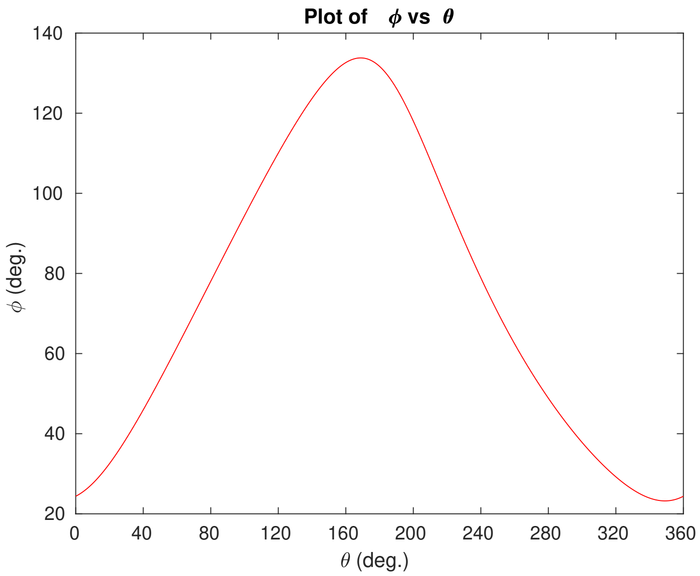
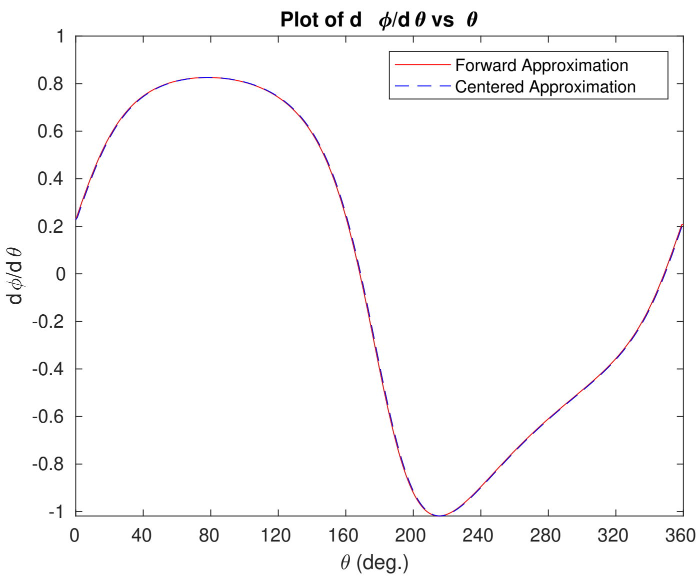
Q1. For the given mechanism, the following plot is obtained for the angle ϕ vs θ. Newton's method from Homework 2, Question 3 has been employed.

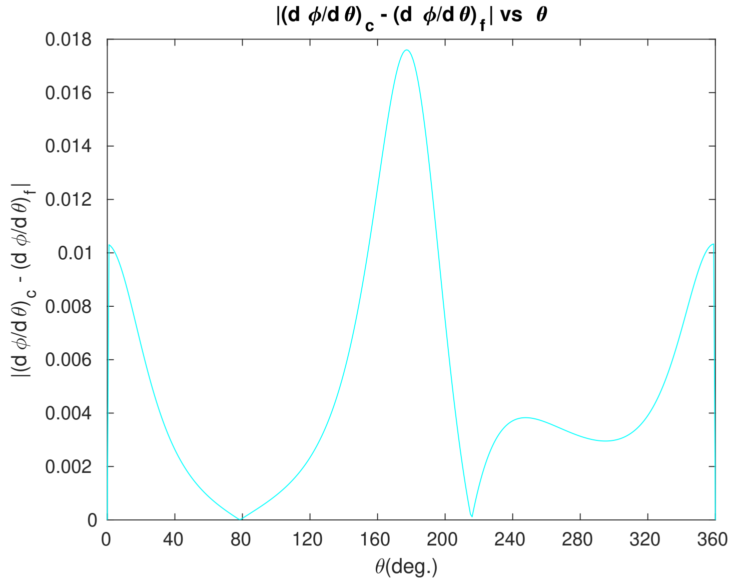


The derivative $d\phi/d\theta$ is obtained using both a forward and centered difference formula. The results are presented below.



