UCL Mechanical Engineering 2020/2021

ENGF0004 Coursework 2

NCWT3

April 16, 2021

1 Question 1

1.1 a

For the line integral to be independent from the path of integration, the following conditions must be fulfilled:

$$I = \int_A^B \left(\frac{\partial u}{\partial x} \, \mathrm{d}x + \frac{\partial u}{\partial y} \, \mathrm{d}y \right) \tag{1.1}$$

$$P\left(x,\, y\right) = \frac{\partial u}{\partial x} \text{ and } Q\left(x,\, y\right) = \frac{\partial u}{\partial y} \tag{1.2}$$

$$\frac{\partial P\left(x,\, y\right)}{\partial y} = \frac{\partial Q\left(x,\, y\right)}{\partial x} \tag{1.3}$$

Considering the integral:

$$I = \int_A^B \left[e^{-\alpha x y} \left(\frac{\alpha - 2}{x} \right) \mathrm{d}x - \frac{1}{\alpha y} \left(e^{-\alpha x y} - 1 \right) \mathrm{d}y \right] \tag{1.4}$$

$$P\left(x,\, y\right) = e^{-\alpha x y} \left(\frac{\alpha - 2}{x} \right) \text{ and } Q\left(x,\, y\right) = -\frac{1}{\alpha y} \left(e^{-\alpha x y} - 1 \right) \tag{1.5}$$

$$\frac{\partial P\left(x,\, y\right)}{\partial y} = -\alpha x \left(\frac{\alpha - 2}{x} \right) e^{-\alpha x y} = \left(2\alpha - \alpha^2 \right) e^{-\alpha x y} \tag{1.6}$$

$$\frac{\partial Q\left(x,\, y\right)}{\partial x} = -\frac{1}{\alpha y} \left(-\alpha y \right) e^{-\alpha x y} = e^{-\alpha x y} \tag{1.7}$$

$$\therefore 2\alpha e^{-\alpha x y} - \alpha^2 e^{-\alpha x y} = e^{-\alpha x y} \tag{1.8}$$

$$e^{-\alpha x y} \left(\alpha^2 - 2\alpha + 1 \right) = 0 \tag{1.9}$$

$$e^{-\alpha x y} = 0 \rightarrow \text{no solutions} \tag{1.10}$$

$$\left(\alpha - 1 \right)^2 = 0 \tag{1.11}$$

$$\alpha = 1 \tag{1.12}$$

1.2 b

Calculating the line integral of 1.13 from $O\left(0,\, 0\right)$ to $A(1,\, e-1)$ along $y = e^x - 1$:

$$I = \int_O^A \left(ye^{-2x}\right)(\mathrm{d}x + \mathrm{d}y) \tag{1.13}$$

$$y = e^x - 1 \tag{1.14}$$

$$\mathrm{d}y = e^x\,\mathrm{d}x \tag{1.15}$$

$$I = \int_0^1 \left((e^x - 1)\left(e^{-2x}\right) + (e^x - 1)\left(e^{-2x}\right)(e^x)\right)\mathrm{d}x \tag{1.16}$$

$$= \int_0^1 \left(e^{-x} - e^{-x} - e^{-2x} + 1\right)\mathrm{d}x \tag{1.17}$$

$$= \int_0^1 \left(1 - e^{-2x}\right)\mathrm{d}x \tag{1.18}$$

$$= \left[x + \frac{e^{-2x}}{2}\right]_0^1 \tag{1.19}$$

$$= 1 + \frac{e^{-2}}{2} - 0 - \frac{1}{2} \tag{1.20}$$

$$I = \frac{1}{2}\left(e^{-2} + 1\right) \tag{1.21}$$

1.3 c

1.3.1 i

$$\underline{F}\left(x,\, y,\, z\right) = \begin{pmatrix} \frac{y}{x^2} \\ \frac{x}{y^2} \end{pmatrix} \tag{1.22}$$

$$\nabla \cdot \underline{F} = \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix} \cdot \begin{pmatrix} \frac{y}{x^2} \\ \frac{x}{y^2} \end{pmatrix} \tag{1.23}$$

$$= \frac{\partial}{\partial x}\left(\frac{y}{x^2}\right) + \frac{\partial}{\partial y}\left(\frac{x}{y^2}\right) \tag{1.24}$$

