ANÁLISE E SIMULAÇÃO DE REENTRADAS ATMOSFÉRICAS CONTROLADAS

RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA
(PIBIC/CNPq/INPE)

Bolsista – Grazielle Cunha Cardoso (ETEP Faculdades, Bolsista PIBIC/CNPq)

E-mail: graziellecunha@yahoo.com.br

Orientador - Dr. Marcelo Lopes de Oliveira e Souza (DMC/ETE/INPE)
E-mail: marcelo@dem.inpe.br

Julho de 2008

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

CAPÍTULO 1 – INTRODUÇÃO E MOTIVAÇÃO

1.1 INTRODUÇÃO

BREVE HISTÓRICO: No período de Agosto de 2007 até Julho de 2008, foram realizados: 1) Nossa introdução ao tema: "Reentradas Atmosféricas Controladas", tratando do retorno controlado de um veículo espacial para dentro da atmosfera da Terra; 2) Um estudo do MATLAB; 3) Um estudo sobre Métodos Numéricos; 3.1) Método de Euler; 3.2) Métodos de Runge-Kutta; 4) O estudo de Introdução à Mecânica Orbital; 5) Reprodução e correção das simulações numéricas feitas pela bolsista anterior; 6) Um estudo sobre Decaimento orbital com Força de Arrasto Atmosférico.

OBJETIVOS E ESTRUTURA DESTE RELATÓRIO: O objetivo deste Relatório Parcial é descrever os trabalhos realizados no período de Agosto de 2007 até o início de Fevereiro de 2008, incluindo: 1) uma introdução ao tema e continuidade aos resultados da pesquisa anterior; 2) uma introdução ao Matlab para elaborar programas que simulem o decaimento orbital controlado usando a transferência inversa de Hohmann e a transferência inversa de Breakwell; 3) uma introdução ao cálculo numérico utilizando os métodos de Euler e Runge-Kutta. 4) uma Introdução à Mecânica Orbital; 5) Reprodução e correção das simulações numéricas feitas pela bolsista anterior; 6) Um estudo sobre Decaimento orbital com Força de Arrasto Atmosférico.

1.2- REENTRADAS ATMOSFÉRICAS CONTROLADAS

A reentrada atmosférica trata do retorno de um veículo espacial, por exemplo, um satélite para dentro da atmosfera da Terra. Se o satélite estiver funcionando e possuir sistema propulsivo, é possível controlar, em parte, o seu decaimento. A ONU solicita aos países e empresas lançadoras de foguetes que colocam satélites em órbita, que planejem meios de desorbitá-los para a reentrada na atmosfera.

Para se ter uma idéia, por ano, caem na Terra entre 150 a 200 toneladas de detritos espaciais artificiais – partes de foguetes, plataformas e satélites, que reentram na atmosfera. A grande maioria, porém, se reduz a cinzas ao cruzar a atmosfera. E o pouco que chega a atingir o solo não costuma causar danos – muito menos ferir pessoas. Mas a minoria já polui as órbitas de interesse.

Por isso torna-se importante o estudo de manobras de atitude e de transferência orbital visando otimizar o decaimento orbital controlado de um satélite; e também o estudo da sua reentrada inteira ou de seus fragmentos, visando impactar uma região segura da superfície da Terra.

CAPÍTULO 2 – OBJETIVO

O objetivo desse trabalho é calcular as manobras de atitude e a distribuição de reentrada atmosférica controlada e estudar a sua evolução no tempo, iniciando com a posição do Centro de Massa-CM. Para simular a geração e a propagação de um satélite em final de órbita e início de reentrada na atmosfera, dividimos a análise e as simulações por dois modelos analíticos em duas fases:

1) Fazer o satélite decair a partir de sua órbita nominal até cerca de 120 km-80 km, realizando transferências orbitais monoimpulsivas no apogeu (transferência inversa de Breakwell), ou biimpulsivas no perigeu e no apogeu (transferências de Hohmann) com consumo mínimo de combustível, possivelmente aproveitando a força de arrasto atmosférico, até atingir uma área segura do Oceano Pacífico.

2) Determinar o melhor posicionamento na órbita a 120 km-80 km de altitude e simular a reentrada do satélite na atmosfera terrestre sem ou com fragmentação, visando atingir uma região segura da superfície da Terra, usualmente no Oceano Pacífico. Depois tentaremos fazer o mesmo por um modelo numérico, para compará-los e ajustar o 1º. ao 2º. Com base nesses dados, objetiva-se, posteriormente, observar e interpretar as propriedades básicas desse processo.

CAPÍTULO 3 – METODOLOGIA

3.1- INTRODUÇÃO

Em 05 de Agosto de 2007, foi iniciado esse projeto de pesquisa seguindo a orientação do Dr. Marcelo Lopes de Oliveira e Souza. O trabalho atual iniciou-se com um estudo: 1) do trabalho feito pela bolsista anterior, 2) do Matlab, 3) de Métodos Numéricos, 4) da apostila de "Introdução a Mecânica Orbital", 5) da reprodução das Simulações Numéricas feitas pela bolsista anterior e 6) de Transferência com Arrasto Atmosférico, bem como a simulação usando programas elaborados no Matlab.

3.2-ESTUDO DO MATLAB

Foi feito um estudo do Matlab com base na apostila "Curso de Matlab 5.1 Introdução à Solução de Problemas de Engenharia" (2003). Estudaram-se matrizes, operações com matrizes, comandos para gerar matrizes com números aleatórios. Trabalhou-se também com números complexos, bem como suas diferentes grafias como também operações simples com esse conjunto de números. Fez-se também a plotagem de alguns gráficos com os dados de algumas matrizes.

3.3- MÉTODOS NUMÉRICOS

3.3.1 – MÉTODO DE EULER

É um método elementar de resolução de equações diferenciais de primeira ordem; por isso, possui uma precisão limitada.

Dada a equação diferencial:

$$\frac{dy}{dx} = f(y,x) \ \forall x \in [x_0, x_f]$$

$y(t_0) = y_0$

A fórmula de aproximação de Euler é:

$$y_1 = y(x_0) + hf(x_0, y(x_0))$$

$$y_{i+1} = y_i + hf(x_i, y_i) = y_i + hf_i \qquad i \geq 1 \qquad (1)$$

Tomemos como exemplo a equação diferencial:

$$\frac{dy}{dx} = f(x,y) = x + y \qquad (2)$$

Para a condição inicial x_0=0, y(x_0=0 temos :0)

$$y = e^x - x - 1 \qquad (3)$$

A solução pelo método de Euler, usando passo h=0.1 e limite de integração x10=1.0, utilizando as seguintes equações:

$$\begin{aligned} y_1 &= y(x_0) + hf(x_0, y(x_0)) \\ y_{i+1} &= y_i + hf(x_i, y_i) = y_i + hf_i \qquad i \geq 1 \end{aligned} \qquad (4)$$

É mostrada na Tabela 1.De acordo com ela :

- A coluna 1 mostra o valor de i .
- A coluna 2 mostra o valor de x_i .
- A coluna 3 mostra o valor de y_i calculado pelas equações (3).
- A coluna 4 mostra o cálculo com equação a (1) da derivada $f(x_i,y_i)$
- A coluna 5 mostra a verdadeira solução analítica de $y(x_i)$, arredondada para quatro algarismos, usando a equação y =e^x - x - 1.
- A coluna 6 mostra o erro global calculado por $E_i = y_i - y(x_i)$.
- A coluna 7 mostra o truncamento de y_i para quatro algarismos.
- A coluna 8 mostra o arredondamento de y_i para quatro algarismos.
- A coluna 9 mostra o valor calculado pela equação (3) usando $y(x_{i-1})$, com o valor verdadeiro $y(x_i)$.
- A coluna 10 mostra o erro de truncamento calculado por Ei = $y(x_{i-1}) - y(x_i)$, com o valor verdadeiro $y(x_i)$.

A coluna 11 mostra o valor do erro de truncamento local calculado por:

$$|e_t|_{\max} = \frac{h^2}{2!} e^{x_i+1} \qquad \text{Sendo } x = x_{i+1}$$

Tabela 1.

		y(0)=0				*H*=0.1				
i (1)	X_i (2)	Y_i (3)	$f(x_i,y_i)$ (4)	$Y(x_i)$ (5)	$E_i = y_i - y(x_i)$ (6)	Y_i (7)	Y_i (8)	$y(x_{i-1})$ (9)	$y(x_{i-1})$ (10)	E^x_{max} (11)
0	0.0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-	-
1	0.1	0.0000	0.1000	0.0052	-0.0052	0.0000	0.0000	0.0000	-0.0052	0.0055
2	0.2	0.0100	0.2100	0.0214	-0.0114	0.0100	0.0100	0.0157	-0.0057	0.0061
3	0.3	0.0310	0.3310	0.0499	-0.0189	0.0310	0.0310	0.0435	-0.0064	0.0067
4	0.4	0.0641	0.4641	0.0918	-0.0277	0.0641	0.0641	0.0849	-0.0069	0.0075
5	0.5	0.11051	0.61051	0.1487	-0.0382	0.1105	0.1105	0.1410	-0.0077	0.0082
6	0.6	0.171561	0.771561	0.2221	-0.0505	0.1715	0.1715	0.2136	-0.0085	0.0091
7	0.7	0.2487171	0.9487171	0.3138	-0.0651	0.2486	0.2488	0.3043	-0.0095	0.0101
8	0.8	0.34358881	1.14358881	0.4255	-0.0819	0.3434	0.3437	0.4152	-0.0103	0.0111
9	0.9	0.457947691	1.357947691	0.5596	-0.1017	0.4577	0.4581	0.5480	-0.0116	0.0123
10	1.0	0.5937424601	1.5937424601	0.7183	-0.1246	0.5934	0.5939	0.7056	-0.0127	0.0136

3.3.2 – MÉTODOS DE RUNGE – KUTTA

São métodos para resolver equações diferenciais de segunda, terceira, quarta, etc. ordens, usando somente cálculos das derivadas de primeira ordem. A aproximação de segunda, terceira e quarta ordem (aproximação com precisão equivalente a Expansão de Taylor em y(x), mantendo o termo em h_2, h_3 e h_4, respectivamente) requer a estimação em dois, três e quatro valores, respectivamente, de x no intervalo $x_i \leq x \leq x_{i+1}$.

