Report on

Assignment 3: Tracking Solutions for Non-Linear BVP using Analytical and Arc-Length Continuation

Submitted in partial fulfillment of the requirements of
ENM 502
Numerical Methods and Modeling
by

Jalaj Maheshwari
M.S.E., Mechanical Engineering and Applied Mechanics

University of Pennsylvania
School of Engineering and Applied Sciences

24th March, 2016

Table of Contents

Table of Figures

1. Introduction

The problem statement of this particular project deals with solving the non-linear boundary value problem represented by Equation 1 using Newton's method and tracking the non-trivial solutions for the range $0 \le \lambda \le 60,\ \varepsilon = 0$ using analytical continuation and for the case where $0 \le \lambda \le 60,\ \varepsilon = 1$ using arc-length continuation.

$$\nabla^2 u + \lambda u(1+u) = \varepsilon \sin(\pi x) \tag{1a}$$

$$u(x,y) = 0 \text{ on all boundaries } (\partial D) \tag{1b}$$

$$D = (0 \le x \le 1) \cup (0 \le y \le 1) \tag{1c}$$

The equation and boundary conditions are first discretized using a centered finite difference approximation. The MATLAB code for generating a set of coupled, non-linear algebraic equations of the form $\underline{R}(\underline{u}) = \underline{0}$ is written. The equation is discretized using a 30x30 uniform finite difference grid.

Test cases are run for both values of ε for the given range of λ to generate curves of the non-trivial solutions to the boundary value problem, first using analytic continuation method corresponding to $\varepsilon = 0$ and using arc length continuation for $\varepsilon = 1$.

These different techniques used to solve a non-linear boundary value problem are looked upon in this project. Newton's method is incorporated in these techniques to track the various non-trivial solutions for the given range of λ.

2. Problem Setup and Formulation

The boundary value problem represented by Equation 1 needs to be discretized before it can be solved further using either of the techniques. This is done using the centered finite difference approximation. The domain is divided into smaller finite elements with the numbering shown as in Figure 1.

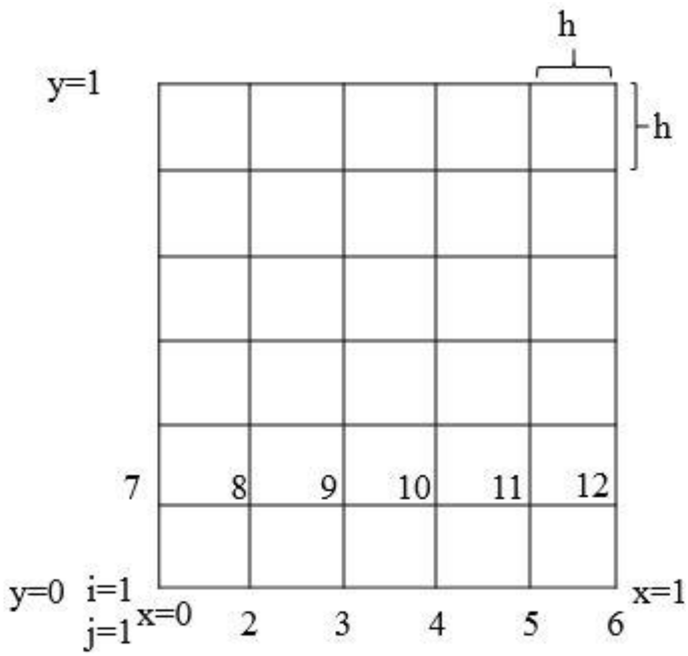


Fig 1: Computational grid generated over domain

Starting from the bottom left corner, the grid points are numbered in the order shown. Here, h_x and h_y represent the grid spacing in the *x* and *y* directions. The boundary value problem described as Equation 1(a) can be solved by generate a set of coupled, non-linear algebraic equations of the form:

$$\underline{R}(\underline{u}) = \underline{0} \tag{2}$$

Thus, for the given boundary value problem:

$$\underline{R}(\underline{u}) \Rightarrow \nabla^2 u + \lambda u(1+u) - \varepsilon \sin(\pi x) = 0 \tag{3a}$$

$$\text{and } \underline{R}(\underline{u}) = u(x, y) \text{ on all boundaries } (\partial D) \tag{3b}$$

An initial guess for the value of *u* is taken. The Jacobian for the problem needs to be evaluated in order to proceed with Newton's method which is found by:

$$J_{ij} = \frac{\partial R_i}{\partial u_j} \tag{4}$$

For Newton's method, there are two steps that are involved in computing the next value:

$$\underline{\underline{J}}\,|_{u_k}\ \delta \underline{u}^k = -\underline{R}(\underline{u_k}) \tag{5a}$$

$$\underline{u^{k+1}} = \underline{u}^k + \delta \underline{u}^k \tag{5b}$$

The new value obtained gets us on the curve of the non-trivial solution. Using analytical or arc-length continuation, by varying the value of λ, we can then trace the curve of the non-trivial solutions.

For analytical continuation, the algorithm is as follows:

- Equation 3a is solved at a given value $\lambda = \lambda_0$ using Newton's method.
- An initial guess is made at $\lambda = \lambda_0 + \delta\lambda$

$$\therefore \underline{u}(\lambda) = \underline{u}(\lambda_0) + \frac{\partial \underline{u}}{\partial \lambda} \cdot \delta + 0(\delta^2) \tag{6}$$

- $\dfrac{\partial \underline{u}}{\partial \lambda}$ is computed as a solution to:

$$\underline{\underline{J}}\Big|_{\lambda_0, u_0} \frac{\partial \underline{u}}{\partial \lambda} = \frac{-\partial R}{\partial \lambda}\Big|_{\lambda_0, u_0} \tag{7}$$

- The next value of u is given as:

$$\underline{u}_2^0 = \underline{u}_1 + \frac{\partial \underline{u}}{\partial \lambda}\Big|_{u_1} (\lambda_2 - \lambda_1) \tag{8}$$

- The Newton's method is used again for this new value to get the point on the curve again.

Using this iterative procedure, the whole curve can be tracked by giving an initial value of λ and then plotting for the given range of λ.

