Erlangung der Doktorwürde, K.K. Technische Hochschule, Vienna.*
4. *Spencer, E. (1967). A Method of Analysis of the Stability of Embankments Assuming Parallel Inter-Slice Forces. Géotechnique Volume 17 Issue 1, March 1967, pp. 11-26.*
5. *Spencer, E. (1973). Thrust Line Criterion in Embankment Stability Analysis. Géotechnique Volume 23 Issue 1, March 1973, pp. 85-100.*

This page is intentionally left blank