Todo Método de Runge-Kutta tem algoritmo da forma:

$$y_{i+1} = y_i + h\phi(x_i, y_i, h) \qquad (5)$$

sendo ∅ o incremento da função para aproximação de f(x,y) sobre intervalo $x_i \leq x \leq x_{i+1}$.

MÉTODO DE RUNGE – KUTTA DE 3ª ORDEM

Sendo k_1, k_2, k_3 aproximações da derivada em diversos pontos sobre o intervalo de integração $[x_i, x_{i+1}]$.

$$\begin{aligned} y_{i+1} &= y_i + \frac{h}{6}(k_1 + 4k_2 + k_3) \\ k_1 &= f(x_1, y_i) \\ k_2 &= f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_1\right) \\ k_3 &= f(x_i + h, y_i + 2hk_2 - hk_1) \end{aligned} \qquad (6)$$

RUNGE – KUTTA 4ª ORDEM (Variante A)

Sendo k_1, k_2, k_3 e k_4 aproximações da derivada em diversos pontos no intervalo de integração $[x_i, x_{i+1}]$.

$$\begin{aligned} y_{i+1} &= y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 &= f(x_1, y_i) \\ k_2 &= f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_1\right) \\ k_3 &= f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_2\right) \\ k_4 &= f(x_i + h, y_i + hk_3) \end{aligned} \qquad (7)$$

RUNGE – KUTTA 4ª ORDEM (Variante B)

Sendo k_1, k_2, k_3 e k_4 aproximação da derivada em diversos pontos no intervalo de integração [x_i, x_{i+1}].

$$
\begin{aligned}
y_{i+1} &= y_i + \frac{h}{8}(k_1 + 3k_2 + 3k_3 + k_4) \\
k_1 &= f(x_1, y_i) \qquad (8) \\
k_2 &= f\left(x_i + \frac{1}{3}h, y_i + \frac{1}{3}hk_1\right) \\
k_3 &= f\left(x_i + \frac{2}{3}h, y_i - \frac{1}{3}hk_1 + hk_2\right) \\
k_4 &= f(x_i + h, y_i + hk_1 - hk_2 + hk_3)
\end{aligned}
$$

O erro dos métodos de Runge-Kutta de quarta ordem é dado por:

$$e_t = Kh_1^5 = \frac{16}{15}(y_{n+1,2} - y_{n+1,1})$$

3.4 – INTRODUÇÃO À MECÂNICA ORBITAL

Durante o período de Agosto de 2007 e início de fevereiro de 2008, desenvolveu - se um estudo de Introdução a Mecânica Orbital com base na Apostila de Kuga e Rao (1995). Foi feita a leitura dos oito primeiros capítulos, como também a compreensão e resolução de alguns exercícios.

A apostila trata de uma introdução às Leis de Newton no espaço, e das leis de Kepler que foram formuladas com a necessidade de estudar o movimento orbital de Marte, já que o método das Órbitas Circulares não se ajustava ao planeta. Então Kepler, baseado nos estudos de Tycho Brahe formulou suas três leis, sendo uma delas a das Órbitas Elípticas.

Depois fez se um estudo sobre o Problema com Dois Corpos, que é determinar a trajetória de um ponto material (satélite) de massa m_1 sujeito à ação de uma força dirigida ao centro de outro corpo com massa m_2, como também a análise do movimento do satélite no espaço em relação a Terra.

Estudou-se também os Sistemas de Coordenadas Celestes: existem quatro sistemas principais para especificar as posições de corpos celestes na esfera celeste sendo eles: Sistema Horizontal (Topocêntrico), Sistema Horário (Topocêntrico ou Geocêntrico), Sistema Equatorial (Geocêntrico), e o Sistema Eclíptico. E para finalizar a parte de Mecânica Orbital, estudou-se Transformações de Coordenadas, que consiste em observar um corpo celeste e sua posição em um sistema e transformar essas medidas para outro sistema (p. ex.: inercial) através de matrizes de transformação.

CAPÍTULO 4 – SIMULAÇÕES NUMÉRICAS

No Relatório Final de julho de 2006 foi realizada uma pesquisa sobre o Observatório Compton de Raios Gama que foi lançado em abril de 1991 e teve sua reentrada na atmosfera da Terra de forma controlada em 4 de julho de 2000, após encerrar suas operações devido a falhas de um giroscópio.

4.1 – SEQUÊNCIA SIMPLIFICADA DA REENTRADA DO CGRO

Em 28 de maio de 2000

- As cargas desnecessárias da configuração / verificação geral da nave espacial (instrumentos) foram desligadas;
- A engenharia testa a execução de comandos;
- Vida de uma órbita: diversos anos.

Em 30 de maio de 2000 – Impulso número 1

- O impulso 1 diminui o perigeu de 510 km para 350 km;
- Vida de uma órbita: ~ 1 ano.

Em 31 de maio de 2000 – Impulso de número 2

- O impulso 2 diminui o perigeu de 350 km para 250 km;
- Vida de uma órbita: ~ 80 dias.

Em 4 de junho de 2000 – Impulso de número 3

- O impulso 3 diminui o perigeu de 250 km para 150 km;
- Vida de uma órbita: ~ 3 dias.

Em 4 de junho de 2000 – Impulso de número 4

- A órbita seguinte, executa o impulso 4 para reentrar na atmosfera;
- A nave espacial reentra no alvo.

De acordo com os dados obtidos da sequência simplificada da reentrada do Observatório Compton de Raios Gama, e Utilizando as equações abaixo:

$$a_2 = \frac{a_1(1+e_1)}{(1+e_2)} \qquad\qquad V^2 = \mu\left(\frac{2}{r} - \frac{1}{a}\right)$$

Obtivemos os resultados mostrados na Tabela 2:

Tabela 2

Sequência Simplificada da Reentrada do CGRO			
Data	**Altura do Perigeu (R_t = 6378 km)**	**Excentricidade (e) e semi – eixo maior (a)**	**ΔV (km/s)**
30 de maio de 2000	510 km – 350 km	e = 0,186 a = 430 km	-2,7343
31 de maio de 2000	350 km – 250 km	e = 0,1667 a = 300 km	-2,9408
04 de junho de 2000	250 km – 150 km	e = 0,25 a = 200 km	-5,3496
04 de junho de 2000	150 km – 84 km	e = 0,282 a = 117 km	-7,8706

Ao reproduzir os cálculos feitos pela bolsista anterior, houve uma contradição com os dados que ela havia obtido e os que eu obtive, sendo esses dados a variação da velocidade.

CAPÍTULO 5 – TRANSFERÊNCIA ORBITAL COM ARRASTO ATMOSFÉRICO

Para demonstrar as transferências com arrasto atmosférico integraremos numericamente as equações obtidas a partir das Leis de Newton envolvendo as forças gravitacional e de arrasto atmosférico.

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = -\frac{\mu}{|\vec{r}|^3}\begin{bmatrix} x \\ y \end{bmatrix} - \frac{Fa}{|\vec{v}|}\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}$$

Onde μ = 398600 km é a constante gravitacional da Terra.

$|\vec{r}| = (x^2 + y^2)^{1/2}$ - é o vetor posição;

$|\vec{v}| = (\dot{x}^2 + \dot{y}^2)^{1/2}$ - é o vetor velocidade.

$Fa = \frac{1}{2}C\rho A v^2$ - equação da força de arrasto atmosférico, onde Cρ é o coeficiente de arrasto; A a área do satélite; ρ é a densidade atmosférica; v é o vetor velocidade.

Foram reproduzidos todos os programas feitos pela bolsista anterior, e foram feitos os mesmos programas, mas variando a área, e a forma geométrica do corpo, para compararmos e provarmos que a força de arrasto depende da área do satélite.

