Name: Aneesh Panchal
SR No: 06-18-01-10-12-24-1-25223
Email ID: aneeshp@iisc.ac.in
Date: November 6, 2024

Homework No: Homework 4
Course Code: DS288
Course Name: Numerical Methods
Term: AUG 2024

Derivation of the function in single variable

Given Four Bar Linkage Problem is of the form,

$$f_1(\theta_2, \theta_3) = r_2 \cos\theta_2 + r_3 \cos\theta_3 + r_4 \cos\theta_4 - r_1 = 0$$

$$f_2(\theta_2, \theta_3) = r_2 \sin\theta_2 + r_3 \sin\theta_3 + r_4 \sin\theta_4 = 0$$

Now here, in given problems we require to find out only θ_2, and instead of (θ_2, θ_3), hence,
From 1^{st} equation we get,

$$\cos\theta_3 = \frac{r_1 - r_2\cos\theta_2 - r_4\cos\theta_4}{r_3}$$

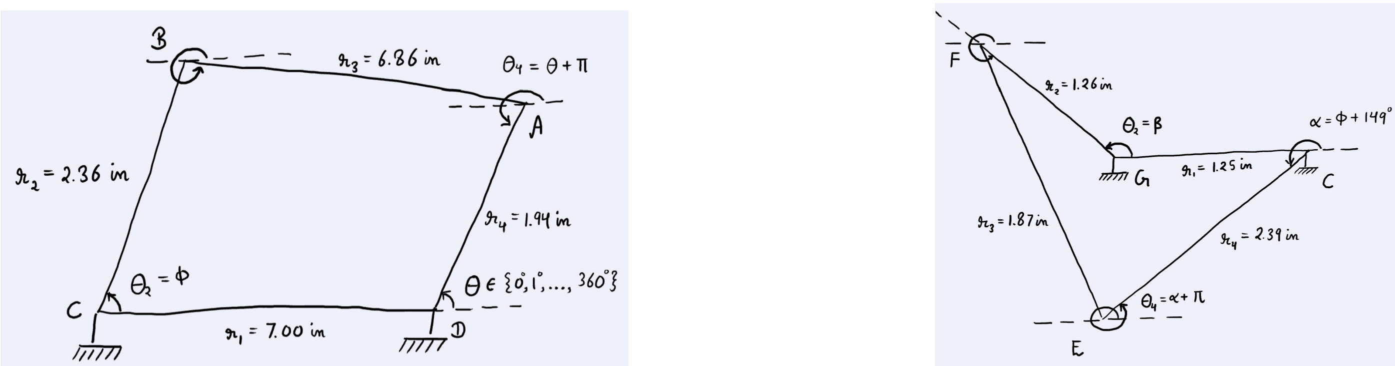
From 2^{nd} equation we get,

$$\sin\theta_3 = \frac{-r_2\sin\theta_2 - r_4\sin\theta_4}{r_3}$$

Using the equation $\sin^2\theta_3 + \cos^2\theta_3 = 1$, we get,

$$f(\theta_2) = r_1^2 + r_2^2 - r_3^2 + r_4^2 + 2\left(r_2 r_4 \sin\theta_2 \sin\theta_4 + r_2 r_4 \cos\theta_2 \cos\theta_4 - r_1 r_2 \cos\theta_2 - r_1 r_4 \cos\theta_4\right) = 0 \tag{1}$$

Hence, for given assignment we will be using Eq. (1) for the function evaluations and Newton iterations.



(a) Similarities and Values relationship of the given **Problem 1** with the Problem 3 of Homework 2.

(b) Similarities and Values relationship of the given **Problem 2** with the Problem 3 of Homework 2.

Figure 1: Correlation of the Problems with Problem 3 of Homework 2.

