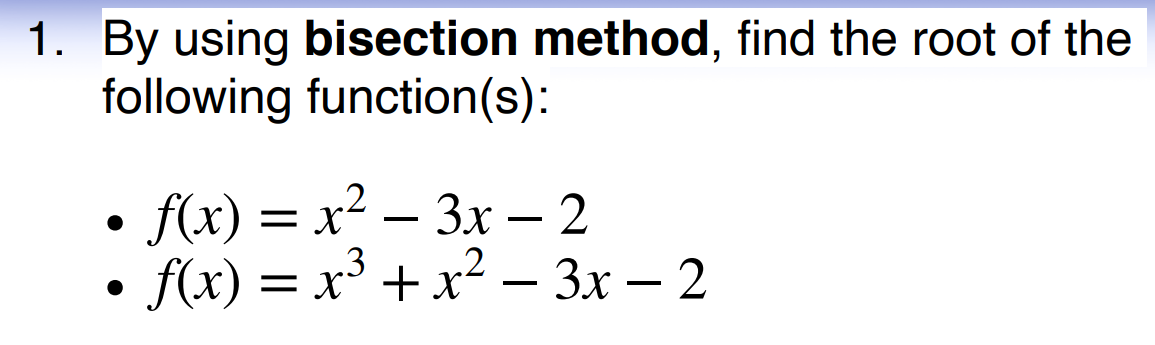
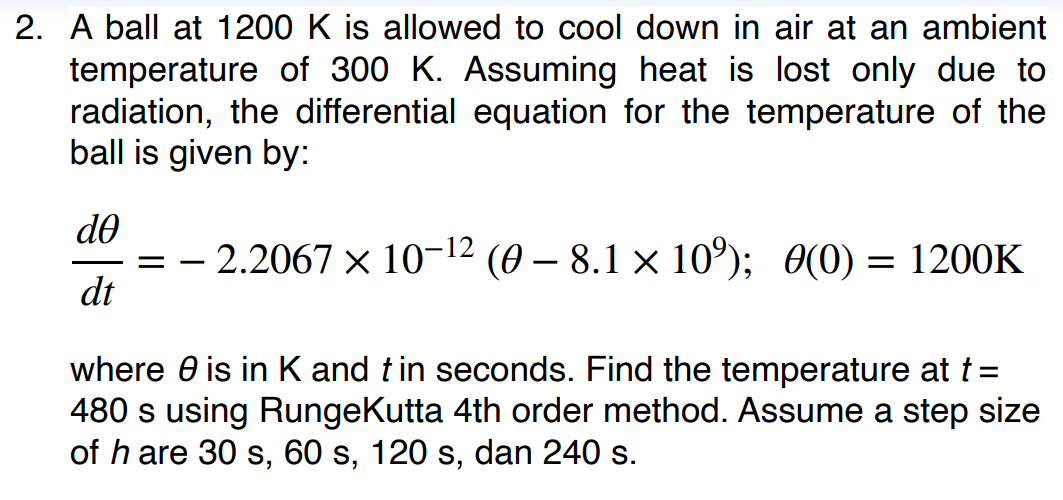
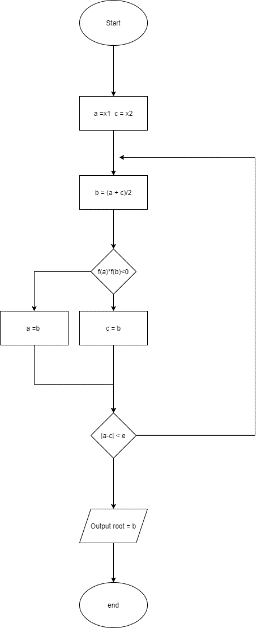
Phil Gandeng 11417001 11 Februari 2021



