

Título: Gravitação na Terra e no Sistema Solar e Métodos
Numéricos

PDPD: Arthur Monteforte

Orientadora: Prof. Dra. Cecilia Chirenti

Centro de Matemática, Computação e Cognição.

Universidade Federal do ABC

2012

Prof. Dra. Cecilia Chirenti

Arthur Monteforte

Resumo

Este projeto busca estudar a gravitação na Terra e no sistema solar, comparar e estudar métodos numéricos capazes de resolver E.D.Os, utilizar uma linguagem de computação para aplicar estes métodos a fim de obter resultados e iniciar o estudo de gravitação universal e órbitas.

Abstract

This project aims to study the gravity on Earth and the solar system, compare and study numerical methods capable of solving ODEs, use a computer language to apply these methods in order to get results and start the study of universal gravitation and orbits.

Sumário	pág.
1. Introdução	06
2. Objetivos	07
3. Métodos	07
3.1. Estudo de Literatura	08
3.2. Métodos numéricos	08
3.2.1. Método de Euler	08
3.2.2. Método de Runge-kutta	09
3.3. Ferramentas utilizadas no projeto	09
3.3.1. Microsoft Excel	09
3.3.2. Java e o ambiente de desenvolvimento integrado (IDE) NetBeans	09
4. Desenvolvimento	10
4.1. Estudo de Balística	10
4.1.1. Dedução das equações de movimento	10
4.1.1.1. Equação da Trajetória	10
4.1.1.2. Altura máxima	12
4.1.1.3. Tempo para atingir o solo	13
4.1.2. Exercícios de balística	13
4.2. Foguete de massa variável	14
4.3. Métodos numéricos, estudo do foguete de massa variável e algoritmos em Java	18
5. Resultados	18
5.1. Gráficos dos estudos de balística	18

5.1.1. Movimento horizontal	19
5.1.2. Movimento vertical	21
5.1.3. Movimento oblíquo	26
5.2. Gráficos dos métodos de Euler e Runge-kutta	26
5.2.1. Método de Euler	26
5.2.1.1. E.D.O. $y' = x$	26
5.2.1.2. E.D.O. $y' = \cos(x)$	29
5.2.2. Método de Runge-kutta	31
5.2.2.1. E.D.O $y' = y - x + 1$	31
5.2.2.2. E.D.O $y' = x$	32
5.2.2.3. E.D.O $y' = \cos(x)$	33
5.3. Resultados dos programas em Java dos métodos de Euler e Runge-kutta	34
5.3.1. Algoritmo do Método de Euler	35
5.3.2. Algoritmo do Método de Runge-kutta	37
5.4 Resultados do caso de um foguete de massa variável	38
5.4.1 Gráficos do foguete de massa variável	38
5.4.2 Algoritmos dos programas do foguete de massa variável	43
6. Cronograma	45
7. Conclusões	46
Apêndices	46
A1. Código fonte (em Java) do Método de Euler	46
A2. Código fonte (em Java) do Método de Runge-kutta	47

A3. Linhas de códigos do estudo do foguete de massa variável	49
A3.1 – Código que entrega a relação de tempo, massa, velocidade e aceleração	49
A3.2 Código que compara a massa obtida pelo cálculo linear e pelo exponencial	51
Referências	53

1. Introdução

O estudo da gravidade é de imensa importância no estudo do Universo, pois nos permite entender melhor o funcionamento não só de fenômenos gravitacionais na Terra, como diversos fenômenos Astrofísicos interessantes, como buracos negros, estrelas de neutrons, explosões de raios gama, de onde as estrelas vêm e como elas evoluem e morrem, como ocorre a formação de galáxias etc. A gravidade é a principal força quando tentamos entender o Universo, pois é a responsável direta sobre a sua configuração.

Apresentamos a seguir uma breve revisão baseada na referência [1].

Indo ao início da história do estudo da gravidade, chegamos a Galileu.

Galileu (1564-1642) viveu em uma época onde havia grande interesse na trajetória de balas de canhão por conta das guerras, e seus estudos tanto em campo quanto em laboratório eram direcionados a entender as leis do movimento. Primeiramente, ele descobriu que *a velocidade com que um corpo cai não depende de seu peso. Em seguida, mediu a velocidade com que os corpos caem e verificou que a aceleração é constante, independente do tempo.*

Além disso, Galileu descobriu que *o movimento horizontal e vertical são independentes, e que a velocidade do movimento horizontal não irá mudar ao menos que uma força horizontal seja aplicada.*

Newton (1646-1716), nascido no ano da morte de Galileu, continuou os estudos do movimento, propondo as três leis do movimento, sendo a primeira lei:

Uma vez que um corpo é posto em movimento, ele irá permanecer em velocidade constante em linha reta ao menos que uma força haja sobre ele.

A segunda lei:

Quando uma força é aplicada em um corpo, a aceleração resultante depende da força e da massa do corpo: quanto maior a força, maior a aceleração; e quanto maior a massa, menor a aceleração.

Matematicamente, esta lei é expressa por

$$F = ma,$$

e que levando em consideração a observação de Galileu sobre a aceleração independe do peso, temos que a força gravitacional é proporcional à massa, nos dando

$$F_{\text{grav}} = mg.$$

Por fim, a terceira lei diz que:

Se uma força é exercida em um objeto, ele exerce uma força de volta de mesma magnitude e direção oposta à da força aplicada. Conhecida como lei da “ação e reação”.

No contexto deste projeto, tais premissas são os pontos iniciais no estudo da gravitação no ambiente terrestre e no sistema solar, como no estudo de balística e de um foguete.

2. Objetivos

Os objetivos deste trabalho são:

- Melhorar a compreensão sobre gravidade na Terra e no sistema solar, e possivelmente incluir conceitos relativísticos.
- Estudar métodos matemáticos capazes de resolver E.D.Os.
- Aprender uma linguagem de programação e utilizá-la para aplicar métodos matemáticos que forneçam resultados para serem analisados.
- Aplicar os métodos numéricos aprendidos e a programação afim de estudar situações físicas.
- Complementar os estudos de cálculo e física do BC&T.

3. Métodos

Neste projeto de PDPD, buscou-se aprofundar o entendimento de conceitos físicos pelo estudo e revisão de literatura; estudar os métodos numéricos de Euler e Runge-kutta, a fim de ser capaz de resolver E.D.Os, aplicá-los

numericamente em uma linguagem computacional e obter equações de movimento de projéteis e de um foguete de massa variável.

3.1. Estudo de Literatura

As leituras iniciais sobre a história do estudo da gravidade e das leis da Mecânica se deram no livro Gravity From The Ground Up de Bernard F. Schutz[1]. No estudo de Balística, foi utilizado o livro Curso de Física Básica: Mecânica de Vol.1 H. Moysés Nussenzveig[2], para resolução de exercícios, obtenção de equações de movimento e melhor compreensão dos conceitos físicos envolvidos. Para o estudo do caso de um foguete de massa variável, utilizou-se o livro Equações Diferenciais Aplicadas à Física de Kleber Daum Machado[3]. Em relação aos métodos numéricos, foram utilizadas as apostilas Cálculo Numérico e Computacional de C.Y. Shigue[4] e a encontrada no link http://www.instructioneducation.info/Mechsub/num2_1.pdf[5]. Para o estudo de Java, utilizou-se o livro Kathy Sierra e Bert Bates, Use a Cabeça! Java[6].

3.2. Métodos numéricos

Um método numérico é uma técnica para integrar uma equação, numa função de intervalo entre L_0 e L . Esse intervalo é a soma das N partes da função a serem integradas, distantes entre si por um avanço (passo) h . Quanto menor o passo melhor a precisão do método e menor é o seu erro. O erro E_x é expresso pela relação $E_x = y - y_i$, onde y é o valor analítico e y_i a aproximação do feita pelo método.

Neste projeto, foram estudados os métodos de Euler e Runge-kutta.[4][5]

3.2.1. Método de Euler

O Método de Euler é um método aproximado de 1ª ordem, isto é, ele aproxima a solução da E.D.O. de 1º grau $y' = f(x)$ por uma função de 1º grau, isto é, por uma reta.

A equação genérica para o cálculo da solução de uma E.D.O. de 1º grau pelo Método de Euler é expressa por[4]:

$$y_{i+1} = y_i + hf(x_i, y_i),$$

$$h = x_{i+1} - x_i.$$

3.2.2. Método de Runge-kutta

Os Métodos de Runge-Kutta consistem em métodos de aproximação de 2ª e 4ª ordem.

Neste projeto, utilizamos o Método de Runge-kutta de 4ª ordem[4][5], sendo que se utilizarmos o mesmo valor de passo (h) para o Método de Euler e de Runge-kutta, conseguiremos resultados mais próximos da solução analítica com o método de Runge-kutta.

$$Y_{i+1} = y_i + h/6(k_1 + 2k_2 + 2k_3 + k_4),$$

$$K_1 = f(x_i, y_i),$$

$$K_2 = f(x_i + h / 2, y_i + hk_1 / 2),$$

$$K_3 = f(x_i + h / 2, y_i + hk_2 / 3),$$

$$K_4 = f(x_i + h, y_i + hk_3).$$

3.3. Ferramentas utilizadas no projeto

3.3.1. Microsoft Excel

O Excel foi utilizado para o estudo de balística, na resolução numérica de problemas; na criação de gráficos de movimento; na aplicação dos métodos numéricos de Euler e Runge-kutta, gerando gráficos das funções estudadas e realizando uma comparação dos valores obtidos entre a solução analítica, o Método de Euler e o de Runge-kutta; e por fim na aplicação do método de Euler no estudo de um foguete de massa variável, obtendo gráficos relacionando as grandezas físicas deste estudo.

3.3.2. Java e o ambiente de desenvolvimento integrado (IDE) NetBeans

A linguagem de programação Java foi escolhida para a criação de programas capazes de aplicar os métodos numéricos de Euler e Runge-kutta, gerando dados que possam ser analisados posteriormente. O NetBeans é o IDE oficial da Sun Microsystems, utilizado para compilar e executar tais programas.

4. Desenvolvimento

4.1. Estudo de balística

Para o estudo de balística, foi utilizado a referência [2], para resolução de exercícios e obtenção das equações de movimento.

4.1.1. Dedução das equações de movimento

O estudo e obtenção das equações de movimento foi realizado com o intuito de aprimorar a compreensão dos conceitos físicos envolvidos no estudo de balística.

4.1.1.1. Equação da Trajetória

Para diferenciar as grandezas vetoriais das grandezas escalares nas equações a seguir, os vetores serão escritos em negrito.

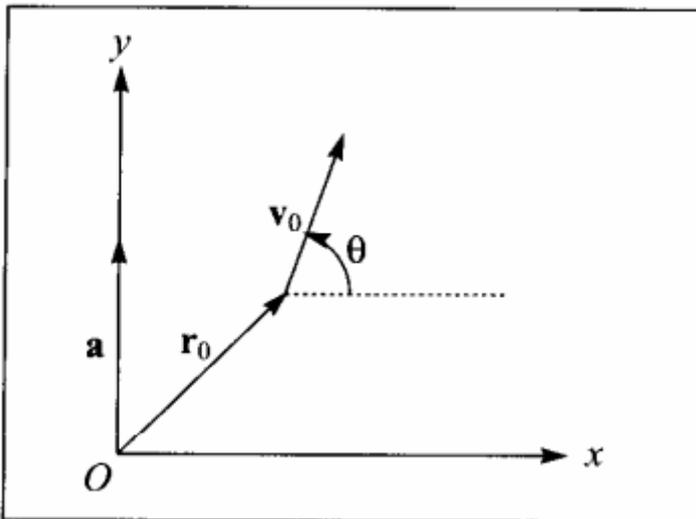


Figura 1: Representação da velocidade (v_0) e deslocamento (r_0) de um corpo em um sistema de coordenadas.[2]

Dividindo v_0 e r_0 em suas componentes na direção x e na direção y, temos que

$$v_0 = v_{0x}i + v_{0y}j, \text{ e } r_0 = x_0i + y_0j.$$

Como velocidades em cada eixo, temos:

$$v_x(t) = v_{0x} \text{ e } v_y(t) = v_{0y} + a(t - t_0),$$

e como equações do deslocamento:

$$x(t) = x_0 + v_{0x}t,$$

$$y(t) = y_0 + v_{0y}t + \frac{1}{2}at^2.$$

Se a aceleração for diferente de zero, teremos como equações de velocidade e deslocamento, respectivamente:

$$v(t) = v_0 + a(t - t_0),$$

e

$$r(t) = v_0(t-t_0) + \frac{1}{2} a(t - t_0)^2.$$

Se a aceleração for nula ($a = 0$), se tratando de um Movimento Retilíneo Uniforme (M.R.U), iremos obter como equação da posição, em cada eixo:

$$x = x_0 + v_{0x}(t - t_0),$$

e

$$y = y_0 + v_{0y}(t - t_0).$$

Dessas duas equações, podemos tirar que:

$$\frac{x-x_0}{v_{0x}} = \frac{y-y_0}{v_{0y}},$$

e

$$t - t_0 = \frac{x-x_0}{v_{0x}}.$$

Assim, somos capazes de obter a Equação da Trajetória (para $a = 0$):

$$y - y_0 = \frac{v_{0y}}{v_{0x}}(x - x_0).$$

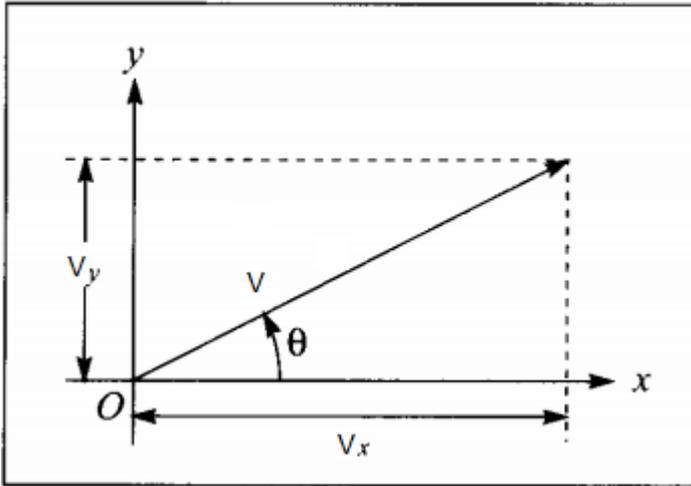


Figura 2: Vetor separado em suas componentes no eixo X e Y.[2]

Conforme representado na Figura 2, de acordo com o ângulo Θ , podemos escrever as equações das velocidades em cada eixo, de forma a ficar $v_{0y} = v_0 \sin \Theta$ e $v_{0x} = v_0 \cos \Theta$ no instante inicial.

Como $\frac{v_{0y}}{v_{0x}} = \tan \Theta$, podemos escrever a Equação da Trajetória como:

$$y - y_0 = \tan \Theta (x - x_0).$$

No caso em que haja aceleração da gravidade, temos como velocidades em cada eixo $v_x = v_0 \cos \Theta$ (pois o movimento horizontal permanece o mesmo, se não houver nenhuma força agindo horizontalmente) e $v_y = v_0 \sin \Theta - gt$.

As posições em cada eixo se dão por $x = v_0 \cos \Theta t$ e $y = v_0 \sin \Theta t - \frac{1}{2} gt^2$.

