Q-4: MATLAB CODE

1] Program numerical solutions to this inflow equation using both the fixed point iterative method and the Newton - Raphson method.
Plot the % error for each iteration step for both the methods v/s iteration number on the same graph.

```
%% Calculation of the inflow ratio at the rotor disk in forward flight from
momentum theory

clear all
close all
clc

% Given Data
alpha = 2*pi/180;               % Disk Angle of Attack (α)
mu = 0.2;                       % Advanced ratio (μ)
Ct = 0.006;                     % Thrust coefficient

%% Fixed Point Iteration
lambda_0 = sqrt(Ct/2);          % Initial guess of inflow ratio (typically hover
                                  inflow ratio)
lambda = lambda_0;
lambda_old = lambda_0;          % Old value of Inflow ratio
maxerrper = 0.0001;             % Maximum error
n = 10;                         % No. of iterations

for i=1:n
    lambda = mu*tan(alpha) + Ct/(2*sqrt(mu^2 + lambda^2));
    errper(i) = abs((lambda - lambda_old)/lambda_old)*100;
    errper_fpi(i) = errper(i);
    if (errper_fpi(i) < maxerrper)
        break;
    end
    lambda_old = lambda;
    i = i+1;
end

% Plotting of data (Fixed Point Iteration)
plot(1:i,errper_fpi,'bo',1:i,errper_fpi,'-r')
xlim([0,6])
ylim([0,70])
title('Inflow ratio (λ) calculation using Momentum theory')
ylabel('% error');
xlabel('No of iteration');
grid on
hold on

%% Newton Raphson Iteration
lambda_0 = sqrt(Ct/2);          % Initial guess of inflow ratio (typically hover
                                  inflow ratio)
lambda = lambda_0;
lambda_old = lambda_0;          % Old value of Inflow ratio
maxerrper = 0.0001;             % Maximum error
n = 10;                         % No. of iterations
```

```
for i=1:n
    f = lambda - mu*tan(alpha) - 0.5*Ct/sqrt(mu^2 + lambda^2);
    df = 1 + 0.5*(Ct*lambda)/((mu^2 + lambda^2)^1.5);
    lambda = lambda - (f/df);
    errper(i) = abs((lambda - lambda_old)/lambda_old)*100;
    errper_nr(i) = errper(i);
    if (errper_nr(i) < maxerrper)
        break;
    end
    lambda_old = lambda;
    i = i+1;
end

% Plotting of data (Newton Raphson Iteration)
plot(1:i,errper_nr,'k*',1:i,errper_nr,'--k')
legend({'λ (Fixed Point Iteration)','','λ (Newton
Raphson)'},'Location','northeast')
hold off
annotation('textbox', [0.55, 0.65, 0.35, 0.15], 'String', "Disk Angle of Attack
(α) = 2°, Advanced ratio (μ) = 0.2, Thrust coefficient (Ct) = 0.006")
```