Supondo que a área exposta à Força de arrasto seja uma circunferência de raio AB=2, com na Figura 1:

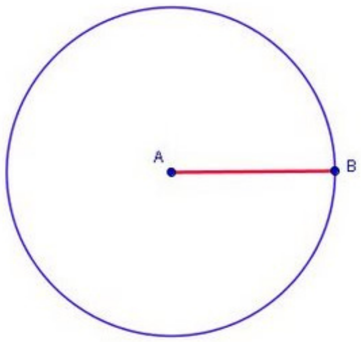

Figura 1 - Circunferência

Obtivemos os seguintes resultados:

O programa " transf_arrasto_modif2 " (vide Apêndice A, Programa 1), simula as transferências orbitais com força de arrasto atmosférico. Para realizar o programa foi utilizado as equações anteriores, os comandos de ode com o método de Runge-Kutta 4 ou 5 no Matlab, densidade (constante) e a densidade variando em função da altitude. Desenvolvendo assim o decaimento da orbital, como mostra as Figuras 2 abaixo:

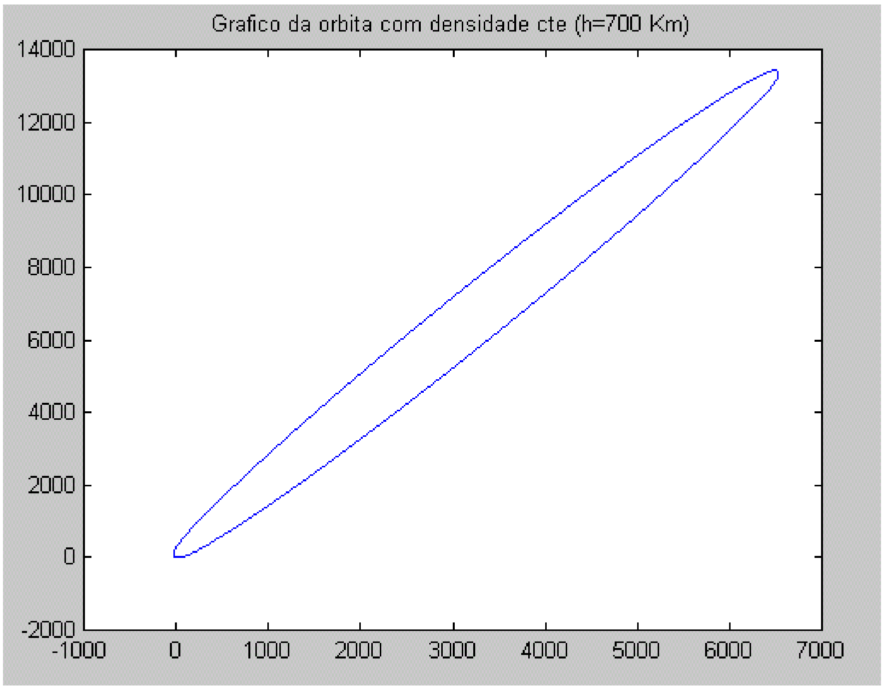

Figura 2- Decaimento Orbital com Força de Arrasto Atmosférico (densidade cte)

A densidade atmosférica em função da altitude como mostra a Tabela 3 :

Tabela 3

Altitude (Km)	Densidade (km/m^3)	Altitude (Km)	Densidade (km/m^3)	Altitude (Km)	Densidade (km/m^3)
0	1.225*10^0	80	1.905*10^-5	200	2.789*10^-10
25	3.899*10^-2	85	8.337*10^-6	250	7.248*10^-11
30	1.774*10^-2	90	3.396*10^-6	300	2.418*10^-11
35	8.279*10^-3	95	1.343*10^-6	350	9.158*10^-12
40	3.972*10^-3	100	5.597*10^-7	400	3.725*10^-12
45	1.995*10^-3	110	9.661*10^-8	450	1.585*10^-12
50	1.057*10^-3	120	2.438*10^-8	500	6.967*10^-13
55	5.821*10^-4	130	8.484*10^-9	600	1.454*10^-13
60	3.206*10^-4	140	3.845*10^-9	700	3.614*10^-14
65	1.718*10^-4	150	2.070*10^-9	800	1.170*10^-14
70	8.770*10^-5	160	1.244*10^-9	900	5.245*10^-15
75	4.178*10^-5	180	5.464*10^-10	1000	3.019*10^-15

A seguir temos a Figura 3 que simula o decaimento orbital onde é usado os dados da Tabela 2 que representa a densidade atmosférica variando em função da altitude.

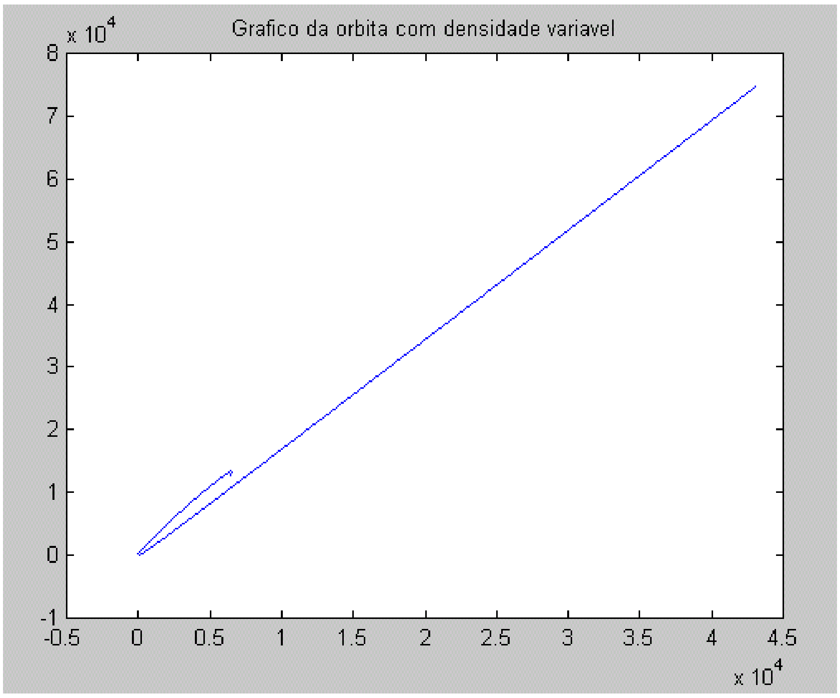

Figura 3- Decaimento Orbital com Densidade Variando

A seguir temos a Figura 4 que está comparando as duas orbitas anteriores para uma melhor compreensão do decaimento.

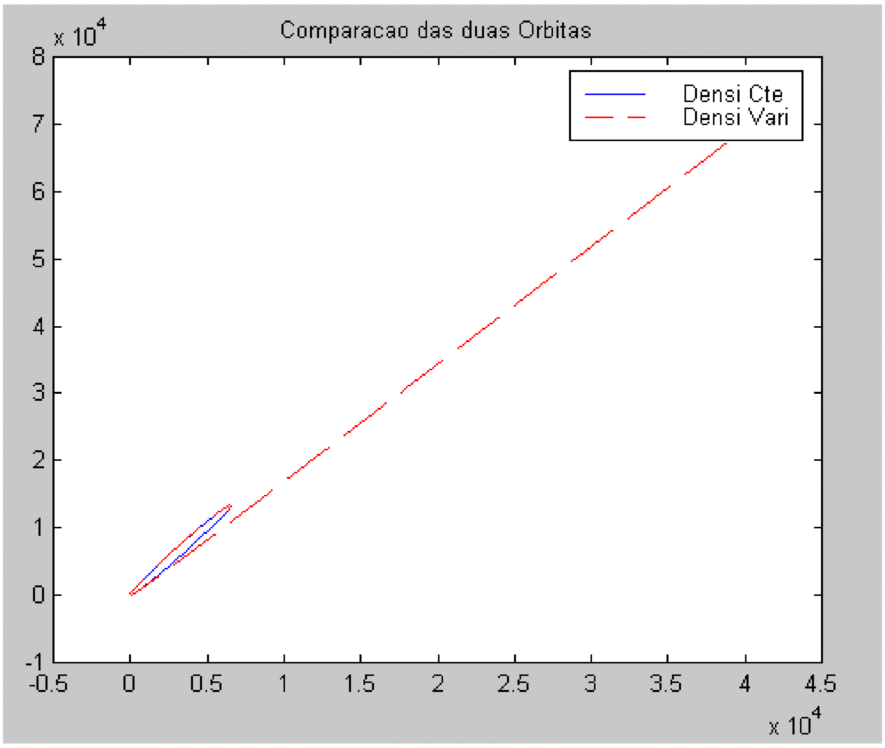

Figura 4- Comparação entre Densidade Constante e Variando

A seguir temos a Tabela 4 que representa o modelo exponencial da densidade:

$$\rho = e^{\frac{-h}{6,8}}$$

com $0 \le h \prec 110$ km, onde h é altitude.

