PMR3401 - Mecânica Computacional para Mecatrônica

Exercício Programa 1

Felipe Augusto Martins Pascutti 10705551
Lucas Fiori Rodrigues Amorim de Oliveira 10770408

Professor: Emilio Carlos Nelli Silva
Professor: Flávio Buiochi

24 de maio de 2021

Conteúdo

1 Introdução

No presente trabalho tem-se por objetivo estudar o comportamento de três diferentes métodos numéricos aprendidos. Para tal, um problema dinâmico de arrancada de veículos será abordado para verificar a estabilidade dos mesmos a partir de soluções obtidas pelos métodos de Euler, Runge-Kutta de 2a ordem e Runge-Kutta de 4a ordem. Posterior análise dos resultados e comparação entre os métodos serão realizadas.

2 Especificação do problema

O problema a ser estudado refere-se a arrancada de um veículo, e almeja determinar se arrancadas instáveis dependem só do motorista ou também do automóvel. Pode-se generalizar o equacionamento da dinâmica do problema a partir da modelagem apresentada na figura 2.1.

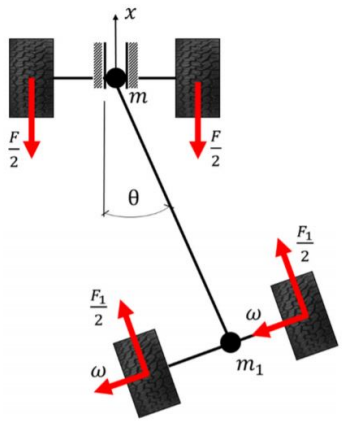

Figura 2.1: Modelo do veículo

As relações que descrevem o comportamento do sistema estão apresentadas nas equações 2.1 e 2.2

$$m_{total}\ddot{x} + m_1 L(\dot{\theta}^2 \cos\theta + \ddot{\theta}\sin\theta) = F_1 \cos\theta - F \tag{2.1}$$

$$m_1 \ddot{x} L \sin\theta + m_1 \ddot{\theta} L^2 + 2I\omega\dot{\theta} = 0 \tag{2.2}$$

É possível adequar esta modelagem a um caso específico alterando-se os valores de m_1 e m (visando equilibrar a distribuição de massa do carro) e também os valores de F e F_1 (visando ajustar-se à tração presente no veículo). Os diferentes casos serão detalhados posteriormente.

Para prosseguir com a análise numérica das equações, faz-se necessário isolar as variáveis $\ddot{\theta}$ e $\ddot{x}$. Assim, é possível utilizar uma abordagem matemática de solução de EDOS simultâneas tal qual visto em aula.

Para a obtenção de $\ddot{\theta}$, pode-se realizar operações entre as equações 2.1 e 2.2. Fazendo-se

2.1 - 2.2 $*(\frac{m_{total}}{m_1 L \sin\theta})$ e isolando a variável em questão, a equação 2.3 é obtida.

$$\ddot{\theta} = \frac{\sin\theta(F_1 \cos\theta - F + \frac{2I\omega\dot{\theta}m_{total}}{m_1 L \sin\theta} - m_1 L\dot{\theta}^2 \cos\theta)}{L(m_1 \sin^2\theta - m_{total})} \tag{2.3}$$

Analogamente, para obter $\ddot{x}$, pode-se fazer 2.1 - 2.2 $*(\frac{\sin\theta}{L})$ e, após isolar a variável em questão, a equação 2.4 é obtida.

$$\ddot{x} = \frac{F_1 \cos\theta - F + \frac{2I\omega\dot{\theta}\sin\theta}{L} - m_1 L\dot{\theta}^2 \cos\theta}{m_{total} - m_1 \sin^2\theta} \tag{2.4}$$

Pode-se introduzir variáveis de forma que $x = y_{x1}$ e $\theta = y_{\theta 1}$.

Desta forma, é possível construir o sistema 2.5.

$$\begin{cases} \dot{y_{x1}} = y_{x2} \\ \dot{y_{x2}} = y_{x3} = \ddot{x} \\ \dot{y_{\theta 1}} = y_{\theta 2} \\ \dot{y_{\theta 2}} = y_{\theta 3} = \ddot{\theta} \end{cases} \tag{2.5}$$

Note que este é um sistema de 4 EDOs de primeira ordem, que pode ser calculado pelos métodos numéricos apresentados em aula. Note, também, que $\ddot{x}$ e $\ddot{\theta}$ já foram previamente explicitados em função das outras variáveis do sistema.

Agora, basta conhecer as constantes do problema e suas condições iniciais para que o mesmo possa ser resolvido.

Do enunciado do EP temos que, para todos os casos, os valores das constantes são:

- $m_{total} = 1939(kg)$;
- $L = 2.95(m)$;
- $I = 1(kgm^2)$;
- $\omega = 10(rad/s)$;
- $\mu = 0,42$;
- $\beta = 0.02$.

Já para as condições inicias, também para todos os casos, temos:

- $\dot{\theta} = 0(rad/s)$;
- $\theta = 10(graus)$;
- $\dot{x} = 0(m/s)$;
- $x = 0(m)$;
- $0 < t < 20(s)$.

O passo a ser adotado nas soluções é de escolha dos alunos e serão explicitados posteriormente. Com as informações apresentadas, pode-se prosseguir aos problemas do enunciado.

3 Primeiro problema

O primeiro problema consiste em realizar as simulações com os três métodos numéricos já apresentados. Adiciona-se, aqui, as condições restantes para a definição do problema:

- $F = \beta mg$;
- $F_1 = \mu m_1 g$;
- $m = 0.6 m_{total}$;
- $m_1 = 0.4 m_{total}$.

Note que para os valores de F e F_1 adotados trata-se de um carro com tração traseira. Os valores de passo Δt, aqui chamados de h, escolhidos pelos integrantes da dupla são: $h = [0.5, 0.1, 0.0001]$.

3.1 Método de Euler

3.1.1 Implementação

O Método de Euler consiste em aproximar dado uma EDO ou um sistemas de EDOs a solução de cada função em um intervalo definido. Para isso, admite-se um passo para calcular uma rotina que fornece o valor da função no próximo ponto a partir do conhecimento do ponto atual e o valor da derivada da função nesse mesmo ponto. Para tal, é necessário conhecer a condição inicial do sistema.

Assim, para o caso de uma função do tipo $\frac{dy}{dx} = f(x, y)$, com condição inicial $y(x_0) = y_0$ a aproximação pelo Método de Euler é expressa por:

$$y(x_{i+1}) = y(x_i) + hf(x_i, y(x_i)) => y_{i+1} = y_i + hf(x_i, y_i) \tag{3.1}$$

Para o sistema 2.5, pode-se aplicar a ideia exposta anteriormente em forma matricial. Possibilitando que todas as EDOs sejam resolvidas simultaneamente. Assumindo a notação entre colchetes como sendo indicador de matrizes, pode-se escrever, baseado na equação

3.1, a equação matricial 3.2 :

$$[Y]_{i+1} = [Y]_i + h[f(t_i, [Y]_i)] \tag{3.2}$$

Onde

$$[Y] = \begin{bmatrix} y_{x1} \\ y_{x2} \\ y_{\theta 1} \\ y_{\theta 2} \end{bmatrix}, [f(t_i, [Y]_i)] = \begin{bmatrix} \dot{y_{x1}} \\ \dot{y_{x2}} \\ \dot{y_{\theta 1}} \\ \dot{y_{\theta 2}} \end{bmatrix} = \begin{bmatrix} f_{x1}(t_i, [Y]_i) \\ f_{x2}(t_i, [Y]_i) \\ f_{\theta 1}(t_i, [Y]_i) \\ f_{\theta 2}(t_i, [Y]_i) \end{bmatrix}$$

Com condições iniciais

$$[Y]_0 = \begin{bmatrix} y_{x10} \\ y_{x20} \\ y_{\theta 10} \\ y_{\theta 20} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 10 \\ 0 \end{bmatrix}$$

A implementação da rotina que realiza tais operações está presente no Apêndice C.