$$= -\frac{2y}{x^3} - \frac{2x}{y^3} \tag{1.25}$$

$$= -2\left(\frac{y}{x^3} + \frac{x}{y^3}\right) \tag{1.26}$$

1.3.2 ii

$$I = \int_1^2 \int_1^2 \left(-2 \left(\frac{y}{x^3} + \frac{x}{y^3} \right) \right) \mathrm{d}x \, \mathrm{d}y \tag{1.27}$$

$$= \int_1^2 \left[-2 \left(\frac{y}{-2x^2} + \frac{x^2}{2y^3} \right) \right]_1^2 \mathrm{d}y \tag{1.28}$$

$$= \int_1^2 \left[-2 \left(-\frac{y}{8} + \frac{2}{y^3} + \frac{y}{2} - \frac{1}{2y^3} \right) \right] \mathrm{d}y \tag{1.29}$$

$$= \int_1^2 \left(-\frac{3y}{4} - \frac{3}{y^3} \right) \mathrm{d}y \tag{1.30}$$

$$= \left[-\frac{3y^2}{8} + \frac{3}{2y^2} \right]_1^2 \tag{1.31}$$

$$= -\frac{3}{2} + \frac{3}{8} + \frac{3}{8} - \frac{3}{2} \tag{1.32}$$

$$I = -\frac{9}{4} \tag{1.33}$$

1.4 d

1.4.1 i

$$I = \int (\sin x \cos y \, \mathrm{d}y + \cos x \sin y \, \mathrm{d}y) \tag{1.34}$$

$$y = 0 \qquad \mathrm{d}y = 0 \tag{1.35}$$

$$I_{AB} = \int_{x=0}^{\pi} (\sin x) \, \mathrm{d}x = [-\cos x]_0^{\pi} = 2 \tag{1.36}$$

$$x = \pi \qquad \mathrm{d}x = 0 \tag{1.37}$$

$$I_{BC} = \int_{y=0}^{\pi} (-\sin y) \, \mathrm{d}y = [\cos y]_0^{\pi} = -2 \tag{1.38}$$

$$\therefore I = I_{AB} + I_{BC} = 2 - 2 = 0 \tag{1.39}$$

1.4.2 ii

$$I = \int (\sin x \cos y \, \mathrm{d}y + \cos x \sin y \, \mathrm{d}y) \tag{1.40}$$

$$y = x \qquad \mathrm{d}y = \mathrm{d}x \tag{1.41}$$

$$I_{AC} = \int_0^{\pi} (\sin x \cos x + \sin x \cos x) \, \mathrm{d}x \tag{1.42}$$

$$= \int_0^{\pi} \big(\sin (2x) \big) \, \mathrm{d}x \tag{1.43}$$

$$I_{AC} = \left[-\frac{1}{2} \cos (2x) \right]_0^{\pi} = \frac{1}{2} - \frac{1}{2} = 0 \tag{1.44}$$

$$\tag{1.45}$$

1.5 e

1.5.1 i

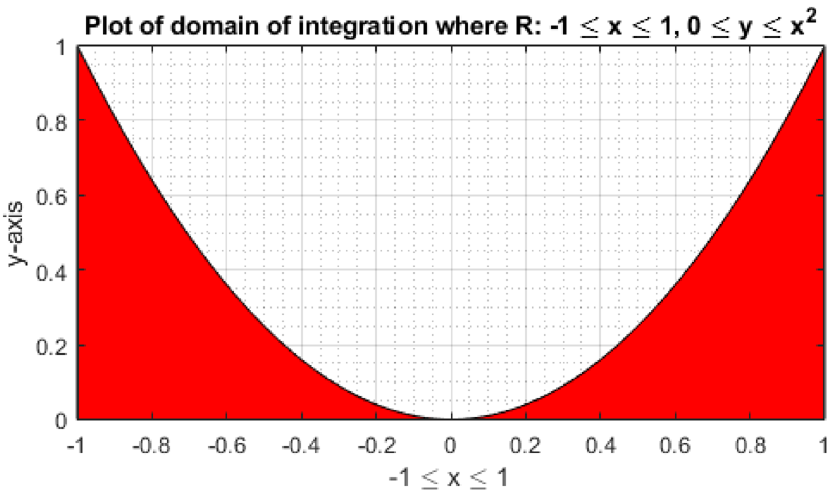

Figure 1: Domain of integration where $R:\ -1 \leq x \leq 1,\ 0 \leq y \leq x^2$.

```
clf
x=-1:0.001:1;
y=x.^2;
A=area(x,y);
set(A(1),'FaceColor','red');
axis('image');
xlabel('-1 \leq x \leq 1')
ylabel('y-axis')
title('Plot of domain of integration where R: -1 \leq x \leq 1, 0 \leq y
    \leq x^2')
grid on
grid minor
```

