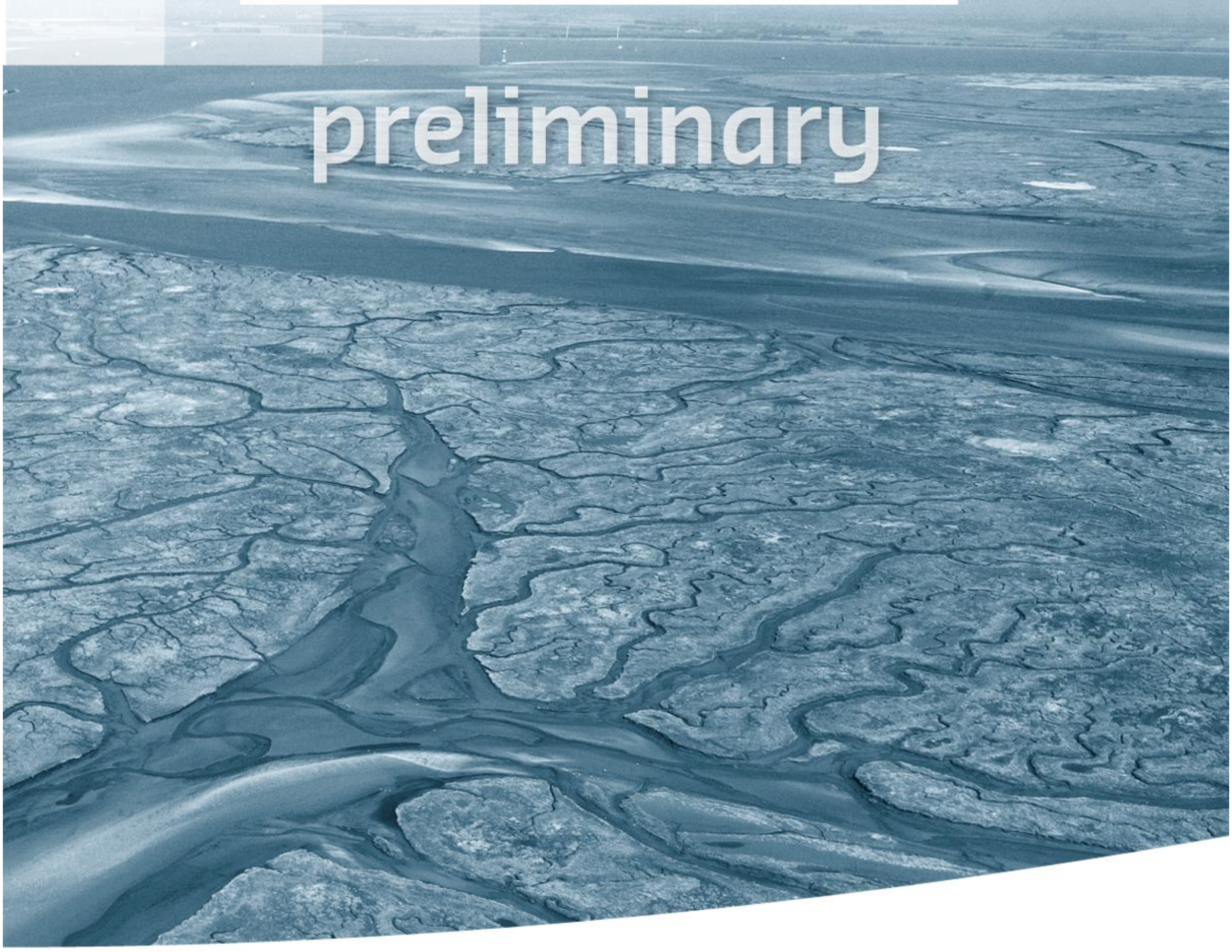
Deltares
Enabling Delta Life

Implementation of a Genetic Algorithm in MStab

For Faster Calculations and Better Accuracy



Implementation of a Genetic Algorithm in MStab

For Faster Calculations and Better Accuracy

Raymond R van der Meij
Hans Sellmeijer

Title
Implementation of a Genetic Algorithm in MStab

Project	**Reference**	**Pages**
1001071-010	1001071-010-GEO-0001	41

Keywords
Genetic Algorithm, MStab, Bishop, Uplift-Van, Jambu, Spencer

Summary
A genetic Algorithm has is implemented in MStab in order to search very efficiently for global minima. This can be a minimum for Bishop's method, Van's method, or even a free surface search using Jambu or Spencer's method.

References
Place references here

Version	Date	Author	Initials	Review	Initials	Approval	Initials
1	2009-02-19	Raymond R van der Meij		Hans Sellmeijer			

State
preliminary
This report is a preliminary report, not a final report and for discussion purposes only. No part of this report may be relied upon by either principal or third parties.

Contents

1 Introduction

Several computational programs are available with which the stability of a soil body can be calculated. In such a program, a slip surface is analyzed with a certain methodology (for example, the method “Bishop”) to determine its stability. The user enters an area in which the program needs to find the circle with the minimal stability factor.

Searching this space in MStab currently happens by calculating all possible slip circles with corresponding tangent lines, and reporting the one with the minimal safety. When the minimal safety factor is at the edge of the grid, the grid is shifted and recalculated. This algorithm has several disadvantages:

- it is sequential and therefore very time consuming;
- there is no guarantee the (global) minimal safety factor will be found;
- a small displacement of the initial grid can lead to fundamentally different answers;
- a small change in boundary conditions can lead to fundamentally different answers;
- much experience and understanding of the method is required, even though this is not obvious.

As mentioned, the search procedure is very time consuming, but on top of that, there’s a risk of getting a fundamentally incorrect answer. In the past, several more advanced search algorithms have been programmed, all with their own disadvantages. Thus far, the method of calculating an entire grid has proven to be the best option.

In the recent past, experience is obtained using new search algorithms. A so-called Genetic Algorithm (GA) seems to be a well-suited method to find the slip circle with the minimal safety factor. The method seems to be faster and better at finding a global minimum. A disadvantage is that the results are not always reproducible. On top of that, there will be a very strong tendency to find the global minimum, while sometimes, a local minimum is interesting as well.

A Genetic Algorithm does not only give the possibility to find the slip surface with the lowest stability factor, but is also able to find a free slip surface. In other words, for some methods like Jambu and Spencer it is possible to find the slip surface with the lowest resistance freely.

The goal of this report is to show the implementation of such a Genetic Algorithm in MStab. First, the GA code is programmed in Matlab together with unit tests to show that the algorithm functions properly. Afterwards, the process is sown where the GA is programmed in such a general way that it can minimize any function in MStab.

Chapter 2, gives the background theory on Genetic Algorithms and shows the algorithm to be implemented in MStab. Chapter 3 shows how this calculation kernel will be implemented in MStab. Chapter 4 contains the unit tests to prove that the algorithm is correct and chapter 5 shows the results of the implementation of the GA related to Bishop’s method. Chapter 6 does so in relation to Van’s method and chapter 7 for a free surface search.

The current document is written without apendixis to keep all the information central. This will change before this report becomes final.

2 The Genetic Algorithm

2.1 Introduction

A Genetic Algorithm (GA) is a procedure to find an optimum value in a search space. GA's perform particularly well in a complex search space with many parameters. It has been proven frequently that it is a very fast search mechanism that is very good in finding a global optimum. These properties make it a valuable addition to the current slope stability program MStab. A GA will be programmed to find the optimum value for different MStab methods.

Two unit tests are defined in chapter 4 in order to prove the correctness of the algorithm. First a very realistic one: an analytical case of Bishop's method. The second case is the so-called "Rastrigin" function. The latter function is a very complex one with many local minima. If a search algorithm works for this function it is very good in finding the global minimum. If the GA works converges quickly and properly for both functions it can be seen as a fast robust extension to the current MStab program. First, these test are performed in Matlab. If these tests are successful, the implementation in MStab can be performed.