3.1.2 Resultados

Utilizando o *software* MatLab, foi possível plotar os gráficos apresentados nas figuras 3.1 (h=0.5), 3.2 (h=0.1) e 3.3 (h=0.0001).

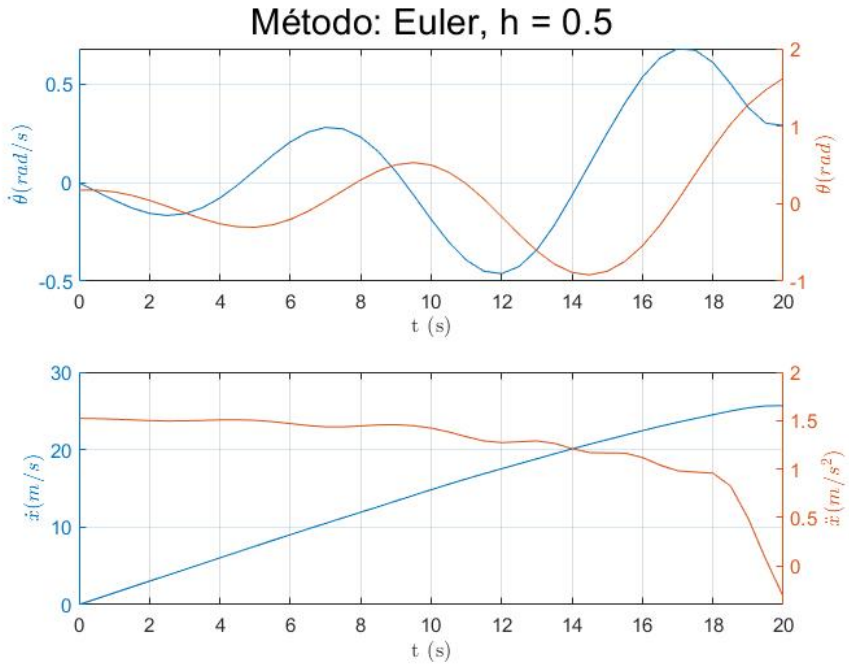

Figura 3.1: Gráficos de $\dot{x}, \ddot{x}, \theta, \dot{\theta}$ para o Método de Euler com h=0.5

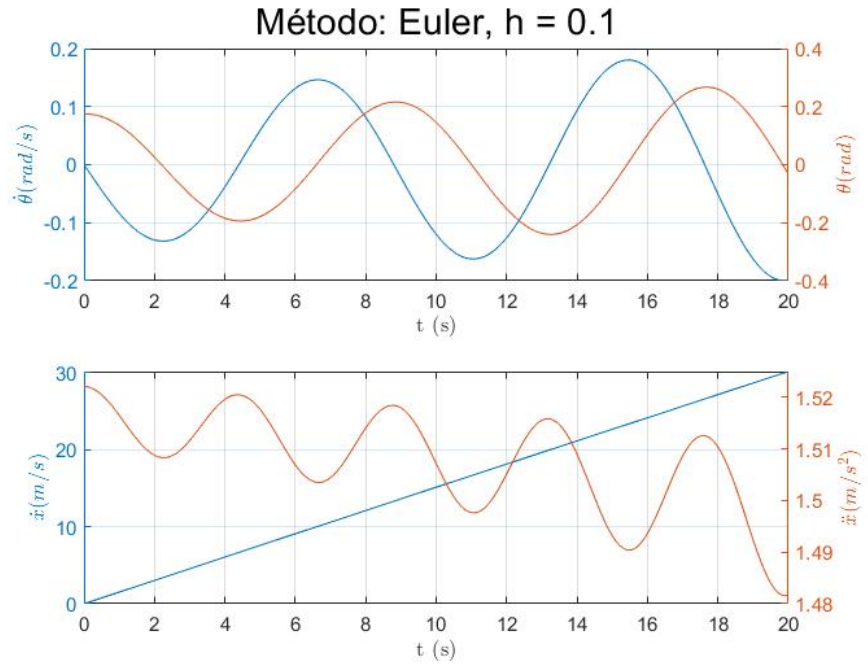

Figura 3.2: Gráficos de $\dot{x}, \ddot{x}, \theta, \dot{\theta}$ para o Método de Euler com h=0.1

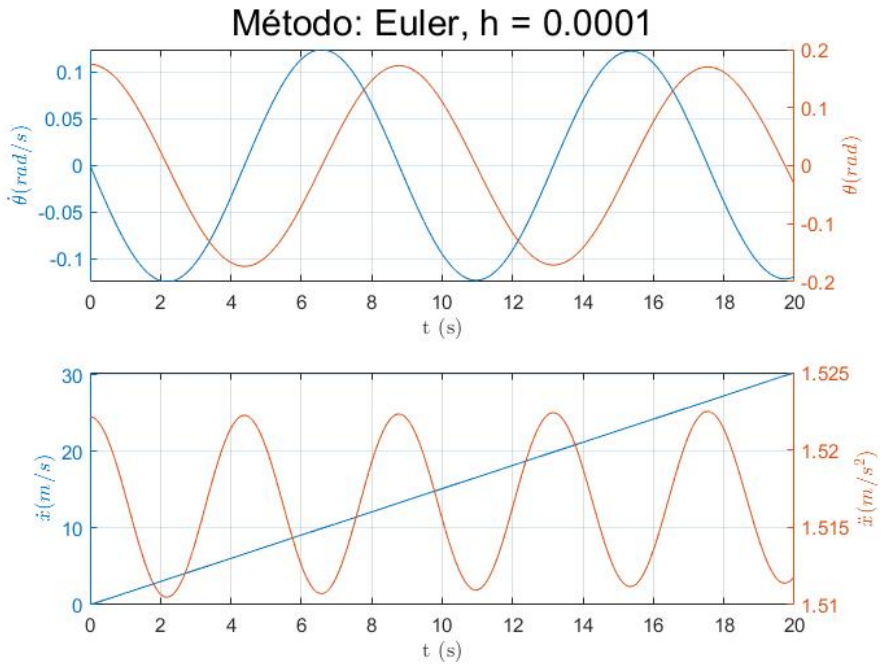

Figura 3.3: Gráficos de $\dot{x}, \ddot{x}, \theta, \dot{\theta}$ para o Método de Euler com h=0.0001

3.2 Método de Runge-Kutta de 2a ordem (RK2)

3.2.1 Implementação

Assim como o Método de Euler, o Método de Runge-Kutta tem por objetivo aproximar as soluções de uma EDO ou de um sistema de EDOs dado. Entretanto, diferentemente deste primeiro que usa apenas a derivada no ponto a ser calculado, o segundo usa uma função incremento (Φ) para melhor aproximar o resultado. A depender do número de termos desta função, têm-se as ordens do método.

Assim, dada uma função $\frac{dy}{dx} = f(x, y)$ com condição inicial $y(x_0) = y_0$ e passo h a aproximação pelo Método de Runge-Kutta é dada por:

$$y_{i+1} = y_i + h\Phi(x_i, y_i, h) \tag{3.3}$$

Onde

$$\Phi(x_i, y_i, h) = ak_1 + bk_2 + ck_3 + dk_4 + ... \tag{3.4}$$

Para o caso de segunda ordem com o Método de Euler Modificado, temos $a = 0$ e $b = 1$:

$$\Phi(x_i, y_i, h) = k_2 \tag{3.5}$$

E, portanto:

$$y_{i+1} = y_i + hk_2 \tag{3.6}$$

Onde

$$k1 = f(x_i, y_i)\ ,\ k2 = f(x_i + h/2, y_i + hk_1/2)$$

Novamente, para o sistema 2.5, pode-se aplicar esta ideia em forma matricial. Assim, partindo-se da equação 3.6, obtém-se a equação matricial 3.7

$$[Y]_{i+1} = [Y]_i + h[k_2] \tag{3.7}$$

Onde

$$[Y] = \begin{bmatrix} y_{x1} \\ y_{x2} \\ y_{\theta 1} \\ y_{\theta 2} \end{bmatrix};$$

$$[k_1] = [f(t_i, [Y]_i)];$$
$$[k_2] = [f(t_i + h/2, [Y]_i + h[k_1]/2)].$$

As condições iniciais, assim como no Método de Euler, são:

$$[Y]_0 = \begin{bmatrix} y_{x10} \\ y_{x20} \\ y_{\theta 10} \\ y_{\theta 20} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 10 \\ 0 \end{bmatrix}$$

A implementação da rotina que realiza tais operações está presente no Apêndice D.