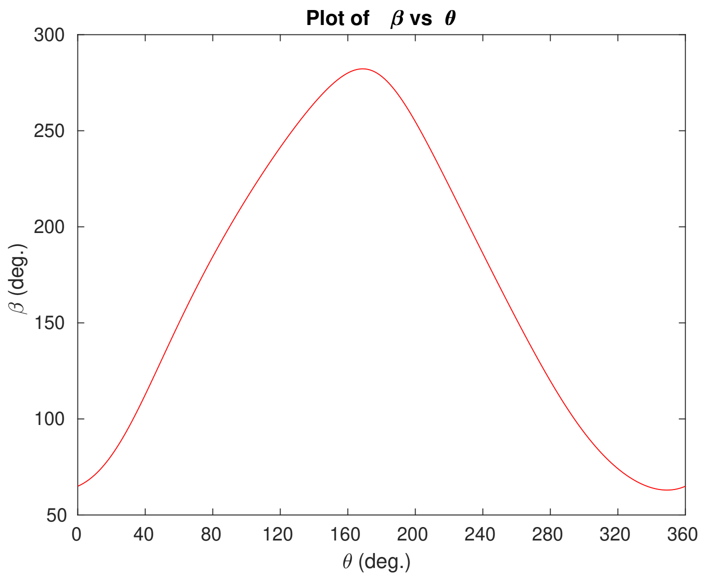
<u>COMPARISON:</u> It can be seen that both the curves almost overlap and the difference is very small. However, since the centered difference method has an error term of the order $O(h^2)$ while the forward difference method has an error term of the order $O(h)$, we expect the centered method formula to be more accurate.

The absolute difference between the results obtained by the two methods is plotted below.

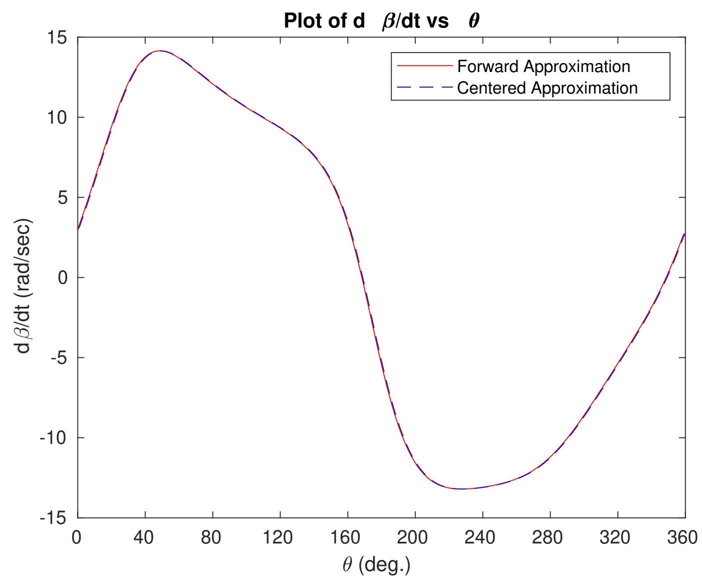


We see that the maximum difference is ≈ 0.02 which is of the order 10^{-2} which is also the same as $O(h)$ since $h = 1^o \approx 0.0175$.

Q2. The obtained plot for the angle β vs θ is shown below.