For Arc-Length continuation, the algorithm is as follows:

- An initial guess for $\underline{u}_0$ *at* $\lambda = \lambda_0$ is used.
- Using Newton's Method, the converged solution at $\underline{u}_0$ for $\lambda = \lambda_0$ is obtained.
- Once the solution at λ_0 is obtained, ALC is used to obtain the solution as a function of λ.
- Compute:

$$\eta(s, \lambda, \underline{u}) = |s - s_0|^2 - \|\underline{u}(s) - \underline{u}(s_0)\|^2 - |\lambda(s) - \lambda(s_0)|^2 \tag{9}$$

- Using this, the augmented Jacobian and augmented residual are obtained:

$$\hat{\underline{\underline{J}}}\Big|_0 = \begin{pmatrix} \dfrac{\partial \underline{R}}{\partial \underline{u}}\Big|_0 & \dfrac{\partial \underline{R}}{\partial \lambda}\Big|_0 \\ \dfrac{\partial \eta}{\partial \underline{u}}\Big|_0 & \dfrac{\partial \eta}{\partial \lambda}\Big|_0 \end{pmatrix} \tag{10a}$$

$$\hat{\underline{R}} = \begin{pmatrix} R \\ \eta \end{pmatrix} \tag{10b}$$

- $\left(\frac{\partial \underline{u}}{\partial s}\right)_0, \left(\frac{\partial \lambda}{\partial s}\right)_0$ can be evaluated by solving for:

$$\left.\hat{\underline{\underline{J}}}\right|_0 \begin{pmatrix} \frac{\partial \underline{u}}{\partial s} \\ \frac{\partial \lambda}{\partial s} \end{pmatrix}_0 = -\left(\frac{\partial \hat{\underline{R}}}{\partial s}\right)_0 \tag{11}$$

- The initial guess for $\lambda_1, \underline{u}_1$ using ALC is thus, given by:

$$\underline{u}_1^0 = \underline{u}_0 + (\delta s)\left(\frac{\partial \underline{u}}{\partial s}\right)_0 \tag{12a}$$

$$\lambda_1^0 = \lambda_0 + (\delta s)\left(\frac{\partial \lambda}{\partial s}\right)_0 \tag{12b}$$

where, $\delta s \equiv s_1 - s_0$ is the arc-length of a small segment of a curve defined by:

$$(\delta S)^2 = (\delta \lambda)^2 + \|\delta \underline{u}\|_2^2 \tag{13}$$

- Once we obtain the initial guess at point $s = s_1, \underline{u} = \underline{u}_1, \lambda = \lambda_1$, we need to iterate using the Newton's Method. For the augmented system, the equation for the Full Newton will be:

$$\left.\left(\hat{\underline{\underline{J}}}^k\right)\right|_S \left.\begin{pmatrix} \delta \underline{u}^k \\ \\ \delta \lambda^k \end{pmatrix}\right|_S = -\left.\left(\hat{\underline{R}}^k\right)\right|_S \tag{14a}$$

$$\lambda^{k+1} = \lambda^k + \delta \lambda^k \tag{14b}$$

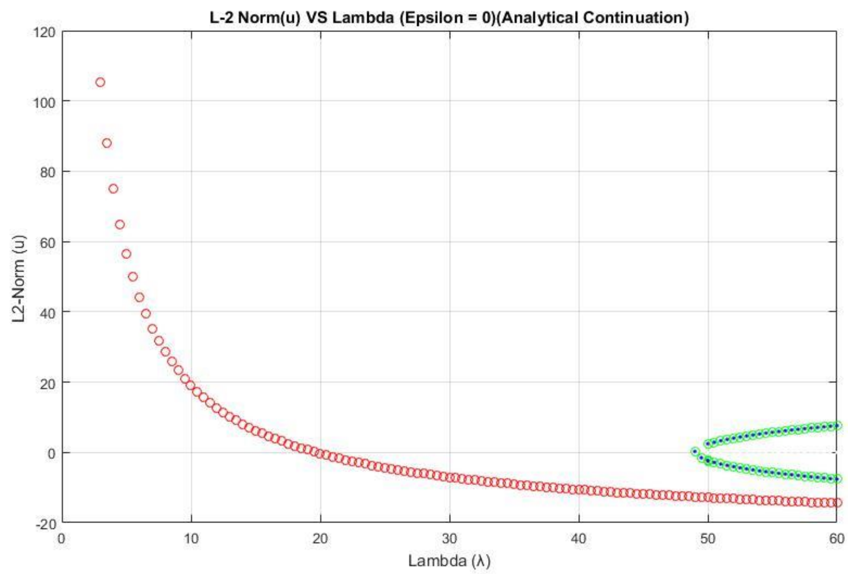
$$\underline{u}^{k+1} = \underline{u}^k + \delta\, \underline{u}^k \tag{14c}$$

This completes a full cycle of the arc-length continuation scheme and further stepping can be achieved by repeating the above sequence of steps.

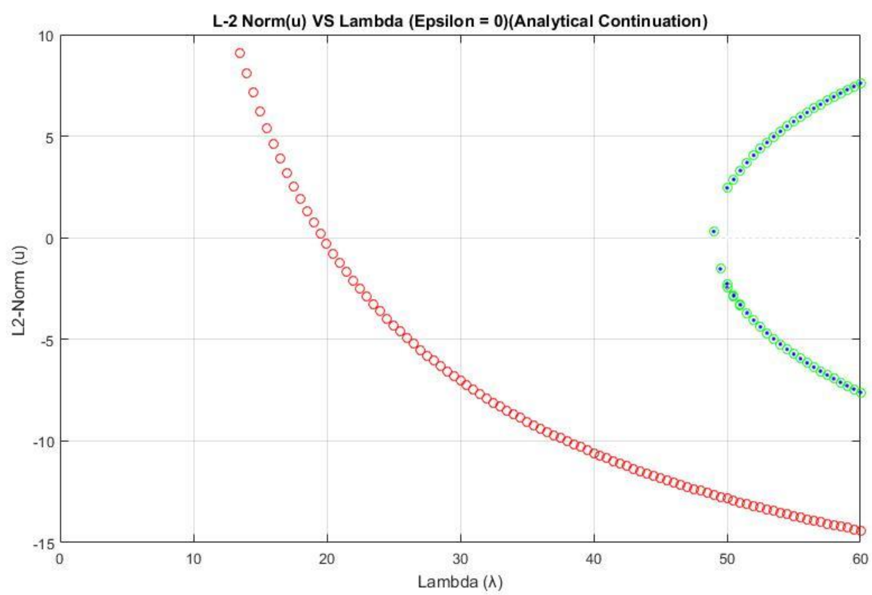
3. Results and Discussion

3.1 Curve Tracking Using Analytical Continuation for ε = 0

For the first part of the project, the various non-trivial solutions to the boundary value problem are tracked using analytical continuation. An initial approximation for λ and the solution is made, which is then used to obtain the point on the non-trivial solution curve by applying Newton's method. Once the initial value on the curve is obtained, the rest of the curve is tracked by changing the value of λ and applying analytical continuation in λ. The curves tracked are shown in Figure 2 for different initial values of λ for the complete range of λ.