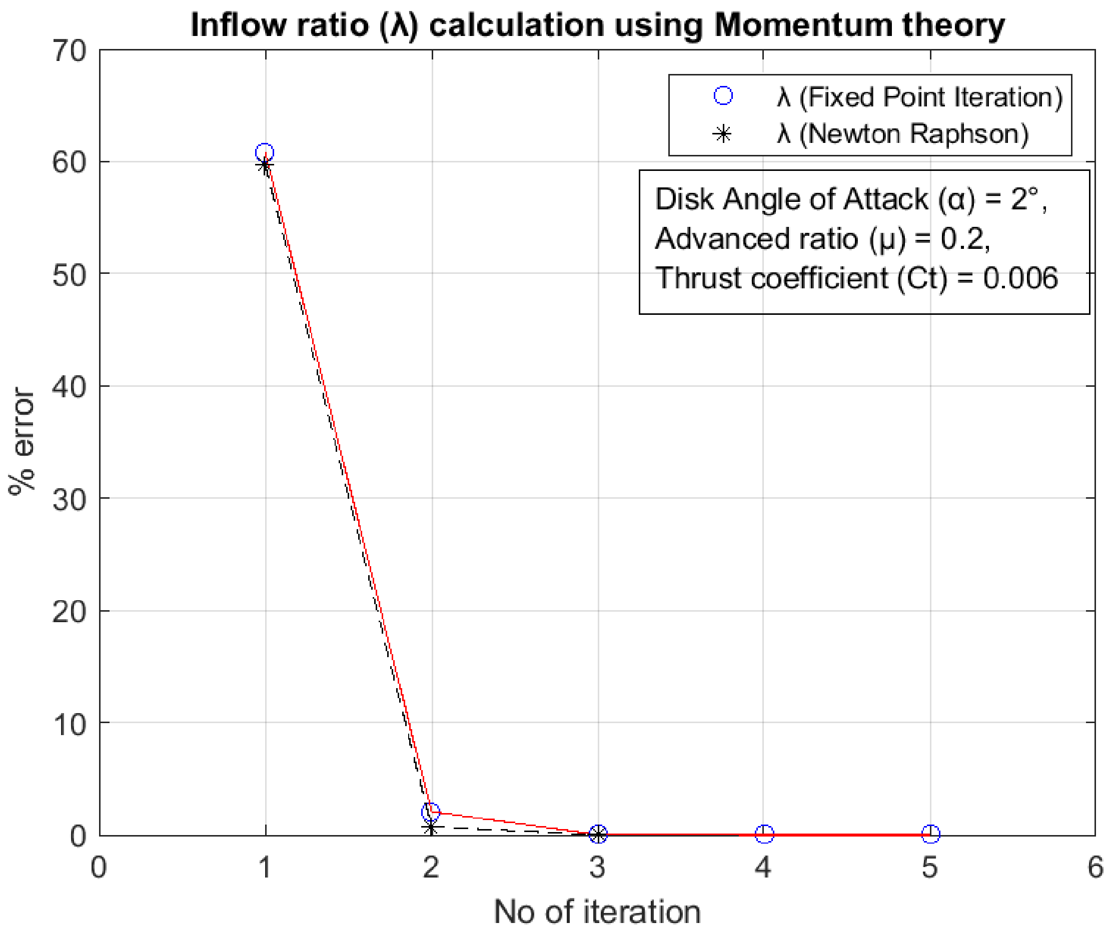

Figure 1 : Plot of the % error for each iteration step using Fixed point iterative method & Newton-Raphson method v/s Iteration Number

2] Describe all the steps and state the convergence criteria used.

Step 1 : First declare the given values (Initial conditions).
Step 2 : Set an number of iterations (Approximately).
Step 3 : To start the fixed point iteration or Newton Raphson method set an initial value (here we typically start with inflow ratio of hovering condition).
Step 4 : Set an error value (here taken 10^-6, randomly), iteration said to be converged if calculation error is below this value.
Step 5 : Start iteration based on the methods of Fixed point Iteration or Newton Raphson method.
Step 6 : For each iteration calculate the percentage error between calculated λ and previous iteration value.
Step 7 : Plot the error in percentage in each step against number of iterations.

3] Check if final answer is affected by choice of initial guess for the inflow ratio on the numerical solution.

Iterations	**Percentage Error (Fixed Point Iteration)**			
λ_initial	**sqrt(2*Ct)**	**sqrt(Ct)**	**sqrt(Ct/2)**	**sqrt(Ct/5)**
1	81.6148	72.9256	60.8352	37.1725
2	8.7818	4.4376	2.0846	0.6066
3	0.0626	0.0336	0.0163	0.0048
4	0.0005	0.0003	0.0001	0
5	0	0	0	

Iterations	**Percentage Error (Newton Raphson)**			
λ_initial	**sqrt(2*Ct)**	**sqrt(Ct)**	**sqrt(Ct/2)**	**sqrt(Ct/5)**
1	79.4138	71.2475	59.7344	36.7162
2	2.9083	1.6906	0.7223	0.1235
3	0.0003	0.0001	0	0
4	0	0		

- We initially guessed our induced inflow value to be similar to that of the hover condition, i.e. $\lambda = \lambda h$ = sqrt(Ct/2).