2.2 About Genetic Algorithms

A GA has a population, which is an abstract representation (a so called "genome") of candidate solutions. This population evolves toward better solutions. The evolution starts from a randomly generated population of individuals and happens in generations. In each generation, the fitness (in the case of stability: the safety factor) of every individual in the population is evaluated. Multiple individuals are selected from the current population to be combined through selection and reproduction (crossover and mutation) to form a new population in the next generation.

The methodology is as follows

1. Choose initial population.
2. Evaluate the fitness (stability) of each individual in the population.
3. Repeat until termination:
   - 3.1. Breed new generation through selection, crossover and mutation.
   - 3.2. Evaluate the individual fitnesses of the offspring.
   - 3.3. Replace worst ranked part of population with offspring.

Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached. Other termination criteria can be:

- a solution is found that satisfies minimum criteria;
- allocated budget (computation time/money) reached;
- the highest ranking solution's fitness is reaching or has reached a plateau such that successive iterations no longer produce better results;
- manual inspection;
- combinations of the above.

This algorithm has first been programmed in Matlab with the following properties:

- tournament selection;
- random selection of parents;
- elitism.

Bishop’s method is implemented analytically as described in 4.1.1 (the function DetermineSafety). Any other function can be implemented like Rastrigin’s function, as described in section 4.2 The results of this implementation are shown in the same chapter regarding the Unit tests.

The methodology described earlier in this section is coded in Matlab. This code is printed in the next section.

2.3 The Matlab algorithm to be implemented in MStab

```
%GA Bishop
%Genetic Algorithm for determining the safety using Bishops method
%
%
%Created jan '09
%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%begin input
%Gen Algo Simulation related variables
PopSize=50;                           % Size of the population
generations=50;                       % # of generations
elite=2;                              % size of the elite
mutation_rate=0.1;                    % threshold value for mutation
genlen=3;                             % length of the genonme

%input
L=15.0;     % 0...200m
H=5.0;      % 0...20m

%search boundaries
bound.min(1)=-15;    % X value
bound.max(1)=30;     % X value
bound.min(2)=0;      % Y value
bound.max(2)=20;     % Y value
bound.min(3)=5;      % R value
bound.max(3)=100;    % R value

%Create the initial random population. Also, determine safety for every
%individual.

fitvec = ones(PopSize);
for i = 1 : PopSize
    population.X(i)=rand*(bound.max(1)-bound.min(1));
    population.Y(i)=rand*(bound.max(2)-bound.min(2));
    population.R(i)=rand*(bound.max(3)-bound.min(3));
    population.F(i)=DetermineSafety(population.X(i), population.Y(i),
      population.R(i), H, L);
    fitvec(i)=population.F(i);
```

```
end

%Begin the optimization algorithm. Loop over the generations. Inside a
%loop, pass the elite and breed the other individuals. in the end, the
%new generation becomes the current generation.

for n = 1 : generations
    %Pass elite
    for j = 1 : elite
        [maxfittest.fitness,Indexmaxfittest]=min(population.F);
        nextgen.X(j)=population.X(Indexmaxfittest);
        nextgen.Y(j)=population.Y(Indexmaxfittest);
        nextgen.R(j)=population.R(Indexmaxfittest);
        nextgen.F(j)=population.F(Indexmaxfittest);
        population.F(Indexmaxfittest)=NaN;
    end

    %Breed rest of population
    for j = (elite+1) : PopSize
        child1 = child(population, PopSize, mutation_rate, genlen,
            bound, H, L);
        child2 = child(population, PopSize, mutation_rate, genlen,
            bound, H, L);
        %put the fittest into next generation
        if child1(4) < child2(4)
            nextgen.X(j)=child1(1);
            nextgen.Y(j)=child1(2);
            nextgen.R(j)=child1(3);
            nextgen.F(j)=child1(4);
        else
            nextgen.X(j)=child2(1);
            nextgen.Y(j)=child2(2);
            nextgen.R(j)=child2(3);
            nextgen.F(j)=child2(4);
        end

    end

    %next generation becomes current generation.
    population=nextgen;
    clear nextgen;
    %for output purposes, make an array of the safety factor over the
      generations.
    [maxfittest.fitness,Indexmaxfittest]=min(population.F);
    safety(n)=population.F(Indexmaxfittest);

end
```

```
function [Safety, X, Y, R] = DetermineSafety(X, Y, R, H, L)
% Determine the safety for the simple dike geometry

% Sizes
XCount = size(X, 1);
YCount = size(Y, 2);
RCount = size(R, 3);
```

```
% Indices for different cases and objects
Index3 = (R > sqrt(X.^2+Y.^2)) & (R < sqrt((X-L).^2+(Y+H).^2));
Index5 = (R > sqrt(X.^2+Y.^2)) & (R > sqrt((X-L).^2+(Y+H).^2)) | Index3;
F = ones(XCount, YCount, RCount);
Alpha = 1000 * ones(XCount, YCount, RCount);

%Safety
FF = (X * L - Y * H + sqrt(max(R.^2 * (H^2+L^2) - (X * H + Y * L).^2,
0))) / (H^2+L^2);
F(Index3) = FF(Index3);
Alpha(Index5) = pi - asin((Y(Index5)+H * F(Index5)) ./ R(Index5));
Alpha(Index5) = Alpha(Index5) - asin(Y(Index5) ./ R(Index5));
Safety = Alpha ./ F ./ (1 - (F.^2*(H^2+L^2)/12 + (F*H/2+Y).^2 + (F*L/2-
X).^2) ./ R.^2);
Safety(~Index5) = NaN;
```

```
function offspring=child(population, pop, mutation_rate, genlen, bound,
H, L)
% Produces a child out of a population by choosing two parents, breeding
% and mutation.


%family=zeros(6,genlen);

    %fam(1,:)=father
    %fam(2,:)=mother
    %fam(3,:)=mask
    %fam(4,:)=breed
    %fam(5,:)=mutation mask
    %fam(6,:)=child

   %breed two families
        ind1=ceil(pop*rand);
        ind2=ceil(pop*rand);
        family(1,:)=[population.X(ind1), population.Y(ind1),
            population.R(ind1)];%store random father in  family
        family(2,:)=[population.X(ind2), population.Y(ind2),
            population.R(ind2)];%store random mother in family family
        family(3,:)=round(rand(genlen,1))+1; %create mask to mix parents
            properties

        for j=1:genlen  %for complete genome
            family(4,j)=family(family(3,j),j); %if mask =1 take father,
                  else take mother
        end

    %mutate

        family(5,:)=rand(genlen,1);

        for j=1:genlen
            if family(5,j)< mutation_rate
            family(6,j)=bound.min(j)+rand*(bound.max(j)-bound.min(j));
            else
                family(6,j)=family(4,j);
            end
        end
        family(6,4)=DetermineSafety(family(6,1), family(6,2),
family(6,3), H, L);
        offspring=family(6,:);
```

3 Vision on the implementation in MStab

In MStab, a very general GA will be implemented in such a way that it can be used to minimize any function the user wants to minimize. This genetic algorithm will be as generic as possible and not contain any information about the function it needs to optimize.

Once the GA has been build, any method (for example, Bishop or Uplift Van) can be connected to this GA and MStab. Figure 3.1 shows four methods, but it can be any function one can think of that is in MStab.

