

1500011370_RKV

December 17, 2018

```
In [1]: import numpy as np
import math

In [2]: def f(t, y):
return y/t-(y/t)**2

In [3]: def actual(t):
return t/(1+math.log(t))

In [4]: def rk4(h, t, w):
k1 = h * f(t, w)
k2 = h * f(t + h/6, w + k1/6)
k3 = h * f(t + 4*h/15, w + 4*k1/75 + 16*k2/75)
k4 = h * f(t + 2*h/3, w + 5*k1/6 - 8*k2/3 + 5*k3/2)
k5 = h * f(t + 5*h/6, w - 165*k1/64 + 55*k2/6 - 425*k3/64
+ 85*k4/96)
k6 = h * f(t + h, w + 12*k1/5 - 8*k2 + 4015*k3/612
- 11*k4/36 + 88*k5/255)
k7 = h * f(t + h/15, w - 8263*k1/15000 + 124*k2/75
- 643*k3/680 - 81*k4/250 + 2484*k5/10625)
k8 = h * f(t + h, w + 3501*k1/1720 - 300*k2/43 + 297275*k3/52632
- 319*k4/2322 + 24068*k5/84065 + 3850*k7/26703)
return np.asarray([k1,k2,k3,k4,k5,k6,k7,k8])

In [7]: a = 1
b = 4
alpha = 1
TOL = 1e-6
hmax = 0.5
hmin = 0.05

In [17]: h = hmax
flag = 1
t = a
w = alpha
ans = np.asarray([t,w,h])
while(flag):
k = rk4(h, t, w)
```

```

error = abs(-k[0]/160 - 125*k[2]/17952 + k[3]/144 -12*k[4]/1955
          - 3*k[5]/44 + 125*k[6]/11592 + 43*k[7]/616)
if error <= TOL:
    t = t + h
    w = w + 13*k[0]/160 + 2375*k[2]/5984 + 5*k[3]/16 + 12*k[4]/85 + 3*k[5]/44
    ans = np.vstack((ans, np.asarray([t,w,h])))
X = 0.9*math.pow((TOL/error),1/5)
if X <= 0.1:
    h = 0.5*h
elif X * h >= hmax:
    h = hmax
else:
    h = h * X

if t >= b:
    flag = 0
elif t + h >= b:
    h = b - t

if flag == 0:
    print(ans)

[[1.00000000e+00 1.00000000e+00 5.00000000e-01]
 [1.50000000e+00 1.06722938e+00 5.00000000e-01]
 [1.99955597e+00 1.18109446e+00 4.99555967e-01]
 [2.49955597e+00 1.30446325e+00 5.00000000e-01]
 [2.99955597e+00 1.42937562e+00 5.00000000e-01]
 [3.49955597e+00 1.55350803e+00 5.00000000e-01]
 [3.99955597e+00 1.67610042e+00 5.00000000e-01]
 [4.00000000e+00 1.67620852e+00 4.44032697e-04]]

```