1.5.2 ii

1.6 f

1.6.1 ii

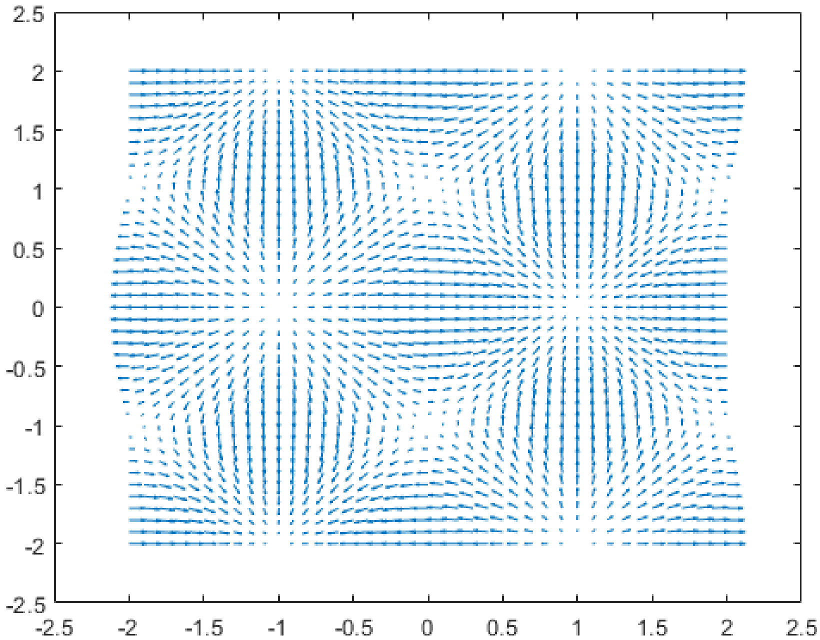

Figure 2:

```
clc
clear
close all

syms x y
z = x*y*exp(-sqrt(x^2 + y^2));
f = (sin((pi/2)*x))*(cos((pi/2)*y));
g = gradient(f,[x,y]);

[X, Y] = meshgrid(-2:0.1:2,-2:0.1:2);
G1 = subs(g(1),[x y],{X,Y});
G2 = subs(g(2),[x y],{X,Y});
quiver(X,Y,G1,G2)
```

2 Question 2

2.1 a

In our series of equations, there are three unknown internal bar forces N_{12}, N_{23}, N_{13}, and three unknown reaction forces, R_{2x}, R_{2y}, R_{3y}. We also have two unknown angles, α and β, and the force F. Given that there are six unknowns that we would like to find and six equations with those variables, the conditions are fulfilled to solve this using matrices. Our answer would be in terms of the variables α, β and F. Values may be assumed for these or we can calculate them, if we have the length of each member.

2.2 b

$$\begin{bmatrix} -\cos\alpha & \cos\beta & 0 & 0 & 0 & 0 \\ -\sin\alpha & -\sin\beta & 0 & 0 & 0 & 0 \\ \cos\alpha & 1 & 0 & 1 & 0 & 0 \\ \sin\alpha & 0 & 0 & 0 & 1 & 0 \\ 0 & -\cos\beta & -1 & 0 & 0 & 0 \\ 0 & \sin\beta & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} N_{12} \\ N_{13} \\ N_{23} \\ R_{2x} \\ R_{2y} \\ R_{3y} \end{bmatrix} = \begin{bmatrix} 0 \\ F \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.1}$$

2.3 c

```
clc
clear
close all

alpha = 0.927295;
beta = 0.643501;
F = 1000;

A = [-cos(alpha) cos(beta) 0 0 0 0;
    -sin(alpha) -sin(beta) 0 0 0 0;
     cos(alpha) 0 1 1 0 0;
     sin(alpha) 0 0 0 1 0;
     0  -cos(beta) -1 0 0 0;
     0 sin(beta) 0 0 0 1];
B = [0; F; 0; 0; 0; 0];

sol = A\B;
```