3.2.2 Resultados

Utilizando o *software* MatLab, foi possível plotar os gráficos apresentados nas figuras 3.4 (h=0.5), 3.5 (h=0.1) e 3.6 (h=0.0001).

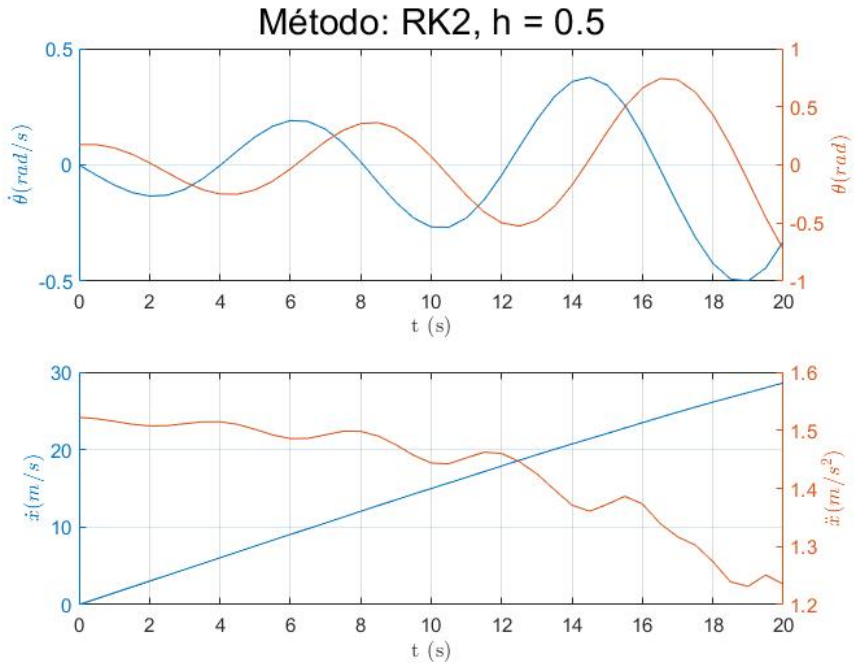

Figura 3.4: Gráficos de $\dot{x}, \ddot{x}, \theta, \dot{\theta}$ para o Método RK2 com h=0.5

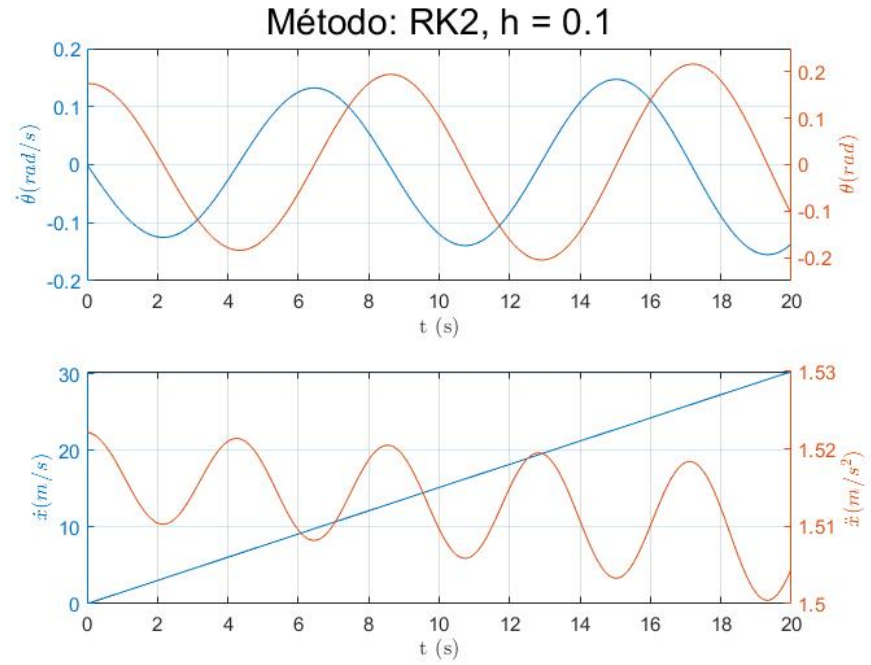

Figura 3.5: Gráficos de $\dot{x}, \ddot{x}, \theta, \dot{\theta}$ para o Método RK2 com h=0.1

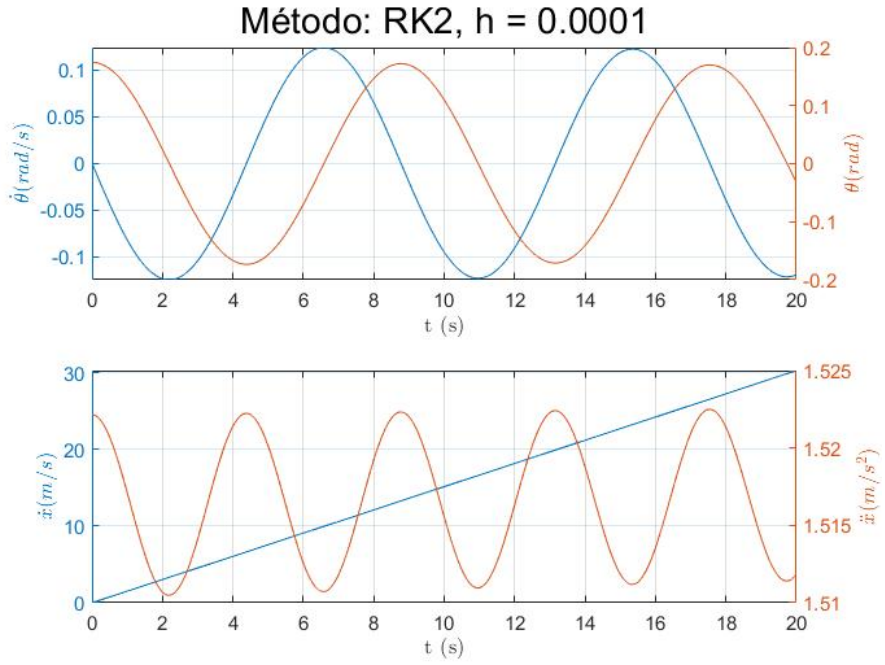

Figura 3.6: Gráficos de $\dot{x}, \ddot{x}, \theta, \dot{\theta}$ para o Método RK2 com h=0.0001

3.3 Método de Runge-Kutta de 4a ordem (RK4)

3.3.1 Implementação

Assim como na seção anterior, é necessário implementar o Método de Runge-Kutta.

Então, para dada função $\frac{dy}{dx}$ com condição inicial $y(x_0) = y_0$ e passo h, temos o apresentado nas equações 3.3 e 3.4.