Tabela 4

Altitude (Km)	Densidade da Tabela (km/m^3)	Densidade (km/m^3)
0	1.225*10^0	1
25	3.899*10^-2	2.53*10^-2
30	1.774*10^-2	1.21*10^-2
35	8.279*10^-3	5.816*10^-3
40	3.972*10^-3	2.788*10^-3
45	1.995*10^-3	1.336*10^-3
50	1.057*10^-3	6.407*10^-4
55	5.821*10^-4	3.071*10^-4
60	3.206*10^-4	1.472*10^-4
65	1.718*10^-4	7.057*10^-5
70	8.770*10^-5	3.383*10^-5
75	4.178*10^-5	1.621*10^-5
80	1.905*10^-5	7.774*10^-6
85	8.337*10^-6	3.726*10^-6
90	3.396*10^-6	1.786*10^-6
95	1.343*10^-6	8.563*10^-7
100	5.597*10^-7	4.105*10^-7
110	9.661*10^-8	9.432*10^-8

O programa "arrast" (vide Apêndice A, Programa 2), simula o decaimento orbital com o modelo exponencial da densidade atmosférica utilizando os dados da tabela anterior, como mostra a Figura 5 abaixo:

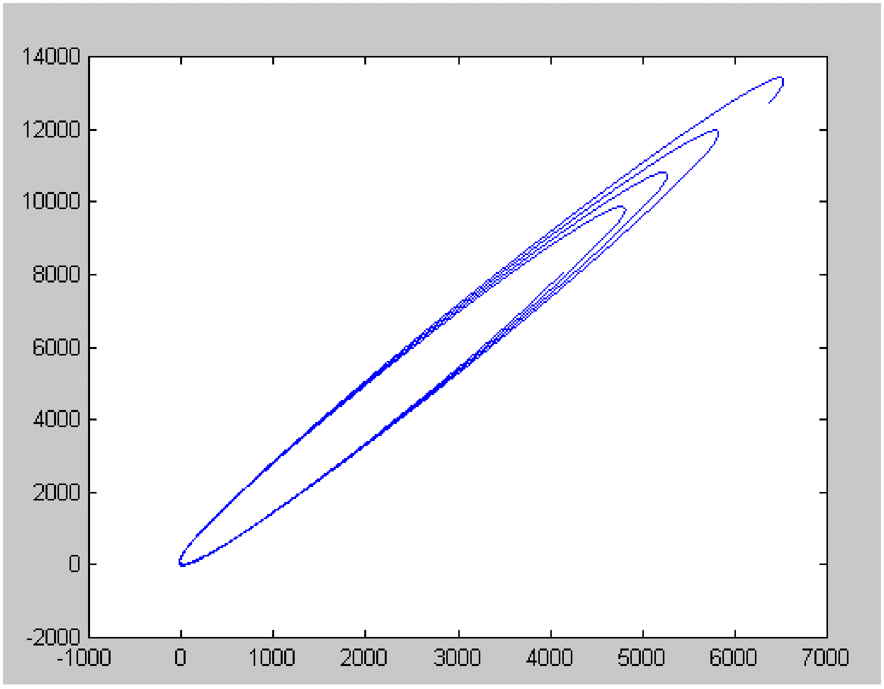

Figura 5- Decaimento Orbital com Modelo Exponencial

Esses programas foram feitos com um satélite fictício com a área de contato de forma esférica. Durante a realização do trabalho foram reproduzidos os mesmos programas com satélites fictícios também, mas com a área de contato em outras formas geométricas, foram baseadas num cubo, num paralelepípedo e numa placa.

Supondo que a área do satélite exposta à força de arrasto atmosférico seja um cubo de lado =4, como mostra a Figura 6:

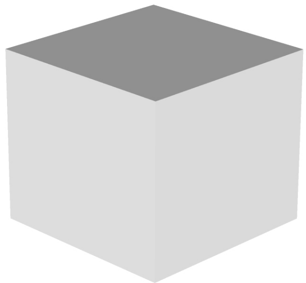

Figura 6 - Cubo

Obtivemos os seguintes resultados:

O programa arrast300 (vide Apêndice A, Programa 3) simula o decaimento orbital com dados da Tabela 3, com um corpo de área cúbica, com densidade variando como mostra a Figura 7:

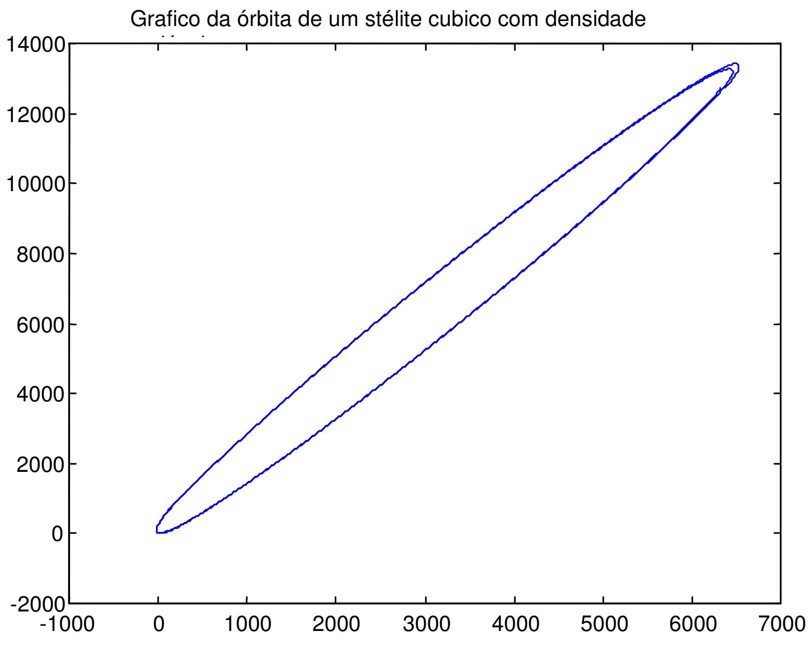

Figura 7 – Decaimento Orbital com Densidade Variando

Fazendo um zoom da Figura 7 temos o que mostra a Figura 8:

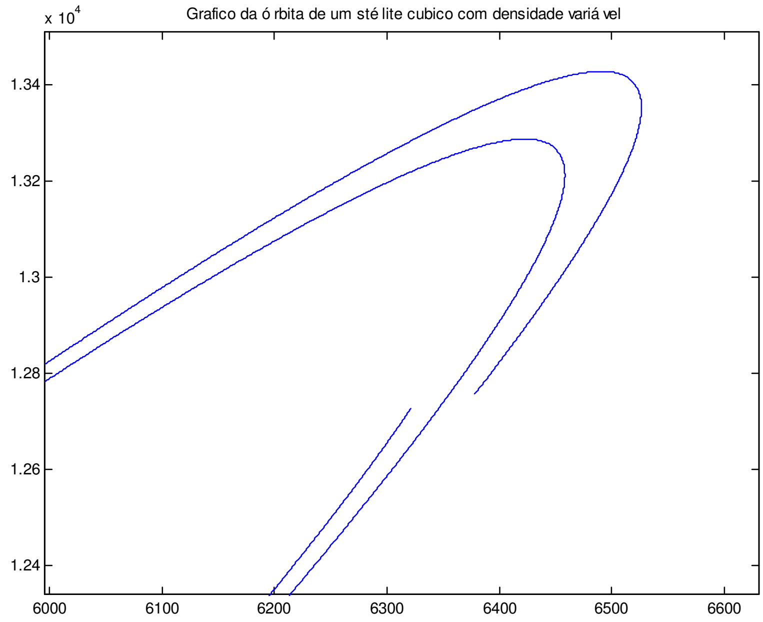

Figura 8 – Detalhe da Figura 7

O programa "arrasto301" (vide Apêndice A, Programa 4), simula o decaimento orbital com o modelo exponencial da densidade atmosférica utilizando os dados da Tabela 4, de um satélite com a área de contato de forma cúbica como mostra a Figura 9 abaixo:

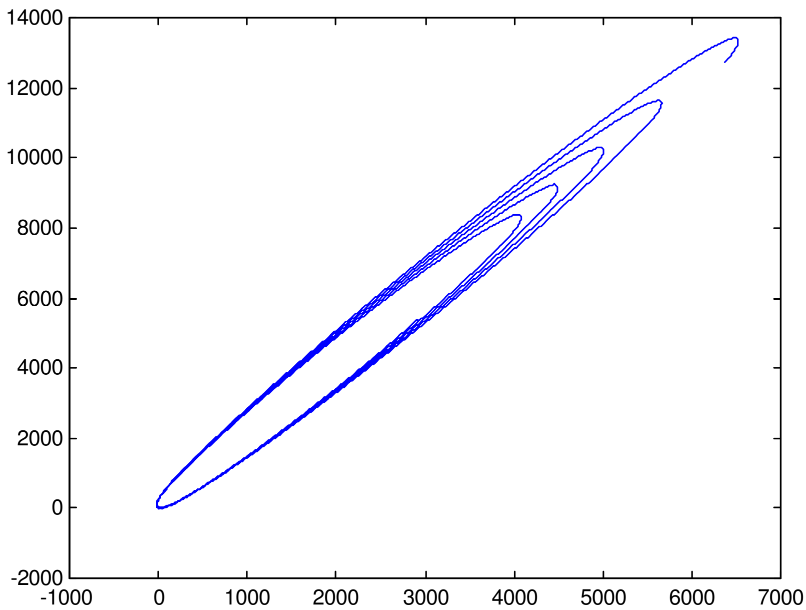

Figura 9 – Decaimento Orbital com modelo exponencial

Supondo que o satélite seja um paralelepípedo ABC, onde A= 4, B=5, C=6, como na figura, e que a menor área seja exposta à força de arrasto atmosférico, como mostra a Figura