Formula used

$$1^{st} \text{ derivative using Forward Difference Scheme: } y_i' = \frac{dy}{dx} = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = \frac{y_{i+1} - y_i}{h} \text{ (for equispaced grids)}$$

$$1^{st} \text{ derivative using Central Difference Scheme: } y_i' = \frac{dy}{dx} = \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}} = \frac{y_{i+1} - y_{i-1}}{2h} \text{ (for equispaced grids)}$$

$$2^{nd} \text{ derivative using Forward Difference Scheme (equispaced grids): } y_i'' = \frac{d^2y}{dx^2} = \frac{y_{i+2} - 2y_{i+1} + y_i}{h^2}$$

$$2^{nd} \text{ derivative using Central Difference Scheme (equispaced grids): } y_i'' = \frac{d^2y}{dx^2} = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}$$

Solution 1

Given values which are needed for solving Four Bar Linkage Problem defined in Eq. (1) are as follows,

1. $r_1 = 7.00$ in
2. $r_2 = 2.36$ in
3. $r_3 = 6.86$ in
4. $r_4 = 1.94$ in

5. θ increments from 0^o to 360^o in gaps of 1^o
6. $\theta_4 = \theta + \pi$
7. $\theta_2 = \phi$ which is to be found out.
8. Assumed initial value, $\theta_2 = 30^o$

Fig. 2 depicts the relationship between ϕ and θ,
Fig. 3 depicts the first derivative of ϕ with respect to θ versus θ and
Fig. 4 depicts the absolute error between values of $d\phi/d\theta$ found using forward and central difference schemes.

Results:

2 graphs obtained using forward difference scheme and central difference scheme turns out to be **nearly equal** with **maximum absolute difference** being **0.0175**.
Here we have equispaced grids which is equal to $\mathbf{h = 1^o = 0.0174533}$ **radians**.
Forward difference scheme have error of order $\mathbf{o(h) = o(0.0174533)}$ and **Central difference scheme** have error of order $\mathbf{o(h^2) = o(0.0003046177)}$.
Analytically, here $h < 1$, which means $h^2 < h$ and hence, $o(h^2) < o(h)$.
From Fig. 4 also it is clear that,

$$o(h) - o(h^2) \approx o(h) = o(0.0174533) \approx o(0.0175)$$

It is clear that central difference scheme yields lesser error. Hence, **central difference scheme is more accurate**.

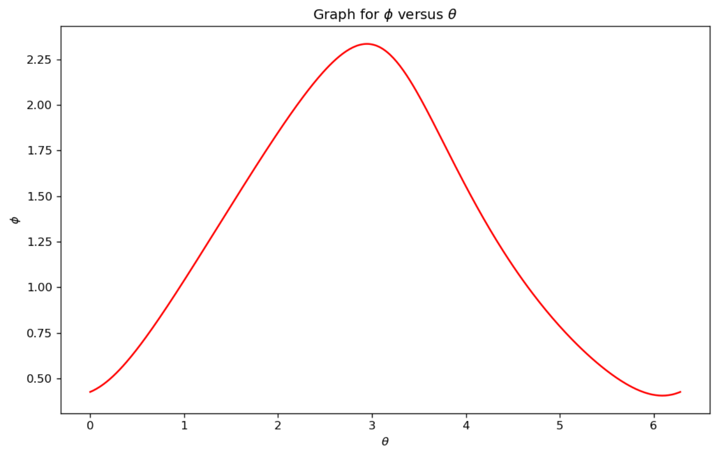


Figure 2: Values of ϕ versus θ (both in radians)

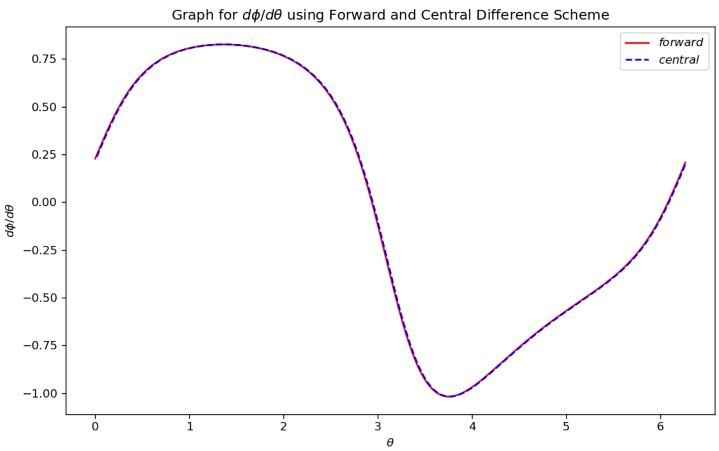


Figure 3: Values of $d\phi/d\theta$ versus θ (θ in radians)