Para o caso de quarta ordem, temos $a = 1/6$, $b = 2/6$, $c = 2/6$ e $d = 1/6$. Dessa forma, a aproximação da função pelo Método RK4 é:

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{3.8}$$

Com

$$k_1 = f(x_i, y_i);$$

$$k_2 = f(x_i + h/2, y_i + hk_1/2);$$

$$k_3 = f(x_i + h/2, y_i + hk_2/2);$$

$$k_4 = f(x_i + h, y_i + hk_3).$$

Novamente, para o sistema 2.5, pode-se aplicar esta ideia em forma matricial. Assim, partindo-se da equação 3.8, obtém-se a equação matricial 3.9

$$[Y]_{i+1} = [Y]_i + \frac{h}{6}([k_1] + 2[k_2] + 2[k_3] + [k_4]) \tag{3.9}$$

Onde

$$[Y] = \begin{bmatrix} y_{x1} \\ y_{x2} \\ y_{\theta 1} \\ y_{\theta 2} \end{bmatrix};$$

$$[k_1] = [f(t_i, y_i)];$$

$$[k_2] = [f(t_i + h/2, y_i + h[k_1]/2)];$$

$$[k_3] = [f(t_i + h/2, y_i + h[k_2]/2)];$$

$$[k_4] = [f(t_i + h, y_i + h[k_3])].$$

Assim como nos métodos anteriores, as condições iniciais são:

$$[Y]_0 = \begin{bmatrix} y_{x10} \\ y_{x20} \\ y_{\theta 10} \\ y_{\theta 20} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 10 \\ 0 \end{bmatrix}$$

A implementação da rotina que realiza tais operações está presente no Apêndice E.

3.3.2 Resultados

Utilizando o *software* MatLab, foi possível plotar os gráficos apresentados nas figuras 3.7 (h=0.5), 3.8 (h=0.1) e 3.9 (h=0.0001).

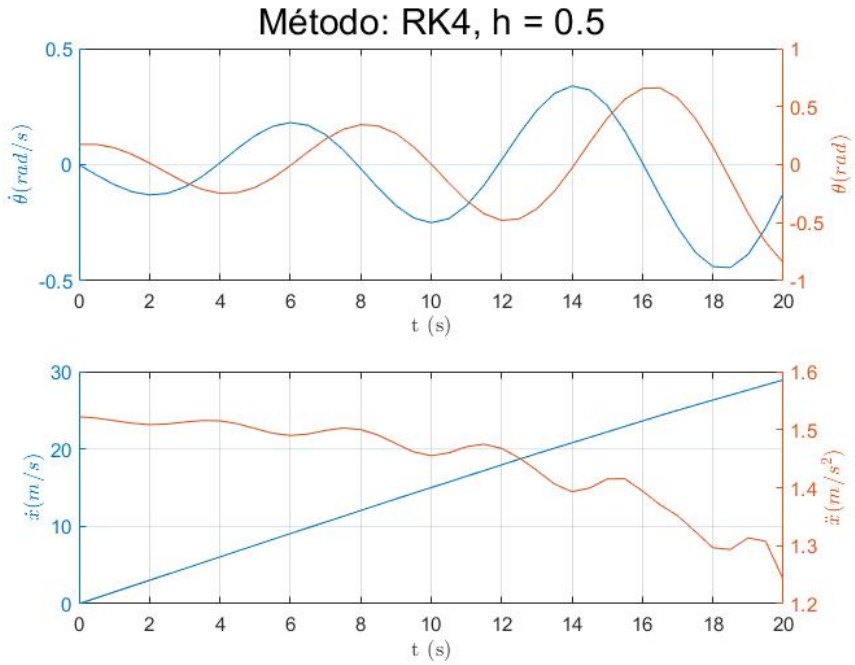

Figura 3.7: Gráficos de $\dot{x}, \ddot{x}, \theta, \dot{\theta}$ para o Método RK4 com h=0.5

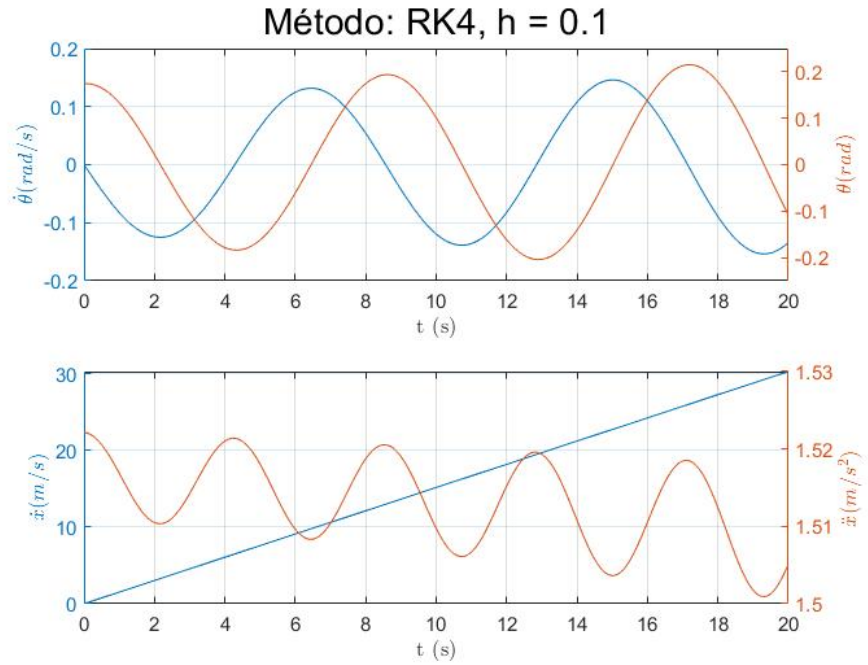

Figura 3.8: Gráficos de $\dot{x}, \ddot{x}, \theta, \dot{\theta}$ para o Método RK4 com h=0.1

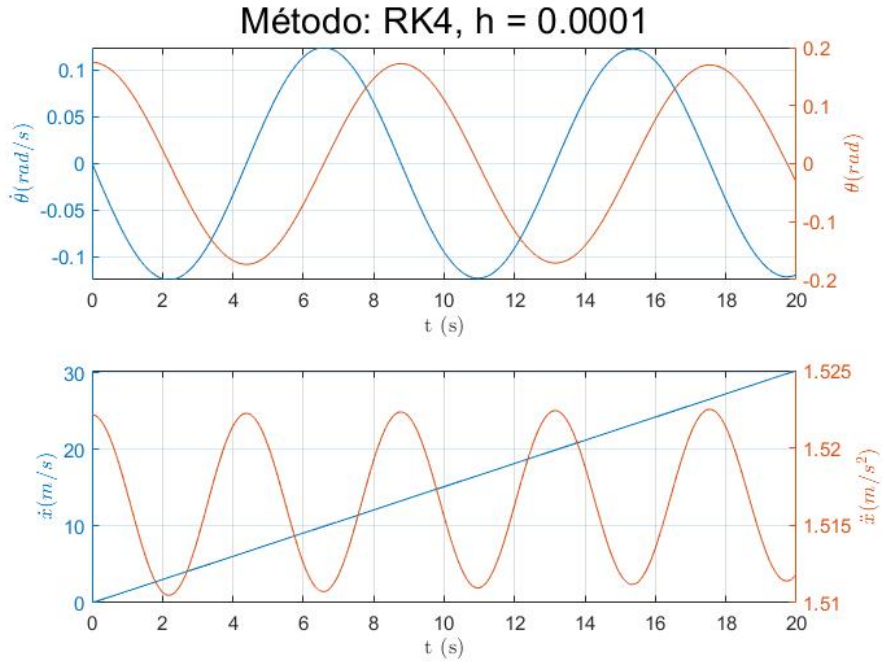

Figura 3.9: Gráficos de $\dot{x}, \ddot{x}, \theta, \dot{\theta}$ para o Método RK4 com h=0.0001

3.4 Discussões

Os gráficos gerados com passo $h = 0.5$ (figuras 3.1, 3.4 e 3.7), ainda que não possam ser utilizados como base de uma interpretação confiável para o problema, evidenciam a relevância do aumento da ordem nos métodos de Runge-Kutta para a convergência da resposta. Em todos os métodos utilizados há convergência plena com o valor de passo $h = 0.0001$ (figuras 3.3, 3.6 e 3.9), isto é, o gráfico correspondente ao parâmetro referido é essencialmente igual para todos os métodos. No entanto, as respostas obtidas com passo $h = 0.5$ são substancialmente diferentes. A divergência é mais expressiva quanto menor é a ordem do método utilizado. Adicionalmente, o gráfico intermediário gerado - cujo valor de passo é $h = 0.1$ - destaca a velocidade com a qual cada um dos métodos converge (figuras 3.2, 3.5 e 3.8). É notável que para o método de Euler o segundo gráfico ainda se distancia consideravelmente da resposta mais refinada dada pelo último gráfico, enquanto à medida que o passo diminui para 0.1 nos métodos de ordem superior, a resposta se aproxima mais da convergência. Sendo assim, são necessários valores de passo cada vez mais baixos para a obtenção de gráficos que permitem graus aceitáveis de confiabilidade à medida que a ordem do método é decrescida.