Na posição no eixo y, substituindo $v_0 \sin \Theta t$ por v_{0y} , e t por $\frac{x-x_0}{v_{0x}}$ (como vimos acima, achando a Equação da Trajetória), obtemos:

$$y - y_0 = v_{0y} \frac{x-x_0}{v_{0x}} - \frac{1}{2} g \frac{(x-x_0)^2}{v_{0x}^2}.$$

Com $x_0 = 0$, $y_0 = 0$ e $\frac{v_{0y}}{v_{0x}} = \tan \Theta$, podemos escrever a posição vertical como:

$$y = \tan(\Theta)x - \frac{1}{2} g \frac{x^2}{v_0^2 \cos^2(\Theta)}.$$

4.1.1.2. Altura máxima

Como na altura máxima atingida a velocidade é nula, temos que $v_y = 0$.

Então, $0 = v_{0y} - gt_{\text{máx}}$.

$$T_{\text{máx}} = v_0 \text{sen} \theta / g.$$

Substituindo esta equação de $T_{\text{máx}}$ na equação da posição vertical obtida anteriormente, temos:

$$y = v_0^2 \text{sen}^2 \theta / 2g.$$

4.1.1.3. Tempo para atingir o solo

Quando o corpo atinge o solo, a posição vertical possui valor nulo, isto é, $y_{\text{máx}} = 0$. Substituindo $y_{\text{máx}} = 0$ na equação

$$y = v_0 \text{sen} \theta t - \frac{1}{2}gt^2,$$

teremos

$$0 = v_0 \text{sen} \theta t - \frac{1}{2}gt^2.$$

Como vimos que $T_{\text{máx}} = v_0 \text{sen} \theta / g$., podemos substituir na equação acima e obter

$$T_{\text{solo}} = 2v_0 \text{sen} \theta / g = 2t_{\text{máx}}.$$

Ou seja, o tempo para atingir o solo é o dobro do tempo para atingir a altura máxima.

4.1.2. Exercícios de balística

Foram realizados exercícios de movimento vertical, horizontal e oblíquo, a fim de aplicar obter numericamente as relações de velocidade, posição e tempo em cada caso, além de expressar tais situações graficamente.

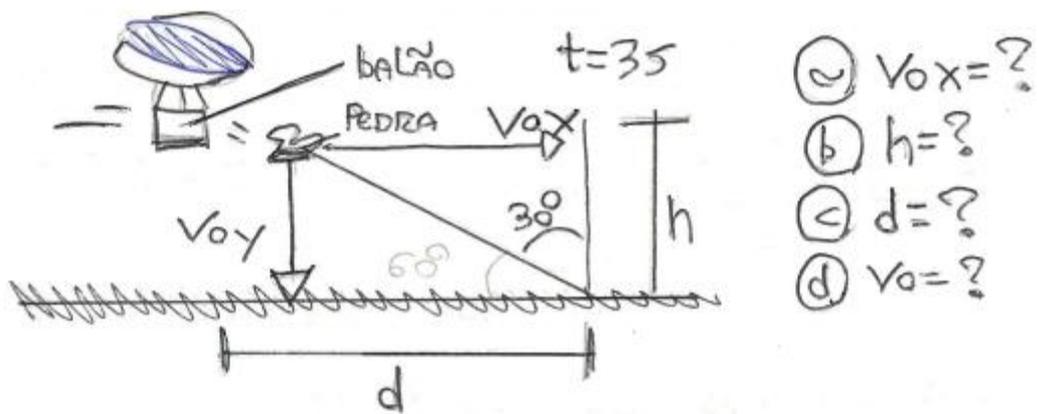


Figura 3: Exemplo de exercício com movimento horizontal e vertical.

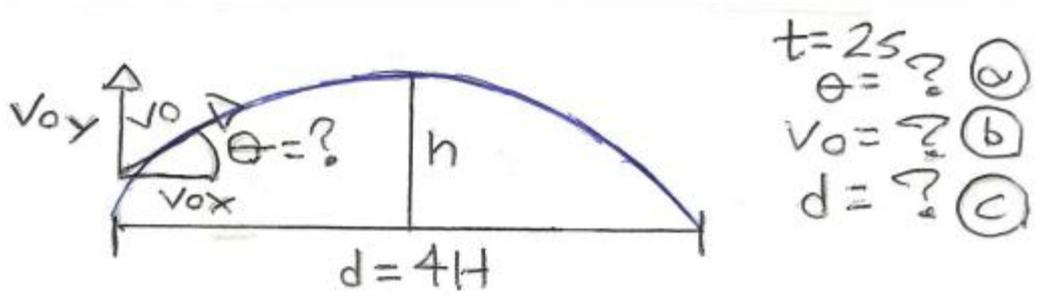


Figura 4: Exemplo de exercício de movimento oblíquo.

4.2. Foguete de massa variável

Os estudos das equações de movimento de um foguete de massa variável se basearam na referência [6].

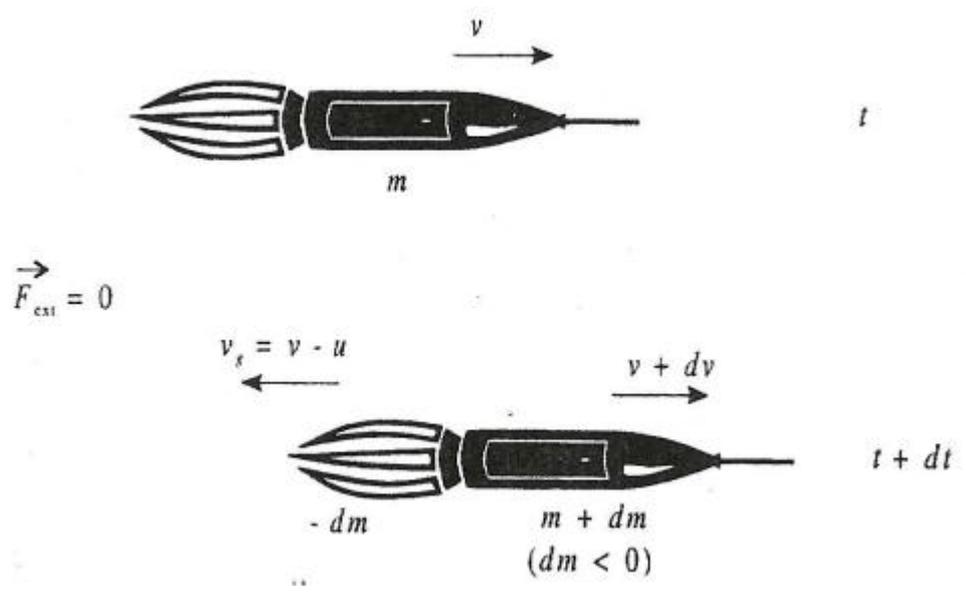


Figura 5: Modelo matemático de um foguete se movendo no espaço, livre de forças externas.
Referência [6], capítulo 3, pag. 73.

Nesta situação, temos m como massa inicial do foguete, t como tempo inicial, v como a velocidade inicial; e no instante seguinte, dm como a massa expelida (sendo $dm < 0$), $v - u$ como a velocidade da massa expelida, $t + dt$ como o aumento do tempo, a velocidade para $v + dv$, e a massa diminui para $m + dm$.

Como o foguete está livre de forças externas, temos que o momento linear é conservado.

Para o tempo t , temos:

$$P(t) = mv.$$

Em $t + dt$, o momento linear é:

$$P(t + dt) = (m + dm)(v + dv) + (-dm)(v - u),$$

sendo $(m + dm)(v + dv)$ o momento linear do foguete, e $(-dm)(v - u)$ o momento linear da massa expelida em forma de gases da combustão.

Desenvolvendo a equação acima, ficará

$$P(t + dt) = mv + mdv + \cancel{dmv} + dmdv + udm - \cancel{dmv}.$$

Como $dmdv$ é o produto de valores infinitesimais, pode ser desconsiderado. E sendo o momento linear constante (por não haver forças externas agindo sobre o foguete), temos que o momento final é igual o inicial. Portanto:

$$P(t) = P(t + dt),$$

$$mv = mv + mdv + udm,$$

$$mdv = - udm,$$

$$dv = -udm/m.$$

Integrando dos dois lados, o lado esquerdo em relação a velocidade e o direito em relação a massa, iremos obter:

$$\int_{v_0}^v dv = -u \left[\int_{m_0}^m dm/m \right],$$

$$v - v_0 = -u \ln\left[\frac{m}{m_0}\right],$$

$$v = v_0 + u \ln\left[\frac{m_0}{m}\right].$$

Por esta equação, podemos observar que o principal fator da velocidade de um foguete é a relação entre sua massa inicial e a massa final. Por este motivo que os foguetes são construídos de forma a liberar as carcaças onde o combustível já foi consumido.

Voltando em $dv = \frac{-u dm}{m}$, podemos dividir os dois lados por dt , obtendo:

$$\frac{mdv}{dt} = \frac{-u dm}{dt}.$$

Como $a = \frac{dv}{dt}$, e $F = ma$, podemos escrever

$$E = \frac{-u dm}{dt},$$

sendo E a força empuxo do motor do foguete. E se considerarmos apenas o foguete como sistema, temos

$$a = \frac{dm}{dt},$$

que é a taxa de consumo de combustível.

No caso de um lançamento vertical, considerando a força da gravidade como força externa, como representado a seguir:

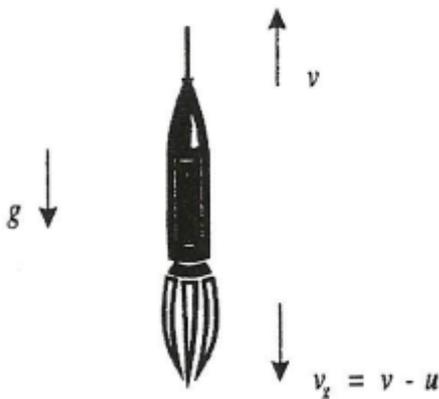


Figura 6: Modelo matemático de um foguete na decolagem, sujeito à força da gravidade. Referência [6], capítulo 3, pag. 75.

Temos que:

$$F_{\text{ext}} = \frac{dp}{dt},$$

$$F_{\text{ext}}dt = p(t + dt) - p(t),$$

$$F_{\text{ext}}dt = (m + dm)(v + dv) + (v - u)(-dm) - mv,$$

$$F_{\text{ext}}dt = mv + mdv + dm v + dm dv - mdv + u dm - mv.$$

Como a força externa é o peso do foguete:

$$- mgdt = mdv + u dm.$$

$$- mg = \frac{mdv}{dt} + \frac{u dm}{dt},$$

$$- dv = - gdt - \frac{u dm}{m}$$

$$\int_{v_0}^v dv = -g \int_{t_0}^t dt - u \int_{v_0}^v \frac{u dm}{m}$$

$$v - v_0 = -g(t - t_0) - u \ln\left[\frac{m}{m_0}\right].$$

Considerando que o foguete partiu do repouso ($v_0 = 0$), e se adotarmos $t_0 = 0$ como tempo inicial, fica

$$v = -gt + u \ln\left[\frac{m_0}{m}\right].$$

Podemos retirar a dependência temporal, e colocando em termos de massa, considerando a taxa de queima do combustível como uma constante, temos:

$$A = \frac{dm}{dt} \text{ ou } dt = \frac{dm}{a}$$

Substituindo em $dv = - gdt - \frac{u dm}{m}$

$$dv = - g\left(\frac{dm}{a}\right) - \frac{u dm}{m}$$

$$\int_{v_0}^v dv = -\frac{g}{a} \int_{m_0}^m dm - u \int_{m_0}^m \frac{u dm}{m}$$

$$V = -\frac{g}{a}(m - m_0) + u \ln\left[\frac{m_0}{m}\right].$$

A altura que o foguete alcança é dada por:

$$a = \frac{dm}{dt}$$

$$dm = a dt$$

$$\int_{m_0}^m dm = a \int_{t_0}^t dt$$

$$m - m_0 = a(t - t_0)$$

$$m - m_0 = at$$

$$m = m_0 - at$$

E a altura

$$y = -\frac{gt^2}{2} - ut + \left(\frac{mu}{a}\right) \ln\left[\frac{m_0}{m}\right].$$

Assim, obtemos as principais equações de um foguete de massa variável, dadas essas duas situações.

4.3. Métodos numéricos, estudo do foguete de massa variável e algoritmos em Java

Utilizando as equações descritas na seção 3.2. métodos numéricos, realizou-se a resolução e representação gráfica de algumas E.D.Os, e posteriormente criou-se dois programas em Java capazes de resolvê-las, um para o método de Euler e outro para o método de Runge-kutta.

Além desse estudo inicial das E.D.Os, foram criados programas para o estudo do caso de um foguete de massa variável (visto na seção 4.2), tendo este programa uma saída externa em que os resultados obtidos são salvos em um arquivo de texto, para análise posterior e criação de gráficos que nos permitam estudar o caso do foguete.

5. Resultados

5.1 Gráficos dos estudos de balística

Após a resolução dos exercícios, foram criados gráficos com o intuito de representar as situações de movimentos horizontais, verticais e oblíquos estudados, utilizando o Excel.

5.1.1. Movimento horizontal

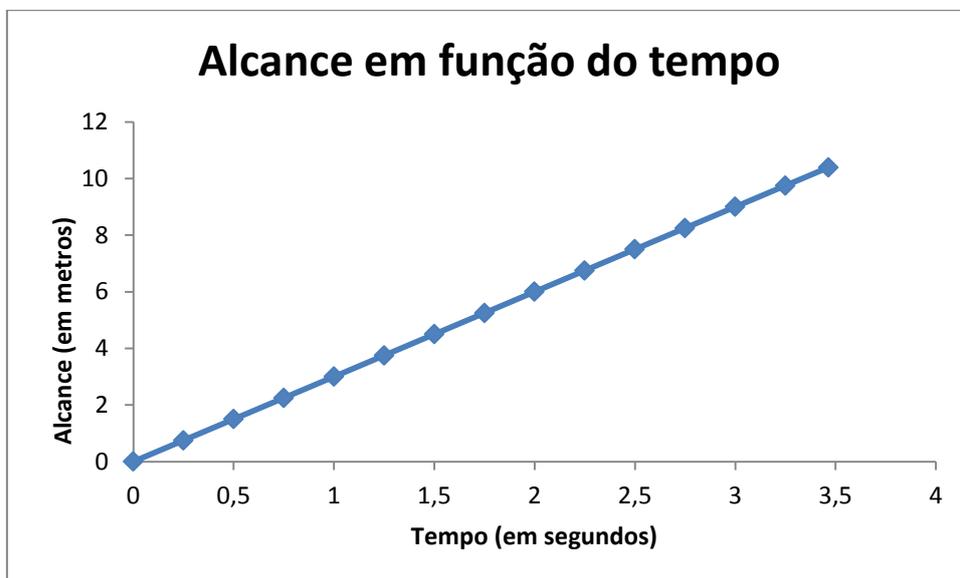


Figura 7: Variação da posição em x com o passar do tempo.

