University of
Massachusetts
UMASS Lowell

22.520 NUMERICAL METHODS FOR PDE'S

Instructor: David J. Willis

Circular Foundations Deflection Analysis

Finite Element Method

Term Project

Report by: *Ana Clara R. Gouveia*

05/13/2015

Table of Contents

Project Summary

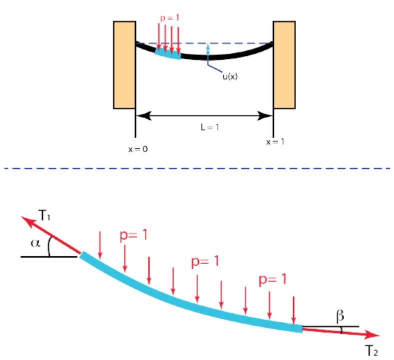
Circular plates are used in several kinds of structural engineering applications. In particular, the use of circular foundations is common practice within specialty construction such as wind turbines and nuclear power plants footings. Structural Engineers must go across several steps in order to efficiently analyze and design shallow foundations. In general the approach taken correlates soil bearing pressure to the foundation materials strength capacity and its supported load.

On the other hand, Geotechnical Engineers must assure that these foundations are safe against overall shear failure in the soil that supports them, and that they do not undergo excessive displacement, i.e. soil settlement. Within the field, the load per unit area of a foundation at which shear failure occurs is named ultimate bearing capacity.

This project aims to develop a comprehensive numerical model to determine the deflection of a circular footing foundation, further correlating results obtained with the method to current analytical solutions. In order to satisfy this project's requirements the numerical model developed for this project uses of polar coordinates to obtain the deflection of the structure in question.

Physical Problem

The deflection of a 1-D string.



Equilibrium in the x-direction (assuming small angles):

$$T_1 \cos(\alpha) = T_2 \cos(\beta) \simeq k\delta x \qquad (1)$$

Equilibrium in the y-direction:

$$T_1 \sin(\alpha) - T_2 \sin(\beta) - p\Delta x = 0 \qquad (2)$$

Divide the Equilibrium in y-direction by

$$\frac{T_1 \sin(\alpha)}{T_1 \cos(\alpha)} - \frac{T_2 \sin(\beta)}{T_2 \cos(\beta)} = \frac{p\Delta x}{k\delta x}$$

$$\tan(\alpha) - \tan(\beta) = \frac{p\Delta x}{k\delta x}$$

Recognizing, $\tan(\alpha)$ is the slope of the string

Taking the limit as $\Delta x \to 0$, we recover the derivative:

$$\frac{\partial^2 u}{\partial x^2} = \frac{p}{k\delta x}$$

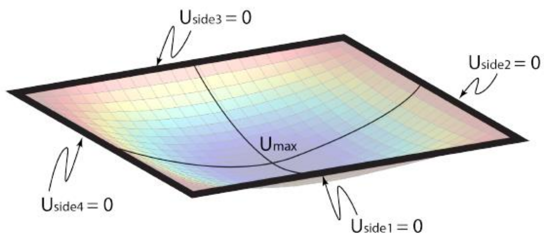
Setting $f = \frac{p}{k\delta x}$, we recover the Poisson equation:

$$\nabla^2 u = f$$

EQ (1)

Let's extend the string equation to a membrane:

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y) \qquad (1)$$



With boundary conditions on both ends of:

$$u_L = u_R = 0 \qquad (2)$$

In this project we'll solve the Laplace equation = f = qu, as the ultimate bearing capacity load caused by the footing.

In this problem it is assumed that the ends of this circular plate is clamped, thus boundary conditions to the problem are the traditional u = 0 at all boundary points.

Numerical Method

In the case of a circular foundation, the numerical approach taken in this project will consist of the development of a polar coordinates method for the problem solution. In a nutshell, this program would use of finite elements for structural analysis. The purpose of the program would consist of solving a 2-D FEM problem to determine the structural deformation of the structure, therefore solving the problem in 3-D. The non-Euclidean nature of the proposed mathematical approach had potentially present itself as a challenge in this project's development, however, the stress distributions obtained with the solution, see Figures 3 and 4, seem to agree with physical expectations.