(a)



(b)

Fig. 2: L-2 Norm(u) versus Lambda ($\varepsilon = 0$) for analytical continuation (a) Normal view, (b) Zoomed view

For making intelligent guesses for the value of λ where the procedure will not give a zero solution, we relate the solution of the boundary value problem to the solution of the linear problem:

$$\nabla^2 u + \lambda u = 0 \tag{15}$$

which makes it an eigenvalue-eigenfunction problem. The solution to this problem comes to be:

$$u(x,y) = A\sin(m\pi x)\sin(n\pi y) \tag{16a}$$

$$\text{where, } m^2\pi^2 + n^2\pi^2 = \lambda \tag{16b}$$

Thus, solutions only exist for discrete values of λ . For different values of *m* and *n*, the curves will be obtained for starting values $2\pi^2$ and $5\pi^2$. The value of $5\pi^2$ is obtained for *m=1, 2* and *n=2, 1* respectively. Hence, we will get two curves overlapping each other. The norm of *u* is a squared quantity and hence will never have negative values. However, to show the curve, it is assumed that on passing the x-axis, the curve keeps continuing in negative norm.

Figure 2 shows the non-trivial solutions corresponding to analytical continuation for $\varepsilon = 0$. We obtain three branches. The tracked curves shown are plotted for $2\pi^2$ and $5\pi^2$. The two $5\pi^2$ curves overlap each other which are shown by different markers and colors to differentiate between the two. Here, the negative sub-branches represent bowl-like solutions while the positive sub-branches represent hill-like solutions.

3.2 Curve Tracking Using Analytical Continuation for ε = 1

The same boundary value problem is solved using the arc-length continuation method as described above. In the case where multiple solutions exist for a given λ , this method is more useful in tracking those solutions. The curves tracked are shown in Figure 3 for different initial values of λ for the complete range of λ .

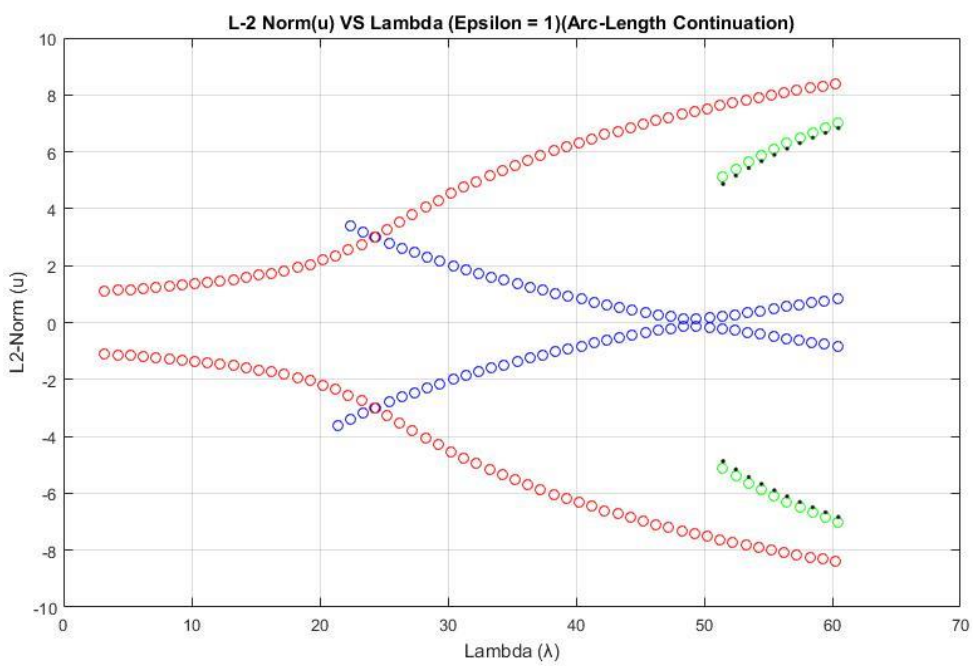
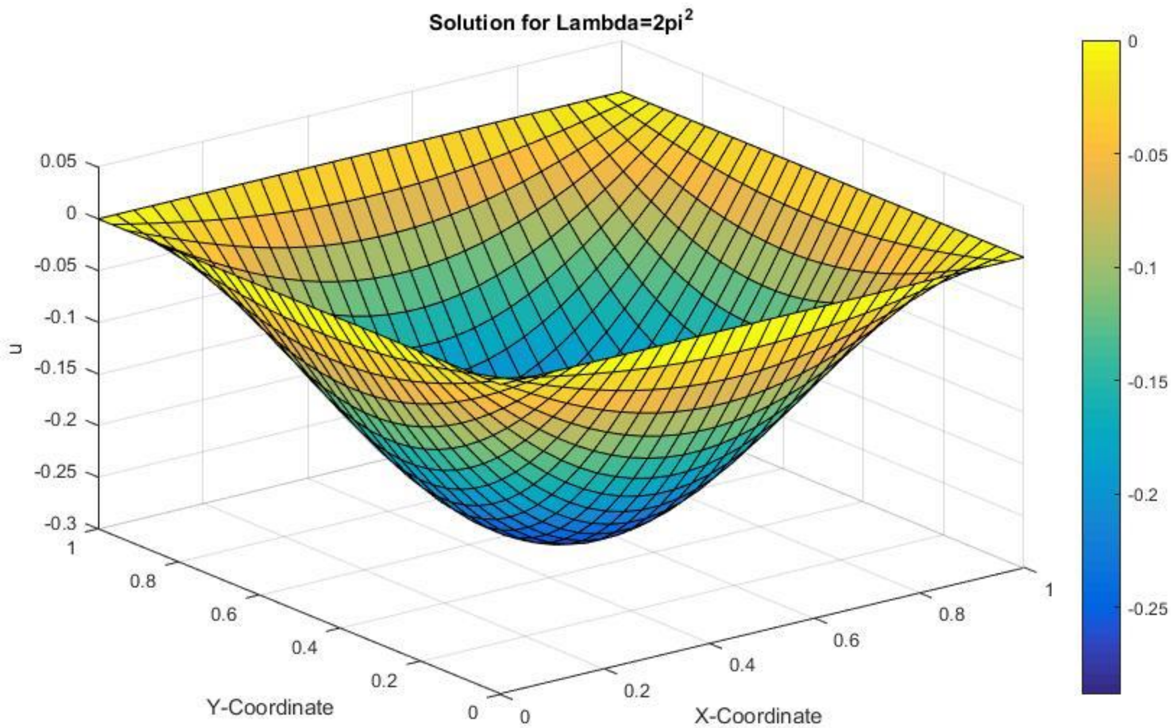