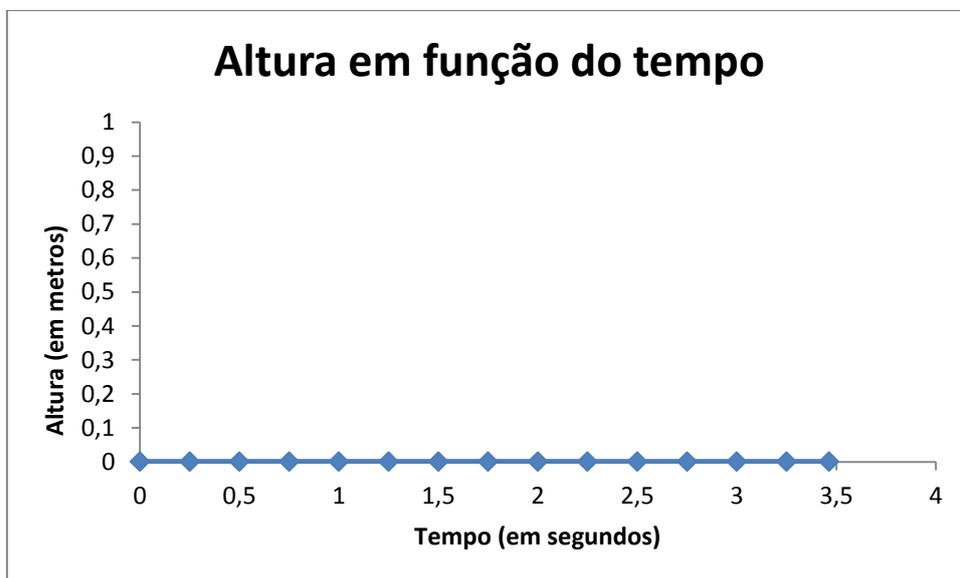


Figura 8: Como não há movimento vertical, a altura não sofre alteração com o passar do tempo.

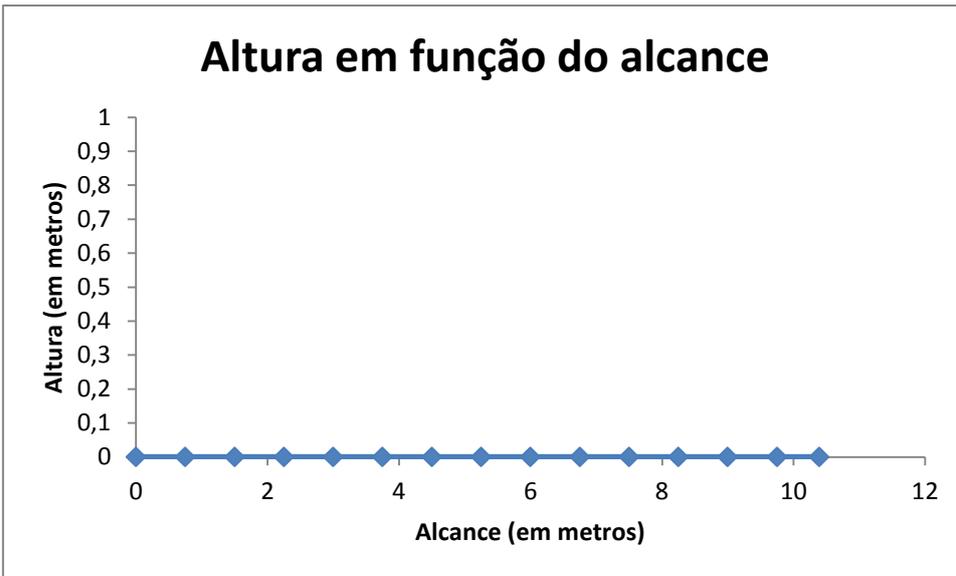


Figura 9: Gráfico do movimento da posição em x pela posição em y . Como não há movimento vertical, apenas ocorre deslocamento na posição x .

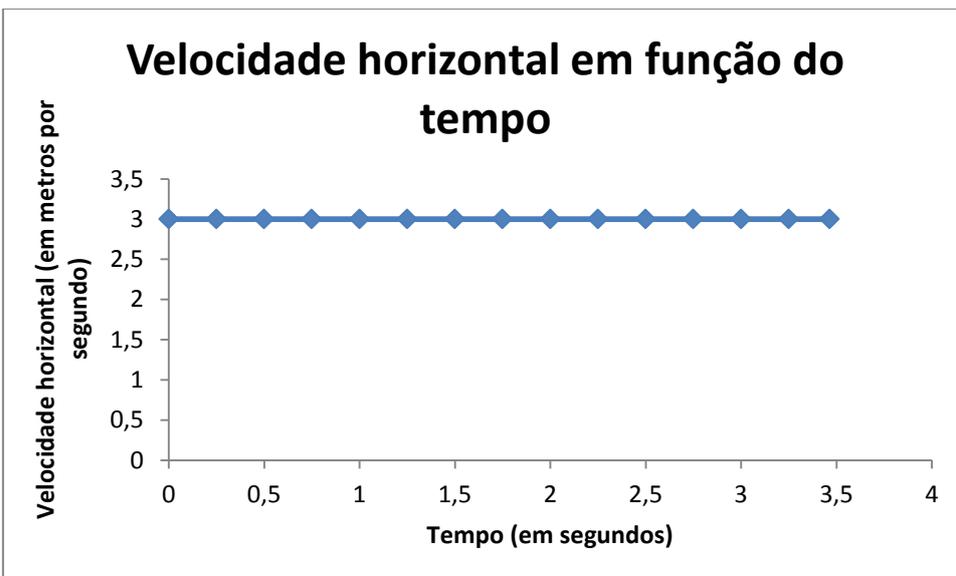


Figura 10: Gráfico da velocidade horizontal em função do tempo. Como dito por Galileu e Newton, não ocorre alteração da velocidade horizontal, ao menos que uma força haja horizontalmente.

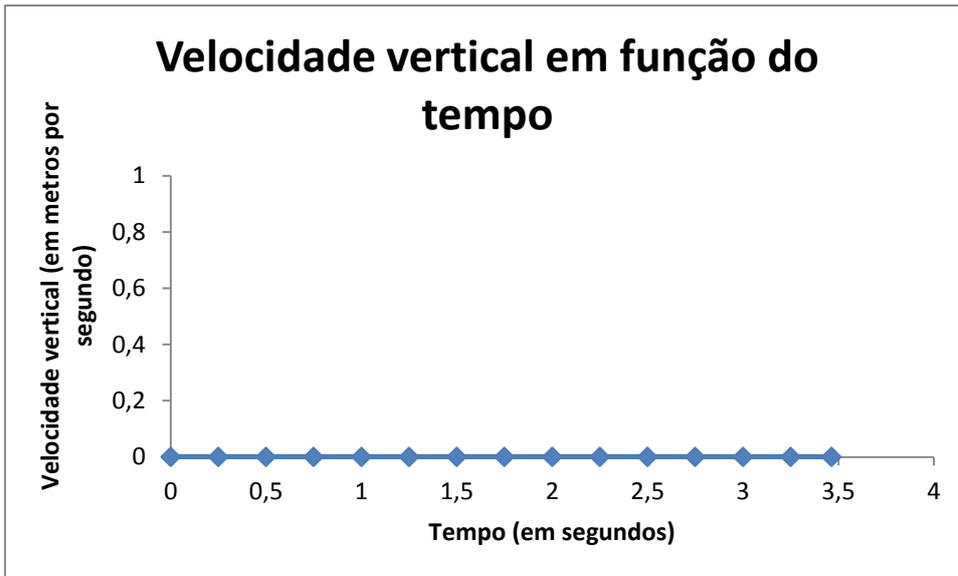


Figura 11: Velocidade vertical em função do tempo. Novamente, como não há movimento horizontal, a velocidade vertical permanece nula todo o tempo.

5.1.2. Movimento Vertical

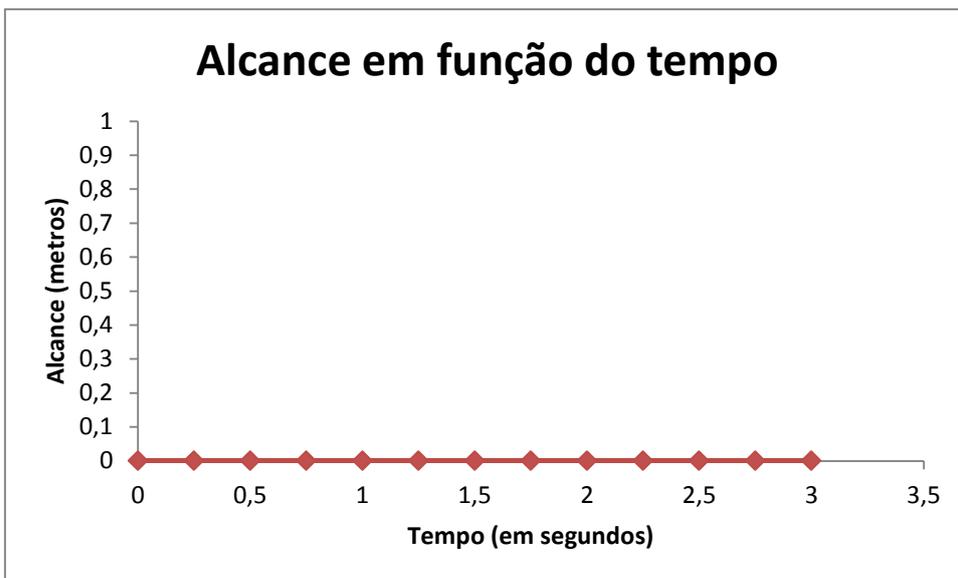


Figura 12: Variação da posição em x com o tempo. Como não há movimento horizontal, a posição em x não sofre alteração.



Figura 13: Variação da posição em y em função do tempo. Como se trata de um movimento de queda, a altura máxima se dá no ponto inicial.

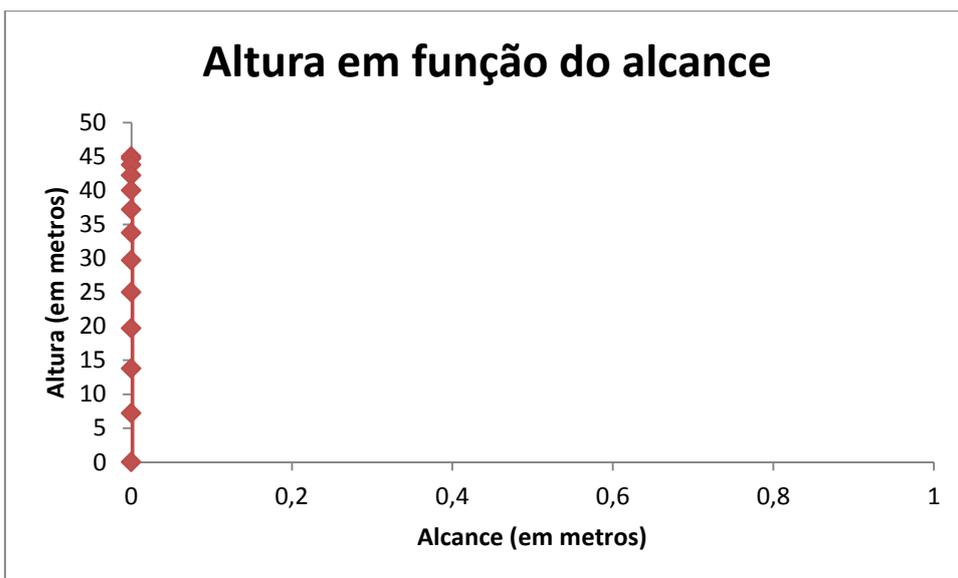


Figura 14: Gráfico do movimento da posição em x pela posição em y . Como não há movimento horizontal, a posição em x não sofre alterações. Além disso, o gráfico demonstra o movimento de uma queda.

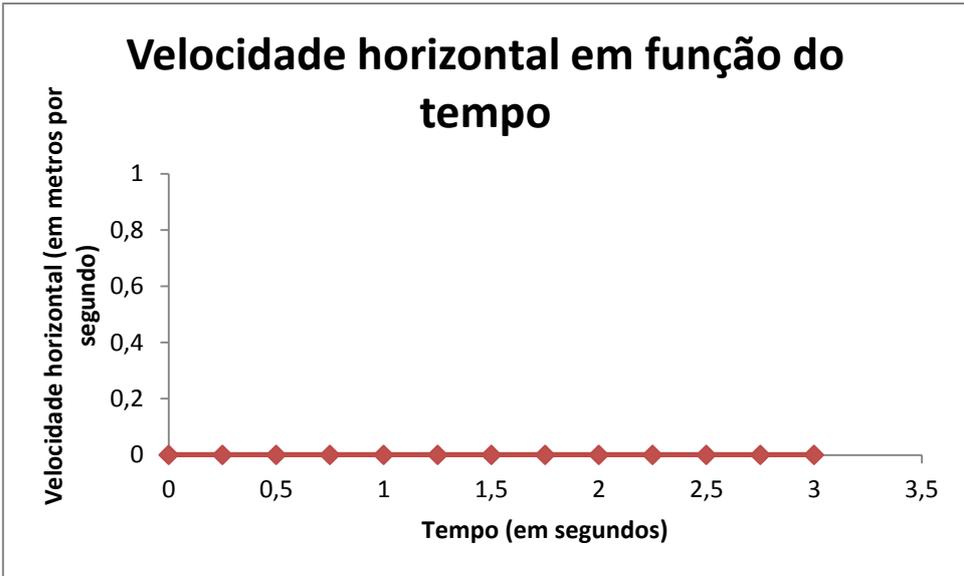


Figura 15: Variação da velocidade horizontal em função do tempo. Como não há movimento horizontal, ela permanece nula.



Figura 16: Variação da velocidade vertical em função do tempo. Como há ação da gravidade, ela aumenta em $9,8m.s^{-2}$.

5.1.3. Movimento oblíquo

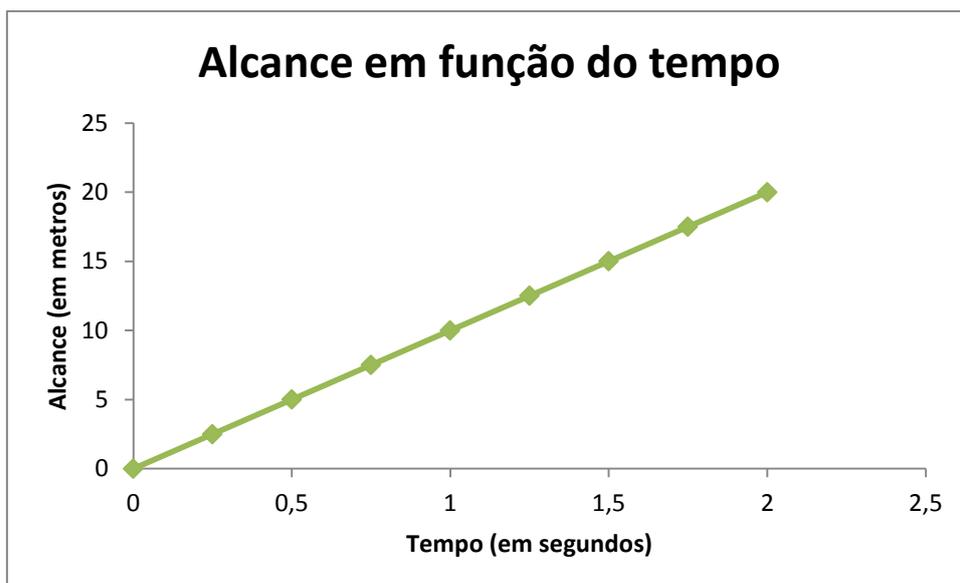


Figura 17: Gráfico da variação da posição em x pelo tempo. Como há movimento horizontal, ocorre deslocamento horizontal com o passar do tempo.



