

Problema de PVI

Marcio Barros e Claudemir Woche

5 de Outubro de 2020

1 Código Runge-Kutta

```
def RungeKutta(h, v, k, m, delta):
    list_t=[]
    S=[]
    S.append([h,v])
    list_t.append(0)
    i=0
    critico=0
    hmax=0
    while(S[i][0]>0):
        i=i+1
        list_t.append(delta*i)
        aux1=S[i-1]+(delta/2)*f(S[i-1], k, m)
        aux2=S[i-1]+delta*(-f(S[i-1],k,m)+2*f(aux1,k,m))
        S.append(S[i-1]+(delta/6)*(f(S[i-1],k,m)+4*f(aux1,k,m)+f(aux2,k,m)))
        if(S[i-1][1]*S[i][1]<0):
            critico=list_t[i-1]+delta/2
            aux1=S[i-1]+(delta/4)*f(S[i-1], k, m)
            aux2=S[i-1]+(delta/2)*(-f(S[i-1],k,m)+2*f(aux1,k,m))
            hmax=(S[i-1]+(delta/12)*(f(S[i-1],k,m)+4*f(aux1,k,m)+f(aux2,k,m)))[0]
        list_t.append(list_t[i-1]+delta/2)
        aux1=S[i-1]+(delta/4)*f(S[i-1], k, m)
        aux2=S[i-1]+(delta/2)*(-f(S[i-1],k,m)+2*f(aux1,k,m))
        S.append((S[i-1]+(delta/12)*(f(S[i-1],k,m)+4*f(aux1,k,m)+f(aux2,k,m))))
    return S[i+1],critico,hmax,list_t[i+1]
```

Figura 1: Código do método de Runge-Kutta

Podemos ver o código onde dado os parametros de altura inicial, velocidade inicial, constante k , massa e o Δt , respectivamente, o código irá obter a sequência de valores S_k e nos devolver os valores do estado final, ponto crítico, altura máxima e o tempo, respectivamente.

Aplicando para os valores de $H_0 = 200$, $v_0 = 5$, $k = 0.25$, $m = 2$, e com diferentes valores de Δt , obtemos os seguintes resultados.

Δt	t	Altura Máxima
0.1	0.45	201.19411183954935
0.01	0.485	201.20024204056818
0.001	0.4845	201.20024080254055
0.0001	0.48495	201.2002420264497

Δt	Tempo Final	Velocidade Final
0.1	7.75	-47.73764156
0.01	7.785	-47.87847853
0.001	7.7895	-4.78965418e+1
0.0001	7.789750000000001	-4.78975450e+1

Δt	Tempo Final	Nível do Mar
0.1	7.75	1.90113247
0.01	7.785	0.2278282
0.001	7.7895	1.23343821e-2
0.0001	7.789750000000001	3.60121224e-4