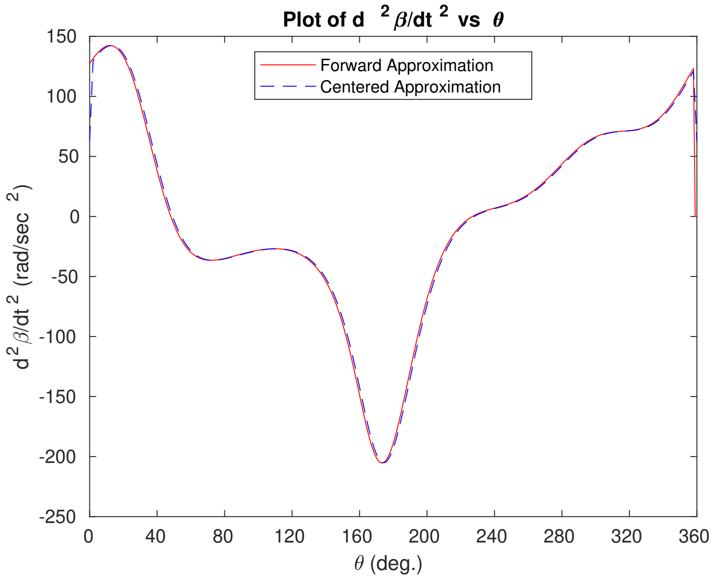


The derivative $d\beta/d\theta$ is obtained using both a forward and centered difference formula. The results are scaled to obtain $d\beta/dt$ using the relation $d\beta/dt = \omega(d\beta/d\theta)$. The results are presented below.



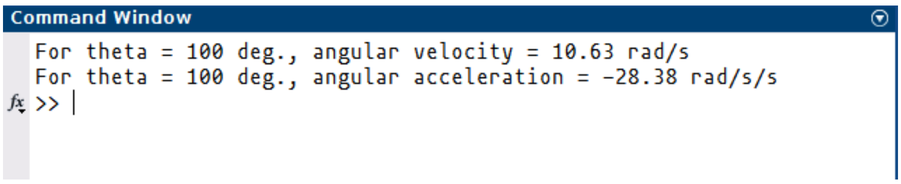
Once again, the curves almost overlap and the difference between the methods is small.

The second derivative $d^2\beta/d\theta^2$ is also obtained using both a forward and centered difference formula. Forward difference method is used with the values of the derivative $d\beta/d\theta$ obtained using the forward difference formula. Similarly, centered difference method is used with the values of the derivative $d\beta/d\theta$ obtained using centered difference method. The results are scaled to obtain $d^2\beta/dt^2$ using the relation $d^2\beta/dt^2 = \omega^2\,(d^2\beta/d\theta^2)$. The results are presented below.



The difference between the two methods is slightly more evident now, since we are comparing results obtained using successive forward and centered difference methods.

<u>ANSWER CHECK:</u> For $\theta = 100^o$ and an angular velocity of the driver link $\omega = 450$ rad/min $= 7.5$ rad/s, we obtain the angular velocity of the agitator shaft $d\beta/dt = 10.63$ rad/s and the angular acceleration $d^2\beta/dt^2 = -28.38$ rad/s^2. These answers are obtained using centered difference methods.