This returned the following:

$$\begin{bmatrix} N_{12} \\ N_{13} \\ N_{23} \\ R_{2x} \\ R_{2y} \\ R_{3y} \end{bmatrix} = \begin{bmatrix} -800 \\ -600 \\ 480 \\ 0 \\ 640 \\ 360 \end{bmatrix} \tag{2.2}$$

2.4 d

```
clc
clear
close all
```

```matlab

alpha = 0.927295;
beta = 0.643501;
F = 1000;

A = [-cos(alpha) cos(beta) 0 0 0 0;
    -sin(alpha) -sin(beta) 0 0 0 0;
    cos(alpha) 0 1 1 0 0;
    sin(alpha) 0 0 0 1 0;
    0  -cos(beta) -1 0 0 0;
    0 sin(beta) 0 0 0 1];
B = [0; F; 0; 0; 0; 0];

[L,U] = lu(A); %splits matrix A such that A = L*U, L is lower triangular, U is upper triangular
y = L\B;
sol = U\y;
```

This returned the following:

$$\begin{bmatrix} N_{12} \\ N_{13} \\ N_{23} \\ R_{2x} \\ R_{2y} \\ R_{3y} \end{bmatrix} = \begin{bmatrix} -800 \\ -600 \\ 480 \\ 0 \\ 640 \\ 360 \end{bmatrix} \tag{2.3}$$

