REPORT

Project I

Helicopter Theory

Uck459E - 22532

Nadir Doğan

110180807

Assist. Prof. Dr. Özge Özdemir

Momentum Theory – Forward Flight

- **Programming Assignment**

A Matlab code was written using the "Fixed Point Iteration Method" and the "Newton Raphson Method" to calculate the inflow rate in forward flight. In the first assignment, the AH-1W Cobra helicopter was chosen and information about this helicopter was given. In line with this information, the required input values for the matlab code are now taken, as discussed below. The program calculates the inflow ratio with the two iteration methods used and tells how many iterations it has made. Then the inflow ratio results are plotted.

- **Numerical Solution to Inflow Ratio**
  - **Fixed Point Iteration Method**

The inflow ratio is calculated by applying the first equation until the error value determined by the second equation falls below a certain value (0.0005 used).

$$\lambda_{n+1} = \mu tan(\alpha) + \frac{C_T}{(2\sqrt{\mu^2 + \lambda_n^2})}$$

$$\epsilon = \left\| \frac{\lambda_{n+1} - \lambda_n}{\lambda_{n+1}} \right\|$$

  - **Newton Raphson Method**

In this method, a function is obtained with λ as in the first equation, and then the derivative of this function is taken. Then the process is repeated with the third equation until the error value is less than (0.0005).

$$f(\lambda) = \lambda - \mu tan(\alpha) - \frac{C_T}{(2\sqrt{\mu^2 + \lambda_n^2})}$$

$$f'(\lambda) = 1 + \frac{C_T}{2}(\mu^2 + \lambda^2)^{(-3/2)}\lambda$$

$$\lambda_{n+1} = \lambda_n - \left[\frac{f(\lambda)}{f'(\lambda)}\right]_n$$

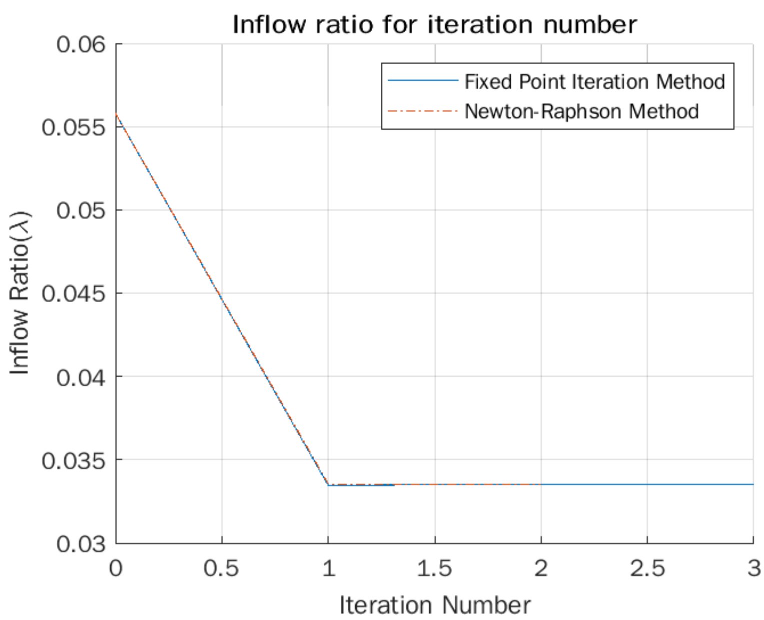
- Results

Fixed Point Iteration Method Inflow ratio: 3.353648e-02, Iteration Number : 3
Newton-Raphson Method Inflow ratio: 3.353648e-02, Iteration Number : 2