1. Algoritma (Bisection Method)



2. Output Keluaran (Bisection Method)

import math

# Fungi yang akan digunakan

def f(x):

return x\*\*2 - 3\*x -2

# Penentuan metode bisection

def bisection(x0,x1,e):

step = 1

condition = True

while condition:

x2 = (x0 + x1)/2

print('Iterasi ke -%d, x2 = %0.6f dan f(x2) = %0.6f' % (step, x2, f(x2)))

if f(x0) \* f(x2) < 0:

x1 = x2

else:

x0 = x2

step = step + 1

condition = abs(f(x2)) > e

print('\nAkar yang dibutuhkan : %0.8f' % x2)

# Baris untuk memasukkan nilai

x0 = input('Titik Awal: ')

x1 = input('Titik Kedua: ')

e = input('Batasan error: ')

# Merubah nilai input menjadi float

x0 = float(x0)

x1 = float(x1)

e = float(e)

# Pengecekan awal nilai bisection

if f(x0) \* f(x1) > 0.0:

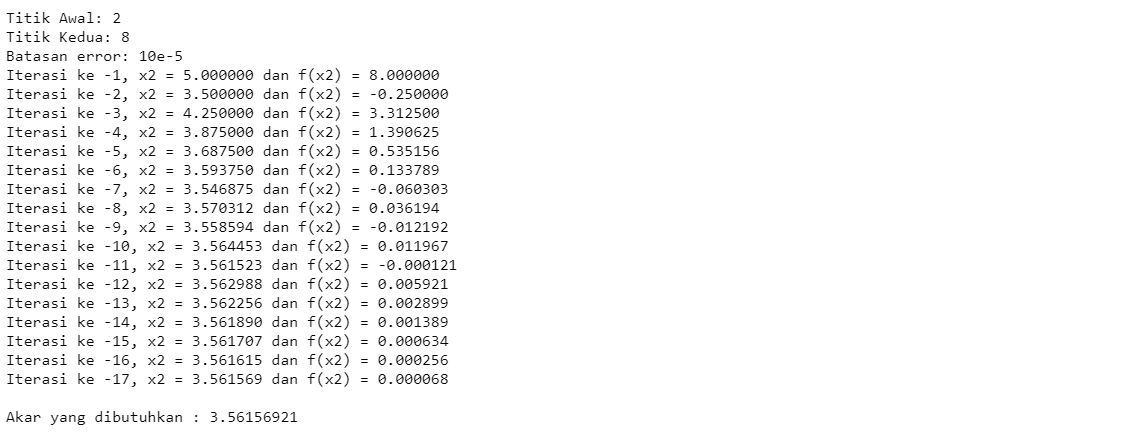
print('Nilai yang dimasukkan tidak masuk dalam kurungan.')

print('Coba dengan nilai baru.')

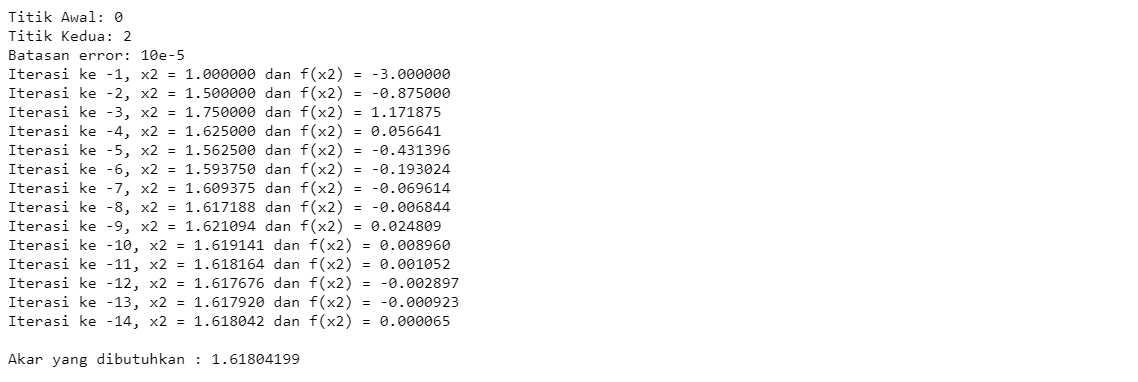
else:

bisection(x0,x1,e)