Quanto ao comportamento específico dos parâmetros escolhidos para serem representados - $\dot{x}, \ddot{x}, \theta, \dot{\theta}$ -, todos exibem comportamento oscilatório exceto $\dot{x}$, que apresenta padrão linear. A amplitude de oscilação de θ e $\dot{\theta}$, entretanto, são mais expressivas com relação à amplitude de $\ddot{x}$, que é consideravelmente baixa. A fim de se obter conclusões a respeito da natureza da perda de controle do veículo, deve ser observado majoritariamente o efeito da aceleração linear média no destracionamento do carro, causado pelo derrapamento em virtude da instabilidade angular estabelecida, fator que implica na perda de controle da direção por parte do operador.

4 Segundo problema

O segundo problema tem por objetivo estudar como diferentes configurações de automóvel alteram a dinâmica do sistema. Dentre as variações possíveis, pode-se alterar a distribuição de massa no veículo (a partir dos fatores m e m_1) e também sua tração (que altera os valores de F e F_1). Para as simulações apresentadas nesta seção, utilizou-se o método RK4 com o passo $h = 0.001$ selecionado pela dupla.

Sabe-se que para tração traseira, temos:

$$F = \beta mg \ , \ F_1 = \mu m_1 g$$

Já para tração dianteira, temos:

$$F = -\mu mg \ , \ F_1 = -\beta m_1 g$$

Por fim, para tração nas 4 rodas, temos:

$$F = -\mu mg \ , \ F_1 = \mu m_1 g$$

Para todos os casos a rotina utilizada foi a referente ao RK4, presente no Apêndice E. O que difere uma simulação da outra são os parâmetros passados às funções, estes devidamente explicitados no Apêndice A.

Os parâmetros dos diferentes casos são apresentados na tabela 4.1.

Casos	Tração	m	m_1
1	Traseira	$0.8m_{total}$	$0.2m_{total}$
2	Dianteira	$0.8m_{total}$	$0.2m_{total}$
3	Dianteira	$0.2m_{total}$	$0.8m_{total}$
4	Traseira	$0.2m_{total}$	$0.8m_{total}$
5.1	4 rodas	$0.2m_{total}$	$0.8m_{total}$
5.2	4 rodas	$0.8m_{total}$	$0.2m_{total}$

Tabela 4.1: Tabela com os diferentes casos

4.1 Caso 1

4.1.1 Resultado

A simulação com os parâmetros apresentados na tabela 4.1 geraram os gráficos apresentados na figura 4.1.

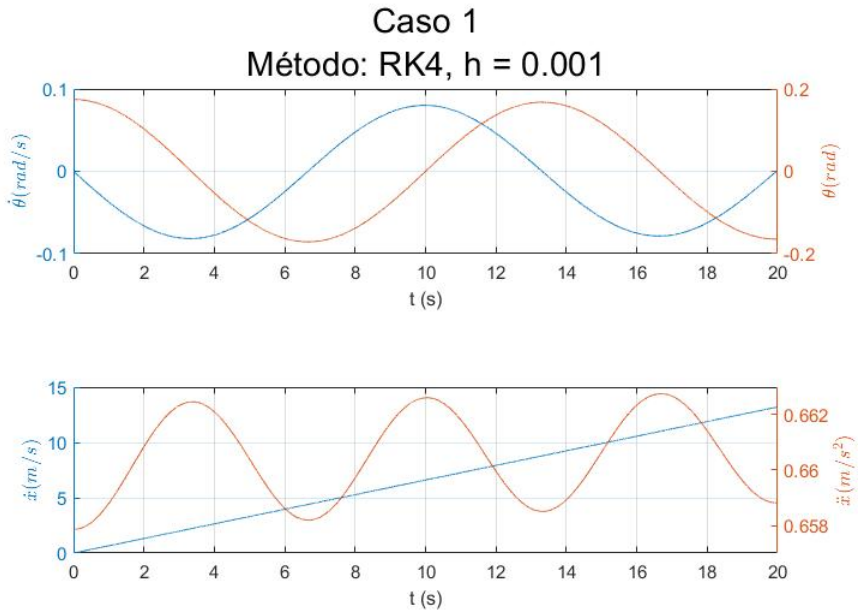

Figura 4.1: Gráficos de $\dot{x}, \ddot{x}, \theta, \dot{\theta}$ para o Método RK4 com h=0.001 - Caso 1

4.2 Caso 2

4.2.1 Resultado

A simulação com os parâmetros apresentados na tabela 4.1 geraram os gráficos apresentados na figura 4.2.

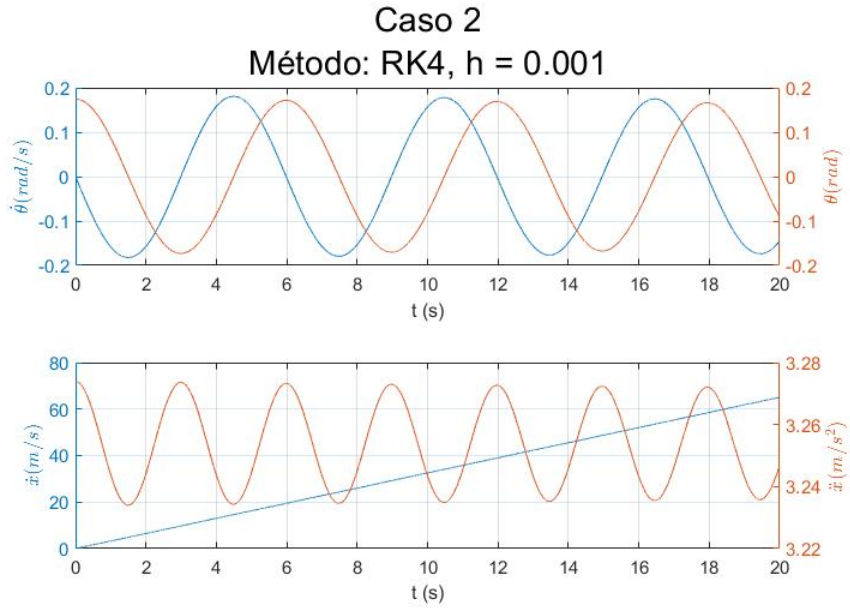

Figura 4.2: Gráficos de $\dot{x}, \ddot{x}, \theta, \dot{\theta}$ para o Método RK4 com h=0.001 - Caso 2

4.3 Caso 3

4.3.1 Resultado

A simulação com os parâmetros apresentados na tabela 4.1 geraram os gráficos apresentados na figura 4.3.