2.5 e

Matlab App Developer was utilised to create a user friendly interface for inputting the Force F, the lengths of each member (as shown in the diagram) and the coefficient matrix (where any mathematical expression can be inputted). The GUI displays the angles α and β as well as a table of values for each of the internal bar and reaction forces. The code is shown below.

```matlab
classdef q2eApp_exported < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure            matlab.ui.Figure
        ForceNLabel         matlab.ui.control.Label
        Force               matlab.ui.control.NumericEditField
        L12mLabel           matlab.ui.control.Label
        L12                 matlab.ui.control.NumericEditField
        L23mLabel           matlab.ui.control.Label
        L23                 matlab.ui.control.NumericEditField
        L13mLabel           matlab.ui.control.Label
        L13                 matlab.ui.control.NumericEditField
        UITable             matlab.ui.control.Table
        FindForcesButton    matlab.ui.control.Button
        betaGaugeLabel      matlab.ui.control.Label
        betaGauge           matlab.ui.control.NinetyDegreeGauge
        alphaGaugeLabel     matlab.ui.control.Label
```

```
        alphaGauge          matlab.ui.control.NinetyDegreeGauge
        ProgrammetocalculateforcesLabel   matlab.ui.control.Label
        UITable2            matlab.ui.control.Table
        Label               matlab.ui.control.Label
    end

    % Callbacks that handle component events
    methods (Access = private)

        % Code that executes after component creation
        function startupFcn(app)
            %initialise table
            ATable = ["-cos(alpha)" "cos(beta)" "0" "0" "0" "0";
                "-sin(alpha)" "-sin(beta)" "0" "0" "0" "0";
                "cos(alpha)" "0" "1" "1" "0" "0";
                "sin(alpha)" "0" "0" "0" "1" "0";
                "0" "-cos(beta)" "-1" "0" "0" "0";
                "0" "sin(beta)" "0" "0" "0" "1"];
            %display table and assign table properties
            set(app.UITable2, 'Visible', 'on');
            set(app.UITable2, 'Data', ATable, 'ColumnFormat',{'char'});
            set(app.UITable2,'ColumnEditable',true(1,6))
        end

        % Button pushed function: FindForcesButton
        function FindForcesButtonPushed(app, event)
            %calculate alpha and beta
            alpha = acos((app.L12.Value^2 + app.L23.Value^2 - app.L13.
                Value^2)/(2*app.L12.Value*app.L23.Value));
            beta = acos((app.L13.Value^2 + app.L23.Value^2 - app.L12.
                Value^2)/(2*app.L13.Value*app.L23.Value));

            %conversion for display gauges
            app.alphaGauge.Value = rad2deg(alpha);
            app.betaGauge.Value = rad2deg(beta);

            %matrix maths
            A = get(app.UITable2, 'Data');
            %convert user inputs into expressions and evaluate
            c = size(A);
            c = c(1)*c(2);
            for i = 1:c
                A(i) = eval(A(i));
            end
            A = str2double(A);
            B = [0; app.Force.Value; 0; 0; 0; 0];
            sol = A\B;
            for i = 1:length(B)
                if sol(i) < 0.01 && sol(i) > -0.01
                    sol(i) = 0;
                end
            end
            namesForces = ["L12";"L13";"L23";"R2x";"R2y";"R3y"];
```

```
            vars = [namesForces sol];

            %output to table
            set(app.UITable, 'Visible', 'on');
            set(app.UITable, 'Data', vars, 'ColumnFormat',{'numeric'});

        end
    end

    % Component initialization
    methods (Access = private)

        % Create UIFigure and components
        function createComponents(app)

            % Create UIFigure and hide until all components are created
            app.UIFigure = uifigure('Visible', 'off');
            app.UIFigure.Position = [100 100 762 598];
            app.UIFigure.Name = 'MATLAB App';

            % Create ForceNLabel
            app.ForceNLabel = uilabel(app.UIFigure);
            app.ForceNLabel.HorizontalAlignment = 'right';
            app.ForceNLabel.Position = [32 403 56 22];
            app.ForceNLabel.Text = 'Force (N)';

            % Create Force
            app.Force = uieditfield(app.UIFigure, 'numeric');
            app.Force.Position = [103 403 100 22];

            % Create L12mLabel
            app.L12mLabel = uilabel(app.UIFigure);
            app.L12mLabel.HorizontalAlignment = 'right';
            app.L12mLabel.Position = [41 370 47 22];
            app.L12mLabel.Text = 'L12 (m)';

            % Create L12
            app.L12 = uieditfield(app.UIFigure, 'numeric');
            app.L12.Position = [103 370 100 22];

            % Create L23mLabel
            app.L23mLabel = uilabel(app.UIFigure);
            app.L23mLabel.HorizontalAlignment = 'right';
            app.L23mLabel.Position = [41 349 47 22];
            app.L23mLabel.Text = 'L23 (m)';

            % Create L23
            app.L23 = uieditfield(app.UIFigure, 'numeric');
            app.L23.Position = [103 349 100 22];

            % Create L13mLabel
            app.L13mLabel = uilabel(app.UIFigure);
            app.L13mLabel.HorizontalAlignment = 'right';
```

```
            app.L13mLabel.Position = [41 328 47 22];
            app.L13mLabel.Text = 'L13 (m)';

            % Create L13
            app.L13 = uieditfield(app.UIFigure, 'numeric');
            app.L13.Position = [103 328 100 22];

            % Create UITable
            app.UITable = uitable(app.UIFigure);
            app.UITable.ColumnName = {'Force'; 'Value (N)'};
            app.UITable.RowName = {};
            app.UITable.Position = [272 59 479 185];

            % Create FindForcesButton
            app.FindForcesButton = uibutton(app.UIFigure, 'push');
            app.FindForcesButton.ButtonPushedFcn = createCallbackFcn(app, @FindForcesButtonPushed, true);
            app.FindForcesButton.Position = [103 292 100 22];
            app.FindForcesButton.Text = 'Find Forces';

            % Create betaGaugeLabel
            app.betaGaugeLabel = uilabel(app.UIFigure);
            app.betaGaugeLabel.HorizontalAlignment = 'center';
            app.betaGaugeLabel.Position = [186 117 29 22];
            app.betaGaugeLabel.Text = 'beta';

            % Create betaGauge
            app.betaGauge = uigauge(app.UIFigure, 'ninetydegree');
            app.betaGauge.Limits = [0 90];
            app.betaGauge.Position = [154 154 90 90];

            % Create alphaGaugeLabel
            app.alphaGaugeLabel = uilabel(app.UIFigure);
            app.alphaGaugeLabel.HorizontalAlignment = 'center';
            app.alphaGaugeLabel.Position = [62 117 35 22];
            app.alphaGaugeLabel.Text = 'alpha';

            % Create alphaGauge
            app.alphaGauge = uigauge(app.UIFigure, 'ninetydegree');
            app.alphaGauge.Limits = [0 90];
            app.alphaGauge.Orientation = 'northeast';
            app.alphaGauge.ScaleDirection = 'counterclockwise';
            app.alphaGauge.Position = [34 154 90 90];

            % Create ProgrammetocalculateforcesLabel
            app.ProgrammetocalculateforcesLabel = uilabel(app.UIFigure);
            app.ProgrammetocalculateforcesLabel.HorizontalAlignment = 'right';
            app.ProgrammetocalculateforcesLabel.FontSize = 20;
            app.ProgrammetocalculateforcesLabel.FontWeight = 'bold';
            app.ProgrammetocalculateforcesLabel.Position = [441 534 306 56];
```

```
            app.ProgrammetocalculateforcesLabel.Text = 'Programme to calculate forces';

            % Create UITable2
            app.UITable2 = uitable(app.UIFigure);
            app.UITable2.ColumnName = {'L12'; 'L13'; 'L23'; 'R2x'; 'R2y'; 'R3y'};
            app.UITable2.RowName = {};
            app.UITable2.ColumnEditable = true;
            app.UITable2.Position = [272 264 479 193];

            % Create Label
            app.Label = uilabel(app.UIFigure);
            app.Label.HorizontalAlignment = 'right';
            app.Label.Position = [202 479 545 56];
            app.Label.Text = {'Please input the force F, the lengths of the members L12, L23 and L13.'; 'The programme will then calculate the values of alpha and beta and display them to you.'; 'If you would like to change the coeffcient matrix, look to the table on the right and adjust as you like.'; 'Click "Find Forces" to calculate the values of the internal bar forces and the reaction forces.'};

            % Show the figure after all components are created
            app.UIFigure.Visible = 'on';
        end
    end

    % App creation and deletion
    methods (Access = public)

        % Construct app
        function app = q2eApp_exported

            % Create UIFigure and components
            createComponents(app)

            % Register the app with App Designer
            registerApp(app, app.UIFigure)

            % Execute the startup function
            runStartupFcn(app, @startupFcn)

            if nargout == 0
                clear app
            end
        end

        % Code that executes before app deletion
        function delete(app)

            % Delete UIFigure when app is deleted
            delete(app.UIFigure)
```

```
        end
    end
end
```

