1 Exercícios

Use o método de integração numérica de Runge-Kutta para obter a aproximação da solução dos seguintes problemas de Cauchy

1.

$$R'(t) = U(t), \quad U'(t) = \frac{35\,(P_v - P_l) - 7U(t)^2\,(5P_l + 5\epsilon_l + 4P_v + 4\epsilon_v)}{7R(t)\,(5P_l + 5\epsilon_l + P_v + \epsilon_v)}, \quad R(0) = 1, \quad U(0) = 0.0001$$

2.

$$R'(t) = U(t), \quad U'(t) = \frac{35\,(P_v - P_l) - 7U(t)^2\,(5P_l + 5\epsilon_l + 4P_v + 4\epsilon_v)}{7R(t)\,(5P_l + 5\epsilon_l + P_v + \epsilon_v)}, \quad R(0) = 0.5, \quad U(0) = 0.146446$$

Adote os seguintes valores para os parâmetros: $P_v = 0.166391$, $P_l = 0.195091$, $\epsilon_v = 1.79634$ e $\epsilon_l = 4.34802$. Note que U é a primeira derivada de R com relação a t.

2 Resolução

O método de Runge-Kutta foi implementado em MATLAB

2.1 Exercício 1

No exercício 1, o valor do passo escolhido foi $h = 0.1$ e o intervalo de simulação foi definido como $t = [0, 12.5]$, pois após este limite superior, o resultado iria pra infinito.

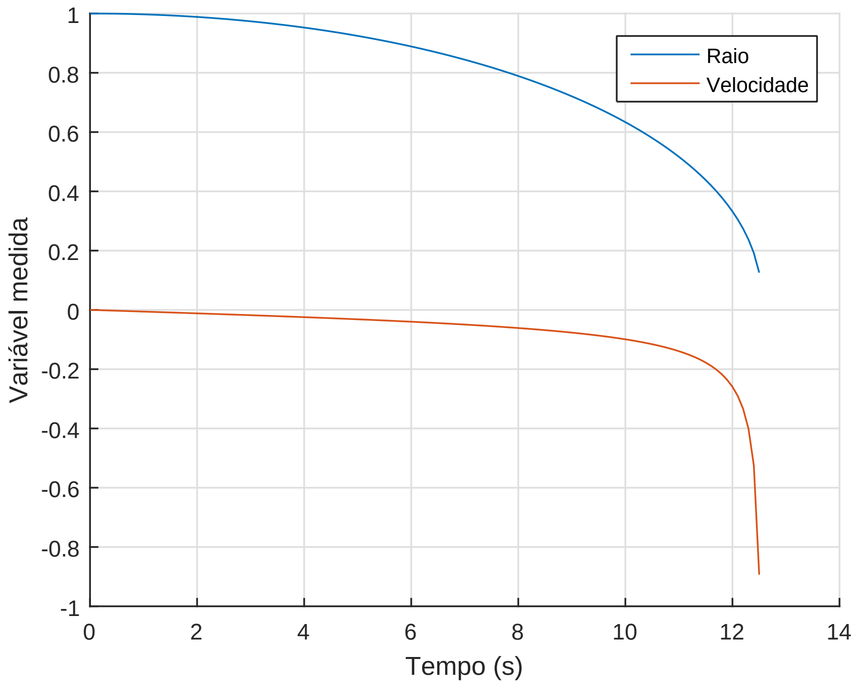

Figura 1: Variáveis medidas no exercício 1.

2.2 Exercício 2

No exercício 2, o valor do passo escolhido foi $h = 0.2$ e o intervalo de simulação foi definido como $t = [0, 22]$, pois após este limite superior, o resultado iria pra infinito.

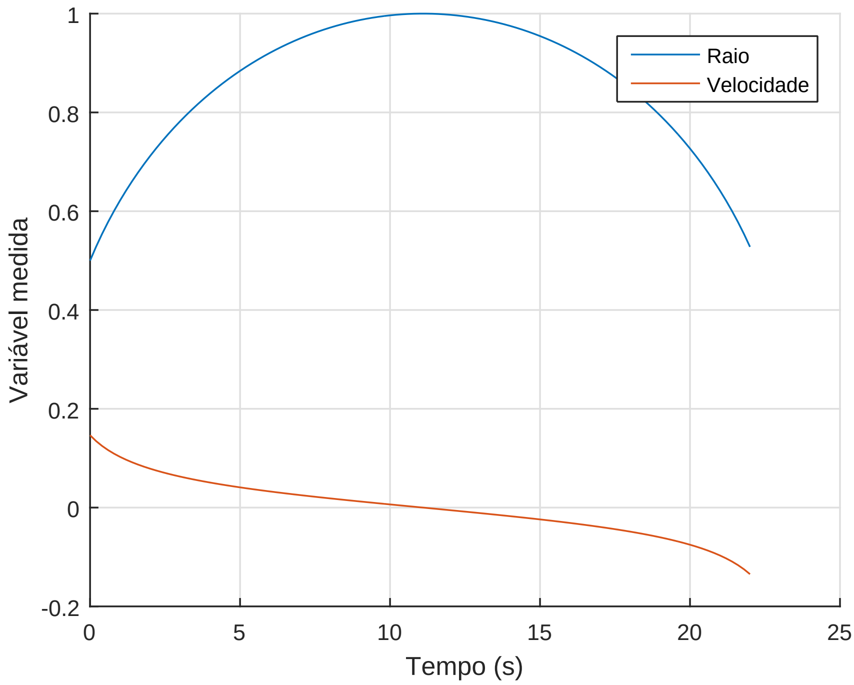

Figura 2: Variáveis medidas no exercício 2.

2.3 Código em MATLAB

```
clear
clc
format long

addpath('../../tools');

%% Parâmetros iniciais da equação da bolha
Pl = 0.195091;
Pv = 0.166391;
el = 4.34802;
ev = 1.79634;

h = 0.1;
x0 = [1 0.0001];
tmin = 0;
tmax = 12.5;

f = @(t,X) [X(2) (35*(Pv-Pl) - 7*X(2)^2*(5*Pl + 5*el + 4*Pv + 4*ev))/(7*X
    (1)*(5*Pl+5*el+Pv+ev))]; % [R;U]
x = rk4(f, h, x0, [tmin tmax]); % Resolve sistema f usando Runge-Kutta de
    quarta ordem

%% Plota as variáveis analisadas
hold on
plot(tmin:h:tmax, x(:, 1));
plot(tmin:h:tmax, x(:, 2));
xlabel('Tempo');
ylabel('Variável medida');
grid on;
legend('Raio', 'Velocidade');
hold off
```

2.4 Runge-Kutta de quarta ordem

```
function [ x ] = rk4( f, h, x0, intv )
%RK4 Summary of this function goes here
%   Detailed explanation goes here

t = intv(1):h:intv(2);                  % Intervalo da simulação
x = zeros(length(t), length(x0));       % Pré-alocação da matriz de resultados
x(1,:) = x0;                            % Condições iniciais

%% Encontra soluções do sistema de eq. diferenciais utilizando o método de
    Runge-Kutta de quarta ordem.
for i = 1:length(t)-1
    k1 = h * f(t(i), x(i, :));
    k2 = h * f(t(i) + h/2, x(i, :) + k1/2);
    k3 = h * f(t(i) + h/2, x(i, :) + k2/2);
    k4 = h * f(t(i) + h, x(i, :) + k3);
    x(i + 1, :) = x(i, :) + (k1 + 2*k2 + 2*k3 + k4)/6;
end

end
```