Analytically, an alternative to determine the maximum deflection[1] occurring at r = 0 is then:

$$w_{\max} = \frac{Pa^2}{16\pi D} \qquad \text{EQ (2)}$$

Where,
P = load
a = area radius
$D = Eh^3 / [12(1 - v^2)]$
E = elastic modulus
h = plate thickness
v = Poisson's ratio

In this report, the numerical model validation is done using Boussinesq's Equation for Vertical Stress:

$$\sigma_z = \sigma_o \left[1 - \left(\frac{z}{\left(R^2 + z^2\right)^{1/2}} \right)^v \right] \qquad \text{EQ (3)}$$

Where,
σ = load
σ_o = area radius
z = depth below soil surface
R = foundation radius

Since this project's analysis is done in the elastic range, the equation used to obtain the soil settlement uses of the stress obtained with Boussinesq's (EQ. 3) as seen in EQ. (4).

Results

In order to design this project's numerical method the following steps were taken. Files associated to each one of these parts are attached to Appendix A of this project and in the virtually shared Matlab folder.

1. **Circular Discretization of Geometry:**

In this step, a for loop with r and theta in small increments was used. Using the Matlab special algorithm Delaunay, allowed for a creation of a fine mesh. Figure 1 represents the resulting meshed structure.

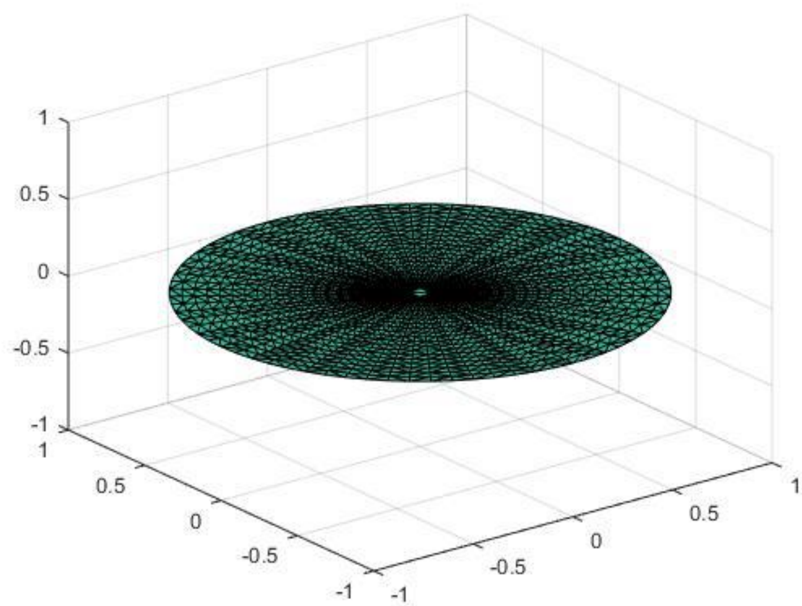


Figure 1. Meshed circular structure

2. **Finite Element Model of the circle**

In this model, instead of using an assumed bearing capacity, qu, the value of force on the domain used is equal to q = f = -1. The purpose of this value is to generate a unit gravitational force representation that can potentially further extrapolate this project results to specific applications, such as q = qu. The solution of the problem yielded a maximum deflection equal to δz = -0.2494.

The m.file associated to this part of the project can be found in Appendix A and shared Matlab folder. Figure 2 represent the finite element model of this membrane, whereas Figure 3 portrays the deflected shape of the membrane. Figure 4 show the plan view of the deflected structure.

