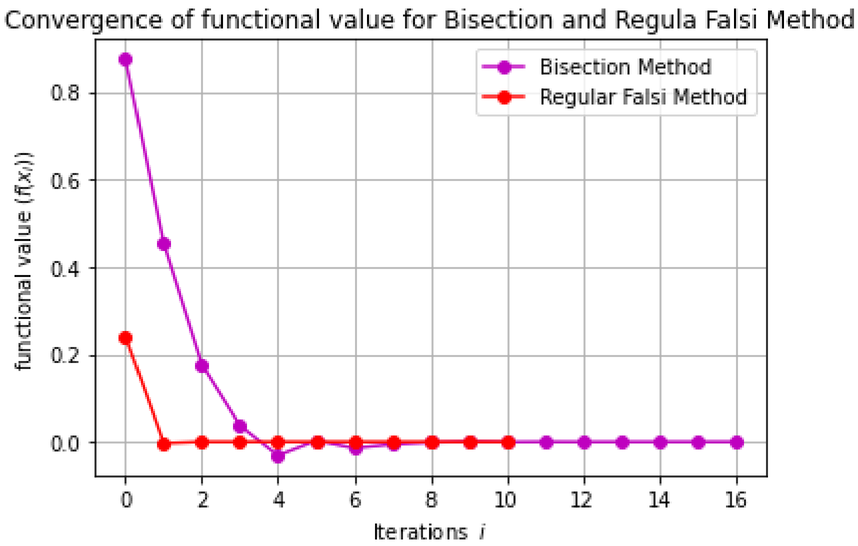
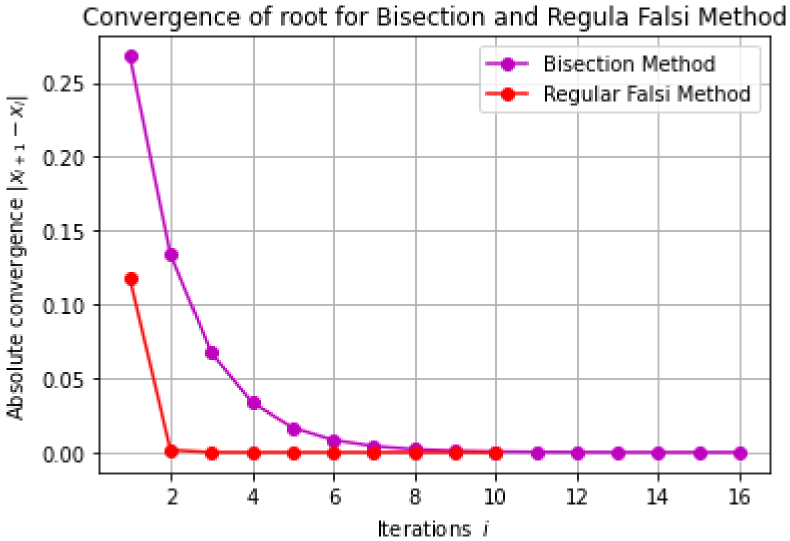
In [2]:

```
#Question 1 :
# Solve the following equation to an accuracy of 10^(-5), starting from an initial gues
s
# interval [1.6, 2.4],log(x/2) - sin(5x/2) = 0.
# If the given interval does not bracket a root, numerically determine an intervalthat
 will.
# Use both Bisection and Regula-falsi method to solve the problem and compare them with
a plot (f(xi) vs i)