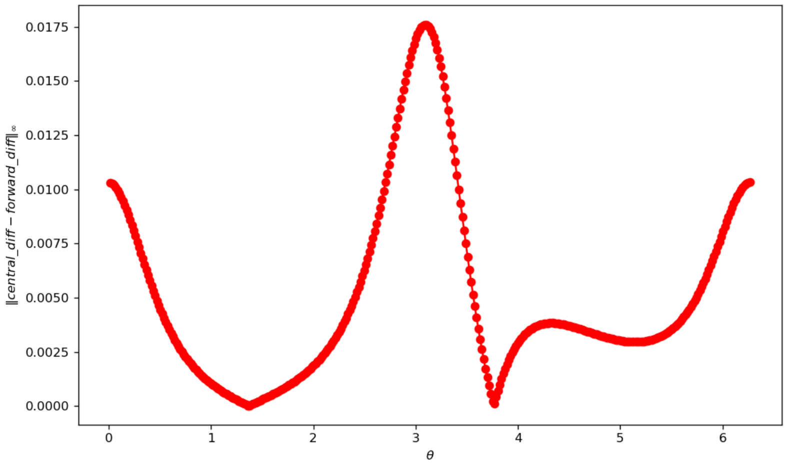


Figure 4: Absolute Difference between $d\phi/d\theta$ found using forward and central difference scheme (θ in radians)

Solution 2

Given values which are needed for solving Four Bar Linkage Problem defined in Eq. (1) are as follows,

1. $r_1 = 1.25$ in
2. $r_2 = 1.26$ in
3. $r_3 = 1.87$ in
4. $r_4 = 2.39$ in
5. $\alpha = \phi + 149^o$ where, ϕ obtained from Question 1
6. $\theta_4 = \alpha + \pi$
7. $\theta_2 = \beta$ which is to be found out.
8. Assumed initial value, $\theta_2 = 30^o$

Fig. 5 depicts the relationship between β and θ,
Fig. 6 depicts the angular velocity versus θ and
Fig. 7 depicts the angular acceleration versus θ relation.
We can find $\frac{d\beta}{d\theta}$ and $\frac{d^2\beta}{d\theta^2}$ similar to the previous question, and other equations required to find angular velocity and angular acceleration are as follows,

$$\text{Angular Velocity (rad/sec)}, \frac{d\beta}{dt} = \omega \frac{d\beta}{d\theta} \quad \text{and} \quad \text{Angular Acceleration (rad/sec}^2\text{)}, \frac{d^2\beta}{dt^2} = \omega^2 \frac{d^2\beta}{d\theta^2}$$

$$\text{Given, } \omega = 450 \text{ rad/minute} = \frac{450}{60} \text{ rad/sec} = 7.5 \text{ rad/sec}$$

Results: At $\theta = 100^o$, obtained values are as follows,

Angular Velocity (Forward difference scheme): **10.599979527512152 rad/sec**
Angular Velocity (Central difference scheme): **10.632997083521774 rad/sec**

Angular Acceleration (Forward difference scheme): $-$**28.126555797477938 rad/sec**^{2}
Angular Acceleration (Central difference scheme): $-$**28.37649913782144 rad/sec**^{2}

Forward and Central difference schemes both yields almost same plots for angular velocity and angular acceleration. But, central difference scheme must be used due to higher accuracy when $h < 1$. Also, we have to make sure that when $h > 1$ we must shift to forward difference scheme for better accuracy.