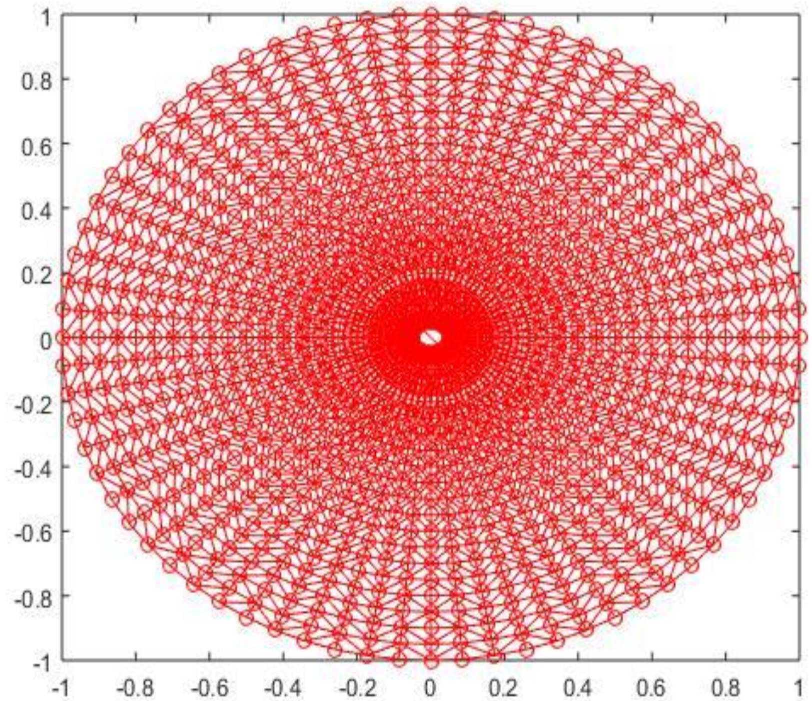


Figure 2. Finite Element Model Representation

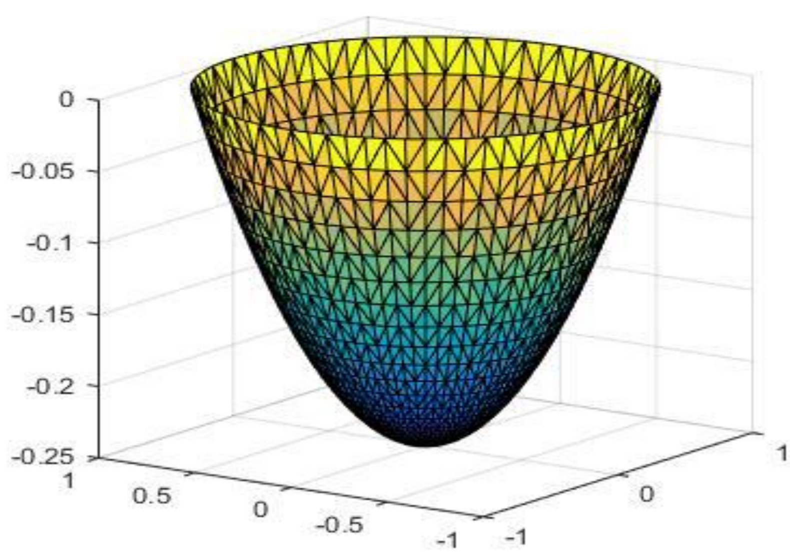


Figure 3. 3-D view of the deflected structure

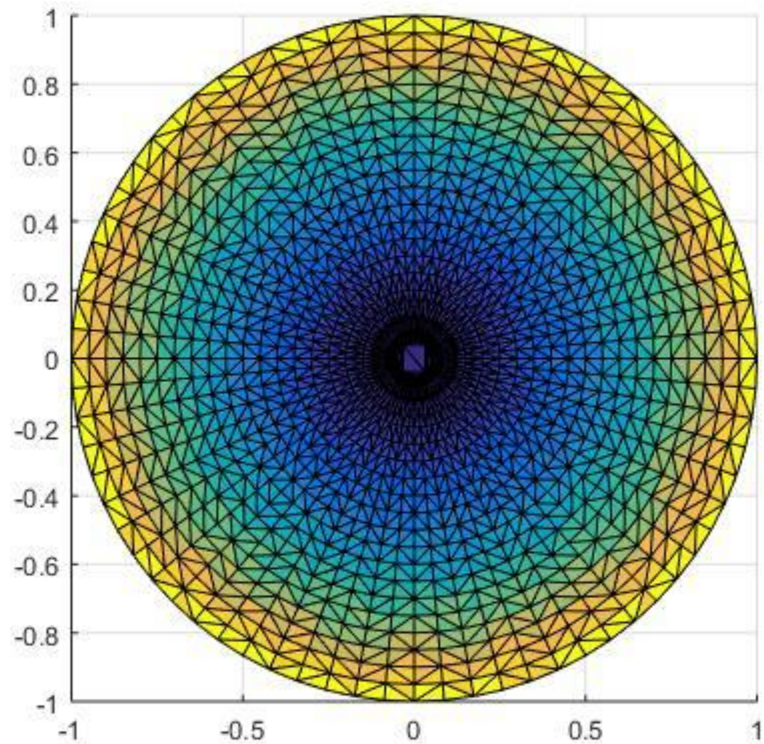


Figure 4. Plan view of stresses distribution in the structure

Validation

Current analytical solutions will be used in order to validate the numerical model developed.