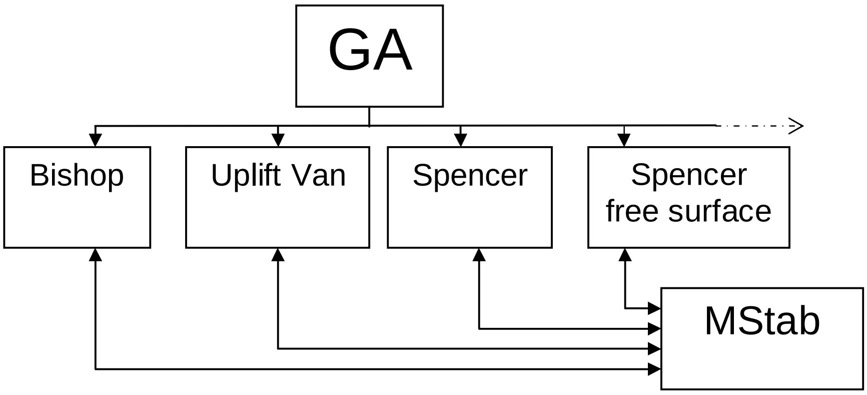


Figure 3.1 Inheritance of several methods to be used in MStab.

The reason to choose for this general approach is obvious. The GA code can be used for different failure mechanisms and should not be programmed separately for every mechanism. This modular approach also gives the possibility to search for a free slip surface instead of a circular one.

4 Testing the GA in Matlab

Because it is much work to implement the GA in MStab, it will first be implemented in Matlab. Two functions will be defined in this chapter and tested in the previously presented algorithm. The results can be compared to the GA that is standard in Matlab as well with the future results from MStab.

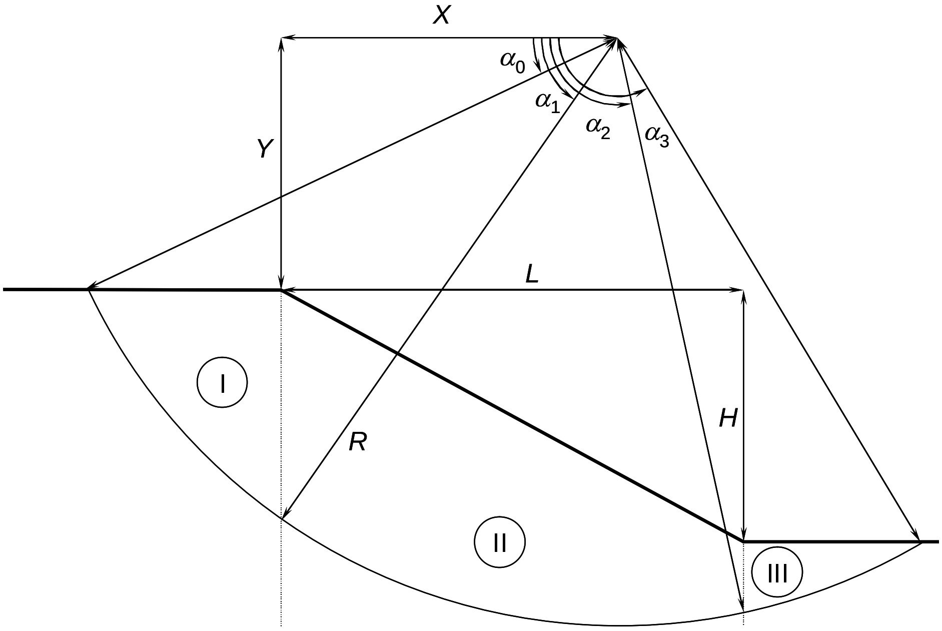
This gives a deeper understanding of the process we are modelling and gives us the certainty that the algorithm performs correctly. On top of that, it gives us a basis for a unit test in MStab.

4.1 Simplified version of Bishop's method

4.1.1 Deriving the equations

Introduction

An analytical formulation of Bishop's method will be derived for a simplified embankment in order to have an analytical safety factor to test the genetic algorithm. Only the crest, slope and surface level of an embankment will be considered. As can be seen in Figure 4.1, the slope is defined with "H" and "L", and the centre of the slip circle with "X" and "Y". The radius of the circle is "R". A circle to be considered will have a centre above the embankment.



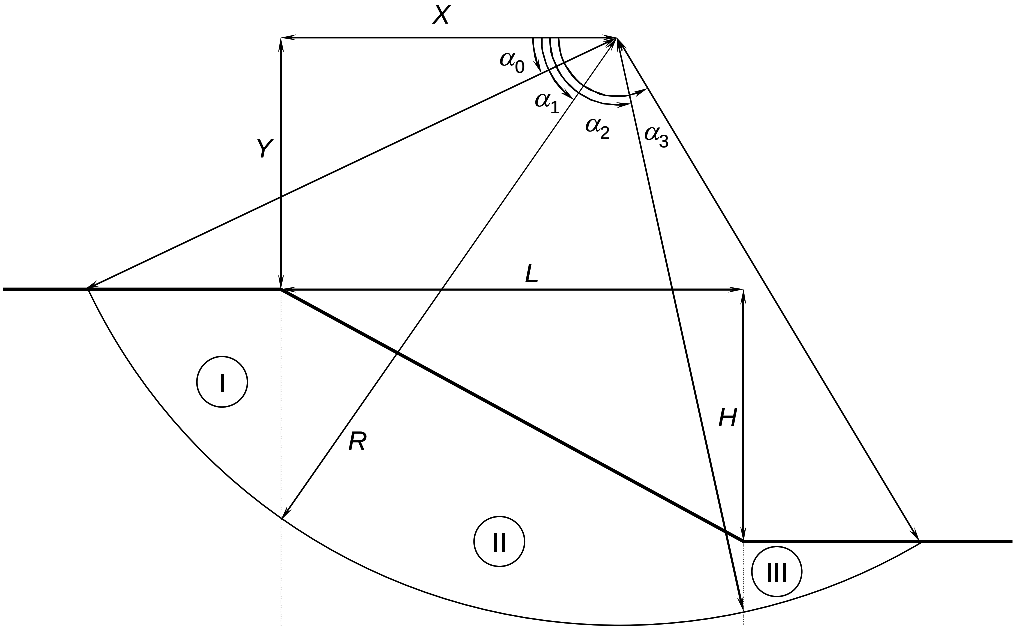


Figure 4.1 *Definition of the parameters related to the slip circle of case 1*

For the purpose of simplicity, no water pressures are considered, so is a homogenous soil with cohesion only and no internal friction. In such a way, an explicit relationship can be derived for the safety factor.

$$F = \frac{M_{\text{reaction}}}{M_{\text{soil}}} \qquad M_{\text{reaction}} = \int_{\alpha_0}^{\alpha_3} R\, c\, R\, \mathrm{d}\alpha \qquad M_{soil} = \int_{\alpha_0}^{\alpha_3} R \cos\alpha\, \gamma_{\text{g}} h\, R \sin\alpha\, \mathrm{d}\alpha \qquad (1)$$

Four types or circles can be distinguished.

1. A circle that enters through the crest and exits on the surface level (drawn in Figure 4.1): $R^2 > X^2 + Y^2$ *and* $R^2 > (L-X)^2 + (H+Y)^2$.
2. A circle that enters through the crest and exits in the slope of the embankment: $R^2 > X^2 + Y^2$ *and* $R^2 < (L-X)^2 + (H+Y)^2$.
3. A circle that enters through the slope and exits on the surface level: $R^2 < X^2 + Y^2$ *and* $R^2 > (L-X)^2 + (H+Y)^2$.
4. A circle that enters and exits through the slope of the embankment: $R^2 < X^2 + Y^2$ *and* $R^2 < (L-X)^2 + (H+Y)^2$.