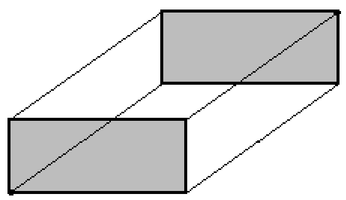

Figura 10 – Paralelepípedo

Obtivemos os seguintes resultados:

O programa paralelepipedo1 (vide Apêndice A, Programa 5) simula o decaimento orbital com dados da Tabela 3, com um corpo em forma de paralelepípedo, densidade variando como mostra a Figura 11:

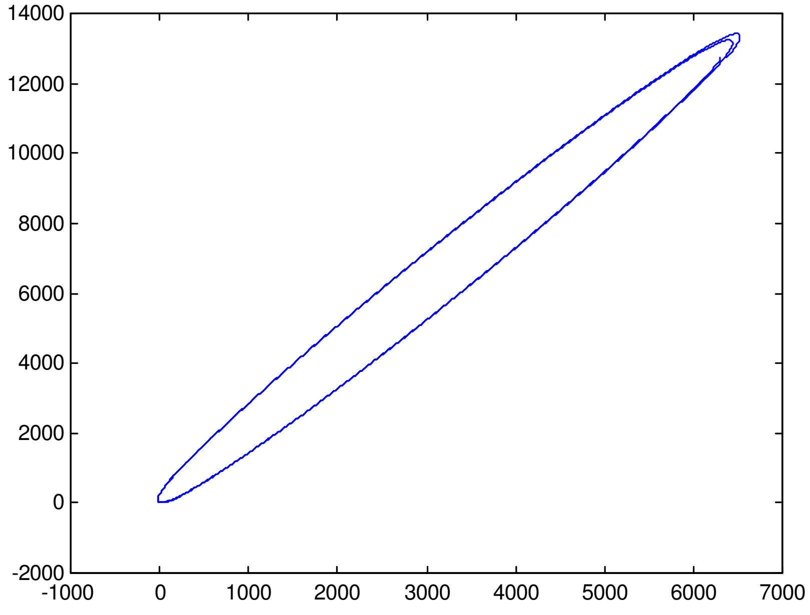

Figura 11 – Decaimento orbital com densidade variando

Fazendo um zoom para uma análise melhor temos o que nos mostra a Figura 12:

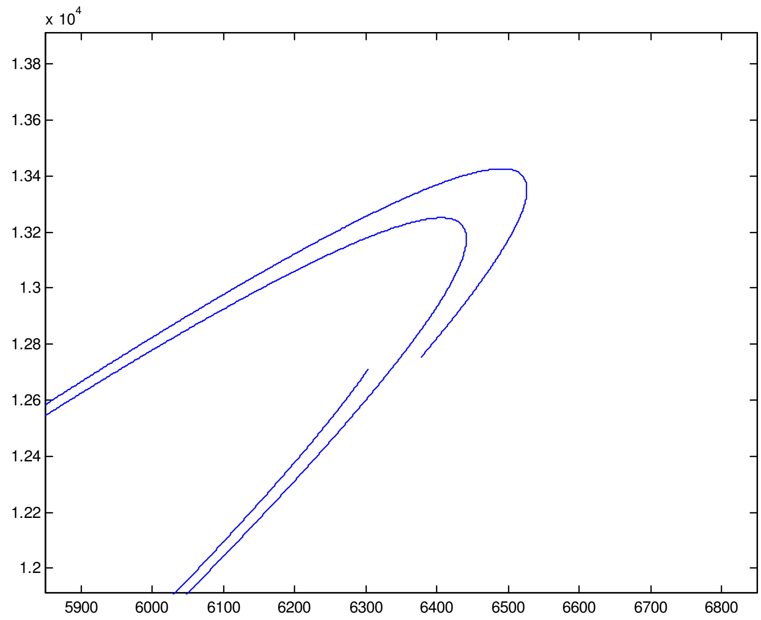

Figura 12 – Detalhe da Figura 11

O programa "paralelepipedo2" (vide Apêndice A, Programa 6), simula o decaimento orbital com o modelo exponencial da densidade atmosférica utilizando os dados da Tabela 4, de um satélite com a área de contato em forma cúbica como mostra a Figura13 abaixo:

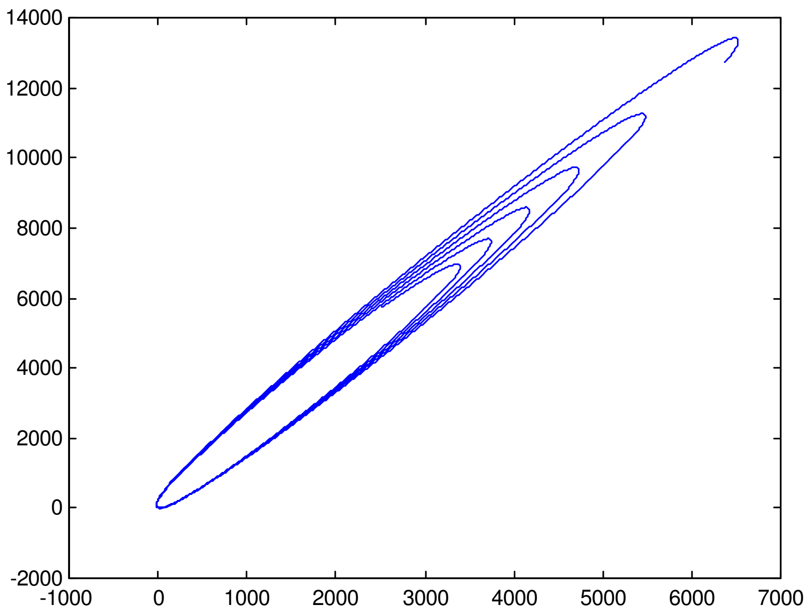

Figura 13 – Decaimento orbital com modelo exponencial

Supondo que o satélite é uma placa abc como mostra a figura, onde a= 1, b=4, c=6 , e que a menor área é exposta à força de arrasto atmosférico.

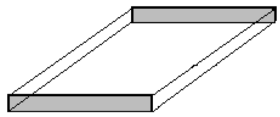

Figura 14 – Placa

Obtivemos os seguintes resultados:

O programa “placa1” (vide Apêndice A, Programa 7) simula o decaimento orbital com dados da Tabela 3, com um corpo em forma de placa, com densidade variando como mostra a Figura 15:

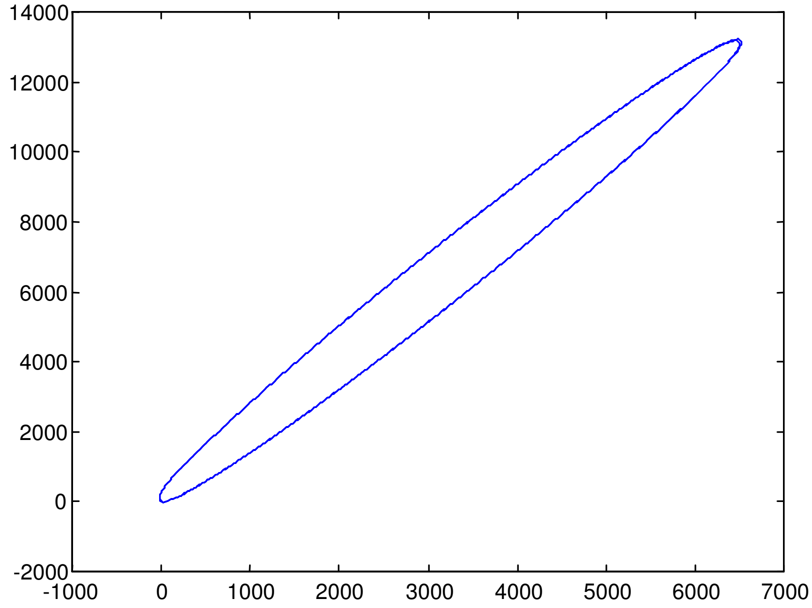

Figura 15 – Decaimento orbital com densidade variando

Fazendo uma zoom na figura acima para melhor visualização temos o que nos mostra a Figura 16:

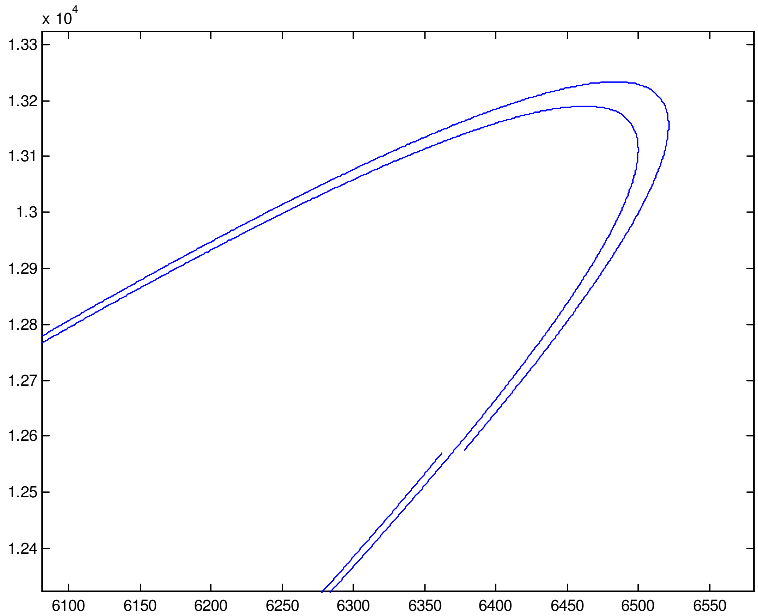

Figura 16 – Detalhe da Figura 15

O programa "placa2" (vide Apêndice A, Programa 8), simula o decaimento orbital com o modelo exponencial da densidade atmosférica utilizando os dados da Tabela 4, de um satélite com a área de contato em forma cúbica como mostra a Figura17 abaixo:

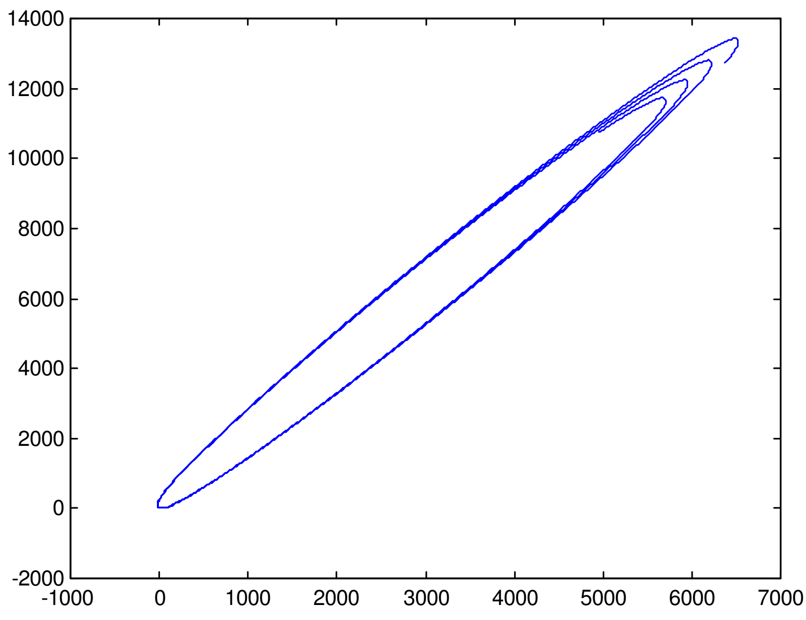

Figura 17– Decaimento Orbital com modelo exponencial

CAPITULO 6 – CONCLUSÕES, COMENTÁRIOS E SUGESTÕES

Este relatório apresenta o trabalho realizado no período de agosto de 2007 à julho de 2008, bem como nossa introdução ao tema, um estudo sobre o Matlab, Métodos Numéricos e Introdução à Mecânica Orbital, que serviram com ferramentas para o desenvolvimento do projeto.

Reproduziu – se as Simulações Numéricas feitas pela bolsista anterior e também a correção de alguns dados, realizou – se também um estudo sobre Transferência Orbital com Arrasto Atmosférico, que a Força que a Atmosfera exerce sobre o satélite quando esse reentra na mesma, essa Força é aplicada na menor área do satélite, já que quando esse reentra tende a expor a menor área. Então supondo satélites fictícios variando suas formas geométricas e consequentemente sua área, podemos observar que quando existe a força de Arrasto o que determina sua intensidade é a variação da densidade atmosférica que varia de acordo com a altitude e a área do satélite. Como fizemos para uma mesma variação de altitude os quatro casos, o que realmente variou e nos ajudou na comparação foi a variação da área.

Com base nestes programas e nestes modelo, objetiva-se, posteriormente, estudar as propriedades básicas desse processo. Assim, será possível analisar os problemas de colisão e interferência dos detritos espaciais com outros objetos encontrados no espaço como satélites, ônibus espaciais, e estações espaciais.

RERÊNCIAS

BOYCE, W. E.; DI PRIMA, R. C. *Equações Diferenciais Elementares e Problemas de Valores de Contorno*. 6ª ed., LTC, Rio de Janeiro, 1997.

CARNAHAN, B.; LUTHER, H. A.; WILKES, J. O. *Applied Numerical Methods***.** New York : John Wiley, 1969, 604 p. : ISBN 471.135070 (enc.)

CARRARA, V. *Aerodinâmica dos satélites artificiais* <www2.dem.inpe.br/val/publicações/carrara_aero_fte_01.pdf>; acessado em 02 de julho de 2008

(Apostila) Curso de MATLAB 5.1*; Introdução à Solução de Problemas de Engenharia*; 2ª edição; Programa Prodenge / Sub-Programa Reenge; Universidade do Estado do Rio de Janeiro.

EDWARDS JUNIOR, C H; PENNEY, D. E.; *Equações diferenciais elementares com problemas de cotorno*. 3. ed. Rio de Janeiro: Prentice Hall, 1995. 643p.

GUEDES, U. T.V.; KUGA, H. K.; SOUZA, M.L.O. *Reentrada Atmosférica.* <www.feg.unesp.br/~orbital/sputinik/capitulo-8.pdf>, acessado em 02 de julho de 2008.

HANSELMAN,D., & LITTLEFIELD,B. *MatLab 5: Guia do Usuário, Versão do Estudante*, Makron Books,1999.

KUGA, H.K., RAO, K.R. *Introdução à Mecânica Orbital,* INPE, São José dos Campos - SP, 1995.

Thomson, W. T.; *Introduction to Space Dynamics*. New York: Courier Dover Publications, 1986, 352p.: ISBN 0486651134 (enc).

Apêndice A – Programas e simulações com força de arrasto atmosférico

Programa 1:

```
%transferencia com arrasto atmosferico
clear all
close all

opt = odeset('AbsTol',0.00001,'RelTol', 0.0000001);

x0 = [6378;12756 ; 0.5; 1.5];     % espaço xo inicial
mu = 398600;                       % a unidade de espaço e o km
r  = [x0(1); x0(2)];                  %x(1)= x = posiçao   , x(4)=y =
posicao
mr = sqrt(x0(1)^2 + x0(2)^2);      % modulo do vetor posiçao
v   = [x0(3); x0(4)];                  %x(3)= xponto=velocidade   ,
x(4)=yponto=velocidade
mv = sqrt(x0(3)^2+x0(4)^2);        % modulo do vetor velocidade
a  = mr*mu/(2*mu-mr*mv^2);        % semi-eixo maior 'a'

periodo = 2*pi*sqrt((a^3)/398600);
tspan   = [0:0.5:2.*periodo];

 [t,x]  = ode45('arrasto1',tspan,x0,opt);

 [t2,x2]  = ode23s('arrasto2',tspan,x0);

 figure(1)
 plot(x(:,1),x(:,2));
 title('Grafico da orbita com densidade cte (h=700 Km)')
 zoom on

 figure(2)
 plot(x2(:,1),x2(:,2));
 title('Grafico da orbita com densidade variavel')
 zoom on

 figure(3)
 plot(x(:,1),x(:,2),'b',x2(:,1),x2(:,2),'r--');
 title('Comparacao das duas Orbitas');
 legend('Densi Cte','Densi Vari')
 zoom on
```

Programa 1.1 :

```
%arrasto1
function dx = f(t,x);

mu     = 398600; % a unidade de espaço e o km(parametro gravt da
terra)
```

```
r     = sqrt(x(1)^2 + x(2)^2); % modulo do vetor posiçao
%v       =  [x(3);  x(4)];              %x(3)=  xponto=velocidade   ,
x(4)=yponto=velocidade
mv    = sqrt(x(3)^2+x(4)^2);   % modulo do vetor velocidade
%rt    = 6378;                  % raio da terra
raio  = 2;                     % raio do satelite
h     = 700;                   %  altura  %h= ra-rt;  ra= h+rt;
cd    = 0.5;                   % coeficiente de arrasto
area  = (pi.*(raio^2));        % area do satelite
densi =3.614*10^-14;            % densidade correspondente a altura
700


f  =  0.5.*cd.*area.*densi.*mv^2;  %  equaçao  da  força  de  arrasto
atmosférico

dx    =[x(3);
        x(4);
        ((-mu/r^3).*x(1))-(f.*x(3)/mv);
        ((-mu/r^3).*x(2))-(f.*x(4)/mv)];      %   vetor coluna
```