Given the following variables:

Load Applied and Physical Characteristics:

Uniform Load Applied: $qo := 1$

Radius: $B := 2 \quad L := B$ Modulus of Elasticity: $E := 1$

Area: $A := \pi \frac{B}{4} = 1.571$ Soil Depth : $z := 1$

Soil Depth : $z := 1$ Soil Embedment Dept: $Df := 0$

Using the Boussinesq relationship, Das, Braja Eq. 5.3, for the increase in stress of a point z due to a loaded circular area:

$$\Delta\sigma := qo \cdot \left[1 - \left[\frac{1}{\left[1 + \left(\frac{B}{2 \cdot z} \right)^2 \right]^{\left(\frac{3}{2} \right)}} \right] \right] = 0.646 \qquad \text{EQ (3)}$$

By linear relationships, displacement δz in z is equal to:

$$\delta z := \Delta\sigma \cdot \frac{z}{E} = 0.646 \ \text{in} \qquad \text{EQ (4)}$$

By using of soil mechanics equations for soil settlement in the elastic range, we have:

If instead of looking at the pressure distribution we compare our numerical value to the estimated settlement based on the Theory of Elasticity, the displacement, i.e. settlement, of a circular area of soil can be obtained with Equation 5.33 from Das, of which the following terms are based on the settlement at the center of the foundation as:

Soil Properties:

Poisson's Ratio: $\mu s := 0.3$

Additional Variables:

$\alpha := 4 \qquad Bp := \frac{B}{2} = 1 \qquad pa := 2000 \quad \frac{lb}{ft^2}$

Values of F1 and F2 can be obtained by using equations mp and np below, by using Table 5.8 and 5.9 in pages 248-251 in the book Foundation Engineering by Das.

$mp := \frac{L}{B} = 1 \qquad np := 2 \cdot \frac{z}{B} = 1 \qquad F1 := 0.408 \qquad F2 := 0.037$

Shape Factor, Is: $Is := F1 + \left[\frac{(1 - 2\mu s)}{(1 - \mu s)}\right] \cdot F2 = 0.429$

The depth factor can be obtained through Table 5.10 in Das:

$F2b := \frac{Df}{B} = 0 \qquad F2c := \frac{B}{L} = 1$

Consequently: $If := .65$

The soil modulus of elasticity can be determined using Eq. 5.42 from Das. Let's assume it is equal to 1 at this point, and apply Das, Braja Eq. 5.33:

$$Se2 := qo \cdot \alpha \cdot Bp \cdot Is \cdot If \cdot \frac{\left(1 - \mu s^2\right)}{Es} = 1.015 \quad in \qquad \text{EQ (6)}$$

From the value obtained with numerical model:

$$\Delta z := -0.2494 \quad ft \qquad \text{EQ (7)}$$

Error Consideration

The values for deflection and soil settlement obtained with equations 4 through 6 in this report were further compared to this project's results, yielding the errors in Table 1.

Numerical Solution: 0.2494 ft = 2.99 in

Table 1. Alternative Methods Results and Error, based on the assumption that the numerical model is the correct answer, and Error2 based on the assumption that the results obtained are the correct answer:

Ref.	Alternative Methods	Results	Error	Error2
Eq 4	Boussinesq (1885)	0.646	0.784	-3.633
EQ 5	Janbu et all (1956) Modified by Christian and Carrier (1978)	0.5	0.833	-4.986
EQ 6	Bowles 1987	1.015	0.661	-1.949

It is necessary to note that the following list of assumptions may have caused the difference between solutions:

- Soil Modulus of Elasticity equal to 1.
- Possible agreement and understanding of units used.
- Need to expand numerical model to accommodate appropriate energy losses, hence developing a more adequate model with Poisson's equation.

References

Das, Braja. *Principles of Foundation Engineering*. Cengage Learning, 2011.

Faraji, Susan and Jerome Connor. *Principles of Structural Engineering*. New York: Springer, 2013.

Lindfield, George and John Penny. *Numerical Methods Using Matlab*. Upper Saddle River: Prentice Hall, 1999.