# and a table showing convergence to a root x_i against number of steps i.



from My_Lib import *

import math
import matplotlib.pyplot as plt

# Function definition
def f(x):
    return math.log(x/2) - math.sin(5*x/2) #Given function


root_val_1, x_error_1,f_x_i_1, iterations_1 = bisection_method(f, 1.6, 2.4, 10**(-5))
#calling bisection function
root_val_2, x_error_2,f_x_i_2, iterations_2 = regula_falsi_method(f, 1.6, 2.4, 10**(-5
)) #calling Regul falsi function



print("Root of the equation via Bisection Method is: ", root_val_1) #Printing root valu
e for bisection
print("Root of the equation via Regula Falsi Method is: ", root_val_2)#Printing root va
lue for Regula Falsi



################################################################################

#
# MAKING TABLES FOR BOTH THE METHODS
#




print ("\n------------Bisection Method--------------")
print("| |x_(i+1)-x_(i)\t|\ti\t|\t")
print ("-----------------------------------------")
for i in range (1,len(iterations_1)):
    print("|\t%.5f"%x_error_1[i],"\t|\t",iterations_1[i],"\t|")

print ("-----------------------------------------")

print ("------------Regula Falsi Method----------")
print("| |x_(i+1)-x_i\t|\t|\ti\t|\t")
print ("-----------------------------------------")
for i in range (1,len(iterations_2)):
    print("|\t%.11f"%x_error_2[i],"\t|\t",iterations_2[i],"\t|")

print ("-----------------------------------------")
```

```
##############################################################################3

#Plotting of Two methods:

plt.plot(iterations_1, f_x_i_1,'m-o', label="Bisection Method")
plt.plot(iterations_2, f_x_i_2,'r-o' ,label="Regular Falsi Method")

plt.title("Convergence of functional value for Bisection and Regula Falsi Method")
plt.xlabel("Iterations  $i$")
plt.ylabel("functional value ($f(x_i)$)")
plt.legend()
plt.grid()
plt.show()

# Removing zeroes from the lists
iterations_2.remove(0)
iterations_1.remove(0)
x_error_2.remove(0)
x_error_1.remove(0)

# plotting convergence of roots

plt.plot(iterations_1, x_error_1,'m-o', label="Bisection Method")
plt.plot(iterations_2, x_error_2,'r-o' ,label="Regular Falsi Method")

plt.title("Convergence of root for Bisection and Regula Falsi Method")
plt.xlabel("Iterations  $i$")
plt.ylabel("Absolute convergence $|x_{i+1}-x_{i}|$")
plt.legend()
plt.grid()
plt.show()
```

```
Root of the equation via Bisection Method is:  2.6231397924202913
Root of the equation via Regula Falsi Method is:  2.6231403354363083

------------Bisection Method-------------
| |x_(i+1)-x_(i)         |        i        |
-----------------------------------------
|       0.26802          |        1        |
|       0.13401          |        2        |
|       0.06700          |        3        |
|       0.03350          |        4        |
|       0.01675          |        5        |
|       0.00838          |        6        |
|       0.00419          |        7        |
|       0.00209          |        8        |
|       0.00105          |        9        |
|       0.00052          |        10       |
|       0.00026          |        11       |
|       0.00013          |        12       |
|       0.00007          |        13       |
|       0.00003          |        14       |
|       0.00002          |        15       |
|       0.00001          |        16       |
-----------------------------------------
------------Regula Falsi Method----------
| |x_(i+1)-x_i  |        |        i        |
-----------------------------------------