The angle “α” rotates around the centre of the circle with as reference the horizontal parallel to the crest and surface level. The circle defines several angles, “α_0” through “α_3”. As can be seen in Figure

4.1

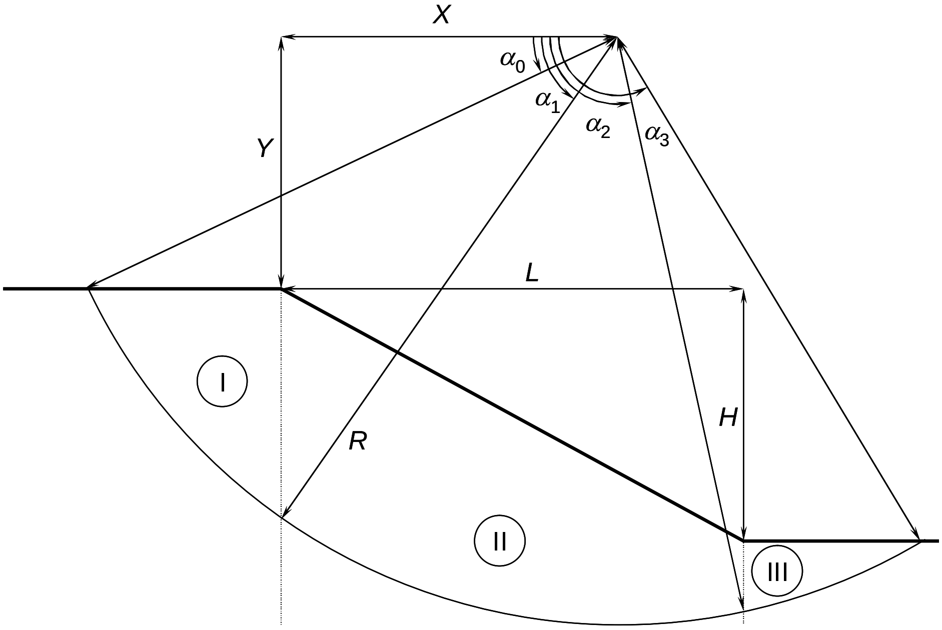


Figure 4.1.

$$\alpha_0 = \arcsin\left(\frac{Y}{R}\right) \qquad \alpha_1 = \arccos\left(\frac{X}{R}\right) \qquad \alpha_2 = \pi - \arccos\left(\frac{L-X}{R}\right) \qquad \alpha_3 = \pi - \arcsin\left(\frac{H+Y}{R}\right) \tag{2}$$

Deriving case 1

Case 1, as drawn in Figure 4.1,

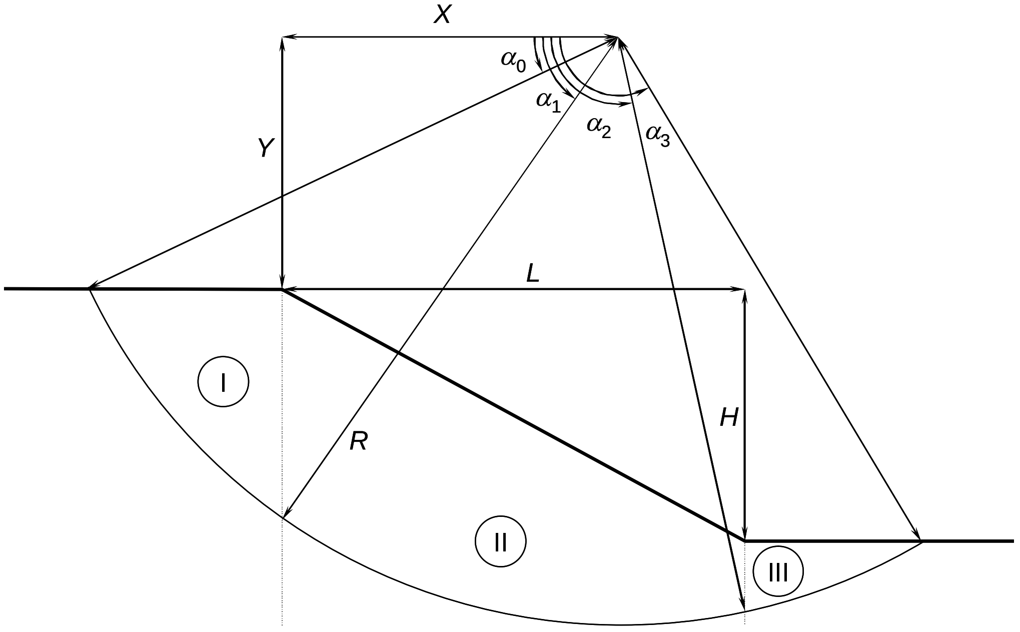


Figure 4.1 is derived first. The height of a strip can be expressed by using these , α's. Because of the shape of the embankment, three area's need to be distinguished: I, II and III (see Figure 4.1

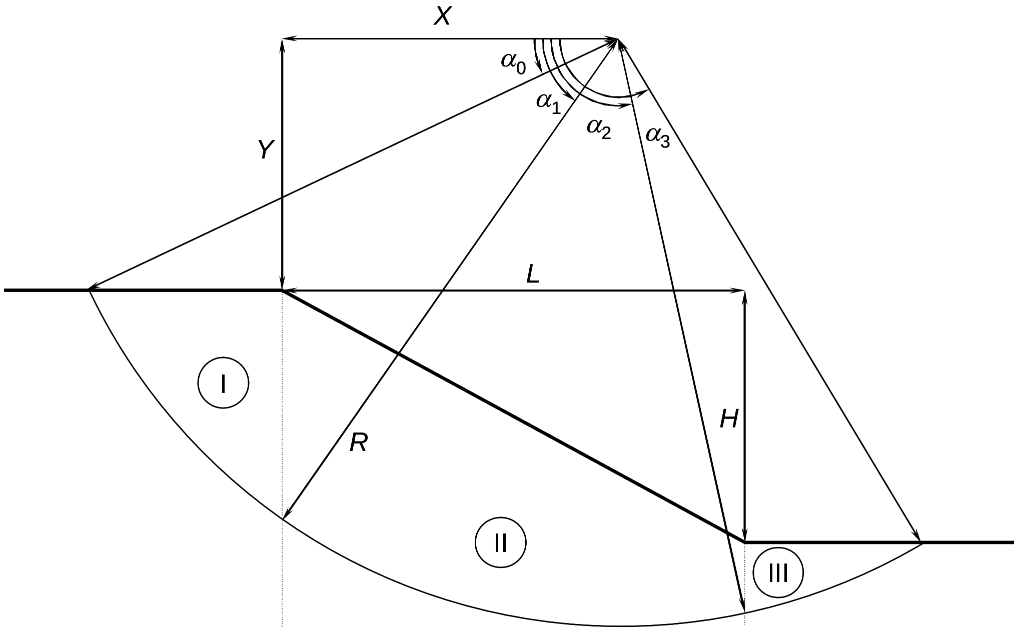


Figure 4.1).

$$
\begin{aligned}
h_I &= R\sin(\alpha) - Y && \mathrm{I}\\
h_{II} &= R\sin(\alpha) - Y - \frac{X - R\cos(\alpha)}{L}H && \mathrm{II}\\
h_{III} &= R\sin(\alpha) - Y - H && \mathrm{III}
\end{aligned}
\tag{3}
$$