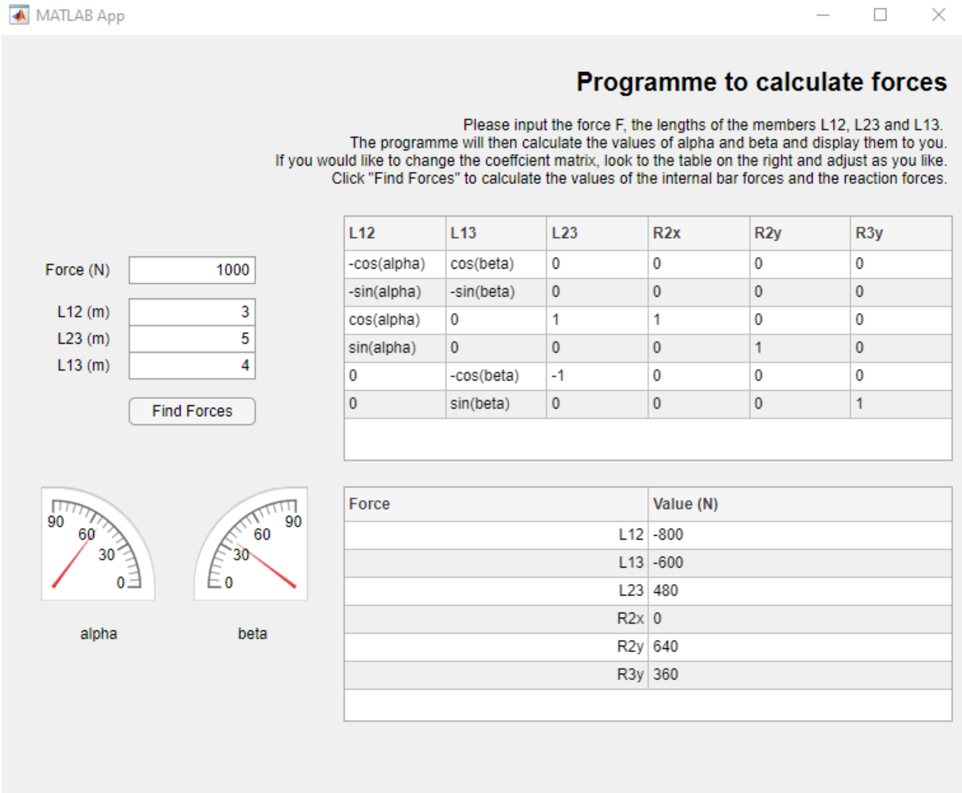

Figure 3: Screenshot from Matlab App, showcasing GUI, input and output parameters.