Appendix A: Matlab Files

Circular Discretization:

```
%CircularDiscretization:

clear all
counter = 1;

for(r = 0.05:0.05:1)
    for(theta = 0:5:355)
        x(counter) = r*cosd(theta);
        y(counter) = r*sind(theta);
        counter = counter + 1;
    end
end

Connectivity = delaunay(x,y);
Vertices = [x',y']

figure
trisurf(Connectivity, x,y,x*0)
```

Deflection Solver:

```
% Finite elements on circle

% discretizing the domain
Vertices
Connectivity

NNodes = length(Vertices);
NElem = length(Connectivity);

% Initialize A and f
A = zeros(NNodes, NNodes);%spalloc(NNodes, NNodes,6*NNodes);
F = zeros(1, NNodes)

% Setup the A-matrix & RHS
for(i=1:NElem)
        % Figure out what the triangle is!!!!!
        ConnectedNodes = Connectivity(i,:);

        V1 = Vertices(ConnectedNodes(1),:);
        V2 = Vertices(ConnectedNodes(2),:);
        V3 = Vertices(ConnectedNodes(3),:);

        plot([V1(1) V2(1) V3(1) V1(1)], [V1(2) V2(2) V3(2) V1(2)],'r-o')

        %pause
        hold on

       % First calculate the area -- Oh No!!!!!!!!!!!!!!!!!!!!!
```

Ana Gouveia
01304683

```
        L1 = [V2-V1,0];
        L2 = [V3-V1,0];
        Area = norm(1/2*cross( L1, L2));

        % Solve the matrix of basis function nodes
        Plane_Mat = [V1(1) V1(2) 1; V2(1) V2(2) 1; V3(1) V3(2) 1];
         RHS1 = [1 0 0]';
         RHS2 = [0 1 0]';
         RHS3 = [0 0 1]';

         abc1 = Plane_Mat\RHS1;
         abc2 = Plane_Mat\RHS2;
         abc3 = Plane_Mat\RHS3;

         a1 = abc1(1);
         b1 = abc1(2);
         c1 = abc1(3);

         a2 = abc2(1);
         b2 = abc2(2);
         c2 = abc2(3);

         a3 = abc3(1);
         b3 = abc3(2);
         c3 = abc3(3);

         % Interactions between basis functions (gradient * gradient)
         ElementMatrix = Area*[a1*a1 + b1*b1  a2*a1 + b2*b1  a3*a1 + b3*b1;
                               a1*a2 + b1*b2  a2*a2 + b2*b2  a3*a2 + b3*b2;
                               a1*a3 + b1*b3  a2*a3 + b2*b3  a3*a3 + b3*b3];

         CN = ConnectedNodes;
         A(CN(1),CN(1)) = A(CN(1),CN(1))  + ElementMatrix(1,1);
         A(CN(1),CN(2)) = A(CN(1),CN(2))  + ElementMatrix(1,2);
         A(CN(1),CN(3)) = A(CN(1),CN(3))  + ElementMatrix(1,3);

         A(CN(2),CN(1))  = A(CN(2),CN(1))  + ElementMatrix(2,1);
         A(CN(2),CN(2))  = A(CN(2),CN(2))  + ElementMatrix(2,2);
         A(CN(2),CN(3))  = A(CN(2),CN(3))  + ElementMatrix(2,3);

         A(CN(3),CN(1))  = A(CN(3),CN(1))  + ElementMatrix(3,1);
         A(CN(3),CN(2))  = A(CN(3),CN(2))  + ElementMatrix(3,2);
         A(CN(3),CN(3))  = A(CN(3),CN(3))  + ElementMatrix(3,3);

         F(CN(1)) = F(CN(1))+Area/3;
         F(CN(2)) = F(CN(2))+Area/3;
         F(CN(3)) = F(CN(3))+Area/3;
end

% Boundary conditions
for(i = 1:NNodes)
    V1 = Vertices(i,1);
    V2 = Vertices(i,2);
```

```
    if( (V1^2 + V2^2)>.99)
        A(i,:) = 0;
        A(i,i) = 1;
        F(i) = 0;
    end

end

Solution = A\-F';

figure
trisurf(Connectivity, Vertices(:,1), Vertices(:,2), Solution)
```