Programa 1.2 :

```
%Arrasto2
function dx = f(t,x);

mu     = 398600; % a unidade de espaço e o km(parametro gravt da
terra)
r     = sqrt(x(1)^2 + x(2)^2); % modulo do vetor posiçao
%v       =  [x(3);  x(4)];              %x(3)=  xponto=velocidade   ,
x(4)=yponto=velocidade
mv    = sqrt(x(3)^2+x(4)^2);   % modulo do vetor velocidade
%rt    = 6378;                  % raio da terra
raio  = 2;                     % raio do satelite
cd    = 0.5;                   % coeficiente de arrasto
area  = (pi.*(raio^2));        % area do satelite
h      = r-6378;

if h > 1000
    densi = 3.019*10^-15;
end
if h >900
    if h<=1000
        densi = 5.245*10^-15;
    end
end
if h >800
    if h<=900
        densi = 1.17*10^-14;
    end
end
```

```
if h >700
    if h<=800
        densi = 3.614*10^-14;
    end
end
if h >600
    if h<=700
        densi = 1.454*10^-13;
    end
end
if h >500
    if h<=600
        densi = 6.967*10^-13;
    end
end
if h >400
    if h<=500
        densi = 3.725*10^-12;
    end
end
if h >300
    if h<=400
        densi = 2.418*10^-11;
    end
end
if h >200
    if h<=300
        densi = 2.789*10^-10;
    end
end
if h >180
    if h<=200
        densi = 5.464*10^-10;
    end
end
if h >160
    if h<=180
        densi = 1.244*10^-9;
    end
end
if h >150
    if h<=160
        densi = 2.070*10^-9;
    end
end
if h >140
    if h<=150
        densi = 3.845*10^-9;
    end
end
if h >130
    if h<=140
        densi = 8.484*10^-9;
```

```
    end
end
if h >120
    if h<=130
        densi = 2.438*10^-8;
    end
end
if h >110
    if h<=120
        densi = 9.661*10^-8;
    end
end
if h >100
    if h<=110
        densi = 5.297*10^-7;
    end
end
if h<=100
    densi = 1.343*10^-6;
end

f  =  0.5.*cd.*area.*densi.*mv^2;  %  equaçao  da  força  de  arrasto
atmosférico

dx    =[x(3);
        x(4);
        ((-mu/r^3).*x(1))-(f.*x(3)/mv);
        ((-mu/r^3).*x(2))-(f.*x(4)/mv)];      %   vetor coluna
```

Programa 2 :

```
%arrast

opt = odeset('AbsTol',0.00001,'RelTol', 0.0000001);

x  = [6378;12756 ; 0.5; 1.5];     % espaço xo inicial
mu = 398600;                      % a unidade de espaço e o km
r  = [x(1); x(2)];                   %x(1)= x = posiçao   , x(4)=y =
posicao
mr = sqrt(x(1)^2 + x(2)^2);       % modulo do vetor posiçao
v   = [x(3); x(4)];                     %x(3)= xponto=velocidade   ,
x(4)=yponto=velocidade
mv = sqrt(x(3)^2+x(4)^2);         % modulo do vetor velocidade
a  = mr*mu/(2*mu-mr*mv^2);        % semi-eixo maior 'a'

periodo = 2*pi*sqrt((a^3)/398600);
tspan   = [0:3.*periodo];

 [t,x]  = ode45('forma',tspan,x,opt);
 plot(x(:,1),x(:,2))
 zoom on
```

Programa 2.1 :

```
%forma
function dx = f(t,x);

mu      = 398600;                          % a unidade de espaço e o
km(parametro gravt da terra)
r     = sqrt(x(1)^2 + x(2)^2); % modulo do vetor posiçao
v       =  [x(3);  x(4)];                 %x(3)=  xponto=velocidade   ,
x(4)=yponto=velocidade
mv    = sqrt(x(3)^2+x(4)^2);    % modulo do vetor velocidade
rt    = 6378;                   % raio da terra
raio  = 2;                      % raio do satelite
%h    = 700;                    %  altura  %h= ra-rt;  ra= h+rt;
cd    = 0.5;                    % coeficiente de arrasto
area  = (pi.*(raio^2));         % area do satelite

%h=[0;
25;30;35;40;45;50;55;60;65;70;75;80;85;90;95;100;110;120;130;140;
 %
150;160;180;200;250;300;350;400;450;500;600;700;800;900;1000];%
altura

%D=[8.44;6.49;6.75;7.07;7.47;7.83;7.95;7.73;7.29;6.81;6.33;6.00;5.
70;5.41;5.38;5.74;6.15;
 %
8.06;11.6;16.1;20.6;24.6;26.3;33.2;38.5;46.9;52.5;56.4;59.4;62.2;6
5.8;79;109;164;225;
 %  268]; % escala de altitude

densi =exp(-100/6.15);
f  =  0.5.*cd.*area.*densi.*mv^2;  %  equaçao  da  força  de  arrasto
atmosférico

dx =    [x(3);
        x(4);
        ((-mu/r^3).*x(1))-(f.*x(3)/mv);
        ((-mu/r^3).*x(2))-(f.*x(4)/mv)];      %   vetor coluna
```

Programa 3

```
%arrasto300
 clear all
 close all
opt = odeset('AbsTol',0.00001,'RelTol',0.0000001);
 x0 = [6378;12756;0.5;1.5];
 mu = 398600;
 r = [x0(1);x0(2)];
 v = [x0(3);x0(4)];
 mr = sqrt(x0(1)^2 + x0(2)^2);
 mv = sqrt(x0(3)^2 + x0(4)^2);
```

```
 a = mr*mu/(2*mu-mr*mv^2);
 periodo = 2*pi*sqrt((a^3)/398600);
 tspan = [0:0.5:2.*periodo];
 [t,x] = ode45('arrasto300',tspan,x0,opt);
 figure
 plot(x(:,1),x(:,2));
 title('Grafico da órbita de um stélite cubico com densidade
variável ');
 zoom on
```

Programa 3.1

```
function dx = f(t,x);
mu = 398600; %a unidade de espaço e o km (parametro gravt da
Terra)
% r = [x(1);x(2)];
mr = sqrt (x(1)^2 + x(2)^2); % modulo do vetor posição
%v = [x(3);x(4)];
mv = sqrt (x(3)^2 + x(4)^2); % modulo do vetor velocidade
%rt = 6378; (raio da Terra);
lado = 4;
cd = 0.5;
area = (lado^2);
h = mr-6378;

if h >300
    %if h<=300
        densi =2* 2.789*10^-10;
    %end
end


if h >200
    if h<=300
        densi = 2.789*10^-10;
    end
end
if h >180
    if h<=200
        densi = 5.464*10^-10;
    end
end
if h >160
    if h<=180
        densi = 1.244*10^-9;
    end
end
if h >150
    if h<=160
        densi = 2.070*10^-9;
    end
end
```

```
if h >140
    if h<=150
        densi = 3.845*10^-9;
    end
end
if h >130
    if h<=140
        densi = 8.484*10^-9;
    end
end
if h >120
    if h<=130
        densi = 2.438*10^-8;
    end
end

if h <=120
    %if h<=130
        densi = 2.438*10^-8/4;
    %end
end


f  =  0.5.*cd.*area.*densi.*mv^2;  %  equaçao  da  força  de  arrasto
atmosférico

dx    =[x(3);
        x(4);
        ((-mu/mr^3).*x(1))-(f.*x(3)/mv);
        ((-mu/mr^3).*x(2))-(f.*x(4)/mv)];      %   vetor coluna
```

Programa 4

```
opt = odeset('AbsTol',0.00001,'RelTol',0.0000001);
 x = [6378;12756;0.5;1.5]; %espaco inicial
 mu = 398600;
 r = [x(1); x(2)];
 mr = sqrt(x(1)^2+x(2)^2);
 v = [x(3);x(4)];
 mv = sqrt(x(3)^2 + x(4)^4);
 a = mr*mu/(2*mu-mr*mv^2);
 periodo = 2*pi*sqrt((a^3)/398600);
 tspan = [0:0.5:3.*periodo];
 [t,x] = ode45('arrasto301',tspan,x,opt);
 plot(x(:,1),x(:,2))
```

Programa 4.1

```
function dx = f(t,x);

mu      = 398600;                        % a unidade de espaço e o
```

```
km(parametro gravt da terra)
r      = sqrt(x(1)^2 + x(2)^2); % modulo do vetor posiçao
v       =  [x(3);  x(4)];                 %x(3)=  xponto=velocidade   ,
x(4)=yponto=velocidade
mv     = sqrt(x(3)^2+x(4)^2);    % modulo do vetor velocidade
rt     = 6378;                   % raio da terra
lado = 4;                       % raio do satelite
%h     = 700;                    %  altura  %h= ra-rt;  ra= h+rt;
cd     = 0.5;                    % coeficiente de arrasto
area  = lado^2;          % area do satelite

%h=[110;120;130;140;150;160;180;200;250;300];% altura

%D=[8.44;6.49;6.75;7.07;7.47;7.83;7.95;7.73;7.29;6.81;6.33;6.00;5.
70;5.41;5.38;5.74;6.15;
 %
8.06;11.6;16.1;20.6;24.6;26.3;33.2;38.5;46.9;52.5;56.4;59.4;62.2;6
5.8;79;109;164;225;
 %  268]; % escala de altitude

densi =exp(-100/6.15);
f  =  0.5.*cd.*area.*densi.*mv^2;  %  equaçao  da  força  de  arrasto
atmosférico

dx =    [x(3);
        x(4);
        ((-mu/r^3).*x(1))-(f.*x(3)/mv);
        ((-mu/r^3).*x(2))-(f.*x(4)/mv)];      %   vetor coluna
```