Figura 18: Gráfico da variação da posição em y em função do tempo. Como há movimento vertical sob influência da aceleração da gravidade, a altura varia até um ponto máximo e então passa a diminuir. Como deduzido matematicamente pela obtenção das equações de movimento anteriormente, o tempo para se chegar ao alcance máximo ($y = 0$) é duas vezes o tempo para se chegar à altura máxima ($v = 0$).

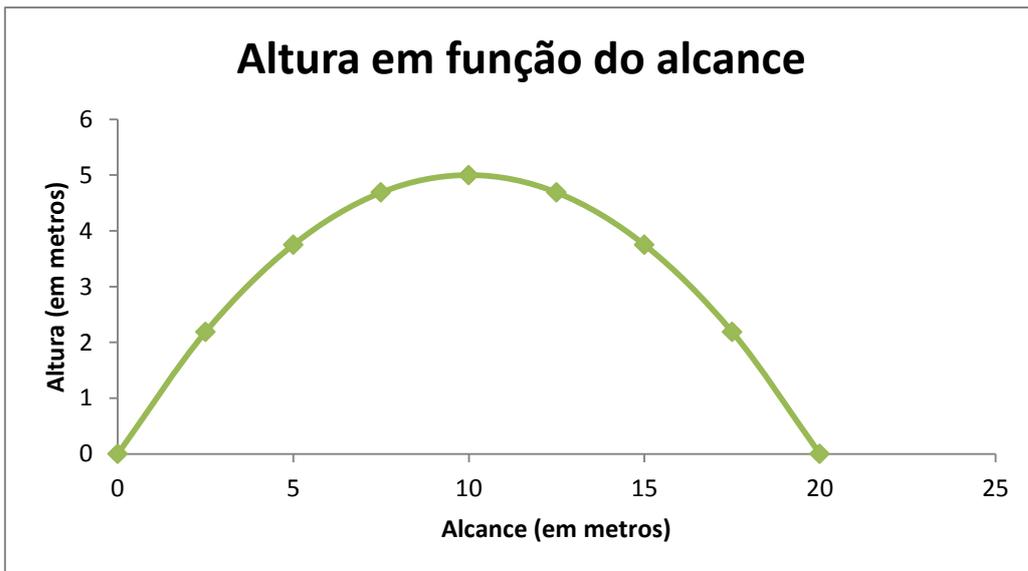


Figura 19: Gráfico da trajetória do movimento da posição em x pela posição em y . Como há movimento nos dois eixos, ambos sofrem variação.

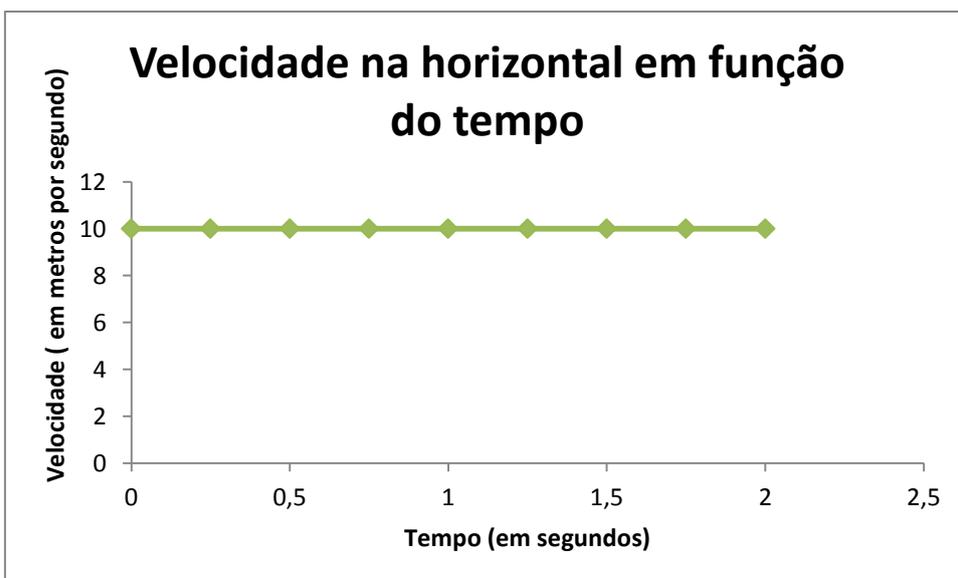


Figura 20: Gráfico da variação da velocidade em x em função do tempo. Novamente, como não há forças agindo horizontalmente, a velocidade não se altera, como dito por Galileu e Newton.

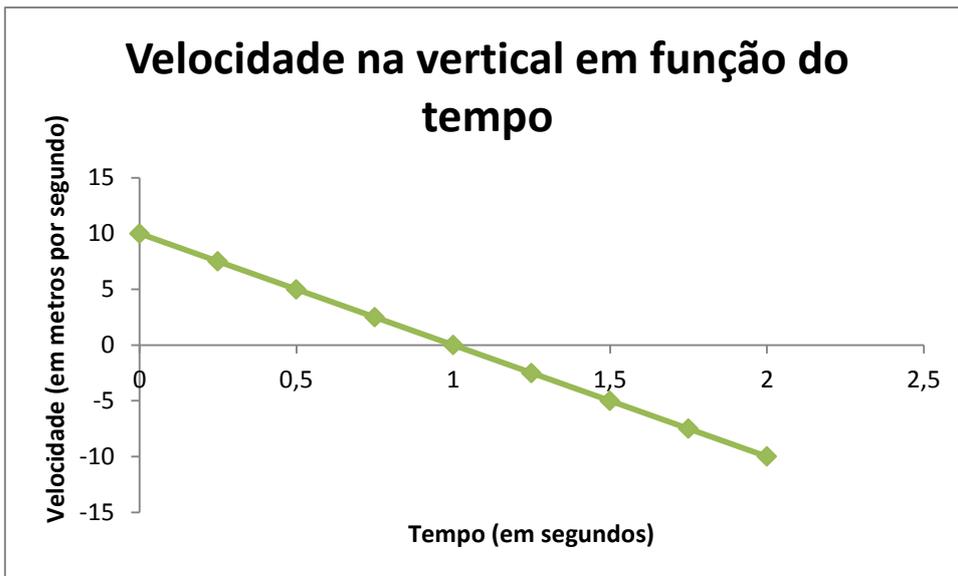


Figura 21: Gráfico da variação da velocidade vertical em função do tempo. Enquanto o corpo sobe, a velocidade é positiva, mas é reduzida pela ação da gravidade, até se tornar nula (altura máxima). O corpo então passa a cair, e a velocidade se torna negativa.

5.2. Gráficos dos métodos de Euler e Runge-kutta

5.2.1. Método de Euler

Neste estudo, aplicou-se o método de Euler em diferentes E.D.Os, e se analisou os diferentes resultados variando o valor do passo (h), além de comparar os resultados obtidos pelo método de Euler com a solução analítica. Nas tabelas, x_i é o valor no eixo x , y_i é o valor em y obtido pelo Método de Euler, e y_{exato} é o valor obtido pela solução analítica. Além disso, a primeira linha de cada tabela se refere as condições iniciais do estudo da E.D.O.

5.2.1.1. E.D.O. $y' = x$

Inicialmente, utilizou-se $h = 4$.

x_i	y_i	y_{exato}
0	2	2
4	18	10
8	50	34

Figura 22: Método de Euler com passo $h = 4$.

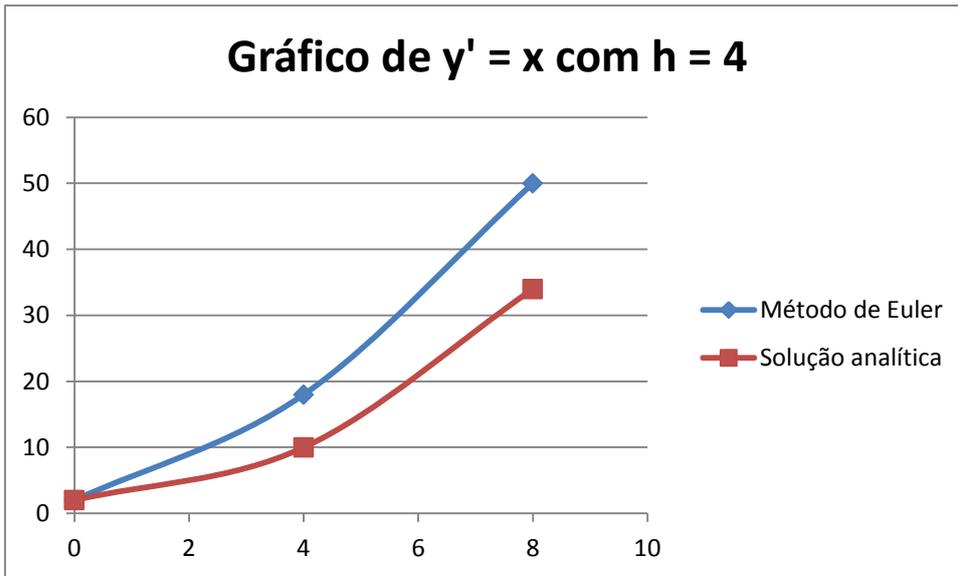


Figura 23: Gráfico comparando o método de Euler com a solução analítica. Como o valor do passo (h) é elevado, o erro se acumula rapidamente.

Utilizando o valor do passo como $h = 2$, obtemos:

x_i	y_i	y_{exato}
0	2	2
2	6	4
4	14	10
6	26	20
8	42	34

Figura 24: Método de Euler com passo $h = 2$.

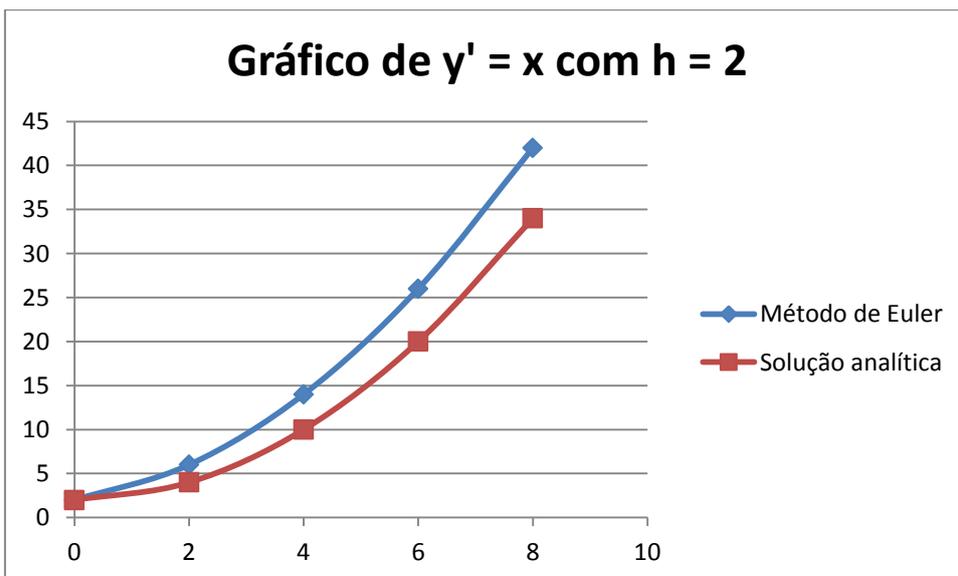


Figura 25: Com a diminuição do valor do passo, o erro se torna menor.

Por fim, utilizando $h = 0,5$, obtemos:

X_i	y_i	y_{exato}
0	2	2
0,5	2,25	2,125
1	2,75	2,5
1,5	3,5	3,125
2	4,75	4
2,5	6	5,125
3	7,5	6,5
3,5	9,25	8,125
4	11,25	10
4,5	13,5	12,125
5	16	14,5
5,5	18,75	17,125
6	21,75	20
6,5	25	23,125
7	28,5	26,5
7,5	32,25	30,125
8	36,25	34

Figura 26: Método de Euler com passo $h = 0,5$.

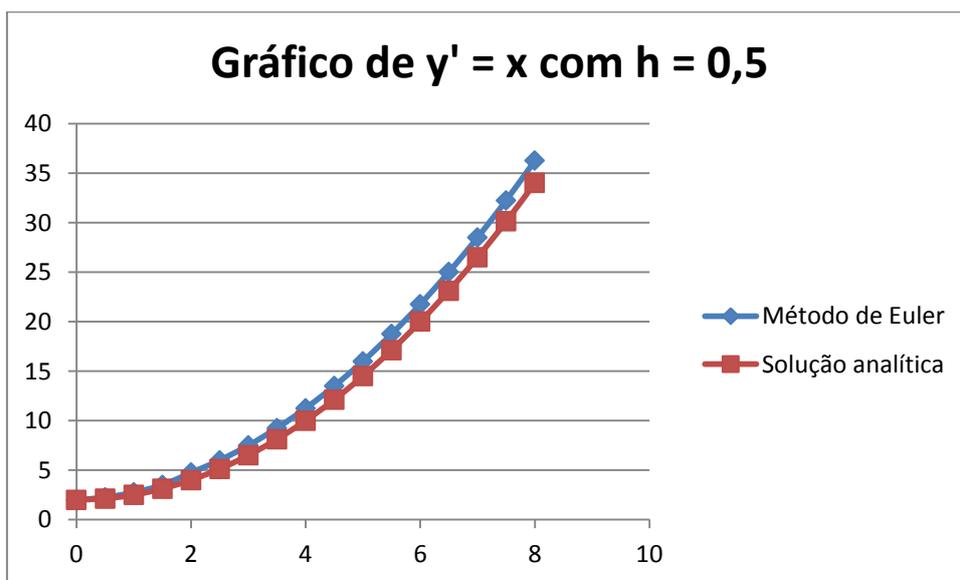


Figura 27: Com a diminuição do valor do passo, podemos ver tanto graficamente quanto pela tabela que os valores são bem mais próximos da solução analítica.

5.2.1.2 E.D.O. $y' = \cos(x)$

Inicialmente, utilizou-se $h = 1$.

x_i	y_i	y_{exato}
0	2,00	2,00
1	3,00	2,841
2	3,540	2,909
3	3,124	2,14
4	2,134	1,243
5	1,480	1,041
6	1,764	1,720
7	2,724	2,656
8	3,478	2,989
9	3,332	2,412
10	2,421	1,455
11	1,582	1,00
12	1,586	1,463
13	2,430	2,420
14	3,338	2,990
15	3,475	2,650
16	2,715	1,712
17	1,757	1,038
18	1,482	1,249
19	2,142	2,149
20	3,131	2,912

Figura 28: Método de Euler com passo $h = 1$.