- For our exploration of initial guess we have tried our two methods with different initial guesses like, λ = sqrt(2*Ct), λ = sqrt(Ct), λ = sqrt(Ct/2) and also λ = sqrt(Ct/5).
- Results from all algorithm: $\lambda = 0.0219$
- Any initial guess will give the same result of λ, only the number of iterations will change.

4] For the given CT and α plot λ vs μ using one of the methods and compare it with the approximate solution of λ valid for $\mu > 0.2$ by plotting it on the same graph using different line types.

4-A] Fixed Point Iteration

```
%% Calculation of inflow ratio at the rotor disk in forward flight from the
momentum theory and Solution of λ using Fixed Point Iteration compared with the
approximate (valid for μ > 0.2) solution of λ

clear all
close all
clc

% Given Data
alpha = 2*pi/180;               % Disk Angle of Attack (α)
Ct = 0.006;                     % Thrust coefficient

%% Fixed Point Iteration
lambda_0 = sqrt(Ct/2);          % Initial guess of inflow ratio (typically hover
inflow ratio)
lambda(1) = lambda_0;

for i = 1:1:301
    mu(i) = 0.002*i - 0.002;                                                  %
0<=μ<=0.6 ~~ varying Advanced ratio (Incompressible Subsonic regime)
    lambda_apprx(i) = mu(i)*tan(alpha) + Ct/(2*mu(i));
    lambda(i+1) = mu(i)*tan(alpha) + Ct/(2*sqrt(mu(i)^2 + lambda(i)^2));
    lambda_fpi(i) = lambda(i+1);
end

% Plotting of data (Fixed Point Iteration)
plot(mu,lambda_fpi,'r')
hold on
plot(mu,lambda_apprx,'--k')
xlim([0 0.6])
ylim([0 0.08])
title('μ v/s λ (Fixed Point Iteration & Approximate Solution)')
xlabel('Advance Ratio, μ')
ylabel('Inflow Ratio, λ')
legend('Fixed Point Iteration','Approximate Solution')
grid on
hold off
annotation('textbox', [0.55, 0.68, 0.34, 0.1], 'String', "Disk Angle of Attack (α)
= 2°, Thrust coefficient (Ct) = 0.006")
```