Programa 5

```
%paralelepipedo
 clear all
 close all
 opt = odeset('AbsTol',0.00001,'RelTol',0.0000001);
 x0 = [6378;12756;0.5;1.5];
 mu = 398600;
 r = [x0(1);x0(2)];
 mr = sqrt(x0(1)^2 + x0(2)^2);
 v = [x0(3);x0(4)];
 mv = sqrt(x0(3)^2 + x0(4)^2);
 a = mr*mu/(2*mu-mr*mv^2);
 periodo = 2*pi*sqrt((a^3)/398600);
 tspan = [0:0.5:2.*periodo];
 [t,x] = ode45('paralelepipedo1',tspan,x0,opt);
 plot(x(:,1),x(:,2));
```

Programa 5.1

```
%Considerando  um paralelepipedo de lados a=4, b=5 e c=6. A menor
area se dá pelos lados ab
```

```
function dx = f(t,x);
mu = 398600; %a unidade de espaço e o km (parametro gravt da
Terra)
% r = [x(1);x(2)];
mr = sqrt (x(1)^2 + x(2)^2); % modulo do vetor posição
%v = [x(3);x(4)];
mv = sqrt (x(3)^2 + x(4)^2); % modulo do vetor velocidade
%rt = 6378; (raio da Terra);
ladoa = 4;
ladob = 5;
cd = 0.5;
area = (ladoa*ladob);
h = mr-6378;

if h >300
    %if h<=300
        densi =2* 2.789*10^-10;
    %end
end


if h >200
    if h<=300
        densi = 2.789*10^-10;
    end
end
if h >180
    if h<=200
        densi = 5.464*10^-10;
    end
end
if h >160
    if h<=180
        densi = 1.244*10^-9;
    end
end
if h >150
    if h<=160
        densi = 2.070*10^-9;
    end
end
if h >140
    if h<=150
        densi = 3.845*10^-9;
    end
end
if h >130
    if h<=140
        densi = 8.484*10^-9;
    end
end
if h >120
    if h<=130
```

```
        densi = 2.438*10^-8;
    end
end

if h <=120
    %if h<=130
        densi = 2.438*10^-8/4;
    %end
end


f  =  0.5.*cd.*area.*densi.*mv^2;  %  equaçao  da  força  de  arrasto
atmosférico

dx     =[x(3);
        x(4);
        ((-mu/mr^3).*x(1))-(f.*x(3)/mv);
        ((-mu/mr^3).*x(2))-(f.*x(4)/mv)];      %   vetor coluna
```

Programa 6

```
opt = odeset('AbsTol',0.00001,'RelTol',0.0000001);
 x = [6378;12756;0.5;1.5]; %espaco inicial
 mu = 398600;
 r = [x0(1);x0(2)];
 r = [x(1); x(2)];
 mr = sqrt(x(1)^2+x(2)^2);
 v = [x(3);x(4)];
 mv = sqrt(x(3)^2 + x(4)^4);
 a = mr*mu/(2*mu-mr*mv^2);
 periodo = 2*pi*sqrt((a^3)/398600);
 tspan = [0:0.5:3.*periodo];
 [t,x] = ode45('paralelepipedo2',tspan,x0,opt);
 plot(x(:,1),x(:,2))
```

Programa 6.1

```
function dx = f(t,x);

mu      = 398600;                      %  a  unidade  de  espaço  e  o
km(parametro gravt da terra)
r      = sqrt(x(1)^2 + x(2)^2); % modulo do vetor posiçao
v        =  [x(3);  x(4)];              %x(3)=  xponto=velocidade   ,
x(4)=yponto=velocidade
mv     = sqrt(x(3)^2+x(4)^2);   % modulo do vetor velocidade
rt     = 6378;                  % raio da terra
ladoa = 4;                      % raio do satelite
ladob = 5;
%h     = 700;                   %  altura  %h= ra-rt;  ra= h+rt;
cd     = 0.5;                   % coeficiente de arrasto
area   = ladoa*ladob;        % area do satelite
```

```
%h=[110;120;130;140;150;160;180;200;250;300];% altura

%D=[8.44;6.49;6.75;7.07;7.47;7.83;7.95;7.73;7.29;6.81;6.33;6.00;5.
70;5.41;5.38;5.74;6.15;
 %
8.06;11.6;16.1;20.6;24.6;26.3;33.2;38.5;46.9;52.5;56.4;59.4;62.2;6
5.8;79;109;164;225;
 %  268]; % escala de altitude

densi =exp(-100/6.15);
f  =  0.5.*cd.*area.*densi.*mv^2;  %  equaçao  da  força  de  arrasto
atmosférico

dx =    [x(3);
         x(4);
         ((-mu/r^3).*x(1))-(f.*x(3)/mv);
         ((-mu/r^3).*x(2))-(f.*x(4)/mv)];      %   vetor coluna
```

Programa 7

```
clear all
 close all
 opt = odeset('AbsTol',0.00001,'RelTol',0.0000001);
 x=[6378;12576;0.5;1.5];
 mu=398600;
 r = [x(1);x(2)];
 mr = sqrt(x(1)^2+x(2)^2);
 v = [x(3);x(4)];
 mv = sqrt(x(3)^2+x(4)^2);
 a = mr*mu/(2*mu-mr-mv^2);
 periodo = 2*pi*sqrt((a^3)/398600);
 tspan = [0:0.5:2*periodo];
 [t,x] = ode45('placa1',tspan,x,opt);
 plot(x(:,1),x(:,2))
```

Programa 7.1

```
%Considerando  uma placa de lados a=1, b=5 e c=6.A menor area se
dá pelos lados ab
function dx = f(t,x);
mu  =  398600;  %a  unidade  de  espaço  e  o  km  (parametro  gravt  da
Terra)
% r = [x(1);x(2)];
mr = sqrt (x(1)^2 + x(2)^2); % modulo do vetor posição
%v = [x(3);x(4)];
mv = sqrt (x(3)^2 + x(4)^2); % modulo do vetor velocidade
%rt = 6378; (raio da Terra);
ladoa = 1;
ladob = 5;
```

```
cd = 0.5;
area = (ladoa*ladob);
h = mr-6378;

if h >300
    %if h<=300
        densi =2* 2.789*10^-10;
    %end
end


if h >200
    if h<=300
        densi = 2.789*10^-10;
    end
end
if h >180
    if h<=200
        densi = 5.464*10^-10;
    end
end
if h >160
    if h<=180
        densi = 1.244*10^-9;
    end
end
if h >150
    if h<=160
        densi = 2.070*10^-9;
    end
end
if h >140
    if h<=150
        densi = 3.845*10^-9;
    end
end
if h >130
    if h<=140
        densi = 8.484*10^-9;
    end
end
if h >120
    if h<=130
        densi = 2.438*10^-8;
    end
end

if h <=120
    %if h<=130
        densi = 2.438*10^-8/4;
    %end
end
```

```
f  =  0.5.*cd.*area.*densi.*mv^2;  %  equaçao  da  força  de  arrasto
atmosférico

dx     =[x(3);
           x(4);
           ((-mu/mr^3).*x(1))-(f.*x(3)/mv);
           ((-mu/mr^3).*x(2))-(f.*x(4)/mv)];       %   vetor coluna
```

Programa 8

```
%placa
 opt = odeset('AbsTol',0.00001,'RelTol',0.0000001);
 x = [6378;12756;0.5;1.5];
 mu = 398600;
 r = [x(1);x(2)];
 mr = sqrt(x(1)^2+x(2)^2)
 v = [x(3);x(4)];
 mv = sqrt(x(3)^2+x(4)^2);
 a = mr*mu/(2*mu-mr*mv^2);
 periodo = 2*pi*sqrt((a^3)/398600);
 tspan = [0:0.5:3.*periodo];
 [t,x] = ode45('placa2',tspan,x,opt);
 plot(x(:,1),x(:,2))
```

Programa 8.1

```
function dx = f(t,x);

mu      = 398600;                        %  a  unidade  de  espaço  e  o
km(parametro gravt da terra)
r       = sqrt(x(1)^2 + x(2)^2); % modulo do vetor posiçao
v        =  [x(3);  x(4)];               %x(3)=  xponto=velocidade   ,
x(4)=yponto=velocidade
mv     = sqrt(x(3)^2+x(4)^2);    % modulo do vetor velocidade
rt     = 6378;                   % raio da terra
ladoa = 1;                       % raio do satelite
ladob = 5;
cd     = 0.5;                    % coeficiente de arrasto
area   = ladoa*ladob;         % area do satelite
densi =exp(-100/6.15);
f  =  0.5.*cd.*area.*densi.*mv^2;  %  equaçao  da  força  de  arrasto
atmosférico
dx =     [x(3);
          x(4);
          ((-mu/r^3).*x(1))-(f.*x(3)/mv);
          ((-mu/r^3).*x(2))-(f.*x(4)/mv)];       %   vetor coluna
```