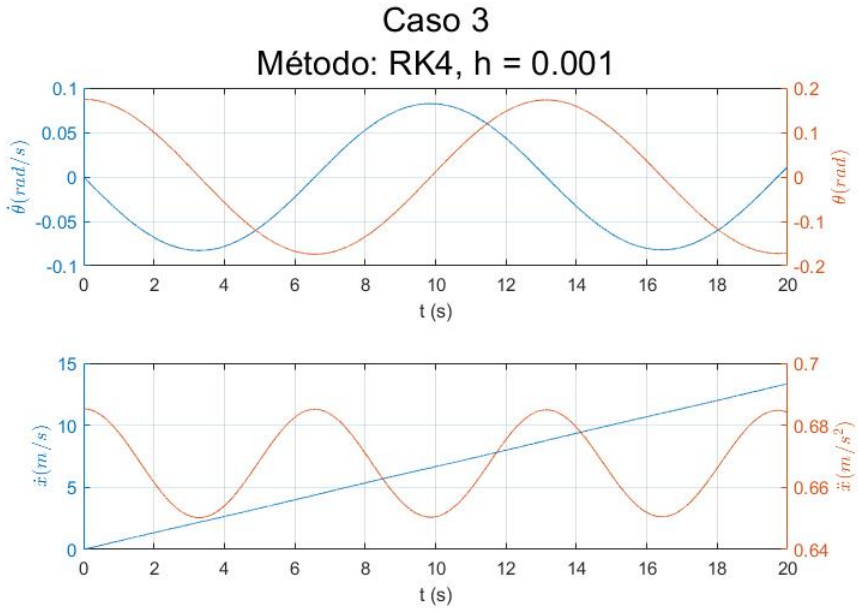

Figura 4.3: Gráficos de $\dot{x}, \ddot{x}, \theta, \dot{\theta}$ para o Método RK4 com h=0.001 - Caso 3

4.4 Caso 4

4.4.1 Resultado

A simulação com os parâmetros apresentados na tabela 4.1 geraram os gráficos apresentados na figura 4.4.

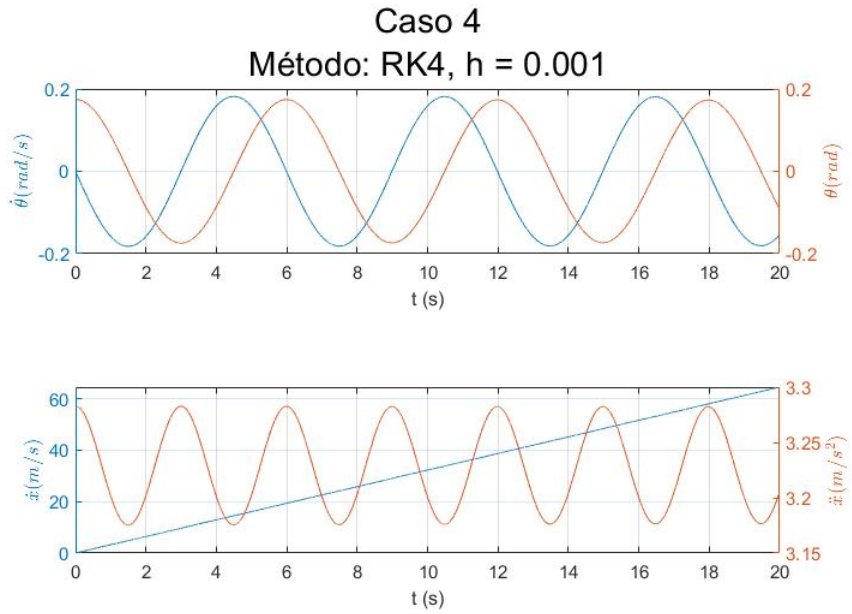

Figura 4.4: Gráficos de $\dot{x}, \ddot{x}, \theta, \dot{\theta}$ para o Método RK4 com h=0.001 - Caso 4

4.5 Caso 5.1

4.5.1 Resultados

A simulação com os parâmetros apresentados na tabela 4.1 geraram os gráficos apresentados na figura 4.5.

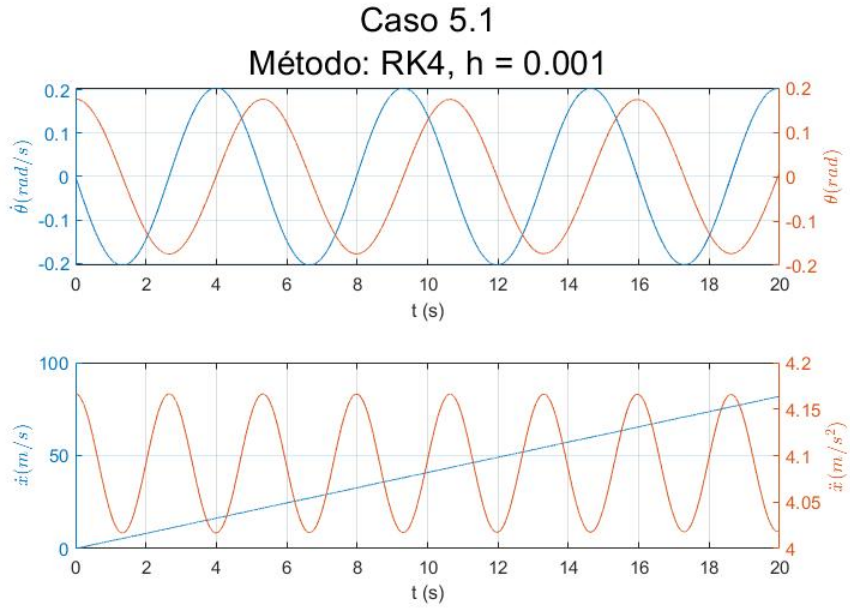

Figura 4.5: Gráficos de $\dot{x}, \ddot{x}, \theta, \dot{\theta}$ para o Método RK4 com h=0.001 - Caso 5.1

4.6 Caso 5.2

4.6.1 Resultados

A simulação com os parâmetros apresentados na tabela 4.1 geraram os gráficos apresentados na figura 4.6.

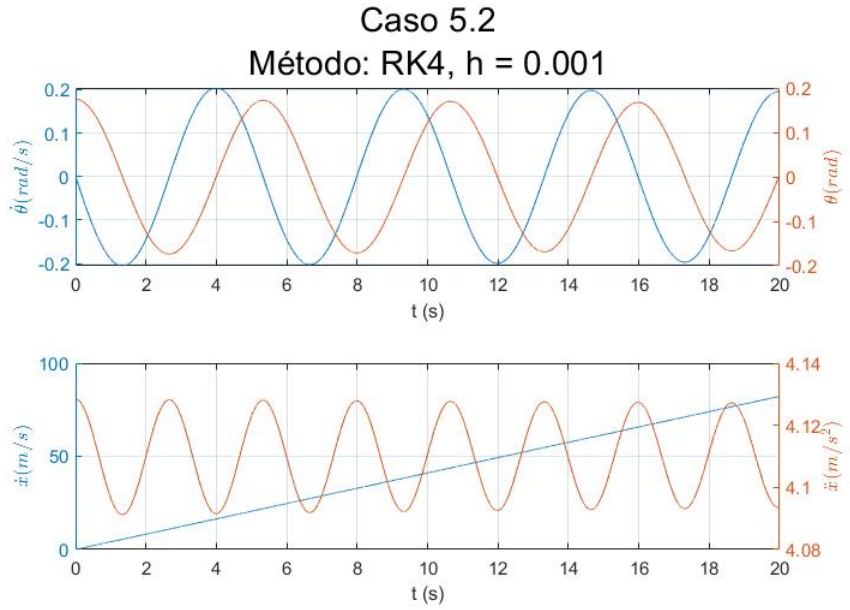

Figura 4.6: Gráficos de $\dot{x}, \ddot{x}, \theta, \dot{\theta}$ para o Método RK4 com h=0.001 - Caso 5.2

4.7 Discussões

Há uma clara similaridade entre os resultados obtidos para o caso 1 e para o caso 3. O primeiro veículo apresenta motor dianteiro e tração traseira enquanto o terceiro apresenta motor traseiro e tração dianteira. Em virtude de, nos dois casos, o motor e tipo de tração se concentrarem em lados opostos do veículo, os resultados são parecidos. Conforme discutido na seção reservada ao primeiro problema, veículos que exibem baixa aceleração linear média pelos gráficos apresentam maior índice de destracionamento no intervalo de tempo que compreende a arrancada, tornando o controle da direção do veículo mais difícil para o operador. Tal efeito acontece nos dois casos referidos, nos quais é possível atribuir alto grau de responsabilidade pelo acidente aos fabricantes.