|       0.11755051207    |        1        |
|       0.00162033161    |        2        |
|       0.00004351404    |        3        |
|       0.00000115777    |        4        |
|       0.00000003080    |        5        |
|       0.00000000082    |        6        |
|       0.00000000002    |        7        |
|       0.00000000000    |        8        |
|       0.00000000000    |        9        |
|       0.00000000000    |        10       |
-----------------------------------------
```

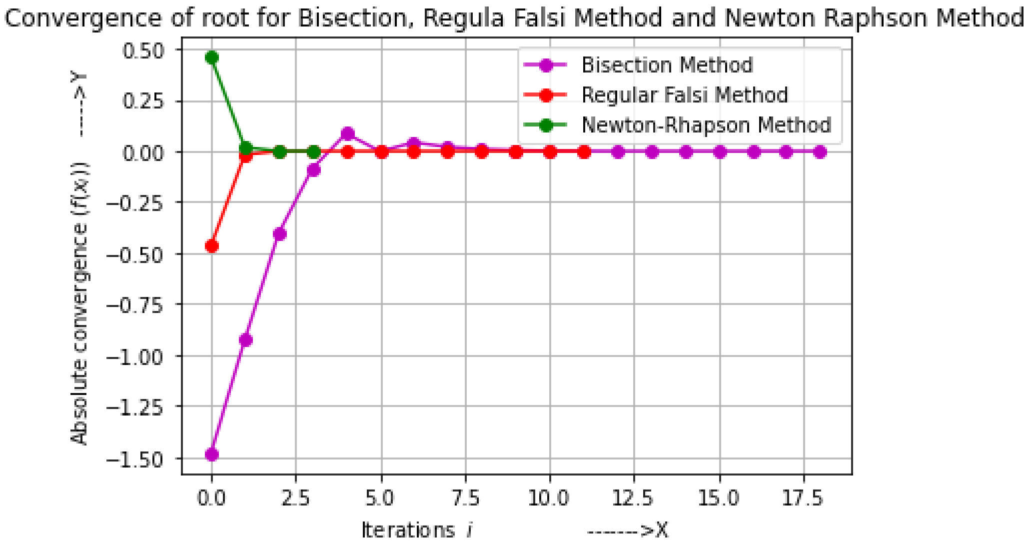



Convergence of root for Bisection and Regula Falsi Method
Bisection Method
Regular Falsi Method
Absolute convergence $|x_{i+1} - x_i|$
Iterations i
0.25
0.20
0.15
0.10
0.05
0.00
2
4
6
8
10
12
14
16

In [1]:

```
#Question : 2
#Make appropriate initial guesses (same interval for Bisection and Regula-falsi)
# and solve the following equation,
# -x - cos x = 0

#Use all three methods Bisection, Regula-falsi and Newton-Raphson to solve it.
#For Newton-Raphson, use x=0.0 as initial guess. Compare all three methods to
#achieve the same accuracy as above in Q1.

from My_Lib import *

import math
import matplotlib.pyplot as plt

# Function definition
def f(x):
    return -x-math.cos(x)   #Given Function


root_val_1, x_error_1,f_x_i_1, iterations_1= bisection_method(f, 1.6, 2.4, 10**(-5))  #
calling bisection
root_val_2, x_error_2,f_x_i_2, iterations_2= regula_falsi_method(f, 1.6, 2.4, 10**(-5))
#calling regula falsi method
root_val_3,f_x_i_3,iterations_3=Newton_Raphson_Method(f,0,10**(-5))#calling Newton Rhap
son method



print("Root of the equation via Bisection Method is: ", root_val_1) #printing roots for
Bisection method
print("Root of the equation via Regula Falsi Method is: ", root_val_2) #Printing roots
 for Regula Falsi method
print("Root of the equation via Newton Rhapson method is:",root_val_3) #Printing roots
 for Newton Rhapson Method


#Plotting of Three methods:

plt.plot(iterations_1, f_x_i_1,'m-o', label="Bisection Method")
plt.plot(iterations_2, f_x_i_2,'r-o' ,label="Regular Falsi Method")
plt.plot(iterations_3, f_x_i_3,'g-o' ,label="Newton-Rhapson Method")
print("\nComparison of convergence of functional value : As we can see for every method
the graph converges to 0.")
plt.title("Convergence of root for Bisection, Regula Falsi Method and Newton Raphson Me
thod")
plt.xlabel("Iterations  $i$                  ------->X")
plt.ylabel("Absolute convergence ($f(x_i)$)    ----->Y")
plt.legend()
plt.grid()
plt.show()
```

```
Root of the equation via Bisection Method is:  -0.7390872530592778
Root of the equation via Regula Falsi Method is:  -0.7390851332151607
Root of the equation via Newton Rhapson method is: -0.7390851333911672

Comparison of convergence of functional value : As we can see for every me
thod the graph converges to 0.
```



In [2]:

```
#Question 3:

#Find the roots (all real) of the following polynomial using the Laguerre’s and
#synthetic division method.
#P(x) = x^4 − 5*x^2 + 4

from My_Lib import*

a_i=[1,0,-5,0,4]  # Co efficients of the given function
list_roots=driver_function_lag(a_i,0,10**(-5)) #calling Lauguerre function, intital gue
ss=0, precision 10^(-5)

print("The roots of the given equation by Lauguerre method :") # Printing the roots

print("x1= %.3f"%list_roots[0])
print("x2= %.3f"%list_roots[1])
print("x3= %.3f"%list_roots[2])
print("x4= %.3f"%list_roots[3])
```

```
The roots of the given equation by Lauguerre method :
x1= 1.000
x2= -1.000
x3= 2.000
x4= -2.000
```