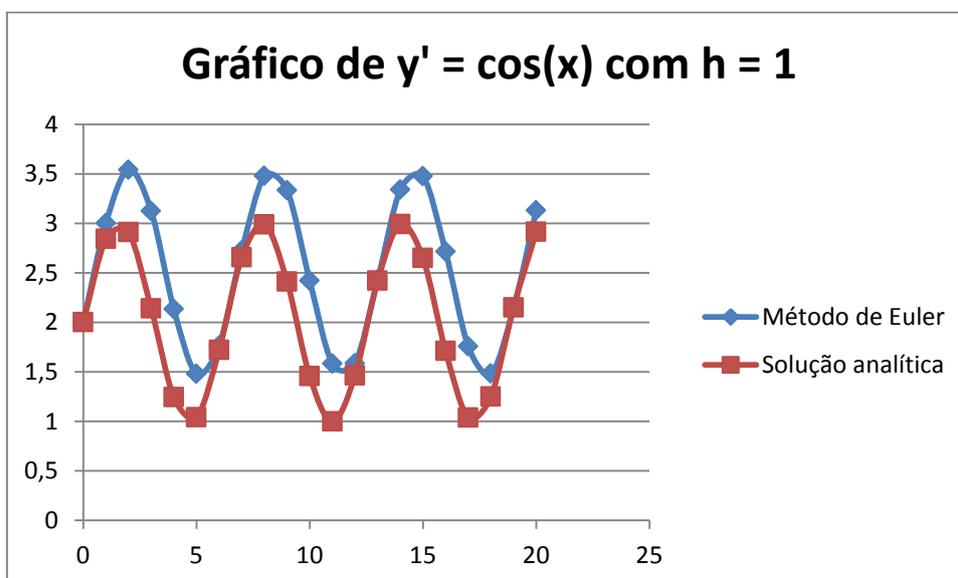


Figura 29: Gráfico comparando o método de Euler com a solução analítica. Neste caso, com este valor de passo, ocorre um erro acumulado acima da solução analítica.

Alterando o passo para $h = 0,25$ temos:

X_i	Y_i	Y_{exato}	9,75	1,586	1,680
0	2,000	2,000	10	1,376	1,455
0,25	2,250	2,247	10,5	1,088	1,120
0,5	2,490	2,479	10,75	1,027	1,030
0,75	2,709	2,681	11	1,028	1,000
1	2,890	2,841	11,25	1,091	1,032
1,25	3,025	2,948	11,5	1,212	1,124
1,5	3,10	2,997	11,75	1,383	1,271
1,75	3,110	2,983	12	1,594	1,463
2	3,070	2,909	12,25	1,831	1,688
2,25	2,966	2,778	12,5	2,081	1,933
2,5	2,808	2,598	12,75	2,327	2,182
2,75	2,608	2,381	13	2,553	2,420
3	2,377	2,141	13,25	2,747	2,631
3,25	2,129	1,891	13,5	2,896	2,803
3,5	1,880	1,649	13,75	2,990	2,925
3,75	1,645	1,428	14	3,025	2,990
4	1,440	1,243	14,25	2,996	2,993
4,25	1,277	1,105	14,5	2,908	2,934
4,5	1,166	1,022	14,75	2,764	2,818
4,75	1,113	1,000	15	2,574	2,650
5	1,104	1,041	15,25	2,350	2,442
5,25	1,232	1,141	15,5	2,105	2,206
5,5	1,409	1,294	15,75	1,855	1,957
5,75	1,624	1,491	16	1,616	1,712
6	1,864	1,720	16,25	1,402	1,484
6,25	2,114	1,966	16,5	1,226	1,288
6,5	2,358	2,215	16,75	1,100	1,136
6,75	2,581	2,450	17	1,031	1,038
7	2,770	2,656	17,25	1,024	1,000
7,25	2,912	2,823	17,5	1,079	1,024
7,5	2,999	2,938	17,75	1,192	1,108
7,75	3,024	2,994	18	1,358	1,249
8	2,988	2,989	18,25	1,564	1,435
8,25	2,892	2,922	18,5	1,799	1,657
8,5	2,741	2,798	18,75	2,048	1,900
8,75	2,546	2,624	19	2,295	2,149
9	2,318	2,412	19,25	2,525	2,389
9,25	2,072	2,173	19,5	2,724	2,605
9,5	1,823	1,924	19,75	2,879	2,783
9,75	1,586	1,680	20	2,981	2,912

Figura 30: Método de Euler com $h = 0,25$.

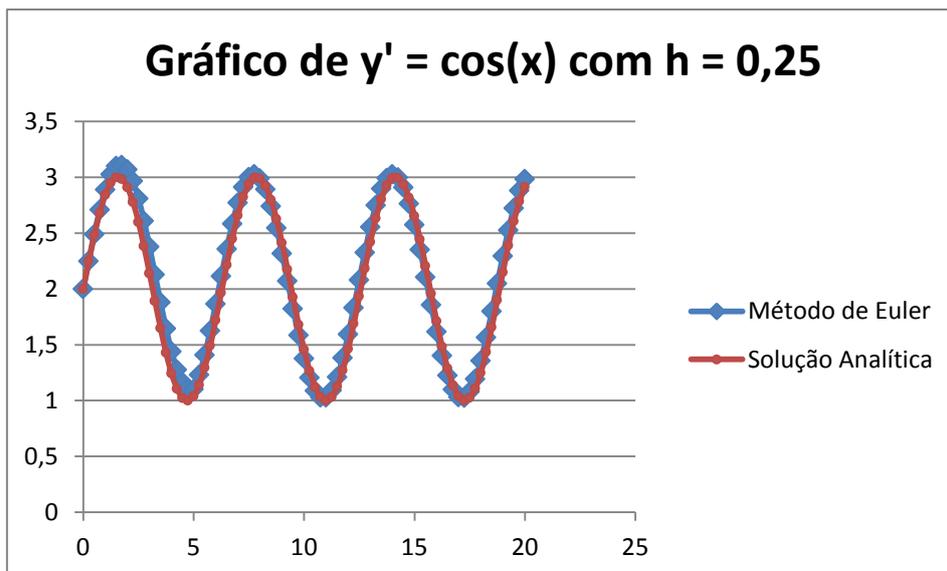


Figura 31: Com este valor pequeno para o passo, os pontos praticamente coincidem.

5.2.2 Método de Runge-kutta

Nesta parte, aplicou-se o Método de Runge-kutta de 4ª ordem para obter soluções de E.D.Os, e realizou-se uma comparação entre este método, o de Euler e a solução analítica. Nas tabelas abaixo, x_i é o valor de x , y_i é o valor obtido pelo método de Runge-kutta, y_{euler} é o valor obtido pelo método de Euler, e y_{exato} é o valor obtido pela solução analítica.

5.2.2.1 E.D.O. $y' = y - x + 1$

Para realização dos dois métodos, utilizou-se como valor do passo $h = 0,2$.

x_i	y_i	y_{euler}	y_{exato}				
				3,6	-36,790	-28,498	-36,776
1	-2	-2	-1,999	3,8	-45,532	-34,7176	-45,514
1,2	-2,464	-2,4	-2,463	4	-56,254	-42,2211	-56,231
1,4	-3,075	-2,92	-3,074	4,2	-69,394	-51,2653	-69,365
1,6	-3,866	-3,584	-3,865	4,4	-85,488	-62,1583	-85,451
1,8	-4,876	-4,420	-4,875	4,6	-105,190	-75,270	-105,143
2	-6,154	-5,464	-6,153	4,8	-129,298	-91,044	-129,237
2,2	-7,760	-6,757	-7,758	5	-158,787	-110,013	-158,710
2,4	-9,765	-8,349	-9,762	5,2	-194,850	-132,815	-194,751
2,6	-12,25	-10,299	-12,25	5,4	-238,940	-160,218	-238,816
2,8	-15,34	-12,679	-15,34	5,6	-292,837	-193,142	-292,680
3	-19,16	-15,575	-19,16	5,8	-358,711	-232,691	-358,513
3,2	-23,87	-19,090	-23,86	6	-439,214	-280,189	-438,963
3,4	-29,66	-23,348	-29,65	6,2	-537,585	-337,226	-537,268

Figura 32: Métodos com $h = 0,2$

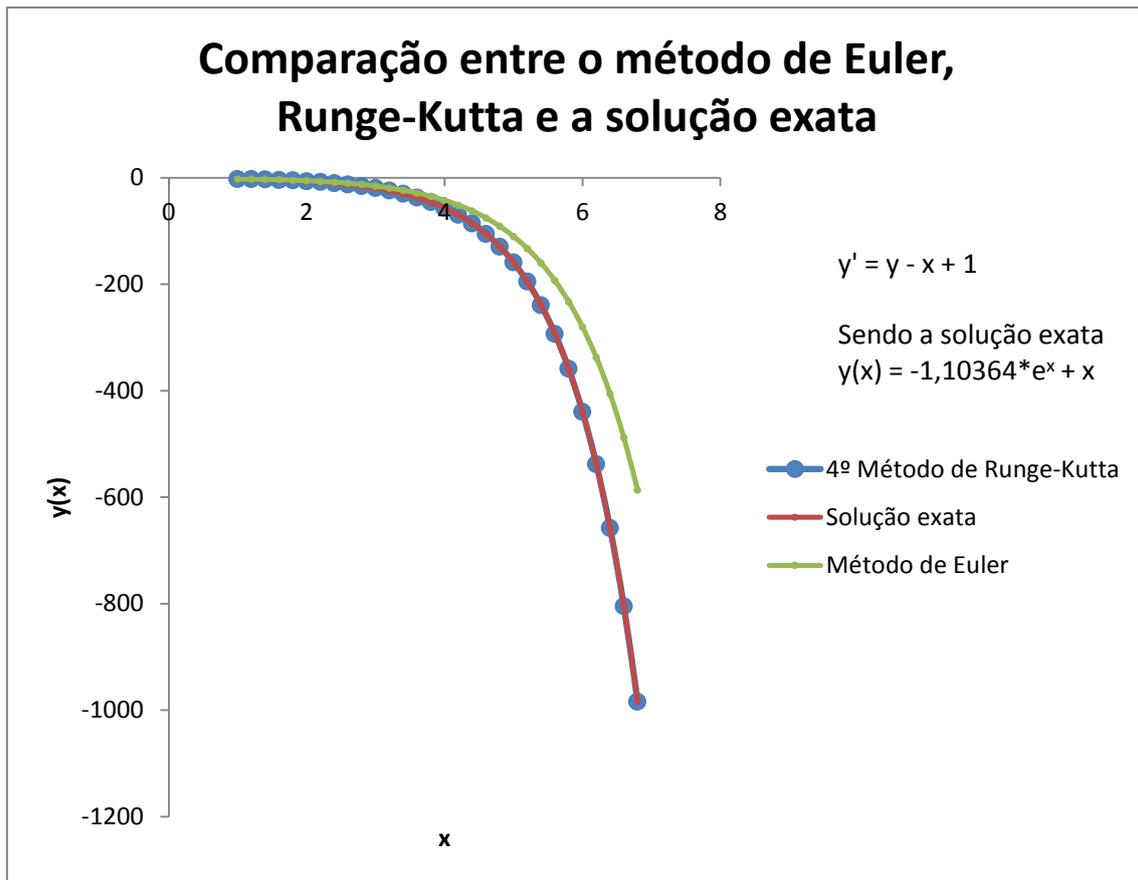


Figura 33: Gráfico de comparação entre o método de Runge-kutta, Euler e a solução analítica. Neste caso, o método de Runge-kutta praticamente coincide com a solução exata, e o método de Euler sofre um grande desvio da solução exata.

5.2.2.2 E.D.O. $y' = x$

Para a realização dos dois métodos, foi utilizado o valor do passo como $h = 0,5$.

X_i	Y_i	Y_{euler}	Y_{exato}	6,5	23,125	24,75	23,125
0	2	2	2	7	26,5	28,25	26,5
0,5	2,125	2,25	2,125	7,5	30,125	32	30,125
1	2,5	2,75	2,5	8	34	36	34
1,5	3,125	3,5	3,125	8,5	38,125	40,25	38,125
2	4	4,5	4,00	9	42,5	44,75	42,5
2,5	5,125	5,75	5,125	9,5	47,125	49,5	47,125
3	6,5	7,25	6,5	10	52	54,5	52
3,5	8,125	9	8,125	10,5	57,125	59,75	57,125
4	10	11	10	11	62,5	65,25	62,5
4,5	12,125	13,25	12,125	11,5	68,125	71	68,125
5	14,5	15,75	14,5	12	74	77	74
5,5	17,125	18,5	17,125	12,5	80,125	83,25	80,125
6	20	21,5	20	13	86,5	89,75	86,5

13,5	93,125	96,5	93,125
14	100	103,5	100
14,5	107,125	110,75	107,125

Figura 34: Métodos com $h = 0,5$.

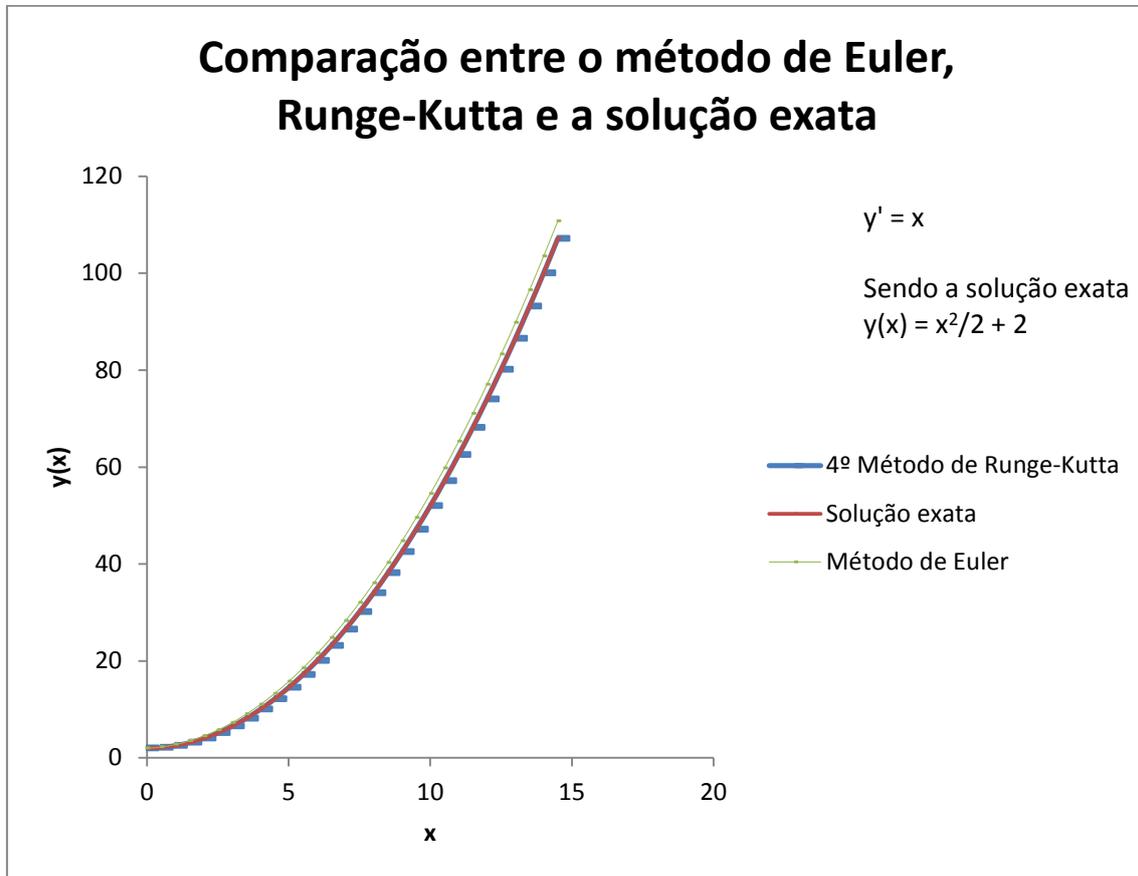


Figura 35: Nesta comparação, o método de Runge-kutta praticamente coincide com os valores da solução exata.