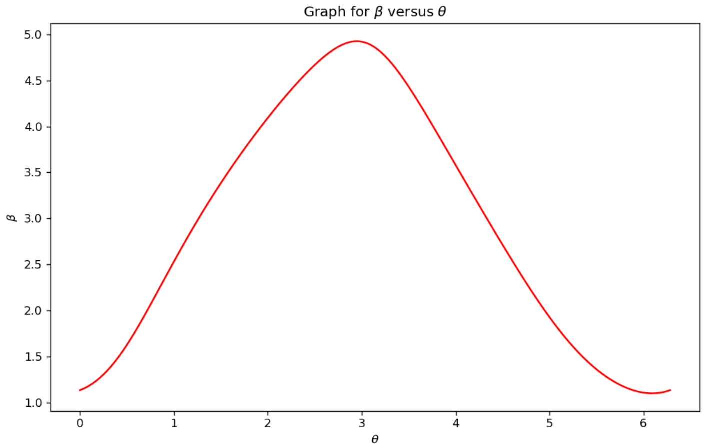


Figure 5: Values of β versus θ (both in radians)

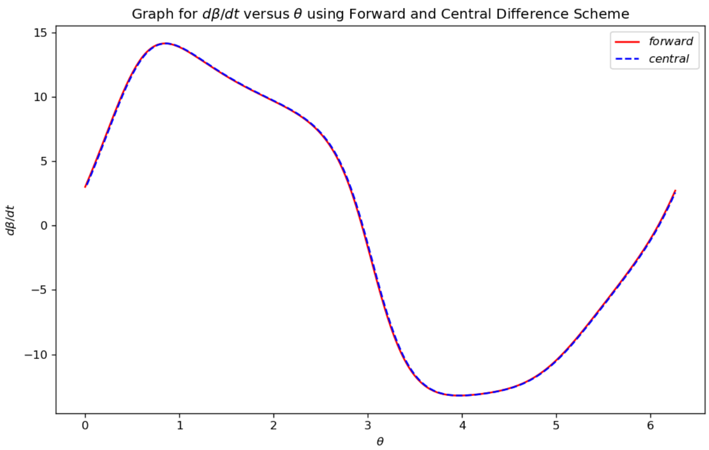


Figure 6: Angular Velocity: Values of $d\beta/dt$ versus θ (θ in radians)

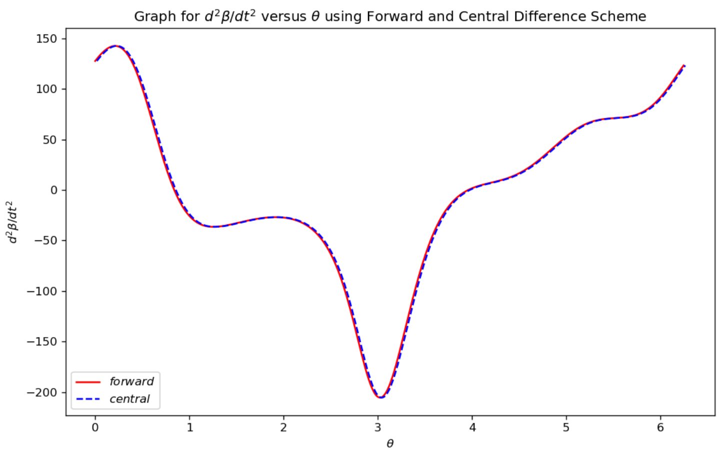
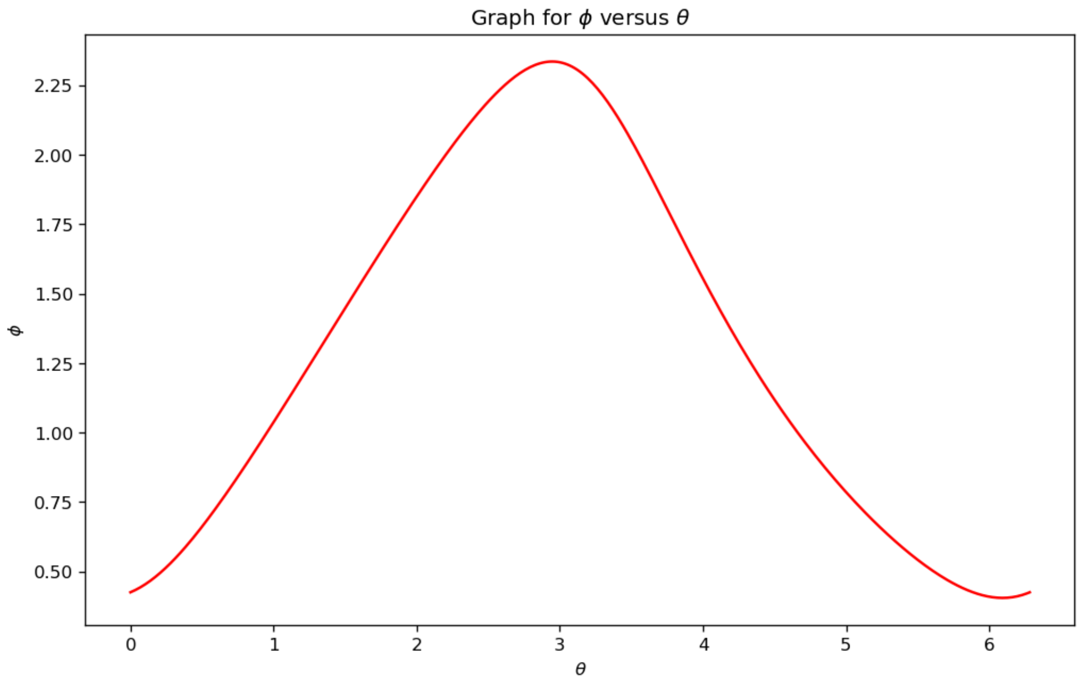