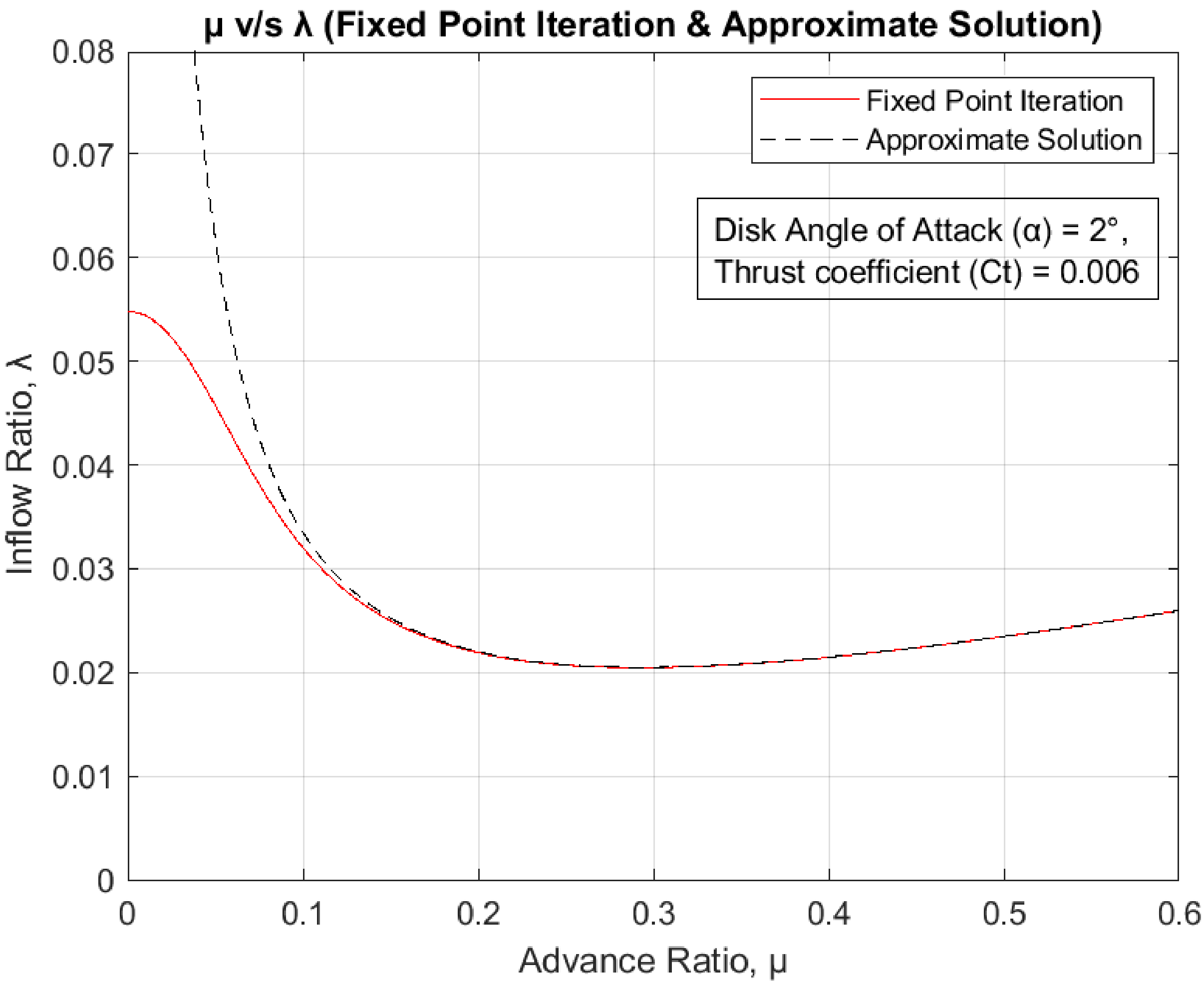

Figure 2 : Comparison of Fixed Point Iteration & Approximate Solution for λ with varying μ

4-B] Newton-Raphson method

```
%% Calculation of inflow ratio at the rotor disk in forward flight from the
momentum theory and Solution of λ using Fixed Point Iteration compared with the
approximate (valid for μ > 0.2) solution of λ

clear all
close all
clc

% Given Data
alpha = 2*pi/180;               % Disk Angle of Attack (α)
Ct = 0.006;                     % Thrust coefficient

%% Newton Raphson
lambda_0 = sqrt(Ct/2);          % Initial guess of inflow ratio (typically hover
inflow ratio)
lambda(1) = lambda_0;

for i = 1:1:301
    mu(i) = 0.002*i - 0.002;                                                   %
0<=μ<=0.6 ~~ varying Advanced ratio (Incompressible Subsonic regime)
    lambda_apprx(i) = mu(i)*tan(alpha) + Ct/(2*mu(i));
    f = lambda(i) - mu(i)*tan(alpha) - 0.5*Ct/sqrt(mu(i)^2 + lambda(i)^2);
    df = 1 + 0.5*(Ct*lambda(i))/((mu(i)^2 + lambda(i)^2)^1.5);
    lambda(i+1) = lambda(i) - (f/df);
    lambda_nr(i) = lambda(i+1);
end

% Plotting of data (Newton Raphson)
plot(mu,lambda_nr,'r')
hold on
plot(mu,lambda_apprx,'--k')
xlim([0 0.6])
ylim([0 0.08])
title('μ v/s λ (Newton Raphson method & Approximate Solution)')
xlabel('Advance Ratio, μ')
ylabel('Inflow Ratio, λ')
legend('Newton Raphson','Approximate Solution')
grid on
hold off
annotation('textbox', [0.55, 0.68, 0.34, 0.1], 'String', "Disk Angle of Attack (α)
= 2°, Thrust coefficient (Ct) = 0.006")
```