Both integrals of the safety factor can be determined analytically under the given boundary conditions. The relevant terms are:

$$
\begin{aligned}
\int_{\alpha_0}^{\alpha_3} \mathrm{d}\alpha &= \alpha_3 - \alpha_0\\
\int_{\alpha_m}^{\alpha_n} \cos\alpha\,\sin\alpha\,\mathrm{d}\alpha &= \int_{\alpha_m}^{\alpha_n} \sin\alpha\,\mathrm{d}(\sin\alpha) = \tfrac{1}{2}\left(\sin^2\alpha_n - \sin^2\alpha_m\right)\\
\int_{\alpha_0}^{\alpha_3} \cos\alpha\,\sin\alpha\,\sin\alpha\,\mathrm{d}\alpha &= \int_{\alpha_0}^{\alpha_3} \sin^2\alpha\,\mathrm{d}(\sin\alpha) = \tfrac{1}{3}\left(\sin^3\alpha_3 - \sin^3\alpha_0\right)\\
\int_{\alpha_1}^{\alpha_2} \cos\alpha\,\cos\alpha\sin\alpha\,\mathrm{d}\alpha &= -\int_{\alpha_1}^{\alpha_2} \cos^2\alpha\,\mathrm{d}(\cos\alpha) = -\tfrac{1}{3}\left(\cos^3\alpha_2 - \cos^3\alpha_1\right)
\end{aligned}
\tag{4}
$$

The angles can be converted to distances:

$$
\begin{aligned}
\int_{\alpha_0}^{\alpha_3} \mathrm{d}\alpha &= \alpha_3 - \alpha_0
& \int_{\alpha_1}^{\alpha_2} \cos\alpha\,\sin\alpha\,\mathrm{d}\alpha &= \tfrac{1}{2}\left\{\left(\frac{X}{R}\right)^2 - \left(\frac{L-X}{R}\right)^2\right\}\\
\int_{\alpha_0}^{\alpha_3} \cos\alpha\,\sin\alpha\,\sin\alpha\,\mathrm{d}\alpha &= \tfrac{1}{3}\left\{\left(\frac{H+Y}{R}\right)^3 - \left(\frac{Y}{R}\right)^3\right\}
& \int_{\alpha_2}^{\alpha_3} \cos\alpha\,\sin\alpha\,\mathrm{d}\alpha &= \tfrac{1}{2}\left\{\left(\frac{H+Y}{R}\right)^2 + \left(\frac{L-X}{R}\right)^2 - 1\right\}\\
\int_{\alpha_1}^{\alpha_2} \cos\alpha\,\cos\alpha\sin\alpha\,\mathrm{d}\alpha &= \tfrac{1}{3}\left\{\left(\frac{X}{R}\right)^3 + \left(\frac{L-X}{R}\right)^3\right\}
& \int_{\alpha_0}^{\alpha_3} \cos\alpha\,\sin\alpha\,\mathrm{d}\alpha &= \tfrac{1}{2}\left\{\left(\frac{H+Y}{R}\right)^2 - \left(\frac{Y}{R}\right)^2\right\}
\end{aligned}
\tag{5}
$$

Hereby, the driving moment is determined:

$$
\begin{aligned}
\frac{M_{\mathrm{reaction}}}{c\,R^2} &= \alpha_3 - \alpha_0\\
\frac{M_{\mathrm{soil}}}{\gamma_{\mathrm{g}} H\,R^2} &= \frac{\dfrac{R}{3H}\left\{\left(\dfrac{H+Y}{R}\right)^3 - \left(\dfrac{Y}{R}\right)^3\right\} - \dfrac{Y}{2H}\left\{\left(\dfrac{H+Y}{R}\right)^2 - \left(\dfrac{Y}{R}\right)^2\right\}}{\dfrac{R}{3L}\left\{\left(\dfrac{X}{R}\right)^3 + \left(\dfrac{L-X}{R}\right)^3\right\} - \dfrac{X}{2L}\left\{\left(\dfrac{X}{R}\right)^2 - \left(\dfrac{L-X}{R}\right)^2\right\}} - \tfrac{1}{2}\left\{\left(\frac{H+Y}{R}\right)^2 + \left(\frac{L-X}{R}\right)^2 - 1\right\}
\end{aligned}
\tag{6}
$$

Or:

$$\frac{M_{\text{reaction}}}{c\,R^2} = \alpha_3 - \alpha_0$$

$$\frac{M_{\text{soil}}}{\gamma_{\text{g}} H\,R^2} = \begin{array}{l} \frac{1}{3}\frac{H^2}{R^2} + \frac{H\,Y}{R^2} + \frac{Y^2}{R^2} - \frac{1}{2}\frac{H\,Y}{R^2} - \frac{Y^2}{R^2} \\ \frac{1}{3}\frac{L^2}{R^2} - \frac{L\,X}{R^2} + \frac{X^2}{R^2} + \frac{1}{2}\frac{L\,X}{R^2} - \frac{X^2}{R^2} \end{array} - \tfrac{1}{2}\left\{\left(\frac{H+Y}{R}\right)^2 + \left(\frac{L-X}{R}\right)^2 - 1\right\} \qquad (7)$$

Or:

$$\frac{M_{\text{reaction}}}{c\,R^2} = \alpha_3 - \alpha_0 \qquad \frac{M_{\text{soil}}}{\frac{1}{2}\gamma_{\text{g}} H\,R^2} = 1 - \frac{1}{3}\frac{H^2}{R^2} - \frac{H\,Y}{R^2} - \frac{Y^2}{R^2} - \frac{1}{3}\frac{L^2}{R^2} + \frac{L\,X}{R^2} - \frac{X^2}{R^2} \qquad (8)$$

Hereby, a simple analytical expression of the safety factor is given for case 1:

$$F = \frac{M_{\text{reaction}}}{M_{\text{soil}}} = \frac{c}{\gamma_{\text{g}} H} \frac{2\left(\alpha_3 - \alpha_0\right) R^2}{R^2 - \frac{1}{12}\left(H^2 + L^2\right) - \left(\frac{1}{2}H + Y\right)^2 - \left(\frac{1}{2}L - X\right)^2} \qquad (9)$$

Deriving case 2

Case 2 is next and depicted in Figure 4.2. Angle “α_3” is no longer relevant and “α_2” becomes another variable.

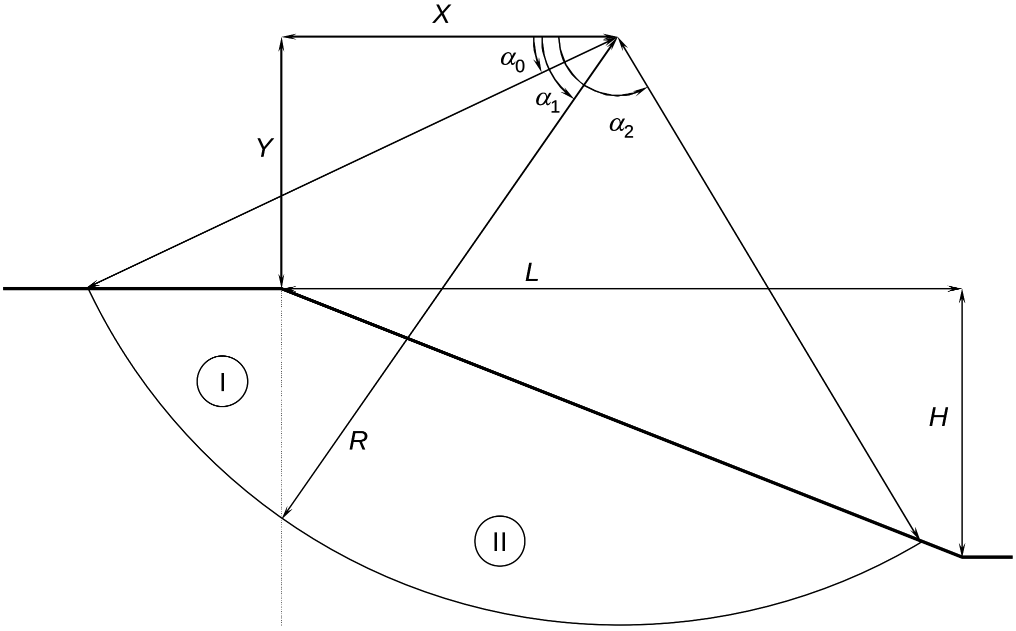


Figure 4.2 *Definition of the parameters related to the slip circle of case 2*

This gives the following relationship:

$$\begin{array}{lll} \sin\alpha_0 = \dfrac{Y}{R} & R\,\dfrac{\sin\alpha_2 - \sin\alpha_0}{H} = F & \multirow{2}{*}{$F = \dfrac{1}{\sqrt{L^2+H^2}}\left[\dfrac{X\,L - Y\,H}{\sqrt{L^2+H^2}} + \sqrt{R^2 - \dfrac{\left(X\,H + Y\,L\right)^2}{\left(L^2+H^2\right)}}\right]$} \\ \cos\alpha_1 = \dfrac{X}{R} & R\,\dfrac{\cos\alpha_1 - \cos\alpha_2}{L} = F & \end{array} \tag{12}$$