Figure 7: Angular Acceleration: Values of $d^2\beta/dt^2$ versus θ (θ in radians)

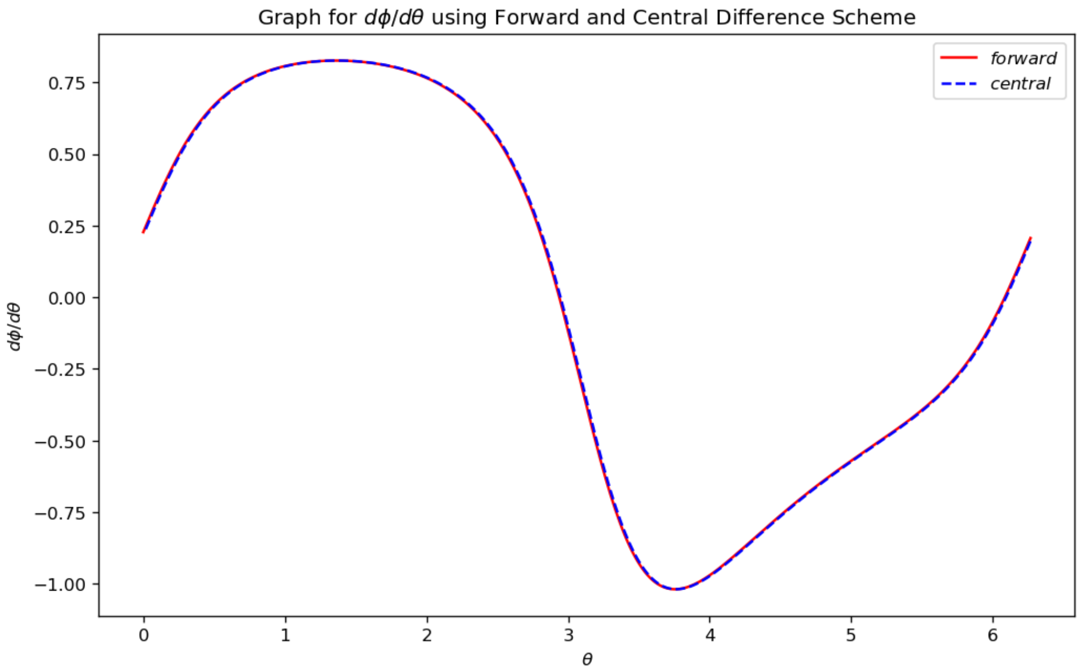
1 Appendix: Assignment 4 Programming

1.1 Ques 1

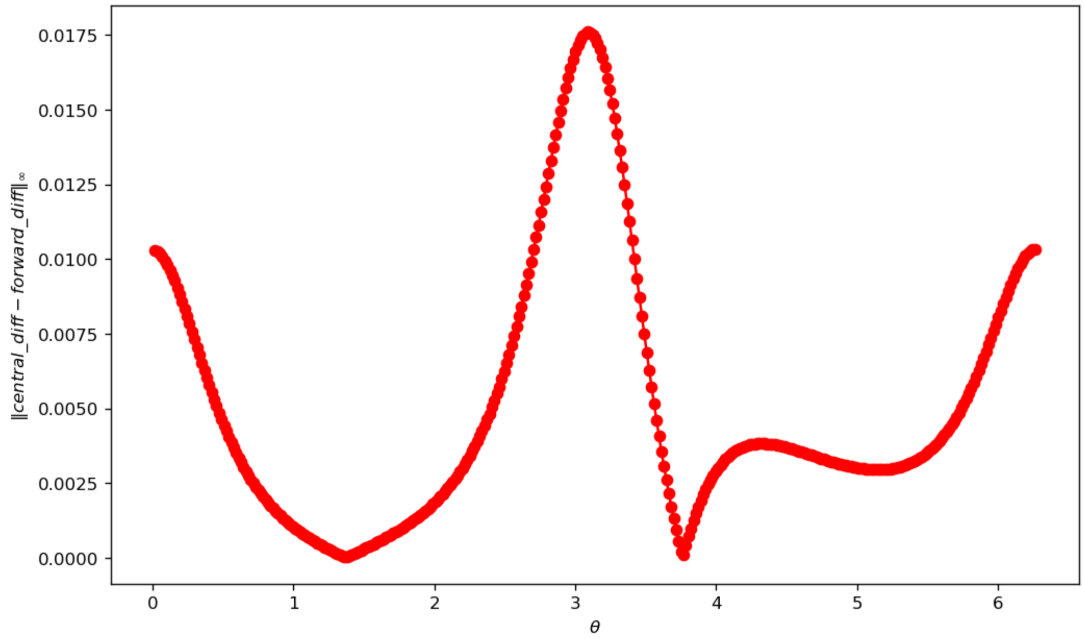
[1]:
```python
import sympy as sp
import math
import matplotlib.pyplot as plt

x = sp.symbols('x')

def find_phi(theta, theta2_val):
    r1, r2, r3, r4 = 7.00, 2.36, 6.86, 1.94
    theta4 = math.pi + theta
    tol = 1e-10
    max_iterations = 100

    f = (r1**2 + r2**2 + r4**2 - r3**2) + 2*(r2*r4*sp.sin(x)*sp.sin(theta4) + r2*r4*sp.
 ↪cos(x)*sp.cos(theta4) - r1*r2*sp.cos(x) - r1*r4*sp.cos(theta4))
    fd = 2*(r2*r4*sp.cos(x)*sp.sin(theta4) - r2*r4*sp.sin(x)*sp.cos(theta4) + r1*r2*sp.
 ↪sin(x))

    iter = 0
    p = [theta2_val]
    while iter < max_iterations:
        p_new = p[iter] - (f.subs(x,p[iter])/fd.subs(x,p[iter]))
        p_new = sp.N(p_new)
        p.append(p_new)
        if abs((p[iter+1]-p[iter])/p[iter+1])<=tol:
            iter = iter+1
            break
        iter = iter+1
    return p[-1]

theta = []
phi = []
theta2_val = sp.rad(30)
for i in range(361):
    theta.append(sp.rad(i))
    theta2_val = find_phi(theta[i], theta2_val)
    phi.append(theta2_val)

plt.figure(figsize=(10, 6), dpi=120)
plt.plot(theta, phi,'r', label = r"$\phi$")
plt.title(r"Graph for $\phi$ versus $\theta$")
plt.xlabel(r"$\theta$")
plt.ylabel(r"$\phi$")
plt.savefig('phi.png')
plt.show()
```