Para os casos 2 e 4, similares pelo efeito oposto ao descrito para os casos 1 e 3, o mesmo não acontece. O fato de o motor e tipo de tração estarem alocados no mesmo lado do veículo implica em acelerações médias consideravelmente maiores, sendo a velocidade alcançada pelos veículos substancialmente maiores. Esse resultado exprime pouco efeito de derrapamento, eximindo a dinâmica dos veículos da culpa pelo acidente. O mesmo ocorre para os casos 5.1 e 5.2, os quais são extremamente similares entre si - resultado que permite afirmar, para veículos com tração nas quatro rodas, que a posição do motor no veículo influencia pouco na sua dinâmica em arrancadas. Os últimos casos citados possuem as maiores acelerações médias e atingem as maiores velocidades entre todos os casos tratados.

5 Conclusão

Métodos numéricos para resolução de equações diferenciais ordinárias são ferramentas extremamente úteis na resolução de problemas de engenharia. A análise dos resultados obtidos pela resolução do problema de instabilidade em veículos proposto permite avaliar a eficiência de três dessas técnicas, a saber - métodos de Runge-Kutta de primeira, segunda e quarta ordem. A velocidade com a qual ocorre a convergência das respostas obtidas por cada um dos métodos com relação à diminuição do parâmetro h (o passo) aumenta conforme a ordem do método utilizado. Quanto maior a ordem, mais rápido ocorre convergência, pois os métodos utilizam cálculos de derivadas cada vez mais refinados para estimar o valor do parâmetro desejado na próxima iteração, sendo valores de passo menores cada vez menos necessários para a obtenção de uma resposta confiável.

O problema buscava verificar se os fabricantes dos veículos em questão poderiam ser responsabilizados pelos acidentes causados pela perda de controle na arrancada. Através da análise dos gráficos gerados com base na resolução das equações diferenciais ordinárias fundamentadas na teoria de mecânica analítica, é possível responsabilizar os fabricantes dos veículos descritos no caso 1 e 3, mas não há evidência física para afirmar o mesmo a respeito dos outros veículos, posto suas estabilidades serem pouco influenciadas pela distribuição de massa e tipo de tração.

6 Apêndice A - Função main

```
clc;
clear all;
close all;

% Constantes
m_tot = 1939; % (kg)
L     = 2.95; % (m)
I     = 1;    % (kg * m^2)
omega = 10;   % (rad/s)
mi    = 0.42;
beta  = 0.02;
g     = 9.8;

% Item 1
% Condicoes iniciais

x0         = 0;
x_dot0     = 0;
theta0     = deg2rad(10);
theta_dot0 = 0;

m     = 0.6*m_tot;
m_1   = 0.4*m_tot;
F     = beta*m*g;
F_1   = mi*m_1*g;

t_0 = 0;
t_f = 20;
h = 1;

t = t_0:h:t_f;

estados0 = [theta_dot0; x_dot0; theta0; x0];

% Metodo de Euler

```

```
[estados, f_hist] = Euler(estados0, h, t, m_tot, m_1, L, F, F_1, I, omega)
    ;
plotGraph(t, estados, f_hist, 'Euler', h, 1, 0)

% RK2

[estados, f_hist] = RK2(estados0, h, t, m_tot, m_1, L, F, F_1, I, omega);
plotGraph(t, estados, f_hist, 'RK2', h, 1, 0)

% RK4

[estados, f_hist] = RK4(estados0, h, t, m_tot, m_1, L, F, F_1, I, omega);
plotGraph(t, estados, f_hist, 'RK4', h, 1, 0)

% Item 2

h = 1;
t = t_0:h:t_f;

% Caso 1
m   = 0.8*m_tot; % Motor dianteiro
m_1 = 0.2*m_tot; % Motor dianteiro
F   = beta*m*g;  % Tracao traseira
F_1 = mi*m_1*g;    % Tracao traseira

[estados, f_hist] = RK4(estados0, h, t, m_tot, m_1, L, F, F_1, I, omega);
plotGraph(t, estados, f_hist, 'RK4', h, 2, 1)

% Caso 2

m   = 0.8*m_tot;   % Motor dianteiro
m_1 = 0.2*m_tot;   % Motor dianteiro
F   = -mi*m*g;     % Tracao dianteira
F_1 = -beta*m_1*g; % Tracao dianteira

[estados, f_hist] = RK4(estados0, h, t, m_tot, m_1, L, F, F_1, I, omega);
plotGraph(t, estados, f_hist, 'RK4', h, 2, 2)

% Caso 3

m   = 0.2*m_tot;   % Motor traseiro
m_1 = 0.8*m_tot;   % Motor traseiro
F   = -mi*m*g;     % Tracao dianteira
F_1 = -beta*m_1*g; % Tracao dianteira

```

```
[estados, f_hist] = RK4(estados0, h, t, m_tot, m_1, L, F, F_1, I, omega);
plotGraph(t, estados, f_hist, 'RK4', h, 2, 3)

% Caso 4

m   = 0.2*m_tot; % Motor traseiro
m_1 = 0.8*m_tot; % Motor traseiro
F   = beta*m*g;  % Tracao traseira
F_1 = mi*m_1*g;    % Tracao traseira

[estados, f_hist] = RK4(estados0, h, t, m_tot, m_1, L, F, F_1, I, omega);
plotGraph(t, estados, f_hist, 'RK4', h, 2, 4)

% Caso 5a

m   = 0.2*m_tot; % Motor traseiro
m_1 = 0.8*m_tot; % Motor traseiro
F   = -mi*m*g;   % Tracao 4 rodas
F_1 = mi*m_1*g;  % Tracao 4 rodas

[estados, f_hist] = RK4(estados0, h, t, m_tot, m_1, L, F, F_1, I, omega);
plotGraph(t, estados, f_hist, 'RK4', h, 2, 5.1)

% Caso 5b

m   = 0.8*m_tot; % Motor dianteiro
m_1 = 0.2*m_tot; % Motor dianteiro
F   = -mi*m*g;   % Tracao 4 rodas
F_1 = mi*m_1*g;  % Tracao 4 rodas

[estados, f_hist] = RK4(estados0, h, t, m_tot, m_1, L, F, F_1, I, omega);
plotGraph(t, estados, f_hist, 'RK4', h, 2, 5.2)
```

7 Apêndice B - Função das derivadas de segunda ordem

```
function [derivadas] = f(theta_dot, theta, m_tot, m_1, L, F, F_1, I, omega
    )
%f: Calcula as derivadas theta_dot2 e x_dot2, oriundas de manipulacao
%   algebrica das equacoes provenientes da modelagem do problema.
%Input:
%   theta_dot = primeira derivada temporal de theta.
%   theta     = angulo theta considerado na modelagem do problema.
%   m_tot, m_1, L, F, F_1, I sao constantes.
%Output:
%   derivadas = vetor que contem o resultado do calculo das derivadas
%               theta_dot2 e x_dot2 nessa ordem.

theta_dot2 = (sin(theta)/(L*((sin(theta)^2)*m_1 - m_tot)))*(F_1*cos(theta
    ) - F - m_1*L*(theta_dot^2)*cos(theta)+ ((2*I*omega*theta_dot*m_tot)/(
    m_1*L*sin(theta))));
x_dot2 = (F_1*cos(theta) - F - m_1*L*(theta_dot^2)*cos(theta) + 2*I*omega
    *theta_dot*sin(theta)/L)/(m_tot - m_1*(sin(theta)^2));

derivadas = [theta_dot2; x_dot2];

end
```