5.2.2.3 E.D.O. $y' = \cos(x)$

Para a realização dos dois métodos, foi utilizado como valor do passo $h = 0,25$.

x_i	y_i	y_{euler}	y_{exato}	2,75	2,381	2,139	2,381
0	2	2	2	3	2,141	1,891	2,141
0,25	2,247	2,242	2,247	3,25	1,891	1,643	1,891
0,5	2,479	2,461	2,479	3,5	1,649	1,408	1,649
0,75	2,681	2,644	2,681	3,75	1,428	1,203	1,428
1	2,841	2,779	2,841	4	1,243	1,040	1,243
1,25	2,948	2,858	2,948	4,25	1,105	0,928	1,105
1,5	2,997	2,876	2,997	4,5	1,022	0,876	1,022
1,75	2,983	2,831	2,983	4,75	1,000	0,885	1,000
2	2,909	2,727	2,909	5	1,041	0,956	1,041
2,25	2,778	2,570	2,778	5,25	1,141	1,084	1,141
2,5	2,598	2,370	2,598	5,5	1,294	1,261	1,29

5,75	1,491	1,477	1,491	9,5	1,924	1,675	1,924
6	1,720	1,717	1,720	9,75	1,680	1,438	1,680
6,25	1,966	1,966	1,966	10	1,455	1,228	1,455
6,5	2,215	2,211	2,215	10,25	1,265	1,059	1,265
6,75	2,450	2,434	2,450	10,5	1,120	0,940	1,120
7	2,656	2,622	2,656	10,75	1,030	0,879	1,030
7,25	2,823	2,764	2,823	11	1,000	0,880	1,000
7,5	2,938	2,851	2,938	11,25	1,032	0,943	1,032
7,75	2,994	2,877	2,994	11,5	1,124	1,064	1,124
8	2,989	2,841	2,989	11,75	1,271	1,235	1,271
8,25	2,922	2,744	2,922	12	1,463	1,446	1,463
8,5	2,798	2,594	2,798	12,25	1,688	1,684	1,688
8,75	2,624	2,398	2,624	12,5	1,933	1,933	1,933
9	2,412	2,171	2,412	12,75	2,182	2,179	2,182
9,25	2,173	1,924	2,173	13	2,420	2,406	2,420

Figura 36: Métodos com $h = 0,25$.

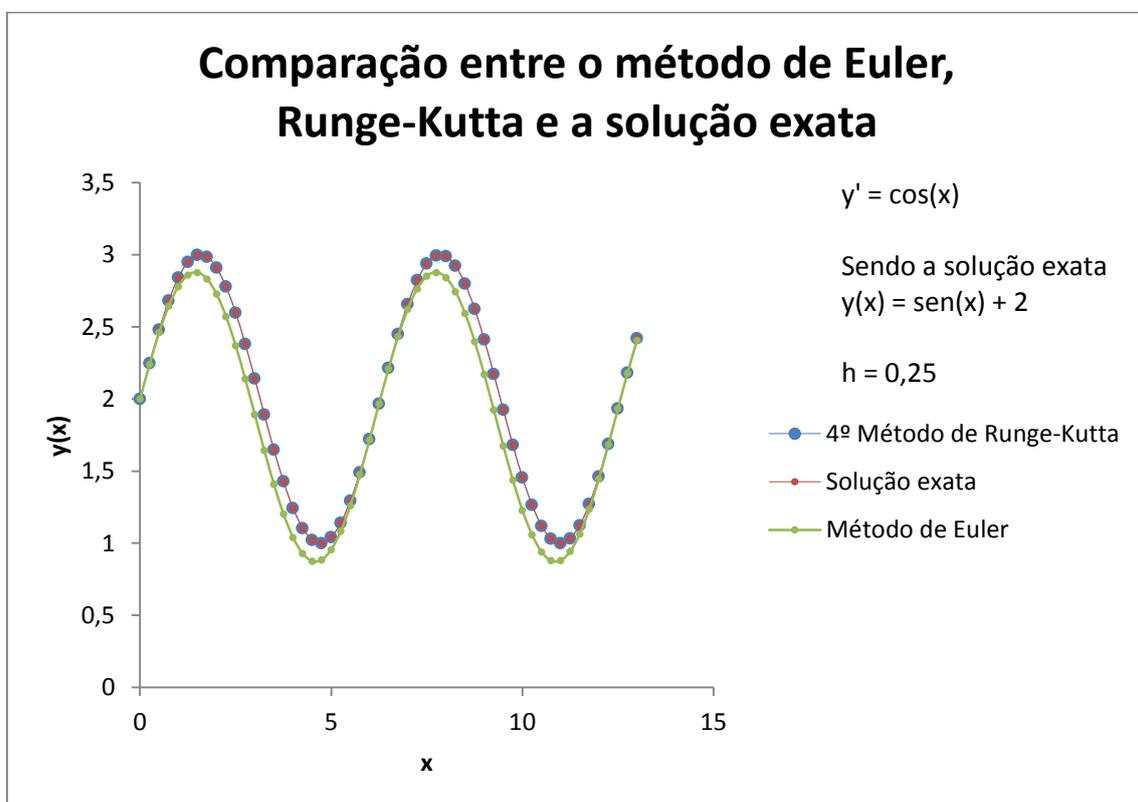


Figura 37: Novamente, o método de Runge-kutta coincide com a solução exata, e o método de Euler sofre um desvio.

5.3 Resultados dos programas em Java dos métodos de Euler e Runge-kutta

Neste projeto, foram criados algoritmos que fossem capazes de executar os métodos de Euler e Runge-kutta. Embora eles já calculem e imprimam os

resultados corretamente, o usuário precisa inserir a equação analisada nas linhas de código. Nos programas feitos posteriormente (para o estudo do foguete), foi adicionado a função de uma saída externa, para que os resultados sejam salvos em um arquivo de texto. As linhas de código destes métodos numéricos estão nos apêndices A1 e A2, e as imagens a seguir foram obtidos no IDE NetBeans.

5.3.1 Algoritmo do Método de Euler

O algoritmo foi executado para resolver a E.D.O. $y' = 1 - x + 4y$, considerando $y(0) = 1$, e o valor do passo $h = 0,1$.

```

41     double b = y0;
42
43     double integral = 1 - a + 4*b;
44     double k1 = 1 - a + 4*b;
45
46     a = x0 + (h/2);
47     b = y0 + (h/2)*k1;
48     double k2 = 1 - a + 4*b;
49
50     b = y0 + (h/2)*k2;
51     double k3 = 1 - a + 4*b;
52
53     a = x0 + h;
54     b = y0 + h*k3;
55     double k4 = 1 - a + 4*b;
56
57     double y = y0 + (h/6)*(k1 + 2*k2 + 2*k3 + k4);
58     //Esta é a fórmula de Runge-Kutta achando os valores yi+1.
59     x0 = x0 + h;
60     System.out.println("O valor de y quando x é igual a " + x0 + " é de " + y);
61     //Aqui será exibido o valor yi+1 para xi+1.
62     y0 = y0 + (h/6)*(k1 + 2*k2 + 2*k3 + k4);

```

Saída - JavaApplication3 (run) ☒ Tarefas

```

Bem-vindo ao programa do Método de Runge-Kutta!
Insira o valor h do passo:
0,1
Agora insira o valor inicial x0 para análise da função:
0
Agora insira o valor final de x na análise da função:
10
Agora insira o valor y0 no ponto x0:
1
O valor de y quando x é igual a 0.1 é de 1.6089333333333333.

```

Figura 38: Programa do Método de Euler, lendo os dados fornecidos pelo usuário.

```

27 System.out.println("Agora insira o valor inicial x0 para análise da função:");
28 //Este será o ponto de x onde a função começará a ser estudada.
29 double x0 = sc.nextDouble();
30
31 System.out.println("Agora insira o valor final de x na análise da função:");
32 //Este será o ponto final de x na análise. Em outras palavras, x0 será acresc
33 double x = sc.nextDouble();
34
35 System.out.println("Agora insira o valor y0 no ponto x0:");
36 //Este será o valor de y no ponto x0 da função.
37 double y0 = sc.nextDouble();
38
39
40 while( x0 < x) {
41     double y = y0 + h*(1-x0+4*y0);
42     y0 = y0 + h*(1-x0+4*y0);
43     x0 = x0 + h;
44     //Esta é a fórmula de Euler achando os valores yi+1.
45     System.out.println("O valor de y quando x é " + x0 + " é de " + y
46     //Aqui será exibido o valor yi+1 para xi+1.
47

```

Saída - JavaApplication3 (run)	Tarefas
O valor de y quando x é 0.1 é de 1.5.	
O valor de y quando x é 0.2 é de 2.19.	
O valor de y quando x é 0.30000000000000004 é de 3.146.	
O valor de y quando x é 0.4 é de 4.4744.	
O valor de y quando x é 0.5 é de 6.324160000000001.	
O valor de y quando x é 0.6 é de 8.903824000000002.	
O valor de y quando x é 0.7 é de 12.505353600000003.	
O valor de y quando x é 0.7999999999999999 é de 17.537495040000003.	
O valor de y quando x é 0.8999999999999999 é de 24.572493056000006.	
O valor de y quando x é 0.9999999999999999 é de 34.41149027840001.	

Figura 39: Programa do Método de Euler, imprimindo os valores y_i para cada x_i , dentro do intervalo inserido pelo usuário.

5.3.2 Algoritmo do Método de Runge-kutta

O algoritmo foi executado para resolver a mesma E.D.O acima: $y' = 1 - x + 4y$, considerando $y(0) = 1$, e o valor do passo $h = 0,1$.

```
41     double b = y0;
42
43     double integral = 1 - a + 4*b;
44     double k1 = 1 - a + 4*b;
45
46     a = x0 + (h/2);
47     b = y0 + (h/2)*k1;
48     double k2 = 1 - a + 4*b;
49
50     b = y0 + (h/2)*k2;
51     double k3 = 1 - a + 4*b;
52
53     a = x0 + h;
54     b = y0 + h*k3;
55     double k4 = 1 - a + 4*b;
56
57     double y = y0 + (h/6)*(k1 + 2*k2 + 2*k3 + k4);
58     //Esta é a fórmula de Runge-Kutta achando os valores yi+1.
59     x0 = x0 + h;
60     System.out.println("O valor de y quando x é igual a " + x0 + " é de " + y);
61     //Aqui será exibido o valor yi+1 para xi+1.
62     y0 = y0 + (h/6)*(k1 + 2*k2 + 2*k3 + k4);
```

Saída - JavaApplication3 (run) Tarefas

```
Bem-vindo ao programa do Método de Runge-Kutta!
Insira o valor h do passo:
0,1
Agora insira o valor inicial x0 para análise da função:
0
Agora insira o valor final de x na análise da função:
10
Agora insira o valor y0 no ponto x0:
1
O valor de y quando x é igual a 0.1 é de 1.6089333333333333.
```

Figura 40: Programa do Método de Runge-kutta, lendo os dados fornecidos pelo usuário.

```

41     double b = y0;
42
43     double integral = 1 - a + 4*b;
44     double k1 = 1 - a + 4*b;
45
46     a = x0 + (h/2);
47     b = y0 + (h/2)*k1;
48     double k2 = 1 - a + 4*b;
49
50     b = y0 + (h/2)*k2;
51     double k3 = 1 - a + 4*b;
52
53     a = x0 + h;
54     b = y0 + h*k3;
55     double k4 = 1 - a + 4*b;
56
57     double y = y0 + (h/6)*(k1 + 2*k2 + 2*k3 + k4);
58     //Esta é a fórmula de Runge-Kutta achando os valores yi+1.
59     x0 = x0 + h;
60     System.out.println("O valor de y quando x é igual a " + x0 + " é de " + y
61     //Aqui será exibido o valor yi+1 para xi+1.
62     y0 = y0 + (h/6)*(k1 + 2*k2 + 2*k3 + k4);

```

Saída - JavaApplication3 (run)	Tarefas
O valor de y quando x é igual a 0.1 é de 1.6089333333333333.	
O valor de y quando x é igual a 0.2 é de 2.5050061511111111.	
O valor de y quando x é igual a 0.30000000000000004 é de 3.8294145091508147.	
O valor de y quando x é igual a 0.4 é de 5.792785270450575.	
O valor de y quando x é igual a 0.5 é de 8.709317547440138.	
O valor de y quando x é igual a 0.6 é de 13.047712629434702.	
O valor de y quando x é igual a 0.7 é de 19.50714785308206.	
O valor de y quando x é igual a 0.7999999999999999 é de 29.130609357370943.	
O valor de y quando x é igual a 0.8999999999999999 é de 43.47395433203548.	
O valor de y quando x é igual a 0.9999999999999999 é de 64.8581068089084.	

Figura 41: Programa do Método de Runge-kutta, imprimindo os valores y_i para cada x_i , dentro do intervalo inserido pelo usuário.

5.4 Resultados do estudo do caso do foguete de massa variável

Nesta parte foram utilizadas as equações de um foguete de massa variável (vistas na seção 4.2) em programas construídos em Java, simulando situações fictícias com valores físicos arbitrários para estudar como a velocidade, a massa e a aceleração de um foguete variam em função do tempo, utilizando o método numérico de Euler. Os códigos dos programas do foguete podem ser encontrados na parte A3 do Apêndice.

5.4.1 Gráficos do foguete de massa variável

Como situação padrão, se atribuiu os valores:

Massa inicial do foguete (combustível e carcaça) de $m_{(0)} = 3000$.

Massa somente da carcaça do foguete de $m_c = 500$.

Taxa de perda de massa do foguete de $u = 50$.

Aceleração do foguete de $a = - 10$.

Velocidade inicial do foguete de $y(0) = 10$.

O tempo inicial de $x_0 = 0$.

E como valor do passo no método de Euler como $h = 0,5$.

Em alguns gráficos, estudou-se o caso em que $m(x)$ é linear, isto é, a taxa de combustível que é queimada em cada instante é um valor fixo, até atingir a massa da carcaça. Em outros gráficos, estudou-se o caso de uma escolha exponencial para a função de $m(x)$, em que a taxa de queima em cada instante diminui conforme a massa do foguete se aproxima do valor de da massa da carcaça m_c . Por fim, se comparou os dois casos.

Observação: Como os gráficos possuem muitos pontos (em alguns casos, chegando a 2500), é inviável postar tabelas com os valores de cada ponto, devido ao espaço que seria ocupado. Portanto, a partir de então serão inseridos apenas os gráficos neste relatório. Porém, utilizando os códigos dos programas anexados e inserindo os valores citados acima, podem-se obter os valores de cada ponto.