MATLAB Code

```
%
%  DS 288 Homework 4
%  Nikhil Jayswal, SR - 16961
%

clear; clc;
close all

% lengths of various links (inches)
DA = 1.94;
GF = 1.26;
AB = 6.86;
DC = 7.00;
CB = 2.36;
CG = 1.25;
EF = 1.87;
CE = 2.39;

% Problem 1

theta = (0:360);
theta4 = (theta*pi/180) + pi;

r1 = DC;
r2 = CB;
r3 = AB;
r4 = DA;

% tolerance
tol = 1e-4;

% iteration counter
iter = ones(size(theta4));

% phi = theta2
theta2 = zeros(size(theta4));
theta3 = zeros(size(theta4));

for i = 1:length(theta4)
```

```
    % initial guess for theta2 and theta3
    if i == 1
        init_theta2 = pi/6;
        init_theta3 = 0;
    else % use previous solutions as initial guess
        init_theta2 = theta2(i-1);
        init_theta3 = theta3(i-1);
    end

    % calculate theta2 and theta3
    while (1 > 0)
        f = -[r2*cos(init_theta2) + r3*cos(init_theta3) + ...
              r4*cos(theta4(i)) - r1; ...
              r2*sin(init_theta2) + r3*sin(init_theta3) + ...
              r4*sin(theta4(i))];
        % compute Jacobian
        J = [-r2*sin(init_theta2) -r3*sin(init_theta3); ...
              r2*cos(init_theta2)  r3*cos(init_theta3)];
        % solve for new theta2 and theta3
        guess = J\f;
        theta2(i) = init_theta2 + guess(1);
        theta3(i) = init_theta3 + guess(2);
        % check if guess is as accurate as required
        if (abs(guess(1)) < tol) && (abs(guess(2)) < tol)
            break
        else
            iter(i) = iter(i) + 1;     % update iteration counter
            init_theta2 = theta2(i);   % reset initial guess for theta2
            init_theta3 = theta3(i);   % reset initial guess for theta3
        end
    end
end

% plot phi vs theta
figure
plot(theta, theta2*180/pi, 'r-')
xlim([0, 360])
xticks(0:40:360)
xlabel('\theta (deg.)')
ylabel('\phi (deg.)')
title('Plot of \phi vs \theta')
```

```
% uncomment to check
%[r2*cos(theta2) + r3*cos(theta3) + r4*cos(theta4) - r1; ...
%r2*sin(theta2) + r3*sin(theta3) + r4*sin(theta4)]

% calculate dphi/dtheta
% forward approximation
h = theta4(2) - theta4(1);
dphi_dtheta_f = zeros(size(theta4));
for i = 1:length(theta4)
    % use backward approximation for last point
    if theta4(i) == theta4(end)
        dphi_dtheta_f(i) = (theta2(i) - theta2(i-1))/h;
    else
        dphi_dtheta_f(i) = (theta2(i+1) - theta2(i))/h;
    end
end
% centered approximation
h = theta4(2) - theta4(1);
dphi_dtheta_c = zeros(size(theta4));
for i = 1:length(theta4)
    % use forward approximation for first point
    if i == 1
        dphi_dtheta_c(i) = (theta2(i+1) - theta2(i))/h;
    % use backward approximation for last point
    elseif theta4(i) == theta4(end)
        dphi_dtheta_c(i) = (theta2(i) - theta2(i-1))/h;
    else
        dphi_dtheta_c(i) = (theta2(i+1) - theta2(i-1))/(2*h);
    end
end

% plot both approximations
figure
plot(theta, dphi_dtheta_f, 'r-')
hold on
plot(theta, dphi_dtheta_c, 'b--')
legend('Forward Approximation', 'Centered Approximation')
xlim([0, 360])
xticks(0:40:360)
xlabel('\theta (deg.)')
```

```
ylabel('d\phi/d\theta')
title('Plot of d\phi/d\theta vs \theta')

% plot difference between the two approximations
figure
diff = (abs(dphi_dtheta_c - dphi_dtheta_f));
plot(theta, diff, 'c-');
xlabel('\theta(deg.)')
ylabel('|(d\phi/d\theta)_c - (d\phi/d\theta)_f|')
xlim([0, 360])
xticks(0:40:360)
title('|(d\phi/d\theta)_c - (d\phi/d\theta)_f| vs \theta')

% Problem 2

alpha = (theta2*180/pi) + 149;
theta4 = (alpha*pi/180) + pi;

r1 = CG;
r2 = GF;
r3 = EF;
r4 = CE;

% tolerance
tol = 1e-4;

% iteration counter
iter = ones(size(theta4));

% beta = theta2
theta2 = zeros(size(theta4));
theta3 = zeros(size(theta4));

for i = 1:length(theta4)
    % initial guess for theta2 and theta3
    if i == 1
        init_theta2 = pi/6;
        init_theta3 = 2*pi - pi/6;
    else % use previous solutions as initial guess