Fig. 3: L-2 Norm(u) versus Lambda ($\varepsilon = 0$) for arc-length continuation

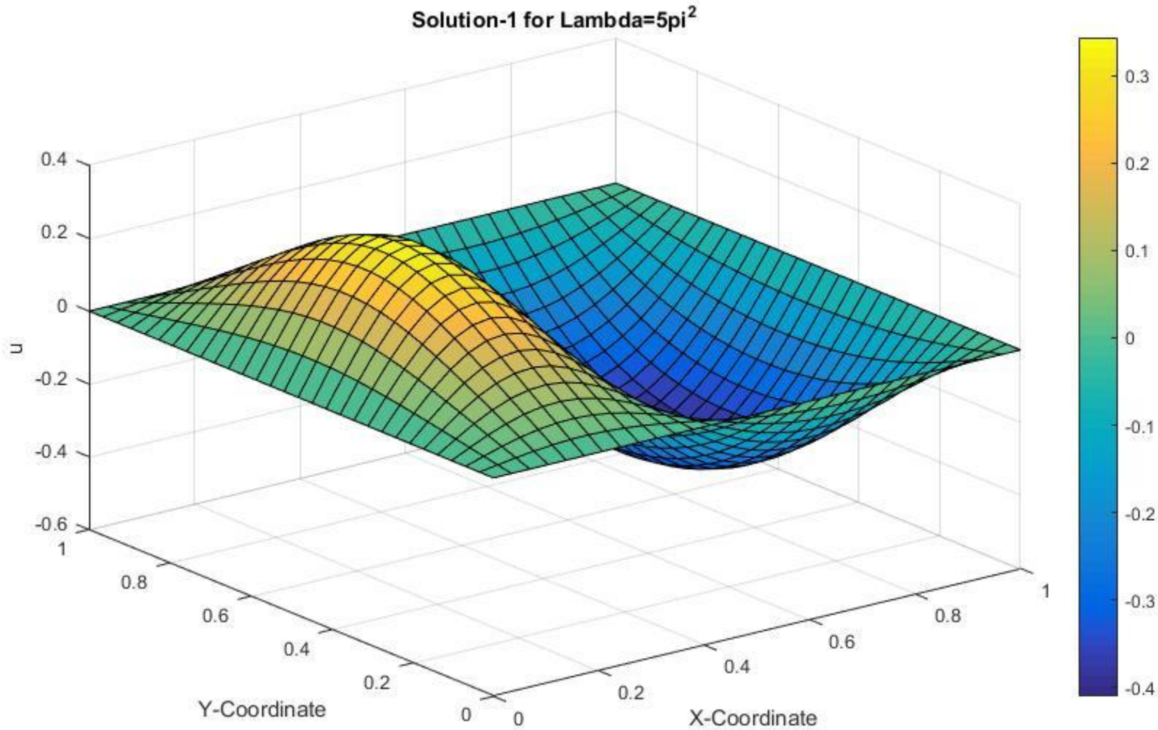
It can be noticed that unlike the analytical continuation curves, the arc-length continuation curves do not pass the x-axis at any point as zero is never a solution due to the weighting function. Thus, it may be affirmed that the graphs obtained using the method of arc-length continuation are correct. Similar to analytical continuation, the curves are obtained by taking initial guesses for λ as well as the length of the arc which is used to trace the curve, *ds*. Further, it may be seen how these curves represent hills or bowls corresponding to the value of λ chosen. In this whole procedure, only the initial λ is taken as an input. Continuation in both *u* and λ is applied to obtain further values along the curve. The subsequent values of λ are governed by the length of the arc, *ds*. We obtain four branches in this case.

3.3 Variation in Contours with Solution Branches

To get a better understanding regarding the change of solution along each branch, the 3D-contours corresponding to analytical continuation and arc-length continuation are plotted and represented in Figure 4.

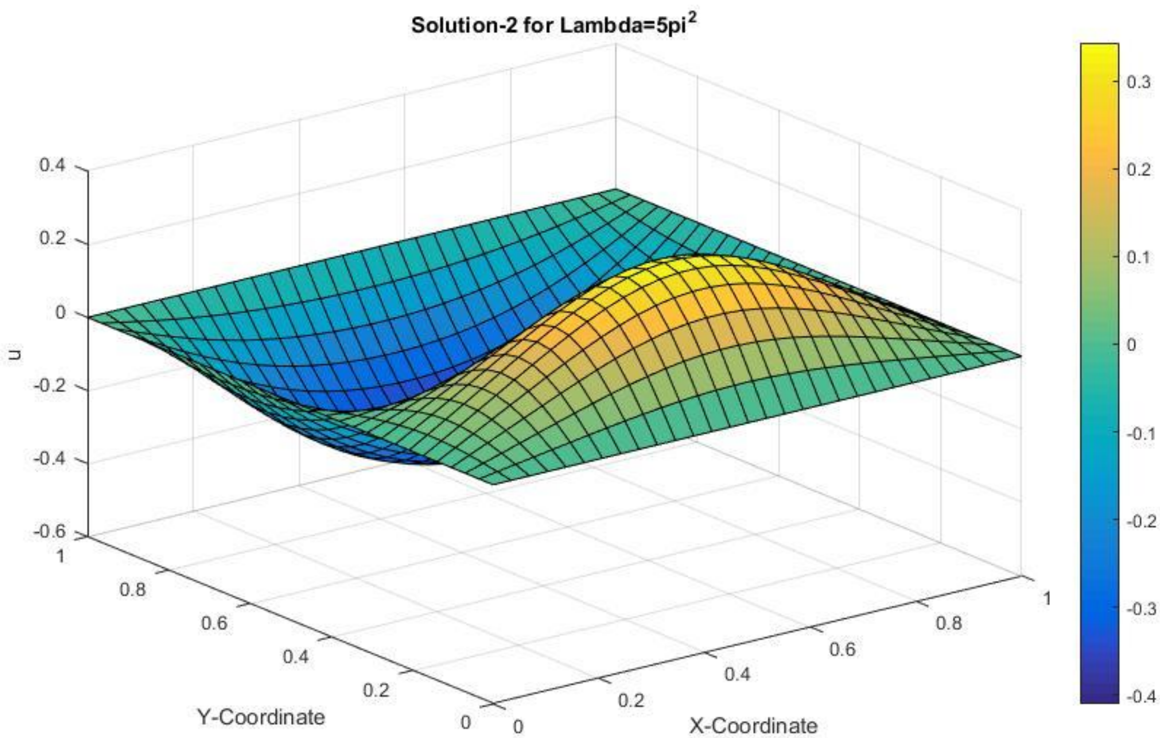


(a)