For the height of the slices, only area I and II are relevant:

$$\begin{array}{lll} h_I & = R\sin\left(\alpha\right) - Y & \mathrm{I} \\ h_{II} & = R\sin\left(\alpha\right) - Y - \dfrac{X - R\cos\left(\alpha\right)}{L}H & \mathrm{II} \end{array} \tag{13}$$

The required integrals have been determined for case I. We continue from equation (4) where the right areas need to be adjusted:

$$\begin{array}{ll} \displaystyle\int_{\alpha_0}^{\alpha_2} \mathrm{d}\alpha = \alpha_2 - \alpha_0 & \displaystyle\int_{\alpha_0}^{\alpha_2} \cos\alpha\,\sin\alpha\,\sin\alpha\;\mathrm{d}\alpha = \tfrac{1}{3}\left(\sin^3\alpha_2 - \sin^3\alpha_0\right) \\ \displaystyle\int_{\alpha_\mathrm{m}}^{\alpha_\mathrm{n}} \cos\alpha\,\sin\alpha\;\mathrm{d}\alpha = \tfrac{1}{2}\left(\sin^2\alpha_\mathrm{n} - \sin^2\alpha_\mathrm{m}\right) & \displaystyle\int_{\alpha_1}^{\alpha_2} \cos\alpha\,\cos\alpha\sin\alpha\;\mathrm{d}\alpha = -\tfrac{1}{3}\left(\cos^3\alpha_2 - \cos^3\alpha_1\right) \end{array} \tag{14}$$

The driving moment (1) can be presented in α's:

$$\frac{M_\mathrm{soil}}{\gamma_\mathrm{g} H\,R^2} = \frac{R}{3H}\left(\sin^3\alpha_2 - \sin^3\alpha_0\right) - \frac{Y}{2H}\left(\sin^2\alpha_2 - \sin^2\alpha_0\right) + \frac{R}{3L}\left(\cos^3\alpha_1 - \cos^3\alpha_2\right) - \frac{X}{2L}\left(\cos^2\alpha_1 - \cos^2\alpha_2\right) \tag{15}$$

From the four terms, the linear component is resolved following (12) and divided by:

$$\frac{M_\mathrm{soil}}{\gamma_\mathrm{g} H\,R^2 F} = \begin{array}{l} +\tfrac{1}{3}\left(\sin^2\alpha_2 + \sin\alpha_2\sin\alpha_0 + \sin^2\alpha_0\right) - \tfrac{1}{2}\sin\alpha_0\left(\sin\alpha_2 + \sin\alpha_0\right) \\ +\tfrac{1}{3}\left(\cos^2\alpha_2 + \cos\alpha_2\cos\alpha_1 + \cos^2\alpha_1\right) - \tfrac{1}{2}\cos\alpha_1\left(\cos\alpha_2 + \cos\alpha_1\right) \end{array} \tag{16}$$

Or:

$$\frac{M_\mathrm{soil}}{\gamma_\mathrm{g} H\,R^2 F} = \tfrac{1}{3} - \tfrac{1}{6}\sin\alpha_0\left(\sin\alpha_2 + \sin\alpha_0\right) - \tfrac{1}{6}\cos\alpha_1\left(\cos\alpha_2 + \cos\alpha_1\right) \tag{17}$$

And using equation (12):

$$\frac{M_\mathrm{soil}}{\gamma_\mathrm{g} H\,F} = \tfrac{1}{3}R^2 - \tfrac{1}{6}Y\left(F\,H + 2Y\right) + \tfrac{1}{6}X\left(F\,L + 2X\right) \tag{18}$$

By expressing the radius in other geometric units, the result looks like case 1:

$$\frac{M_\mathrm{soil}}{\tfrac{1}{2}\gamma_\mathrm{g} H\,F} = R^2 - Y\left(H\,F + Y\right) + X\left(L\,F - X\right) - \tfrac{1}{3}F^2\left(H^2 + L^2\right) \qquad R^2 = \left(H\,F + Y\right)^2 + \left(L\,F - X\right)^2 \tag{19}$$

And the safety factor for case 2 is derived:

$$F = \frac{M_{\text{reactie}}}{M_{\text{grond}}} = \frac{c}{\gamma_{\text{g}} H} \frac{1}{F} \frac{2\left(\alpha_2 - \alpha_0\right) R^2}{R^2 - \frac{1}{12} F^2 \left(H^2 + L^2\right) - \left(\frac{1}{2} H F + Y\right)^2 - \left(\frac{1}{2} L F - X\right)^2} \qquad (20)$$

Deriving case 3

Case 3 is next and depicted in Figure 4.3. Angle “α_0” is no longer relevant and “α_1” becomes a variable:

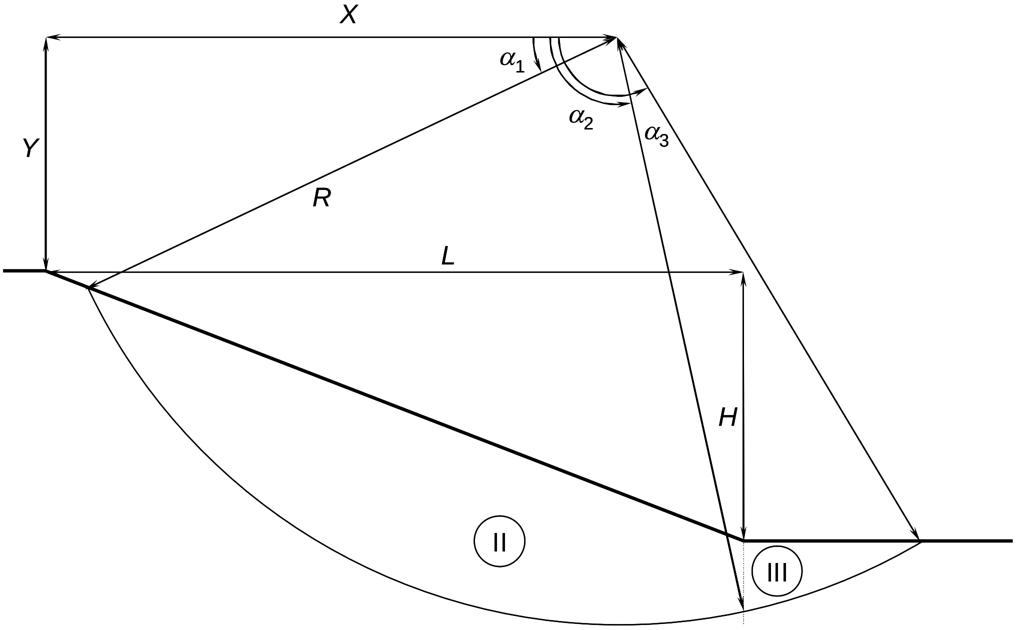


Figure 4.3 Definition of the parameters related to the slip circle of case 3

$$\begin{aligned} \alpha_2 &= \pi - \arccos\left(\frac{L-X}{R}\right) \qquad & \frac{R \sin\alpha_1 - Y}{H} &= 1-F \\ \alpha_3 &= \pi - \arcsin\left(\frac{H+Y}{R}\right) \qquad & \frac{X - R\cos\alpha_1}{L} &= 1-F \end{aligned} \qquad 1-F = \frac{1}{\sqrt{L^2+H^2}}\left[\frac{X\,L - Y\,H}{\sqrt{L^2+H^2}} - \sqrt{R^2 - \frac{\left(X\,H+Y\,L\right)^2}{\left(L^2+H^2\right)}}\right] \qquad (22)$$