        init_theta2 = theta2(i-1);
        init_theta3 = theta3(i-1);
```

```
    end

    % calculate theta2 and theta3
    while (1 > 0)
        f = -[r2*cos(init_theta2) + r3*cos(init_theta3) + ...
              r4*cos(theta4(i)) - r1; ...
              r2*sin(init_theta2) + r3*sin(init_theta3) + ...
              r4*sin(theta4(i))];
        % compute Jacobian
        J = [-r2*sin(init_theta2) -r3*sin(init_theta3); ...
              r2*cos(init_theta2)  r3*cos(init_theta3)];
        % solve for new theta2 and theta3
        guess = J\f;
        theta2(i) = init_theta2 + guess(1);
        theta3(i) = init_theta3 + guess(2);
        % check if guess is as accurate as required
        if (abs(guess(1)) < tol) && (abs(guess(2)) < tol)
            break
        else
            iter(i) = iter(i) + 1;     % update iteration counter
            init_theta2 = theta2(i);   % reset initial guess for theta2
            init_theta3 = theta3(i);   % reset initial guess for theta3
        end
    end
end

% convert all values to between 0 and 2*pi
for i = 1:length(theta2)
    theta2(i) = theta2(i) + 10*pi;
end

% plot beta vs phi
figure
plot(theta, theta2*180/pi, 'r-')
xlim([0, 360])
xticks(0:40:360)
xlabel('\theta (deg.)')
ylabel('\beta (deg.)')
title('Plot of \beta vs \theta')

% uncomment to check
```

```
%[r2*cos(theta2) + r3*cos(theta3) + r4*cos(theta4) - r1; ...
%r2*sin(theta2) + r3*sin(theta3) + r4*sin(theta4)]

% calculate dbeta/dtheta
% forward approximation
dbeta_dtheta_f = zeros(size(theta4));
for i = 1:length(theta4)
    % use backward approximation for last point
    if theta4(i) == theta4(end)
        dbeta_dtheta_f(i) = (theta2(i) - theta2(i-1))/h;
    else
        dbeta_dtheta_f(i) = (theta2(i+1) - theta2(i))/h;
    end
end
% centered approximation
dbeta_dtheta_c = zeros(size(theta4));
for i = 1:length(theta4)
    % use forward approximation for first point
    if i == 1
        dbeta_dtheta_c(i) = (theta2(i+1) - theta2(i))/h;
    % use backward approximation for last point
    elseif theta4(i) == theta4(end)
        dbeta_dtheta_c(i) = (theta2(i) - theta2(i-1))/h;
    else
        dbeta_dtheta_c(i) = (theta2(i+1) - theta2(i-1))/(2*h);
    end
end

% % plot both approximations
% figure
% plot(theta, dbeta_dtheta_f, 'r-')
% hold on
% plot(theta, dbeta_dtheta_c, 'b--')
% legend('Forward Approximation', 'Centered Approximation')
% xlim([0, 360])
% xticks(0:40:360)
% xlabel('\theta (deg.)')
% ylabel('d\beta/d\theta')
% title('Plot of d\beta/d\theta vs \theta')

% angular velocity of driver link = dtheta/dt = 450 rad/min
```

```
omega = 450/60; % rad/sec

% plot dbeta_dt
figure
plot(theta, omega * dbeta_dtheta_f, 'r-')
hold on
plot(theta, omega * dbeta_dtheta_c, 'b--')
legend('Forward Approximation', 'Centered Approximation')
xlim([0, 360])
xticks(0:40:360)
xlabel('\theta (deg.)')
ylabel('d\beta/dt (rad/sec)')
title('Plot of d\beta/dt vs \theta')

% calculate d(dbeta/dtheta)/dtheta
% forward approximation
ddbeta_dtheta_f = zeros(size(theta4));
for i = 1:length(theta4)
    % use backward approximation for last point
    if theta4(i) == theta4(end)
        ddbeta_dtheta_f(i) = (dbeta_dtheta_f(i) - dbeta_dtheta_f(i-1))/h;
    else
        ddbeta_dtheta_f(i) = (dbeta_dtheta_f(i+1) - dbeta_dtheta_f(i))/h;
    end
end
% centered approximation
ddbeta_dtheta_c = zeros(size(theta4));
for i = 1:length(theta4)
    % use forward approximation for first point
    if i == 1
        ddbeta_dtheta_c(i) = (dbeta_dtheta_c(i+1) - ...
                              dbeta_dtheta_c(i))/h;
    % use backward approximation for last point
    elseif theta4(i) == theta4(end)
        ddbeta_dtheta_c(i) = (dbeta_dtheta_c(i) - ...
                              dbeta_dtheta_c(i-1))/h;
    else
        ddbeta_dtheta_c(i) = (dbeta_dtheta_c(i+1) - ...
                              dbeta_dtheta_c(i-1))/(2*h);
    end
end
```