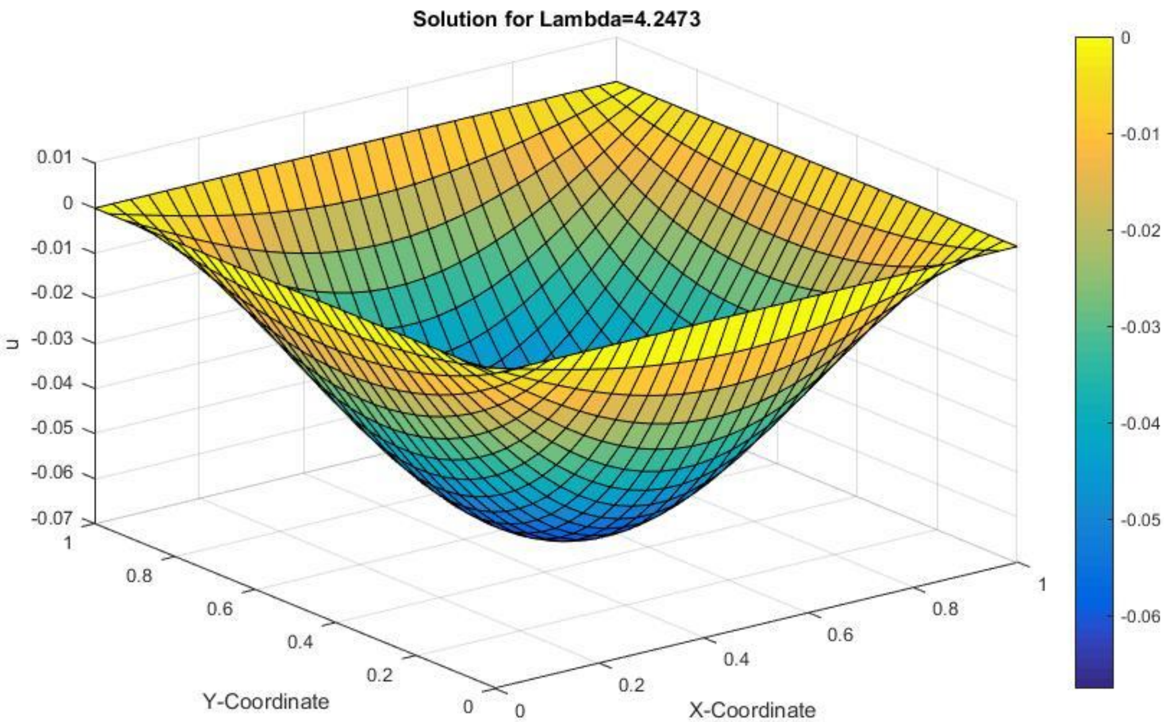
Solution-1 for Lambda=5pi^2
u
Y-Coordinate
X-Coordinate

(b)