The other angles are defined in the first case. Only areas II and III are relevant for determining the height of a slice.

$$\begin{aligned} h_{II} &= R\sin\left(\alpha\right) - Y - H + \frac{L - X + R\cos\left(\alpha\right)}{L} H \qquad && \text{II} \\ h_{III} &= R\sin\left(\alpha\right) - Y - H && \text{III} \end{aligned} \qquad (23)$$

The inverse notation is more convenient:

$$
\begin{aligned}
\alpha_2' &= \arccos\left(\frac{L-X}{R}\right) & \frac{Y+H-R\sin\alpha_1'}{H} &= F \\
\alpha_3' &= \arcsin\left(\frac{H+Y}{R}\right) & \frac{L-X-R\cos\alpha_1'}{L} &= F
\end{aligned}
\qquad
F = \frac{\dfrac{(L-X)L+(Y+H)H}{\sqrt{L^2+H^2}} + \sqrt{R^2 - \dfrac{\{(L-X)H-(Y+H)L\}^2}{\left(L^2+H^2\right)}}}{\sqrt{L^2+H^2}} \tag{24}
$$

And the required integrals have been determined in the previous case. In equation (4), the right areas are substituted:

$$
\begin{aligned}
\int_{\alpha_1}^{\alpha_3} \mathrm{d}\alpha &= \alpha_1'-\alpha_3' & \int_{\alpha_1}^{\alpha_3} \cos\alpha\,\sin\alpha\,\sin\alpha\,\mathrm{d}\alpha &= \tfrac{1}{3}\left(\sin^3\alpha_3'-\sin^3\alpha_1'\right) \\
\int_{\alpha_\mathrm{m}}^{\alpha_\mathrm{n}} \cos\alpha\,\sin\alpha\,\mathrm{d}\alpha &= \tfrac{1}{2}\left(\sin^2\alpha_\mathrm{n}'-\sin^2\alpha_\mathrm{m}'\right) & \int_{\alpha_1}^{\alpha_2} \cos\alpha\,\cos\alpha\,\sin\alpha\,\mathrm{d}\alpha &= \tfrac{1}{3}\left(\cos^3\alpha_2'-\cos^3\alpha_1'\right)
\end{aligned} \tag{25}
$$

The driving moment (1) is expressed in α's:

$$
\frac{M_\mathrm{soil}}{\gamma_\mathrm{g} H\,R^2} = \frac{R}{3H}\left(\sin^3\alpha_3'-\sin^3\alpha_1'\right) - \frac{Y+H}{2H}\left(\sin^2\alpha_3'-\sin^2\alpha_1'\right) + \frac{R}{3L}\left(\cos^3\alpha_2'-\cos^3\alpha_1'\right) - \frac{L-X}{2L}\left(\cos^2\alpha_2'-\cos^2\alpha_1'\right) \tag{26}
$$

Or:

$$
\frac{M_\mathrm{soil}}{\gamma_\mathrm{g} H\,R^2 F} = \begin{array}{l} +\tfrac{1}{3}\left(\sin^2\alpha_3'+\sin\alpha_3'\sin\alpha_1'+\sin^2\alpha_1'\right)-\tfrac{1}{2}\sin\alpha_3'\left(\sin\alpha_3'+\sin\alpha_1'\right) \\ +\tfrac{1}{3}\left(\cos^2\alpha_2'+\cos\alpha_2'\cos\alpha_1'+\cos^2\alpha_1'\right)-\tfrac{1}{2}\cos\alpha_2'\left(\cos\alpha_2'+\cos\alpha_1'\right) \end{array} \tag{27}
$$

Or:

$$
\frac{M_\mathrm{soil}}{\gamma_\mathrm{g} H\,R^2 F} = \tfrac{1}{3}-\tfrac{1}{6}\sin\alpha_3'\left(\sin\alpha_3'+\sin\alpha_1'\right)-\tfrac{1}{6}\cos\alpha_2'\left(\cos\alpha_2'+\cos\alpha_1'\right) \tag{28}
$$

After equation (22) follows:

$$
\frac{M_\mathrm{soil}}{\gamma_\mathrm{g} H\,F} = \tfrac{1}{3}R^2-\tfrac{1}{6}\left(H+Y\right)\left(2H+2Y-H\,F\right)-\tfrac{1}{6}\left(L-X\right)\left(2L-2X-L\,F\right) \tag{29}
$$

The result will look like the previous section by expressing the radius in other geometric values:

$$
\begin{aligned}
\frac{M_\mathrm{soil}}{\tfrac{1}{2}\gamma_\mathrm{g} H\,F} &= R^2-\tfrac{1}{3}F^2\left(H^2+L^2\right)-\left(H+Y\right)\left(H+Y-H\,F\right)-\left(L-X\right)\left(L-X-L\,F\right) \\
R^2 &= \left(Y+H-H\,F\right)^2+\left(L-X-L\,F\right)^2
\end{aligned} \tag{19}
$$

And the safety factor for case 3 is derived:

$$F = \frac{M_{\text{reaction}}}{M_{\text{soil}}} = \frac{c}{\gamma_{\text{g}} H} \frac{1}{F} \frac{2\left(\alpha_2 - \alpha_0\right) R^2}{R^2 - \frac{1}{12} F^2 \left(H^2 + L^2\right) - \left(H + Y - \frac{1}{2} H F\right)^2 - \left(L - X - \frac{1}{2} L F\right)^2} \tag{20}$$

Deriving case 4

Case 4 is similar. Only area II is relevant. And both angles “α_1” and “α_2” become variables.

$$h_{II} = R \sin\left(\alpha\right) - Y - H + \frac{L - X + R\cos\left(\alpha\right)}{L} H \quad II \tag{21}$$

More…

4.1.2 Graphical representation of the simplified Bishop’s method

The previously derived function can be shown graphically, for example with a height of the embankment of 5 meters and a length of 15 meters. In Figure 4.4, on can se the slope and above the circles representing lines of an equal safety factor. There is a minimum halfway the slope at about 2 meters above the crest of the embankment.

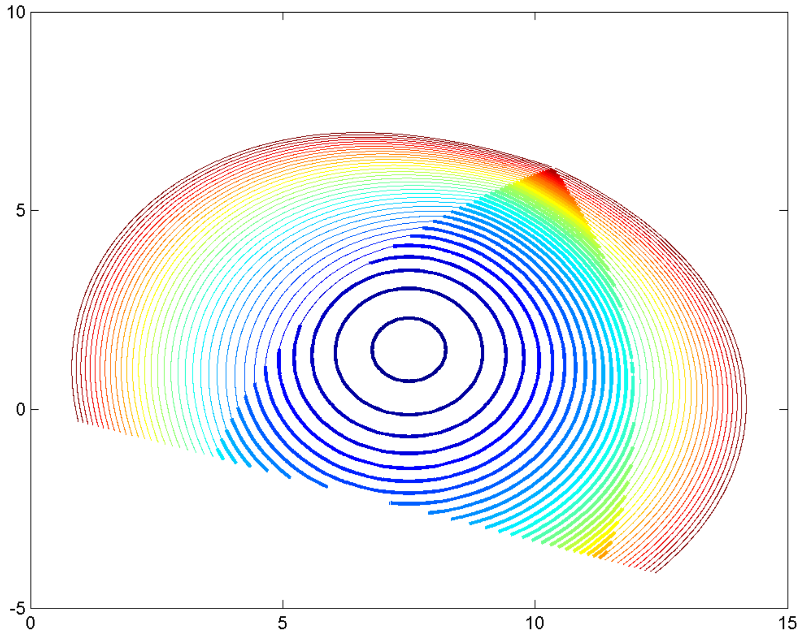


Figure 4.4 lines of equal stability factor above a slope if only circles of types 1, 2 and 3 are considered

The center points of the circles that enters through the crest and exit through the surface level (so called “case II” are printed bold. To the left side, there are points that are of type I, and to the right of type 2.