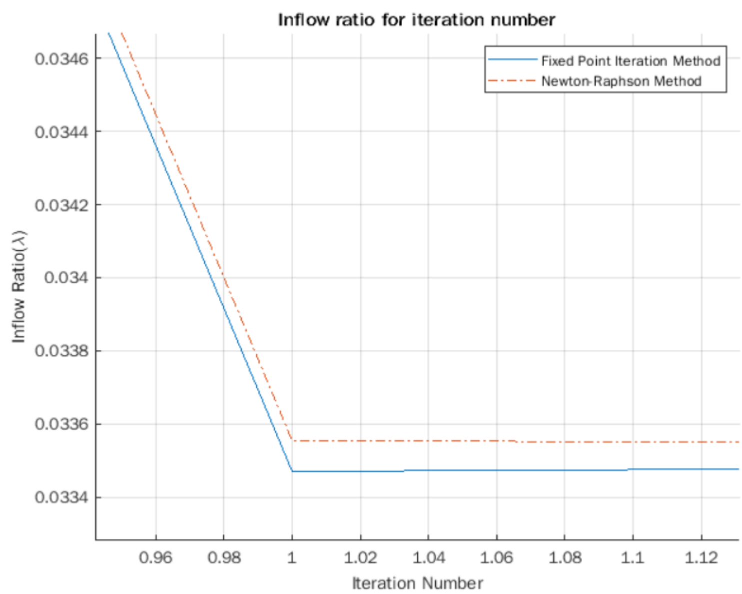
Command Window

```
Fixed Point Iteration Method Inflow ratio: 3.353648e-02, Iteration Number : 3
Newton-Raphson Method Inflow ratio: 3.353648e-02, Iteration Number : 2
>>
```

Plot



Closer Look



General characteristics	
Crew	2: pilot, CPG (co-pilot/gunner)
Fuselage length	44 ft 7 in (13.6 m)
Length, rotors turning	
Rotor diameter	48 ft (14.6 m)
Height	13 ft 5 in (4.1 m)
Disc area	530.83 ft² (168.1 m²)
Empty weight	10,920 lb (4,953 kg)
Max takeoff weight	14,750 lb (6,690 kg)
Powerplant	2× General Electric T700 turboshaft, 1,680shp (1,300 kW) each
Rotor systems	2 blades on main rotor, 2 blades on tail rotor
Performance	
Maximum speed	190 knots (218 mph, 352 km/h)
Range	317 nm (365 mi, 587 km)
Service ceiling	12,200 ft (3,720 m)
Rate of climb	1,620 ft/min (8.2 m/s)

AH-1W Cobra

```
clc
clear

Gtow= 4953*9.81;                    % N
R = 14.6/2;                         % m -- rotor diameter
Ro = 1.226;                         % kg/m^3
Area = 168.1;                       % m^2
Voo = 97.7;                         % m/s -- 352 km/h --> 97.7 m/s
Vtip = R*2*pi*(360/60);             % m/s tip velocity
a1 = 0*pi/180;                      % hover con.
a2 = 4*pi/180;                      % forward flight
Th = Gtow/cos(a1);                  % N -- Thrust hover con.
Cth = Th / ((1/2)*Ro*Area*Vtip^2);  % Thrust coefficient hover con.
L(1) = sqrt(Cth/2);                 % hover condition Lambda
T = Gtow/cos(a2);                   % N -- Thrust forward flight
Ct = T / ((1/2)*Ro*Area*Vtip^2);    % Thrust coefficient forward flight
Mu = (Voo*cos(a2))/Vtip;            % advance ratio

for i=1:100 % 100 = iteration number -- FIXED POINT ITERATION METHOD
   L(i+1)=(Mu*tan(a2))+(Ct/(2*sqrt(Mu^2+L(i)^2)));
   err = abs((L(i+1)-L(i))/L(i+1));
   forplot(i+1) = i;
   if err < 0.0005                 % error estimate
       fprintf('Fixed Point Iteration Method Inflow ratio: %d, Iteration
       Number : %d \n', L(i+1), double(i))
       break
   end
end
L2(1) = L(1);
for k=1:100 % 100 = iteration number -- NEWTON-RAPHSON METHOD
   fL2(k+1) = L2(k)-(Mu*tan(a2))-(Ct/(2*sqrt(Mu^2+L2(k)^2)));
   fL2dot(k+1) = 1 + Ct*L2(k)*((Mu^2 + L2(k)^2)^(-3/2))/2;
   L2(k+1) = L2(k) - ((fL2(k+1))/fL2dot(k+1));
   forplot2(k+1) = k;
```

```
    err2 = abs((L2(k+1)-L2(k))/L2(k+1));
    if err2 < 0.0005                % error estimate
        fprintf('Newton-Raphson Method Inflow ratio: %d, Iteration Number :
        %d \n', L2(k+1), double(k))
        break
    end
end
    hold on
    grid on
    plot(forplot,L)
    plot(forplot2,L2,'-.')
    legend('Fixed Point Iteration Method','Newton-Raphson Method')
    title('Inflow ratio for iteration number');
    xlabel('Iteration Number');
    ylabel('Inflow Ratio(\lambda)');
```