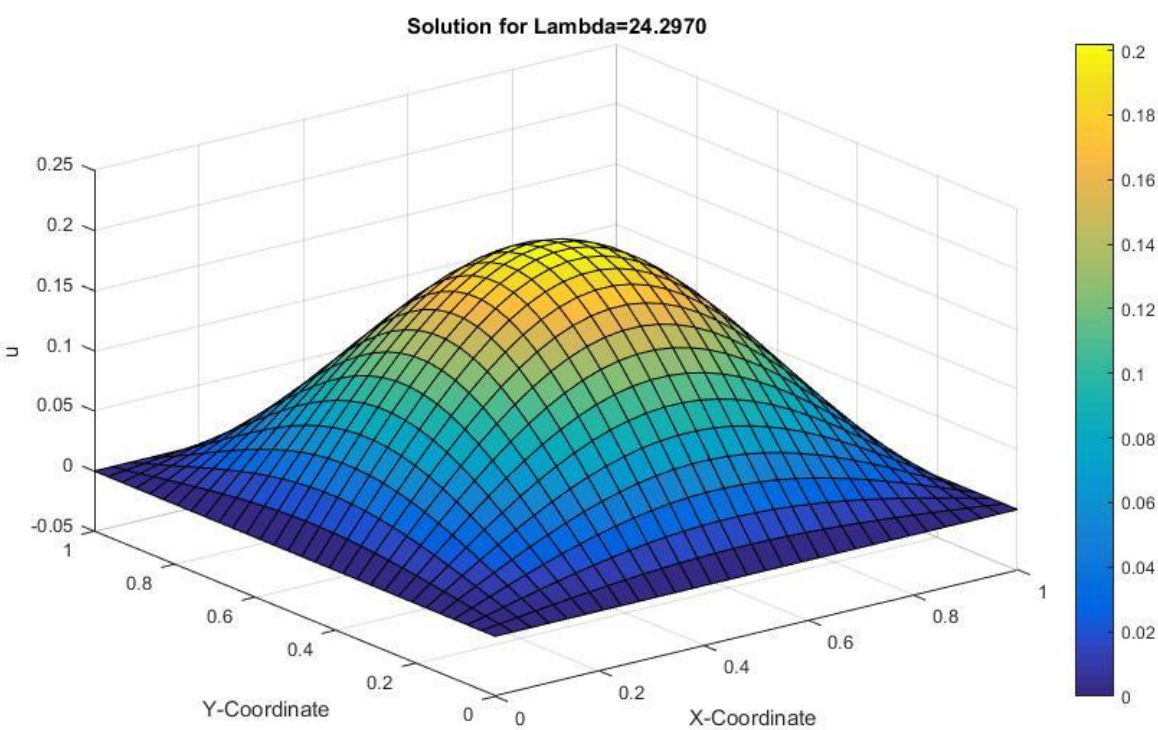


Solution-2 for Lambda=5pi^2
u
Y-Coordinate
X-Coordinate

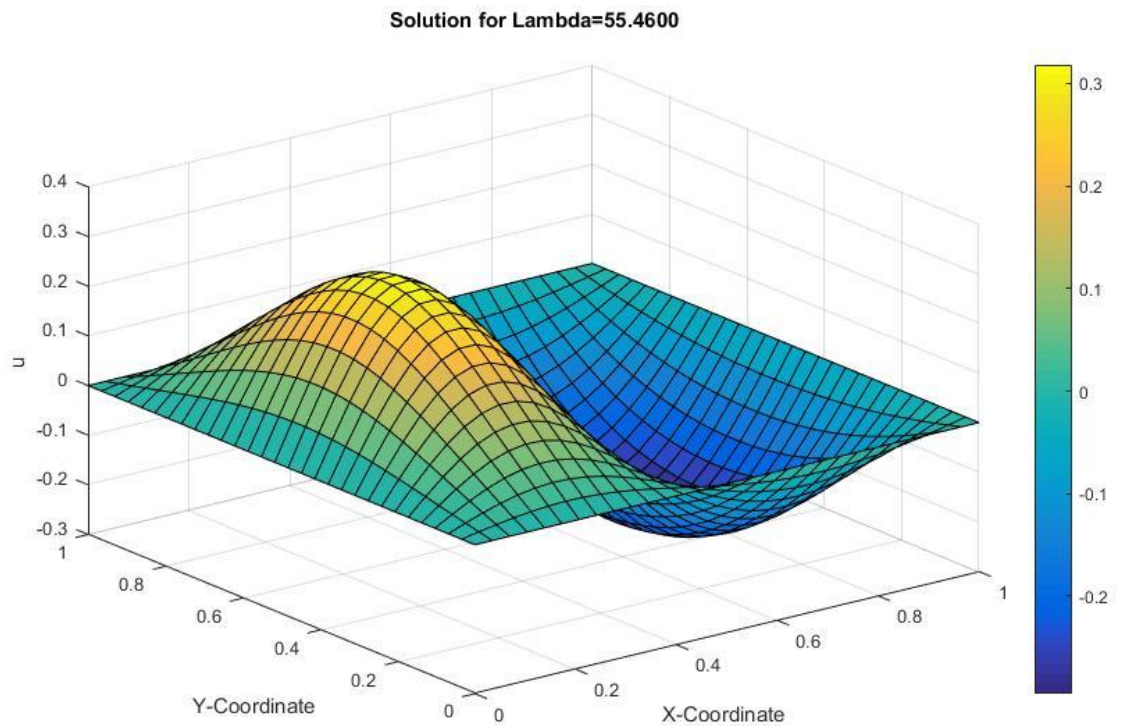
(c)


Solution for Lambda=4.2473
u
Y-Coordinate
X-Coordinate
0.01
0
-0.01
-0.02
-0.03
-0.04
-0.05
-0.06
-0.07
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
0
-0.01
-0.02
-0.03
-0.04
-0.05
-0.06

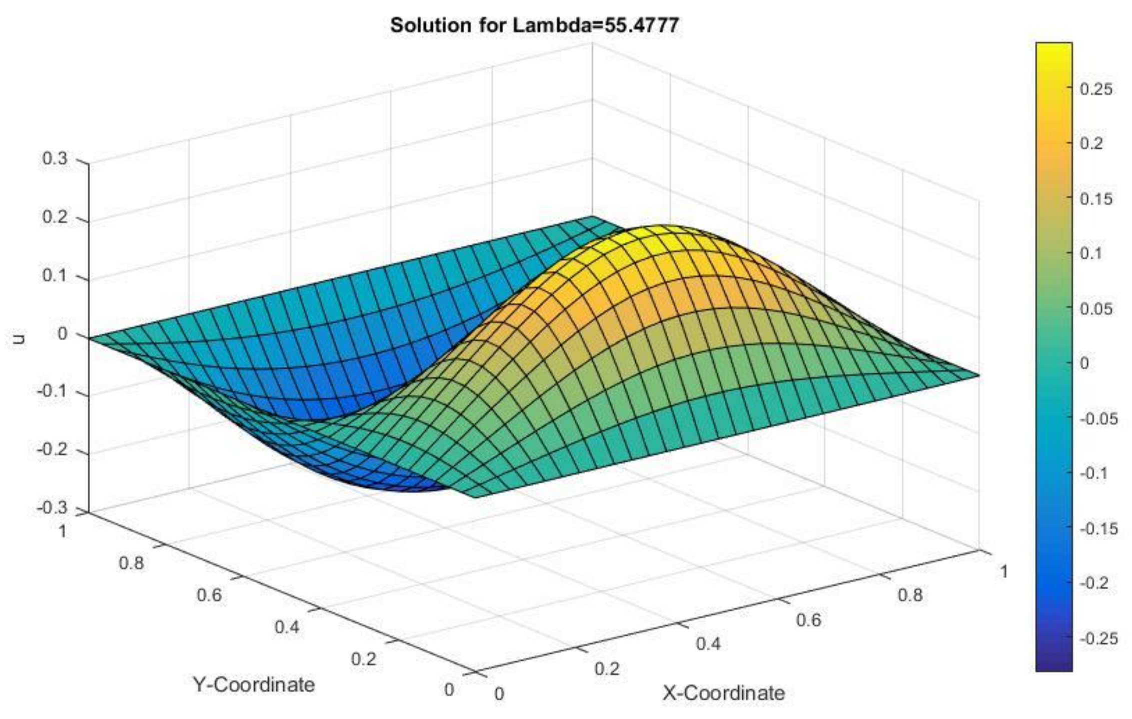
(d)


Solution for Lambda=24.2970
u
Y-Coordinate
X-Coordinate
0.25
0.2
0.15
0.1
0.05
0
-0.05
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0

(e)



(f)



(g)

Fig. 4: Contour plots for solutions for ε = 0, a) $2\pi^2$, b) $5\pi^2$, c) $5\pi^2$ and for ε = 1, d) $\lambda = 4.2473$, e) $\lambda = 24.2950$, f) $\lambda = 55.4600$, g) $\lambda = 55.4777$

For different values of λ on the curves, the corresponding distribution of *u* throughout the geometry is shown in Figure 4. For different values of λ along the same curve, the shape will remain the same, i.e. consist of hills and bowls. Again, as mentioned earlier, hills are represented as positive quantities whereas bowls are represented as negative quantities. These contours give us an idea how the solution seems to vary as we jump from one solution branch to another. Since the maximum range of λ is limited to 60, the plots above are obtained. However, corresponding to higher values of λ, it may be possible to obtain two hills and two valleys as the distribution of *u* throughout the domain. In the case where $\varepsilon = 1$, the various solutions are obtained by changing the integer vales of *m* and *n* specified in Equation 16b. Hence, two bifurcation curves are obtained for values of *m=1, 2* and *n=2, 1* respectively shown in Figure 4f and 4g.