The shape becomes a little more complex if also circles of type 4 are considered because besides the global (circular) minimum, there is also an additional local minimum in the form a line, see Figure 4.5.

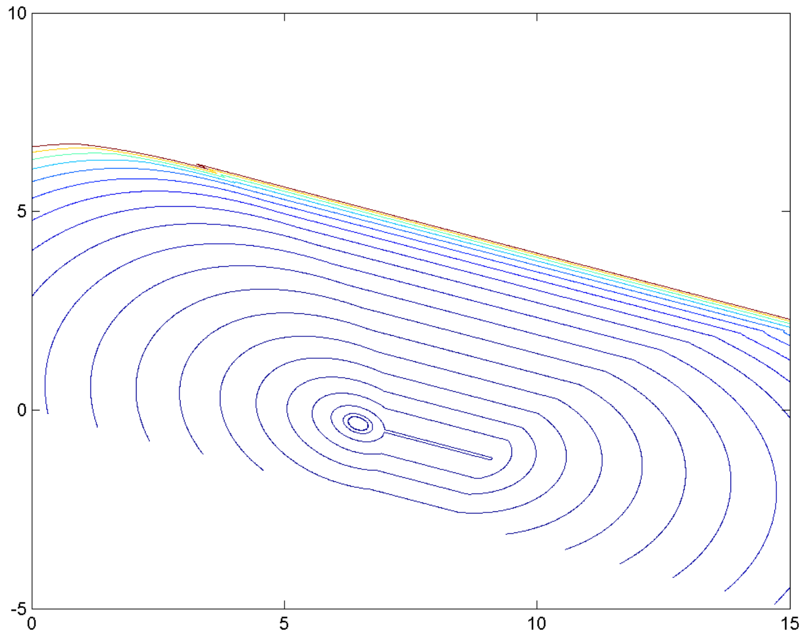


Figure 4.5 lines of equal stability factor above a slope if only circles of types 1, 2, 3 and 4 are considered

On top of the line minimum (which makes hill climbing difficult), there is a discontinuity at the left top of the figure. Figure 4.6 zooms in on this discontinuity. This discontinuity happens when a very thin slice passes through the point where the crest intersects the slope of the embankment.

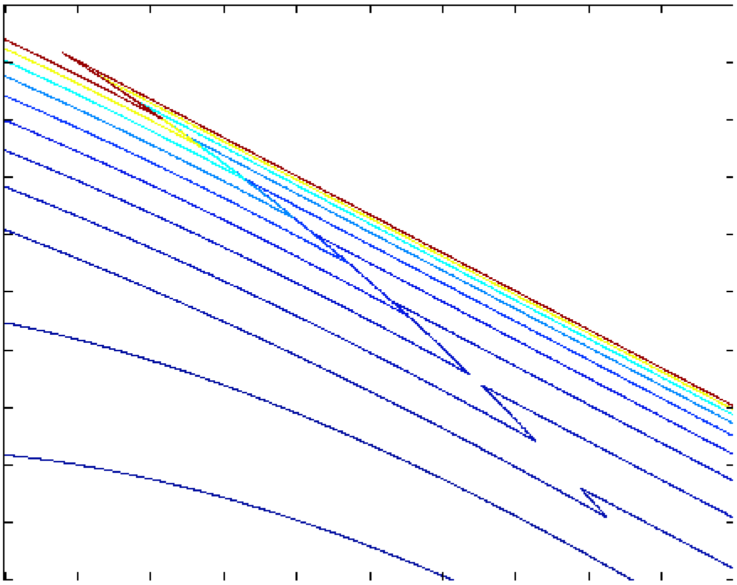


Figure 4.6 Discontinuities at the left top corner

4.1.3 Results from Bishop’s unit test

The GA programmed for this purpose as described in section 2.3 is run 100 times on Bishop’s method, as derived above. The population is set to 50 as is the number of generations. The average optimum value of the analysis is 2,779 with a standard deviation of 0,000171.

The average value can be assumed to be correct as it is the same value as given by the GA build in by Matlab.

4.2 Rastrigin’s function

4.2.1 The function

Rastrigin’s function is one with a large number of local maxima and local minima. The global minimum is at (0,0) and has a value of 0. For a search mechanism, it is very difficult to find the global minimum in this function because there are so many local minima. The function is defined in *(21)*.

$$Y(\overline{x}) = A\cdot n + \sum_{i=1}^{n} x_i^2 - A\cdot\cos(\omega\cdot x_i) \qquad (21)$$

$$A = 10;\ \omega = 2\cdot\pi;\ x_i \in [-5.12, 5.12]$$

The function is a combination of a parabola and a cosine function. The boundaries considered are between -5 and 5. In the unit test, the two dimensional case is used.

4.2.2 Graphical representation of the function

The function is drawn in Figure 4.7.

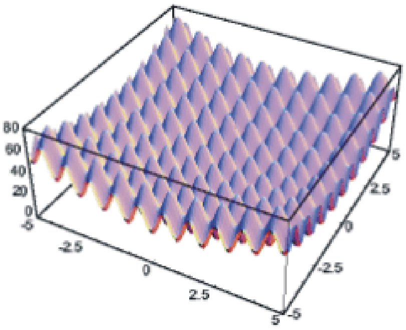


Figure 4.7 Rastrigin’s function

4.2.3 Results from Rastrigin’s unit test

The GA as presented in section 2.3 is run 100 times to see if the results are as desired.

When this test runs with a population of 50 individuals and 50 generations, the average value of the minimum is 0,128 and the standard deviation is 0,158. This error is too large compared to the one with Bishop's problem. In other words, a larger population is required and/ or more iteration steps need to be made to converge to the right minimum.

With a population of 100 individuals and 100 generations, the average value is 0,0073 and the standard deviation is 0,0099. This accuracy becomes close to level of accuracy in the previous Bishop's example.

The GA supplied by Matlab converges better to the optimum than the GA presented in this report. It is a future possibility to change the algorithm if more speed is required. The matlab algorithm uses more refined techniques than tournament selection and random selection of the parents. If the GA from section 2.3 needs to be quickened, such techniques should be implemented.

Comment [m1]: Dit sommetje moet nog gemaakt worden, maar hier twijfel ik niet aan…

4.3 Conclusions

Thus far, an analytical solution for a simple case of Bishop's method has been presented as well as a complex function with many local minima. The simplified analytical Bishop's function has proven to be rather complex with discontinuities and different types of minima, even for the simplest case.

Both functions can be minimized with a genetic algorithm. The first tests show that a population of 50 solutions and a total of 50 generations should be sufficient to find the global minimum in case of a stability problem with a rather smooth solution space of around 200 square metres.

Both functions presented can clearly be much better optimized with a genetic algorithm than a grid procedure. Implementation of the GA in MStab will lead to much faster calculation time and more accurate results.

5 The generic GA in MStab

6 Bishop

7 Uplift Van

8 Jambu

9 Free surface search

10 Conclusions