[2]:
```
def forward(phi, theta):
    diff = []
    for i in range(len(phi)-1):
        diff.append((phi[i+1] - phi[i])/(theta[i+1] - theta[i]))
    return diff

def central(phi, theta):
    diff = []
    for i in range(1, len(phi)-1):
        diff.append((phi[i+1] - phi[i-1])/(theta[i+1] - theta[i-1]))
    return diff

forward_derivative = forward(phi, theta)
central_derivative = central(phi, theta)

plt.figure(figsize=(10, 6), dpi=120)
plt.plot(theta[:len(theta)-1], forward_derivative, 'r', label = r"$forward$")
plt.plot(theta[1:len(theta)-1], central_derivative,'b--', label = r"$central$")
plt.legend()
plt.title(r"Graph for $d\phi / d\theta$ using Forward and Central Difference Scheme")
plt.xlabel(r"$\theta$")
plt.ylabel(r"$d\phi / d\theta$")
plt.savefig('diff1.png')
plt.show()
```



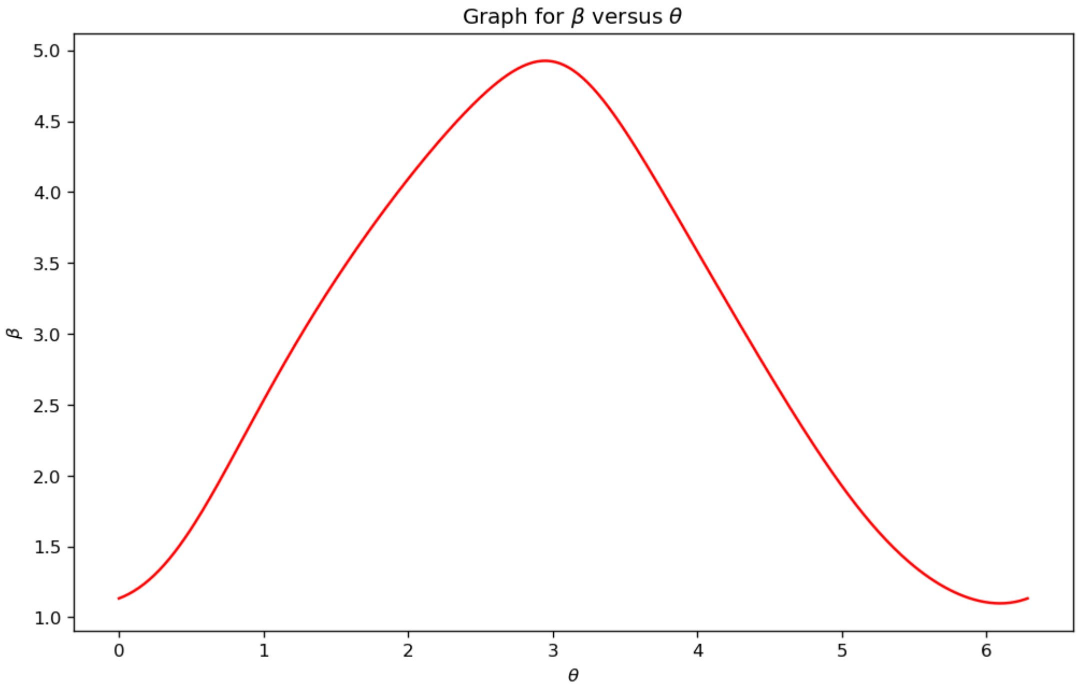
[3]:
```
error = []
for i in range(len(central_derivative)):
    error.append(abs(central_derivative[i] - forward_derivative[i+1]))
plt.figure(figsize=(10, 6), dpi=120)
plt.plot(theta[1:len(theta)-1], error,'ro-')
plt.xlabel(r"$\theta$")
plt.ylabel(r"$\|central\_diff - forward\_diff\|_\infty$")
plt.savefig('abserr1.png')
plt.show()
```