5.4.1.1 Massa do foguete em função do tempo

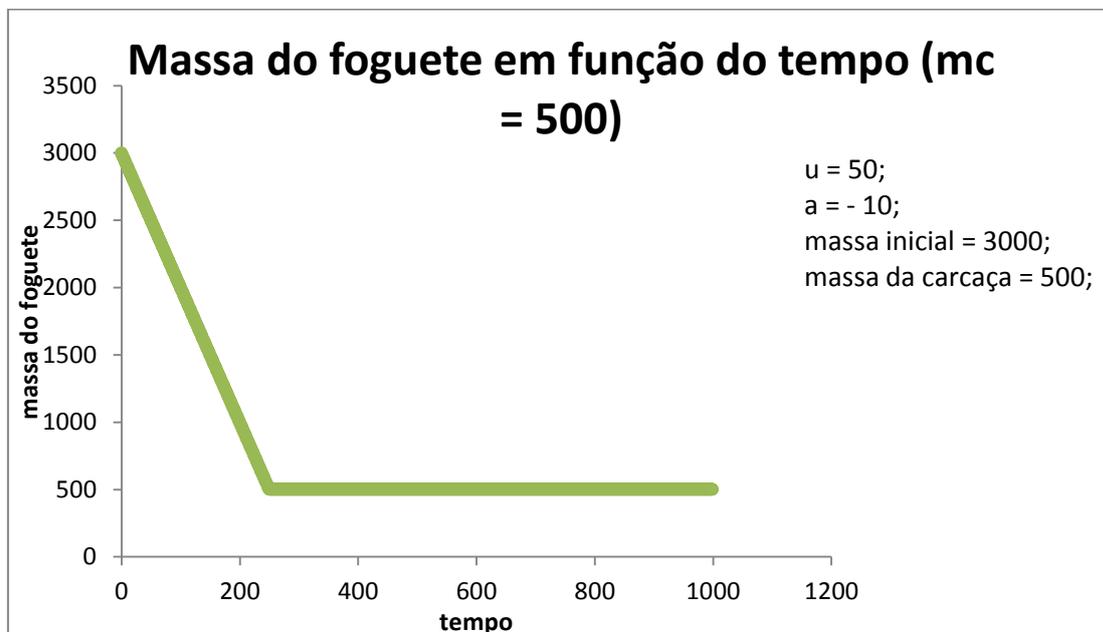


Figura 42: Neste gráfico, a massa do foguete $m(x)$ é diminuída linearmente até o combustível acabar. Após este momento, ela permanece constante em $m_c = 500$.

Os cálculos do gráfico seguinte foram realizados utilizando uma escolha exponencial para a função $m(x)$:

$$m(x) = (m_0 - m_c) * e^{-\frac{x_0}{\tau}} + m_c.$$

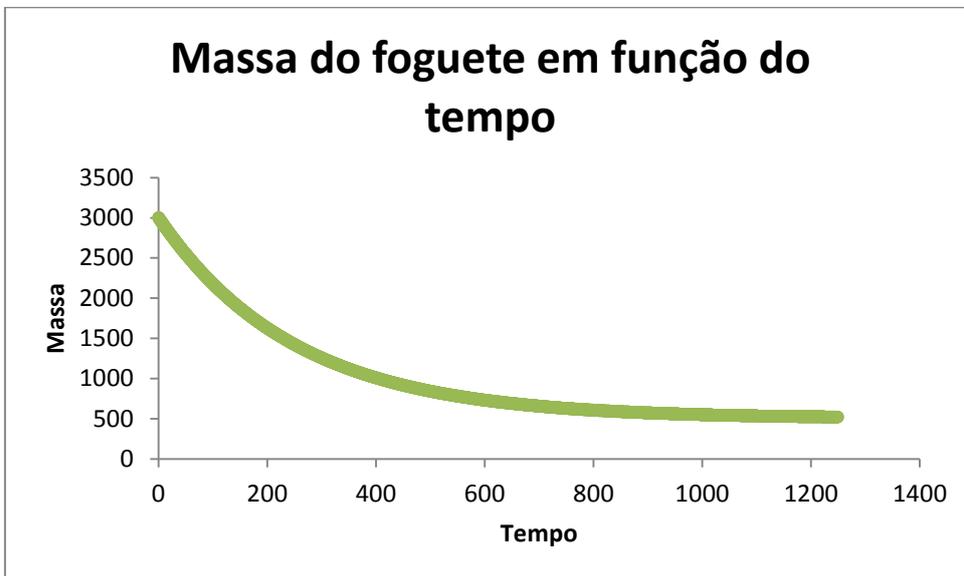


Figura 43: Neste cálculo foi utilizado $m(x)$ exponencial. Diferente do caso anterior, neste caso a massa do foguete vai diminuindo cada vez menos até ser limitada pela massa da carcaça.

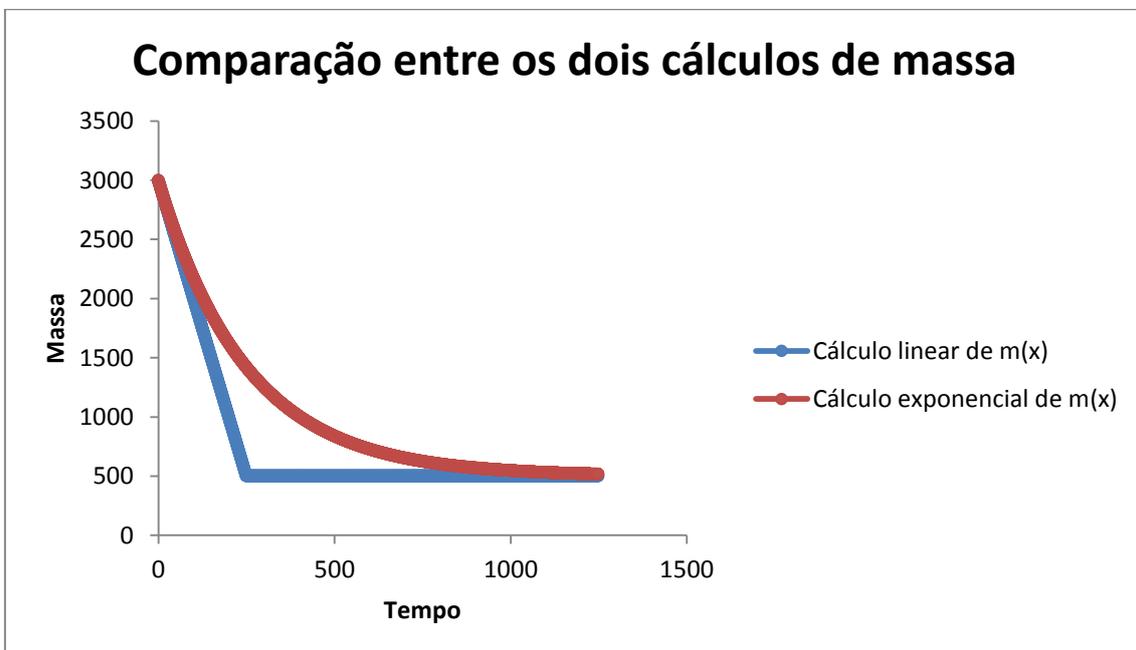


Figura 44: Comparação entre a diminuição da massa com $m(x)$ linear e $m(x)$ exponencial. No caso do cálculo linear, a massa do foguete diminui até atingir o valor da massa da carcaça, enquanto no caso exponencial a massa vai diminuindo cada vez menos com o passar do tempo, tendendo a massa da carcaça.

5.4.1.2 Velocidade do foguete em função do tempo

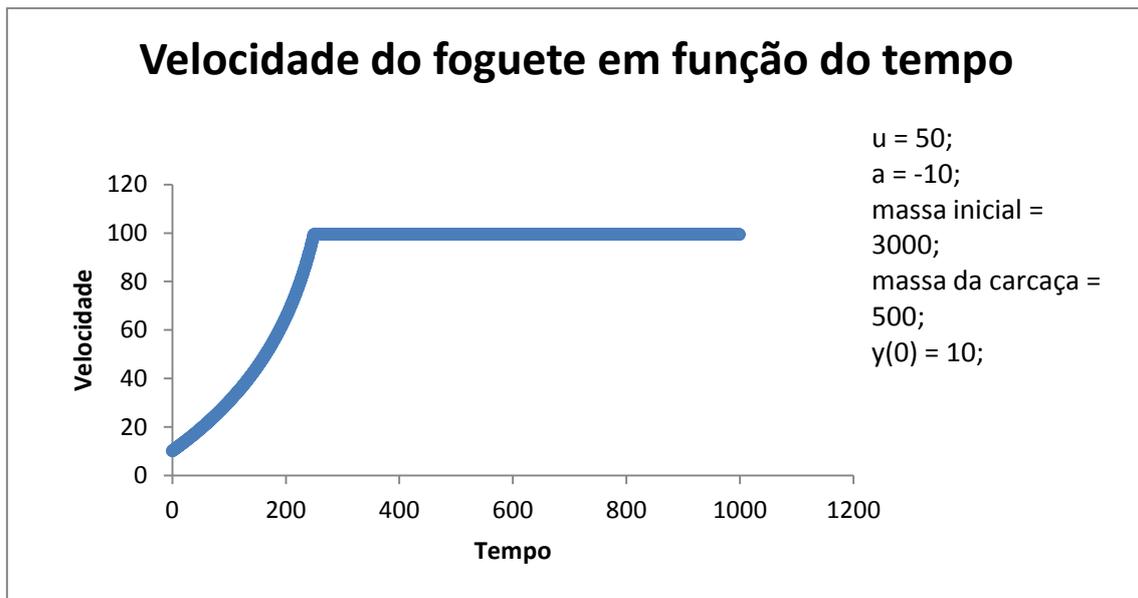


Figura 45: Gráfico da velocidade do foguete em função do tempo, utilizando $m(x)$ linear. A velocidade aumenta exponencialmente até um instante entre 200 e 300, pois quanto mais o combustível é queimado, mais aumenta a diferença entre $m(0)$ e $m(x)$, aumentando a velocidade em que o foguete se desloca (conforme visto na seção 4.2). Após este instante, todo o combustível foi consumido e a massa $m(x)$ atingiu o valor da carcaça, assim a velocidade para de aumentar, ficando contínua.

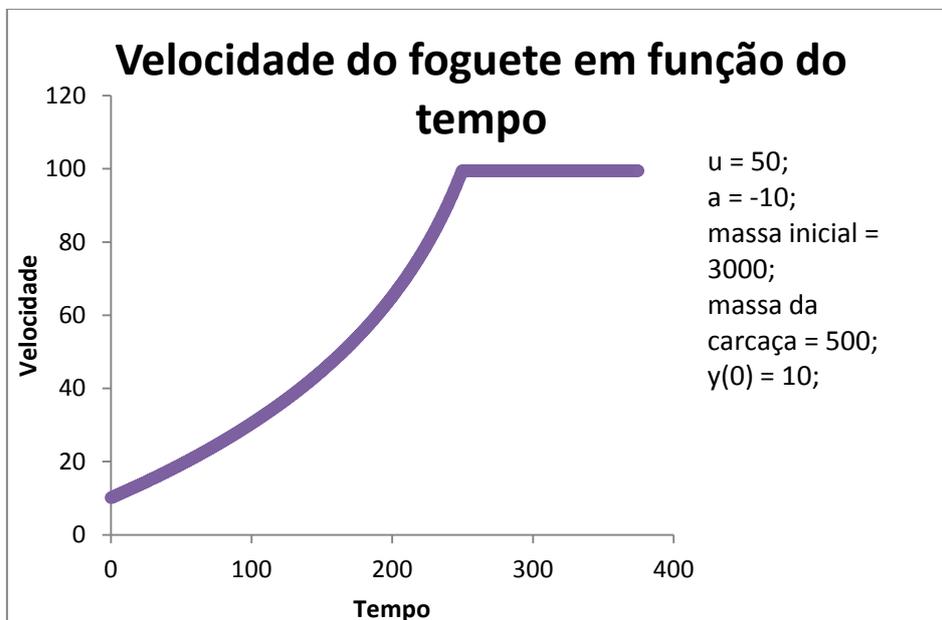


Figura 46: Mesmo caso da figura 42, porém o gráfico está em um intervalo de tempo menor, para melhor visualização da parte exponencial.

Para o gráfico seguinte, utilizou-se o método de Euler, no caso $y_{i+1} = y_i + h \cdot f(x_i)$, sendo $f(x) = -\frac{u \cdot dm/dx}{m \cdot x}$.

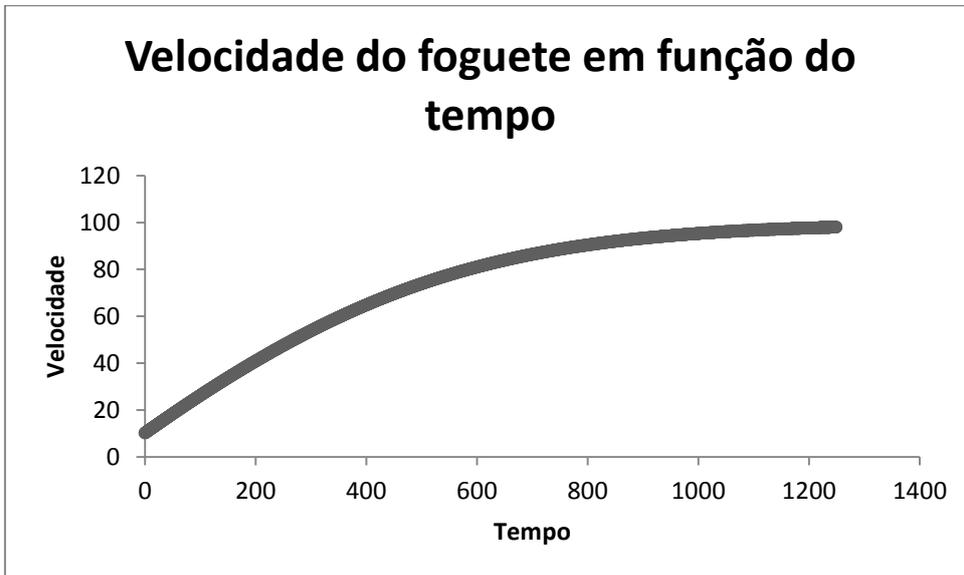


Figura 47: Neste gráfico, os cálculos foram realizados escolhendo-se $m(x)$ como exponencial. A velocidade aumenta cada vez menos conforme a massa do foguete se aproxima da massa da carcaça, tendendo a um valor constante.

5.4.1.3 Variação da aceleração do foguete em função do tempo



Figura 48: Este gráfico mostra o quanto a aceleração muda em função do tempo, utilizando $m(x)$ linear. Conforme o tempo avança, a aceleração se torna cada vez maior, até o instante $t = 250$, em que a massa do foguete atingiu a da carcaça, ele para de acelerar e a velocidade fica contínua.

Neste gráfico, utilizou-se o cálculo $f(x) = -\frac{u \cdot dm dx}{mx}$.