8 Apêndice C - Função Euler

```
function [estados, f_hist] = Euler(estados, h, t, m_tot, m_1, L, F, F_1, I
    , omega)
%Euler: Resolve uma EDO generica atraves do metodo de Euler.
%Input:
%   estados = vetor que contem os valores calculados na iteracao
%             anterior (ou os valores iniciais para o caso de ser a
    primeira iteracao),
%             a saber: theta_dot, x_dot, theta e x nessa ordem.
%   h       = valor do passo considerado.
%   t       = vetor do tempo.
%   m_tot, m_1, L, F, F_1, I sao constantes passadas como entrada para que
%   seja realizado o calculo das derivadas de segunda ordem.
%Output:
%   estados = vetor com os valores calculados em todas as iteracoes do
%             metodo.
%   f_hist  = vetor que contem os valores de x_dot2 calculados em todas as
%             iteracoes do m todo.

K1 = [0; 0; 0; 0];

for n=1:length(t)-1
    derivadas = f(estados(1, n), estados(3, n), m_tot, m_1, L, F, F_1, I,
    omega);
    K1(1:2) = derivadas; % theta_dot2 (i) e x_dot2 (i)
    f_hist(n) = K1(2);
    K1(3:4) = estados(1:2, n); % theta_dot (i) e x_dot (i)
    estados(:, n+1) = estados(:, n) + (h * K1);
end

    derivadas = f(estados(1, length(estados)), estados(3, length(estados))
    , m_tot, m_1, L, F, F_1, I, omega);
    f_hist(length(f_hist) + 1) = derivadas(2);

end
```

9 Apêndice D - Função RK2

```
function [estados, f_hist] = RK2(estados, h, t, m_tot, m_1, L, F, F_1, I,
    omega)
%Input:
%   estados = vetor que contem os valores calculados na iteracao
%             anterior (ou os valores iniciais para o caso de ser a
    primeira iteracao),
%             a saber: theta_dot, x_dot, theta e x nessa ordem.
%   h       = valor do passo considerado.
%   t       = vetor do tempo.
%   m_tot, m_1, L, F, F_1, I sao constantes passadas como entrada para que
%   seja realizado o calculo das derivadas de segunda ordem.
%Output:
%   estados = vetor com os valores calculados em todas as iteracoes do
%             metodo.
%   f_hist  = vetor que contem os valores de x_dot2 calculados em todas as
%             iteracoes do m todo.

K1 = [0; 0; 0; 0];
K2 = [0; 0; 0; 0];

for n=1:length(t)-1
    derivadas = f(estados(1, n), estados(3, n), m_tot, m_1, L, F, F_1, I,
    omega);
    K1(1:2) = derivadas; % theta_dot2 (i) e x_dot2 (i)
    f_hist(n) = K1(2);
    K1(3:4) = estados(1:2, n); % theta_dot (i) e x_dot (i)
    derivadas = f(estados(1,n) + (h/2)*K1(1), estados(3, n) + (h/2)*K1(3),
     m_tot, m_1, L, F, F_1, I, omega);
    K2(1:2) = derivadas;
    K2(3:4) = estados(1:2, n) + (h/2)*K1(3:4);
    estados(:, n+1) = estados(:, n) + (h * K2);
end

    derivadas = f(estados(1, length(estados)), estados(3, length(estados))
    , m_tot, m_1, L, F, F_1, I, omega);
    f_hist(length(f_hist) + 1) = derivadas(2);
end
```

10 Apêndice E - Função RK4

```
function [estados, f_hist] = RK4(estados, h, t, m_tot, m_1, L, F, F_1, I,
    omega)
%Input:
%   estados = vetor que contem os valores calculados na iteracao
%             anterior (ou os valores iniciais para o caso de ser a
    primeira iteracao),
%             a saber: theta_dot, x_dot, theta e x nessa ordem.
%   h       = valor do passo considerado.
%   t       = vetor do tempo.
%   m_tot, m_1, L, F, F_1, I sao constantes passadas como entrada para que
%   seja realizado o calculo das derivadas de segunda ordem.
%Output:
%   estados = vetor com os valores calculados em todas as iteracoes do
%             metodo.
%   f_hist  = vetor que contem os valores de x_dot2 calculados em todas as
%             iteracoes do m todo.

K1 = [0; 0; 0; 0];
K2 = [0; 0; 0; 0];
K3 = [0; 0; 0; 0];
K4 = [0; 0; 0; 0];

for n=1:length(t)-1
    derivadas = f(estados(1, n), estados(3, n), m_tot, m_1, L, F, F_1, I,
    omega);
    K1(1:2) = derivadas; % theta_dot2 (i) e x_dot2 (i)
    f_hist(n) = K1(2);
    K1(3:4) = estados(1:2, n); % theta_dot (i) e x_dot (i)
    derivadas = f(estados(1,n) + (h/2)*K1(1), estados(3, n) + (h/2)*K1(3),
     m_tot, m_1, L, F, F_1, I, omega);
    K2(1:2) = derivadas;
    K2(3:4) = estados(1:2, n) + (h/2)*K1(3:4);
    derivadas = f(estados(1,n) + (h/2)*K2(1), estados(3, n) + (h/2)*K2(3),
     m_tot, m_1, L, F, F_1, I, omega);
    K3(1:2) = derivadas;
    K3(3:4) = estados(1:2, n) + (h/2)*K2(3:4);
```

```
32      derivadas = f(estados(1,n) + h*K3(1), estados(3, n) + h*K3(3), m_tot,
        m_1, L, F, F_1, I, omega);
        K4(1:2) = derivadas;
34      K4(3:4) = estados(1:2, n) + h*K3(3:4);

36      K = K1 + 2*K2 + 2*K3 + K4;
        estados(:, n+1) = estados(:, n) + (h/6 * K);
38  end

40      derivadas = f(estados(1, length(estados)), estados(3, length(estados))
        , m_tot, m_1, L, F, F_1, I, omega);
        f_hist(length(f_hist) + 1) = derivadas(2);
42
    end
```

11 Apêndice F - Função plotagem

```
function plotGraph(t, estados, f_hist, metodo, h, item, caso)
%plotGraph: Imprime a resposta de theta e theta_dot em um grafico e x_dot e
% x_dot2 em outro grafico, todos em uma mesma figura.
%Input:
%   t       = vetor do tempo.
%   estados = vetor com os valores calculados em todas as iteracoes do
%             metodo utilizado.
%   f_hist  = vetor que contem os valores de x_dot2 calculados em todas as
%             iteracoes do metodo utilizado.
%   metodo  = nome do metodo utilizado.
%   h       = valor do passo considerado.
%   item    = numero do item correspondente.
%   caso    = caso correspondente se o item for o 2, caso contrario nao e
%             utilizado.

figure

subplot(2, 1, 1);
yyaxis left
plot(t, estados(1, :));
ylabel('$\dot{\theta}(rad/s)$')
yyaxis right
plot(t, estados(3, :));
ylabel('$\theta(rad)$');
xlabel('t (s)')
if item == 1
    title(['M todo: ', metodo, ', h = ', num2str(h)], 'FontWeight', '
    Normal', 'FontSize', 17);
else
    title({['Caso ', num2str(caso)], ['M todo: ', metodo, ', h = ',
    num2str(h)]}, 'FontWeight', 'Normal', 'FontWeight', 'Normal', 'FontSize
    ', 17);
end
grid on;

subplot(2, 1, 2);
```

```
yyaxis left
plot(t, estados(2, :));
ylabel('$\dot{x}(m/s)$')
yyaxis right
plot(t, f_hist);
ylabel('$\ddot{x}(m/s^2)$')
xlabel('t (s)')
grid on;


end
```