2.6 f

Code was written to generate a table of data:

```
clc
clear
close all
%forces
F = [1000 3000 4500];

%lengths of members
L12 = [6 8 5];
L23 = [10 12 8];
L13 = [9 7 4];

%initalise matrix
sol = zeros(9,10);

%initalise counter
counter = 0;

%nested loops, iterates between F and then between A1, A2, A3 and stores
    in sol matrix
```

```
for i = 1:3
    for j = 1:3
        %calculate alpha and beta
        alpha = acos((L12(j)^2 + L23(j)^2 - L13(j)^2)/(2*L12(j)*L23(j)));
        beta = acos((L13(j)^2 + L23(j)^2 - L12(j)^2)/(2*L13(j)*L23(j)));

        %calculate A and B matrices
        A = [-cos(alpha) cos(beta) 0 0 0 0;
             -sin(alpha) -sin(beta) 0 0 0 0;
             cos(alpha) 0 1 1 0 0;
             sin(alpha) 0 0 0 1 0;
             0 -cos(beta) -1 0 0 0;
             0 sin(beta) 0 0 0 1];
        B = [0; F(i); 0; 0; 0; 0];

        %generate result
        temp = (A\B)';

        %increment counter
        counter = counter + 1;

        %store result
        sol(counter, 5:10) = temp;
    end
end

%table formatting
sol(:,1) = repelem(F',3,1);
sol(:,2) = repmat(L12',3,1);
sol(:,3) = repmat(L13',3,1);
sol(:,4) = repmat(L23',3,1);

%swap L13 and L23 columns
v = sol(:, 7);
sol(:, 7) = sol(:, 6);
sol(:, 6) = v;

%clean up values
for i=1:numel(sol)
    if sol(i) < 0.01 && sol(i) > -0.01
        sol(i) = 0;
    end
end

%table generation
T = array2table(sol);
T.Properties.VariableNames = {'Force', 'L12', 'L23', 'L13', 'N12', 'N13',
    'N23', 'R2x', 'R2y', 'R3y'};
```

	1 Force	2 L12	3 L23	4 L13	5 N12	6 N13	7 N23	8 R2x	9 R2y	10 R3y
1	1000	6	9	10	-815.7246	373.8738	-464.1192	0	725	275.0000
2	1000	8	7	12	-799.0757	661.7346	-861.7938	0	447.9167	552.0833
3	1000	5	4	8	-1.0504e+03	958.4751	-1.1153e+03	0	429.6875	570.3125
4	3000	6	9	10	-2.4472e+03	1.1216e+03	-1.3924e+03	0	2175	825.0000
5	3000	8	7	12	-2.3972e+03	1.9852e+03	-2.5854e+03	0	1.3438e+03	1.6562e+03
6	3000	5	4	8	-3.1512e+03	2.8754e+03	-3.3459e+03	0	1.2891e+03	1.7109e+03
7	4500	6	9	10	-3.6708e+03	1.6824e+03	-2.0885e+03	0	3.2625e+03	1.2375e+03
8	4500	8	7	12	-3.5958e+03	2.9778e+03	-3.8781e+03	0	2.0156e+03	2.4844e+03
9	4500	5	4	8	-4.7267e+03	4.3131e+03	-5.0189e+03	0	1.9336e+03	2.5664e+03

Table 1: Table of data generated from MATALB, showing forces in three configuration with three different loads.

Force (N)	L12 (m)	L13	L23	N13 (N)	N23	N13	R2x	R2y	R3y
1000	6	9	10	-815.7	373.9	-464.1	0	725.0	275.0
1000	8	7	12	-799.1	661.7	-861.8	0	447.9	552.1
1000	5	4	8	-1050.4	958.5	-1115.3	0	429.7	570.3
3000	6	9	10	-2447.2	1121.6	-1392.4	0	2175.0	825.0
3000	8	7	12	-2397.2	1985.2	-2585.4	0	1343.8	1656.3
3000	5	4	8	-3151.2	2875.4	-3346.0	0	1289.1	1710.9
4500	6	9	10	-3670.8	1682.4	-2088.5	0	3262.5	1237.5
4500	8	7	12	-3595.8	2977.8	-3878.1	0	2015.6	2484.4
4500	5	4	8	-4726.7	4313.1	-5018.9	0	1933.6	2566.4

Table 2: Table formatted as requested in question.