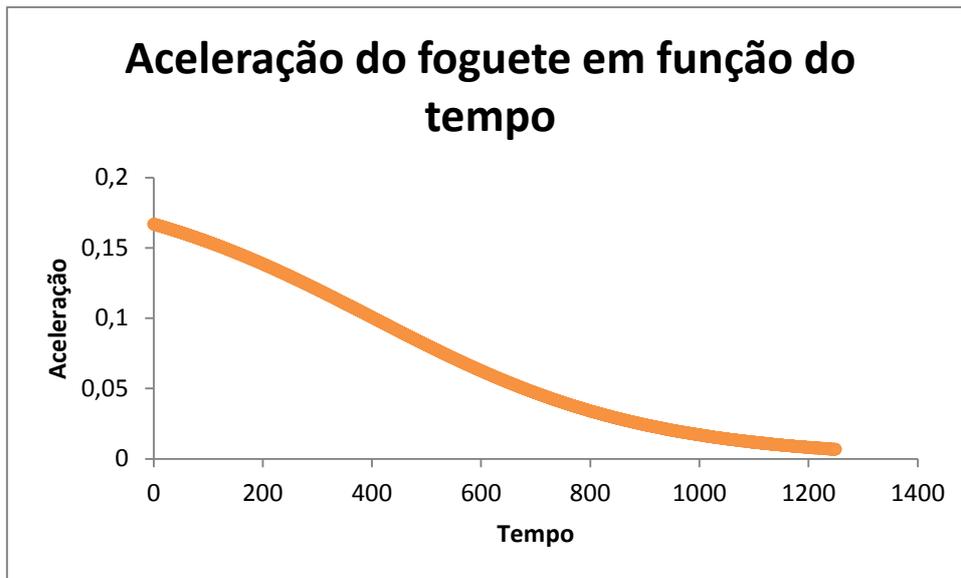


Figura 49: Neste caso, escolheu-se $m(x)$ como sendo exponencial. A aceleração diminui cada vez mais até atingir próximo do instante em que todo o combustível foi consumido. Após isto, ela vai diminuindo cada vez menos, tendendo a zero.

5.4.2 Algoritmos dos programas do foguete de massa variável

O primeiro programa para o estudo da situação de um foguete de massa variável foi construído de forma que o usuário insere os valores físicos no código do programa, e ao executá-lo irá ser pedido que insira o instante de tempo inicial da análise, a velocidade inicial do foguete e o valor do passo h que se deseja utilizar para aplicação do método de Euler. Já o segundo programa (do cálculo da massa) foi construído de forma que basta o usuário inserir no código os valores físicos iniciais desejados e o intervalo de tempo a ser estudado, que os cálculos serão automaticamente realizados, sem que o programa peça valores. Ambos salvam os dados obtidos em um arquivo de texto externo.

```

60     try{
61         // Create file
62         FileWriter fstream = new FileWriter("euler_teste41.txt");
63         BufferedWriter out = new BufferedWriter(fstream);
64         while( x0 < x){
65             //if(x0 < xc){
66                 //
67                 //mx = m0+a*x0;
68                 //dmdx = a;
69                 mx = (m0 - mc)*Math.exp(-x0/tau)+ mc;
70                 dmdx = (m0 - mc)*(-1/tau)*Math.exp(-x0/tau);
71                 fx0 = - (u*dmdx)/mx;
72             //}
73             //else{
74                 //fx0 = 0;
75                 //mx = mc;
76             //}
77
78             double y = y0 + h*fx0;
79             x0 = x0 + h;

```

Saída - euler_teste3 (run) #3 Tarefas

```

run:
Bem-vindo ao programa do estudo do Foguete de Massa Variável!
Insira o valor h do passo:

```

euler_teste3 (run) #3 running...

Figura 50: Programa principal do foguete de massa variável pedindo que o usuário insira os dados. Este programa fornece a relação de tempo, massa, velocidade e aceleração.

```

47     BufferedWriter out = new BufferedWriter(fstream);
48     while (x0 < x){
49         if(x0 < xc){
50             mx_linear = m0 + a*x0;
51             mx = (m0 - mc)*Math.exp(-x0/tau)+ mc;
52         }
53         else{
54             mx_linear = m0 + a*xc;
55             mx = (m0 - mc)*Math.exp(-x0/tau)+ mc;
56         }
57         //out.write("O valor de y no ponto " + x0 + " é " + y + ".");
58         out.write(String.format("%8.4f\t %8.4f \t %8.4f \n", x0, mx_linear, mx));
59         out.newLine();
60         x0 = x0 + h;
61         //Aqui é transferido para o arquivo externo os valores de yi+1 para xi+1 pelo método de Euler..
62     }
63     //Close the output stream
64     out.close();
65 }catch (Exception e){//Catch exception if any
66     System.err.println("Error: " + e.getMessage());

```

Saída - massa_foguete (run) Tarefas

```

run:
CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos)

```

Figura 51: Outro programa para o foguete de massa variável, sendo esse exclusivo para comparação dos dados da massa.

6 Cronograma

Este projeto possui a duração de 10 meses, de 01/10/2011 a 31/07/2012. No começo do projeto foi proposto o seguinte cronograma:

- 01/10/2011 a 31/11/2011 Estudo e revisão da literatura. Trabalho inicial com um programa gráfico, para apresentar os resultados do estudo balístico.

Aconteceu conforme o previsto, houve o estudo referente a história do estudo da gravidade e das leis da Mecânica; o estudo de como obter equações de movimento e a realização de exercícios de balística, representados em um programa gráfico.

- 01/12/2011 a 31/01/2012 Estudo das equações de movimento do foguete e dos métodos numéricos para a resolução das equações.

Nesta etapa, aconteceu o estudo dos métodos numéricos de Euler e Runge-kutta, para resolução de equações e expressão de tais resultados em um programa gráfico, além do estudo da diferença de precisão entre os métodos, e as diferenças de precisão com a alteração do valor do passo. Ocorreu um pequeno atraso com relação ao estudo das equações de movimento do foguete.

- 01/02/2012 a 31/03/2012 Testes numéricos e interpretação dos resultados. Elaboração do Relatório Parcial.

Nesta etapa, ocorreu o estudo das equações de movimento do foguete de massa variável pendente da etapa anterior, e iniciou-se o aprendizado da linguagem de programação Java, utilizando-a para criar algoritmos capazes de aplicar os métodos matemáticos de Euler e Runge-kutta e gerar resultados. Houve um pequeno atraso em relação a interpretação dos resultados através de tais algoritmos. Além disso, ocorreu a elaboração do Relatório Parcial.

- 01/04/2012 a 31/07/2012 Introduzir o conceito de velocidade de escape. Estudo da gravitação universal e órbitas. Elaboração do Relatório Final.

Nesta etapa, o projeto se focou mais no aprimoramento dos programas em Java capazes de resolver E.D.Os utilizando os métodos matemáticos, criando novos programas para o estudo do caso físico do foguete de massa variável e adicionando novas funções a estes programas (sendo a principal uma saída externa de dados). Além disso, ocorreu a elaboração do Relatório Final. Provavelmente o projeto avançou menos nessa etapa devido a greve das Universidades.

7 Conclusões

Com este projeto, foi possível estudar a história da gravidade; as leis da Mecânica relacionadas ao movimento terrestre, sendo possível expressar os resultados obtidos graficamente; estudar as equações de movimento de um foguete de massa variável; aprender e comparar métodos numéricos capazes de resolver E.D.Os; utilizar uma linguagem de programação para aplicar tais métodos e simular com os programas criados uma situação física de um foguete de massa variável.

Apêndice

A1 Código fonte (Java) do Método de Euler

Segue a baixo o programa criado, com os campos onde se insere a E.D.O. preenchidos para resolver a E.D.O. citada no exemplo (em Resultados):

```
package metodoeuler;

import java.util.Scanner;

public class metodoeuler {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Bem-vindo ao programa do Método de Euler!");

        System.out.println("Insira o valor h do passo:");

        //Quanto menor o valor inserido, mais preciso será o cálculo de Euler.

        double h = sc.nextDouble();

        System.out.println("Agora insira o valor inicial x0 para análise da função:");
```

```

//Este será o ponto de x onde a função começará a ser estudada.

double x0 = sc.nextDouble();

System.out.println("Agora insira o valor final de x na análise da função:");

//Este será o ponto final de x na análise. Em outras palavras, x0 será acrescido de h até chegar neste
ponto.

double x = sc.nextDouble();

System.out.println("Agora insira o valor y0 no ponto x0:");

//Este será o valor de y no ponto x0 da função.

double y0 = sc.nextDouble();

while( x0 < x) {

    double y = y0 + h*(1-x0+4*y0);

    y0 = y0 + h*(1-x0+4*y0);

    x0 = x0 + h;

    //Esta é a fórmula de Euler achando os valores yi+1.

    System.out.println("O valor de y quando x é " + x0 + " é de " + y + ".");

    //Aqui será exibido o valor yi+1 para xi+1.

}

}

}

```

A2 Linhas de código do Método de Runge-kutta

Segue a baixo o programa criado, com os campos onde se insere a E.D.O. preenchidos para resolver a E.D.O. citada no exemplo (em Resultados):

```

package metodorungekutta;

public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);

    System.out.println("Bem-vindo ao programa do Método de Runge-Kutta!");

    System.out.println("Insira o valor h do passo:");

    double h = sc.nextDouble();

    //Quanto menor o valor inserido, mais preciso será o cálculo de Runge-Kutta.

    System.out.println("Agora insira o valor inicial x0 para análise da função:");

```

```

//Este será o ponto de x onde a função começará a ser estudada.

double x0 = sc.nextDouble();

System.out.println("Agora insira o valor final de x na análise da função:");

//Este será o ponto final de x na análise. Em outras palavras, x0 será acrescido de h até chegar neste ponto.

double x = sc.nextDouble();

System.out.println("Agora insira o valor y0 no ponto x0:");

//Este será o valor de y no ponto x0 da função.

double y0 = sc.nextDouble();

while( x0 < x) {

    double a = x0;

    double b = y0;

    double integral = 1 - a + 4*b;

    double k1 = 1 - a + 4*b;

    a = x0 + (h/2);

    b = y0 + (h/2)*k1;

    double k2 = 1 - a + 4*b;

    b = y0 + (h/2)*k2;

    double k3 = 1 - a + 4*b;

    a = x0 + h;

    b = y0 + h*k3;

    double k4 = 1 - a + 4*b;

    double y = y0 + (h/6)*(k1 + 2*k2 + 2*k3 + k4);

    //Esta é a fórmula de Runge-Kutta achando os valores yi+1.

    x0 = x0 + h;

    System.out.println("O valor de y quando x é igual a " + x0 + " é de " + y + ".");

    //Aqui será exibido o valor yi+1 para xi+1.

    y0 = y0 + (h/6)*(k1 + 2*k2 + 2*k3 + k4);

}

}

}

```

A3 – Linhas de códigos do estudo do foguete de massa variável

A3.1 – Código que entrega a relação de tempo, massa, velocidade e aceleração

```
package euler_teste3;

import java.io.BufferedWriter;

import java.io.FileWriter;

import java.util.Scanner;

public class Euler_teste3 {

    public static void main(String[] args) {

        // TODO code application logic here

        double fx0;

        double dmdx, mx;

        Scanner sc = new Scanner(System.in);

        System.out.println("Bem-vindo ao programa do estudo do Foguete de Massa Variável!");

        System.out.println("Insira o valor h do passo:");

        //Quanto menor o valor inserido, mais preciso será o cálculo de Euler.

        double h = sc.nextDouble();

        System.out.println("Agora insira o valor inicial x0 para análise da função:");

        //Este será o ponto de x onde a função começará a ser estudada.

        double x0 = sc.nextDouble();

        //System.out.println("Agora insira o valor final de x na análise da função:");

        //Este será o ponto final de x na análise. Em outras palavras, x0 será acrescido de h até chegar neste ponto.

        //double x = sc.nextDouble();

        System.out.println("Agora insira o valor y0 no ponto x0:");

        //Este será o valor de y no ponto x0 da função.
```

```

double y0 = sc.nextDouble();

//este é o método que será chamado para definir os valores de f(x0) que serão aplicados no método de Euler.

double mc = 500;

//massa da carcaça do foguete.

double u = 50;

//perda de carga.

double m0 = 3000;

//massa inicial do foguete (combustível e carcaça).

double a = - 10;

//aceleração do foguete

double xc = (mc - m0)/a;

//definindo o momento em que todo o combustível foi consumido.

double tau = xc;

double x = 5*tau;

{
try{
// Create file

FileWriter fstream = new FileWriter("euler_teste41.txt");

BufferedWriter out = new BufferedWriter(fstream);

while( x0 < x){

//if(x0 < xc){

//

//mx = m0+a*x0;

//dmdx = a;

mx = (m0 - mc)*Math.exp(-x0/tau)+ mc;

dmdx = (m0 - mc)*(-1/tau)*Math.exp(-x0/tau);

fx0 = - (u*dmdx)/mx;

```

```

    //}

    //else{

        //fx0 = 0;

        //mx = mc;

    //}

    double y = y0 + h*fx0;

    x0 = x0 + h;

    //out.write("O valor de y no ponto " + x0 + " é " + y + ".");

    out.write(String.format("%8.4f \t %8.4f \t %8.4f \t %8.4f \n", x0, mx, y, fx0));

    out.newLine();

    //Aqui é transferido para o arquivo externo os valores de yi+1 para xi+1 pelo método de Euler..

    y0 = y;

    }

//Close the output stream

out.close();

}catch (Exception e){//Catch exception if any

System.err.println("Error: " + e.getMessage());

}

}

}

}

```

A3.2 Código que compara a massa obtida pelo cálculo linear e pelo exponencial

```

package massa_foguete;

import java.io.BufferedWriter;

import java.io.FileWriter;

public class Massa_foguete {

    public static void main(String[] args) {

```

```

// TODO code application logic here

double h = 0.5;

//passo

double x0 = 0;

//tempo inicial

double mx_linear, mx;

//massa do foguete em função do tempo.

double mc = 500;

//massa da carcaça do foguete.

double u = 50;

//perda de carga.

double m0 = 3000;

//massa inicial do foguete (combustível e carcaça).

double a = - 10;

//aceleração do foguete

double xc = (mc - m0)/a;

//definindo o momento em que todo o combustível foi consumido.

double tau = xc;

double x = 5*tau;

{
try{

// Create file

FileWriter fstream = new FileWriter("massa_foguete5.txt");

BufferedWriter out = new BufferedWriter(fstream);

while (x0 < x){

if(x0 < xc){

mx_linear = m0 + a*x0;

```

```

        mx = (m0 - mc)*Math.exp(-x0/tau)+ mc;
    }

    else{

        mx_linear = m0 + a*xc;

        mx = (m0 - mc)*Math.exp(-x0/tau)+ mc;

    }

    //out.write("O valor de y no ponto " + x0 + " é " + y + ".");

    out.write(String.format("%8.4ft %8.4f \t %8.4f \n", x0, mx_linear, mx));

    out.newLine();

    x0 = x0 + h;

    //Aqui é transferido para o arquivo externo os valores de yi+1 para xi+1 pelo método de Euler..

    }

//Close the output stream

out.close();

}catch (Exception e){//Catch exception if any

System.err.println("Error: " + e.getMessage());

}

}

}

}

```

Referências

- [1] B. Schutz, Gravity from the Ground Up, Cambridge Univeristy Press, Cambridge (2003), capítulos 1 e 2.
- [2] H. Moyses Nussenzveig, Curso de Física Básica vol. 1 Mecânica, Edgard Blücher, São Paulo (2002).
- [3] Kathy Sierra e Bert Bates, Use a Cabeça! Java, 2ª edição, (2007).
- [4] C.Y. Shigue, Apostila de Cálculo Numérico e Computacional.

[5] http://www.instructioneducation.info/Mechsub/num2_1.pdf

[6] Kleber Daum Machado, Equações Diferenciais Aplicadas à Física, Editora UEPG, 2ª edição, (2000).