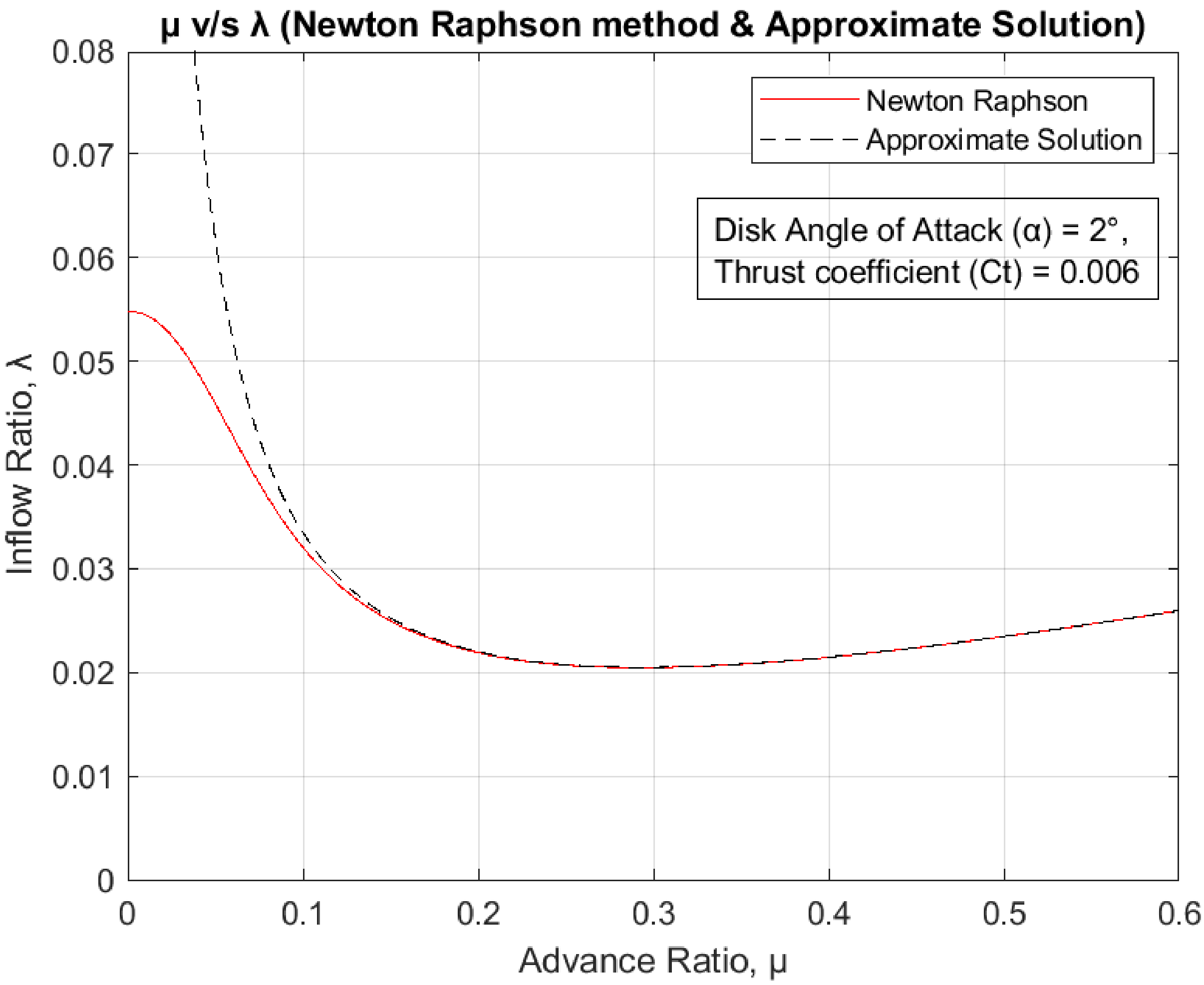

Figure 3 : Comparison of Newton Raphson & Approximate Solution for λ with varying μ