1.2 Ques 2

[4]:
```
x = sp.symbols('x')

def find_betax(alpha, theta2_val):
    r1, r2, r3, r4 = 1.25, 1.26, 1.87, 2.39
    theta4 = math.pi + alpha
    tol = 1e-10
    max_iterations = 100

    f1 = (r1**2 + r2**2 + r4**2 - r3**2) + 2*(r2*r4*sp.sin(x)*sp.sin(theta4) +
 ↪r2*r4*sp.cos(x)*sp.cos(theta4) - r1*r2*sp.cos(x) - r1*r4*sp.cos(theta4))
    f1p = 2*(r2*r4*sp.cos(x)*sp.sin(theta4) - r2*r4*sp.sin(x)*sp.cos(theta4) +
 ↪r1*r2*sp.sin(x))

    iter = 0
    p = [theta2_val]
    while iter < max_iterations:
        p_new = p[iter] - (f1.subs(x,p[iter])/f1p.subs(x,p[iter]))
        p_new = sp.N(p_new)
        p.append(p_new)
        if abs((p[iter+1]-p[iter])/p[iter+1])<=tol:
            iter = iter+1
            break
        iter = iter+1
    return p[-1]

alpha = phi
```

```python
for i in range(len(phi)):
    alpha[i] = phi[i] + sp.rad(149)

beta = []
theta2_val = sp.rad(30)
for i in range(len(alpha)):
    theta2_val = find_betax(alpha[i], theta2_val)
    beta.append(theta2_val)

plt.figure(figsize=(10, 6), dpi=120)
plt.plot(theta, beta,'r', label = r"$\beta$")
plt.title(r"Graph for $\beta$ versus $\theta$")
plt.xlabel(r"$\theta$")
plt.ylabel(r"$\beta$")
plt.savefig('beta.png')
plt.show()
```



[5]:
```python
forward_deri = forward(beta, theta)
central_deri = central(beta, theta)

forward_der = forward_deri
central_der = central_deri
for i in range(len(forward_deri)):
    forward_der[i] = 7.5*forward_deri[i]
for i in range(len(central_der)):
    central_der[i] = 7.5*central_deri[i]
```

```
# Specific for equispaced grids
def forwardDouble(beta, theta):
    diff = []
    for i in range(len(beta)-2):
        diff.append((beta[i+2] - 2*beta[i+1] + beta[i])/((theta[i+1] - theta[i])**2))
    return diff

# Specific for equispaced grids
def centralDouble(beta, theta):
    diff = []
    for i in range(1, len(beta)-1):
        diff.append((beta[i+1] - 2*beta[i] + beta[i-1])/((theta[i+1] - theta[i])**2))
    return diff

forward_dderi = forwardDouble(beta, theta)
central_dderi = centralDouble(beta, theta)

forward_dder = forward_dderi
central_dder = central_dderi
for i in range(len(forward_dderi)):
    forward_dder[i] = (7.5**2)*forward_dderi[i]
for i in range(len(central_dderi)):
    central_dder[i] = (7.5**2)*central_dderi[i]

print("\nForward difference scheme (1st derivative): ", float(forward_der[100]))
print("Central difference scheme (1st derivative): ", float(central_der[99]))
print("\nForward difference scheme (2nd derivative): ", float(forward_dder[100]))
print("Central difference scheme (2nd derivative): ", float(central_dder[99]), "\n")

plt.figure(figsize=(10, 6), dpi=120)
plt.plot(theta[:len(theta)-1], forward_deri, 'r', label = r"$forward$")
plt.plot(theta[1:len(theta)-1], central_deri,'b--', label = r"$central$")
plt.legend()
plt.title(r"Graph for $d\beta / dt$ versus $\theta$ using Forward and Central␣
 ↪Difference Scheme")
plt.xlabel(r"$\theta$")
plt.ylabel(r"$d\beta / dt$")
plt.savefig('diff2.png')
plt.show()

plt.figure(figsize=(10, 6), dpi=120)
plt.plot(theta[:len(theta)-2], forward_dderi, 'r', label = r"$forward$")
plt.plot(theta[1:len(theta)-1], central_dderi,'b--', label = r"$central$")
plt.legend()
plt.title(r"Graph for $d^2\beta / dt^2$ versus $\theta$ using Forward and Central␣
 ↪Difference Scheme")
plt.xlabel(r"$\theta$")
plt.ylabel(r"$d^2\beta / dt^2$")
plt.savefig('ddiff2.png')
plt.show()
```

```
Forward difference scheme (1st derivative):  10.599979527512152
Central difference scheme (1st derivative):  10.632997083521774
```

```
Forward difference scheme (2nd derivative):  -28.126555797477938
Central difference scheme (2nd derivative):  -28.37649913782144
```