3.4 Dependence of Solution on Forcing Function

As seen from Figures 2 and 3, the non-trivial solution value changes due to the change in the forcing function of the equation. In the first case, the forcing function is set to zero as $\varepsilon = 0$ whereas, the forcing function becomes $\sin(\pi x)$ as $\varepsilon = 1$. As we keep changing the value of ε, zero will no longer be a solution to the boundary value problem. Further, the solution tends to come very close to zero and the starts moving away from zero as was the case while applying arc-length continuation. Thus, by analyzing the non-trivial solutions for the two mentioned cases, we can sketch the solution structure for say $\varepsilon \ll 1$, which is shown in Figure 5.

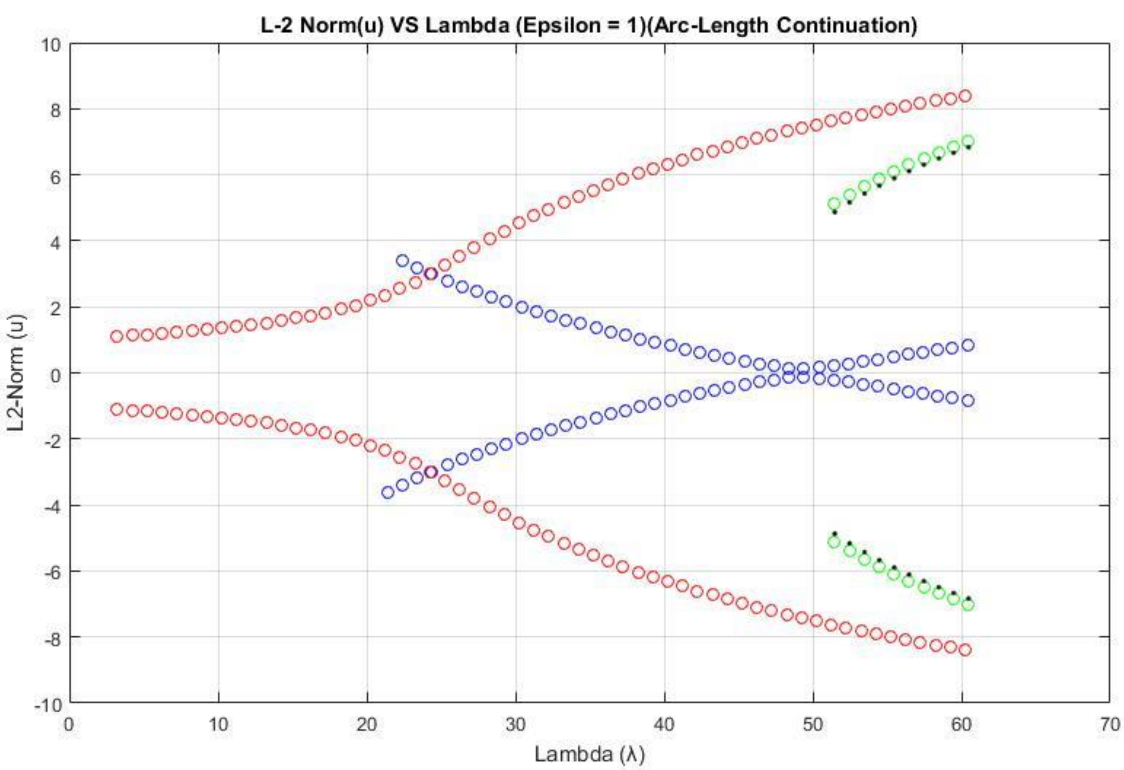


Fig. 5: Estimate for non-trivial solutions for $\varepsilon \ll 1$

4. Conclusion

We can conclude that the solution for a non-linear boundary value problem can be obtained by discretizing the domain and using the analytical continuation method and the arc-length continuation method, incorporating Newton's method. The solution tracking for non-linear BVPs is difficult and a correct estimate for the initial value of the parameter, in this case λ , needs to be made before we can start plotting our solutions. More often than not, the solution corresponding to $\varepsilon = 0$ converged to zero. Thus, to obtain an initial guess, we can use the linearized problem's solution as mentioned in the report. Suing this solution, we can get as close to the curve as possible and can further jump on the curve using Newton's method. After this, the analytical continuation method or the arc-length continuation method may be used to track the non-trivial solution.

In case of analytic continuation, by choosing as initial value of λ , and then applying Newton's method, we can jump on the curve provided we start with the appropriate value of λ . Further, on applying the technique, it gives us points right below the curve we need to plot. The exact points on the curve are obtained by Newton's method again. This iterative procedure thus, tracks the complete non-trivial solution for the range of λ starting with an initial guess.

The arc-length continuation differs in the sense that it uses two points which are to be provided as inputs, which are further used to track subsequent points along the curve. Here the length of the arc and the initial value of λ are given as an input. Thus subsequent values of λ and guesses for u are found by computing the augmented residual and Jacobian. Using this technique, curves with multiple solutions for a given λ can be plotted.

These techniques can thus, be used to find the non-trivial solution for any boundary value problem. On plotting the norm versus λ for each of the cases, we notice the formation of hills and bowls which simply tell us the variation in the solution of u throughout the domain using either of these techniques. As we keep increasing the value of λ , the number of hills, or bowls keeps increasing corresponding to that solution. The 3D contours represented in Figure 4 provide a clear thought on how the solution varies throughout the domain. This similar technique may further be incorporated in solving other non-linear boundary value problems that govern different physical nature.