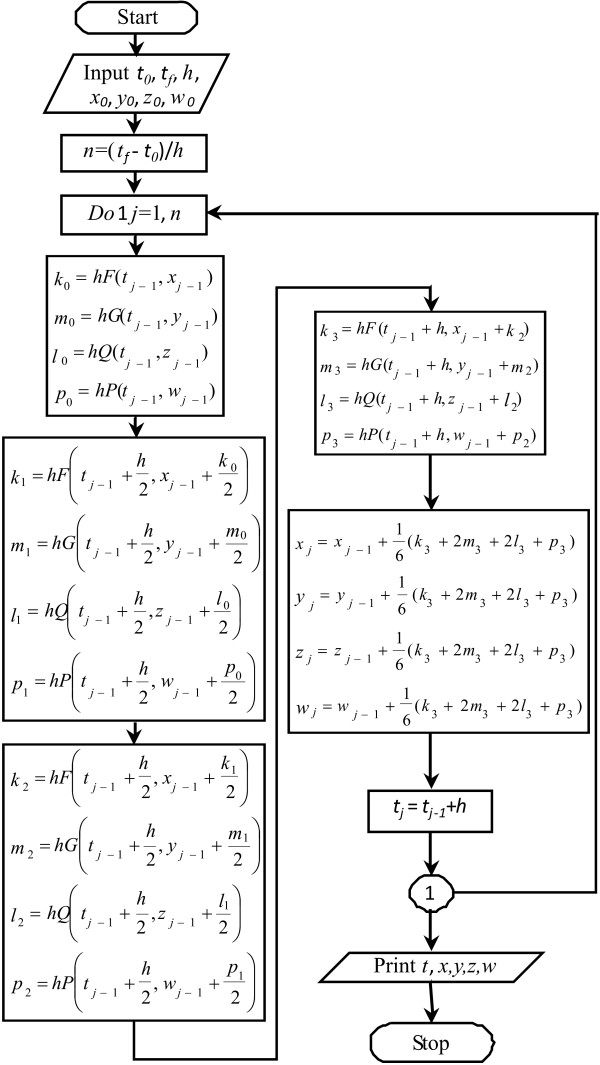
Untuk persamaan pertama outputnya:



Untuk persamaan kedua outputnya:



1. Algortima ( Runge – Kutta)



2.

# RK-4 method python program

# function to be solved

def f(x,y):

return (-2.2067 \* (10\*\*-12) \* (y\*\*4-81\*(10\*\*9)))

# or

# f = lambda x: x+y

# RK-4 method

def rk4(x0,y0,xn,n):

# Calculating step size

h = (xn-x0)/n

print('\n--------SOLUTION--------')

print('-------------------------')

print('x0\ty0\tyn')

print('-------------------------')

for i in range(n):

k1 = h \* (f(x0, y0))

k2 = h \* (f((x0+h/2), (y0+k1/2)))

k3 = h \* (f((x0+h/2), (y0+k2/2)))

k4 = h \* (f((x0+h), (y0+k3)))

k = (k1+2\*k2+2\*k3+k4)/6

yn = y0 + k

print('%.4f\t%.4f\t%.4f'% (x0,y0,yn) )

print('-------------------------')

y0 = yn

x0 = x0+h

print('\nAt x=%.4f, y=%.4f' %(xn,yn))

# Inputs

print('Enter initial conditions:')

x0 = float(input('x0 = '))

y0 = float(input('y0 = '))

print('Enter calculation point: ')

xn = float(input('xn = '))

print('Enter number of steps:')

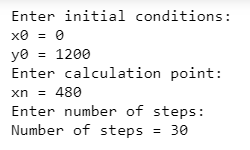
step = int(input('Number of steps = '))

# RK4 method call

rk4(x0,y0,xn,step)

Hasil Output :

* Ketika step = 30





* Ketika step = 120