5. Appendix

The MATLAB code for the discretization of the boundary value problem, the analytical continuation and the arc length continuation is given as follows:

```
% Newton's Method

function [u] = Newton(u00,m,k,e)
    h = 1/(m-1);
    n = m*m;

    c = 0;
    while c < 15
        [J,R,~,~] = Jacobian(u00,m,h,k,e);
        deltau = J\(-R);
        u = u00 + deltau;
        u00=u;
        c=c+1;
    end
end

% Jacobian

function [J,R,dRdk,dRde] = Jacobian(u0,m,h,k,e)
n=m^2;
R=zeros(n,1);
J=zeros(n,n);
dRdk=zeros(n,1);
X = zeros(m,1);
dRde=zeros(n,1);

for i = 1:m
        for j = 1:m
            l=(j-1)*(m)+i;
            X(i,1)=(i-1)*h;

            if i==1 || i==m || j==1 || j==m
                R(l) = u0(l);
                J(l,l) = 1;
            else
                R(l) = ((u0(l+1)+u0(l-1)+u0(l+m)+u0(l-m)
4*u0(l))/(h^2))+k*u0(l)+k*(u0(l)^2)-e*sin(pi*(X(i,1)));
                dRdk(l)=u0(l)+(u0(l)^2);
                dRde(l)=-sin(pi*(X(i,1)));
                J(l,l+1)=1/(h^2);
                J(l,l-1)=1/(h^2);
                J(l,l+m)=1/(h^2);
                J(l,l-m)=1/(h^2);
                J(l,l)=(-4)/(h^2)+k+2*k*u0(l);
            end
        end
end
```

```
% Analytical Continuation

function analytic(k,dk,upperk,lowerk)
    m=30;
    h = 1/(m-1);
    n = m*m;
    for i = 1:m                                        % Discretizing the domain
        for j = 1:m
            l=(j-1)*(m)+i;
            X(i,1)=(i-1)*h;
            Y(j,1)=(j-1)*h;
            u00(l,1)=sin(pi*X(i,1))*sin(pi*Y(j,1)); % Initial guess for u
        end
    end
    e=0;
    dudk=zeros(n,1);
    [u0] = Newton(u00,m,k,e);                          % Getting on the curve

    while k<upperk                            % Plotting points ahead of lambda
        y=u0;
        [J,R,dRdk,~] = Jacobian(u0,m,h,k,e);           % Applying analytical
continuation
        dudk=J\-dRdk;
        u1=u0+dudk*dk;                 % Finding new point just below the curve
        k=k+dk
        [u0] = Newton(u1,m,k,e);                       % Converging to curve

        if abs(min(u0))>max(u0)
            plot(k,-1*norm(u1),'go');
            hold on;
        else
            plot(k,norm(u1),'go');
            hold on;
        end
    end

     while k>lowerk                            % Plotting points behind lambda
         y=u0;
         [J,R,dRdk] = Jacobian(u0,m,h,k,e);           % Applying analytical
continuation
         dudk=J\-dRdk;
         u1=u0-dudk*dk;                 % Finding new point just below the curve
         k=k-dk;
         [u0] = Newton(u1,m,k,e);                     % Converging to curve

         if abs(min(u0))>max(u0)
            plot(k,-1*norm(u1),'go');
            hold on;
        else
            plot(k,norm(u1),'go');
            hold on;
         end

     end
end
    % Arc Length Continuation
```

```
function arclength(k1,ds)

    m=30;
    h = 1/(m-1);
    n = m*m;
    Z = zeros(n+1,1);
    for i = 1:m                                  % Discretizing the domain
        for j = 1:m
            l=(j-1)*(m)+i;
            X(i,1)=(i-1)*h;
            Y(j,1)=(j-1)*h;
            u00(l,1)=sin(pi*X(i,1))*sin(pi*Y(j,1));    % Initial guess for u
        end
    end
    e=0;                                             % Keeping epsilon=0
    dudk=zeros(n,1);
    [u_exact] = Newton(u00,m,k1,e);

    y=u_exact;
    [J,R,dRdk,dRde] = Jacobian(u_exact,m,h,k1,e);    % Applying analytical
continuation
    dude=J\-dRde;
    u_guess=u_exact+dude*1;
    e=1;                                             % Changing epsilon=1
    [u1] = Newton(u_guess,m,k1,e);               % Finding point 1 on curve

    k2=k1+0.5;                                  % Increment in lambda = 0.5
    [J,R,dRdk,~] = Jacobian(u1,m,h,k2,e); % Applying analytical continuation
    dudk=J\-dRdk;
    u_guess=u1+dudk*0.5;
    [u2] = Newton(u_guess,m,k2,e);               % Finding point 2 on curve

    s1=0;
    s2 = sqrt(((k2-k1)^2)+norm(u2-u1))+s1;

    while k2<60
        [Jh,dRhds,Rh] = ALCJacobian(u1,u2,s1,s2,k1,k2,m,R,dRdk);
% Finding augmented J and R
        Z = Jh\-dRhds;
        for i = 1:n
            duds(i,1) = Z(i,1);
        end
        dkds = Z(n+1,1);
        u3g = u2 + ds*duds;                        % Finding guess for u3
        k3g = k2 + ds*dkds;                        % Finding guess for k3
        [u3,k3] = augNewton(Jh,Rh,n,u3g,k3g,u2,k2,m,R,dRdk,s1,s2);
% Converging to find u3 and k3
        plot(k3,norm(u3),'k.');
        hold on;




        % Contour
```

```
        if 55<k3<56
            for i = 1:m
                for j = 1:m
                    l=(j-1)*(m)+i;
                    c(j,i) = u1(l);
                end
            end
            figure:surf(X,Y,c)
        end

        u1 = u2;                  % Switching parameters to compute next arc
        k1 = k2;
        s1 = s2;
        u2 = u3;
        k2 = k3;
        s3 = s2+ds;
        s2 = s3;
    end
end

% Arc Length Jacobian

function [Jh,dRhds,Rh] = ALCJacobian(u1,u2,s1,s2,k1,k2,m,R,dRdk)
n=m*m;
h = 1/(m-1);
Jh=zeros(n+1);
Rh=zeros(n+1,1);
dRhds=zeros(n+1,1);
dcdu=zeros(n,1);

c = (s2-s1)^2-norm(u2-u1)^2-(k2-k1)^2;              % Calculating eta

for i = 1:m                                 % Calculating augmented J and R
        for j = 1:m
            l=(j-1)*(m)+i;

            dcdk = -2*(k2-k1);
            dcdu(l)=-2*(u2(l)-u1(l));
            dcds = 2*(s2-s1);

            Rh(l)=R(l);
            Rh(901)=c;
            dRhds(l)=0;
            dRhds(901)=dcds;

            if i==1 || i==m || j==1 || j==m
                Jh(l,l) = 1;

            else
                Jh(l,l+1)=1/(h^2);
                Jh(l,l-1)=1/(h^2);
                Jh(l,l+m)=1/(h^2);
                Jh(l,l-m)=1/(h^2);
                Jh(l,l)=(-4)/(h^2)+k2+2*k2*u2(l);
            end
```

```
            Jh(n+1,l)=dcdu(l);
            Jh(1,n+1)=dRdk(l);
            Jh(n+1,n+1)=dcdk;
        end
end

% Augmented Newton

function [u3,k3] = augNewton(Jh,Rh,n,u3g,k3g,u2,k2,m,R,dRdk,s1,s2)

Q = zeros(n+1,1);
deltau=zeros(n,1);
Q=Jh\(-Rh);
c=0;
while c < 15
    [Jh,~,Rh] = ALCJacobian(u2,u3g,s1,s2,k2,k3g,m,R,dRdk);
    Q=Jh\(-Rh);
    for i=1:n
        deltau(i,1) = Q(i,1);
    end
    deltak = Q(n+1,1);
    u3 = u3g + deltau;
    k3 = k3g + deltak;
    c=c+1;
end
end
```