



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA  
**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**



## **INFLUÊNCIA DO COEFICIENTE DE ARRASTO NA IDENTIFICAÇÃO DO CORREDOR DE REENTRADA**

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

**Alex Machado da Silva (Universidade Federal do ABC, Bolsista  
PIBIC/CNPq)**

**E-mail: alex.machado@aluno.ufabc.edu.br**

**Ulisses Thadeu Vieira Guedes (DMC/INPE, Orientador)**

**E-mail: ulisses@dem.inpe.br**

**Julho de 2011**

## SUMÁRIO

<b>RESUMO.....</b>	<b>4</b>
<b>1.INTRODUÇÃO.....</b>	<b>5</b>
1.1-Objetivos.....	6
<b>2.FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>7</b>
2.1-Força Gravitacional.....	7
2.2- Velocidade em Uma Orbita Circular de raio R.....	7
2.3- Força de Arrasto.....	8
2.4- Modelo Exponencial de Variação de Densidade.....	8
2.5- Método de Runge-Kutta.....	9
2.6- Método de Aproximação Newton-Raphson.....	10
<b>3-METODOLOGIA.....</b>	<b>11</b>
3.1-Parte 1.....	11
3.2-Parte 2.....	12

<b>4-RESULTADOS E DISCUSSÕES.....</b>	<b>14</b>
4.1-Parte 1.....	14
4.2-Parte 2.....	16
<b>5. CONCLUSÃO.....</b>	<b>27</b>
<b>6. REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>28</b>

**ANEXOS**

## RESUMO

Este trabalho, iniciado em agosto de 2010, tem como objetivo estudar o efeito do coeficiente de arrasto na trajetória de reentrada de um corpo na trajetória de reentrada, considerando modelo de campo gravitacional com harmônico zonal até ordem 6. Ao reentrar na atmosfera um corpo está sujeito a ação de forças tais como o campo gravitacional e força aerodinâmica. No estudo considera-se um corpo esférico inerte (sem rotação) reduzindo a força aerodinâmica à sua componente de arrasto, que está diretamente ligada ao coeficiente de arrasto. Utilizando um modelo computacional já existente, foi possível saber a influência direta das forças no corpo durante a reentrada. Assim modificando, o coeficiente de arrasto (de 0.01 a 2.5) foram realizadas diversas simulações, obtendo como resultado diferentes tempos e diferentes locais de queda.

## 1- INTRODUÇÃO

Em 1957 foi lançado pela URSS o primeiro satélite artificial, a entrar em órbita ao redor da Terra. Desde então, já ocorreram várias missões espaciais com as mais diversas finalidades e o aumento de detritos espaciais passaram a ser motivo de grande preocupação nos últimos anos. Colisões na velocidade orbital podem ser altamente danosas, tanto em órbita prejudicando o funcionamento de satélites e estruturas espaciais, como em uma possível reentrada de tais detritos na atmosfera. O ônibus espacial Discovery teve que realizar uma manobra evasiva de uma colisão durante a STS-48, em que procedeu-se no acionamento de um subsistema de segurança da espaçonave para evitar a colisão com detritos de um satélite russo, Kosmos. A Estação Espacial Internacional também já teve que realizar diversas manobras evasivas. Janelas de ônibus espaciais já retornaram a Terra com danos provenientes do espaço, com a possibilidade de a janela estilhaçar-se durante a reentrada. Estima-se que uma lasca de tinta do tamanho de um grão de sal, orbitando a uma velocidade de 14.400 km/h, pode abrir uma significativa cratera de 2,5 cm de diâmetro (VENTUROLI,1999).

Em maio de 2000, duas esferas metálicas caíram em fazendas na África do Sul. A agência espacial estadunidense admitiu que ambos os objetos pertenciam a um foguete Delta lançado em 1996. O primeiro detrito, que pesava cerca de 30 kg, caiu na Cidade do Cabo e contou com a presença de testemunhas. O segundo objeto, pousou três dias depois do primeiro e pesava cerca de 55 kg.

Diante disto, surgem algumas preocupações referentes a reentrada de estruturas espaciais/detritos na atmosfera: como as coordenadas que este atingirá o solo e se a área atingida é povoada.

Para saber as coordenadas de colisão é necessário modelar a reentrada do objeto considerando diversos fatores, como a influência do campo gravitacional, radiação solar, campo magnético e arrasto.

No Presente trabalho estudaremos os efeitos da reentrada especialmente para um satélite. Utilizando um modelo computacional existente, e a partir de condições orbitais em baixa altitude, foi testado um conjunto de valores do Coeficiente de Arrasto (de 0.5 a 2.5)

verificando e avaliando a partir de quais condições o satélite reentrará na atmosfera e alcançará o solo.

### **1.1- Objetivos**

- Estudar o efeito do Coeficiente de Arrasto na trajetória de reentrada atmosférica de um corpo esférico sujeito a variações de densidade, pressão e temperatura decorrentes da reentrada.

-Verificar e avaliar a partir de quais condições o satélite reentrará na atmosfera e alcançará o solo considerando um modelo de campo gravitacional com harmônico zonal até 6.

## 2. FUNDAMENTAÇÃO TEÓRICA

Um satélite ao reentrar na Atmosfera está sujeito a uma série de Forças. Para o modelo desenvolvido foram consideradas as forças de arrasto e a gravitacional:

### 2.1 Força Gravitacional

É uma força fundamental de atração que age entre todos os objetos por causa de suas massas, isto é, a quantidade de matéria de que são constituídos, tem a direção paralela a linha que une o centro de massa dos corpos.

No nosso caso consideramos essa força na Interação entre dois corpos a Terra e o Satélite.

Considerando a Terra como uma esfera uniforme de massa M. A intensidade da força gravitacional entre os dois corpos é:

$$\vec{F} = -G \frac{Mm}{r^2} \hat{r} \quad (1)$$

Onde:

G= constante gravitacional=  $6,67 \times 10^{-11} \text{ N.m}^2/\text{kg}^2$ .

M= massa da Terra  $5,98 \times 10^{24} \text{ kg}$ .

m= massa do satélite.

r = distância entre os dois corpos.

### 2.2 Velocidade em Uma Orbita Circular de raio R

Para que um satélite que esteja em uma orbita circular permaneça estável sem a ação da força de arrasto, é necessário que a sua velocidade obedeça a seguinte equação:

$$V = (GM/R)^{0,5} \quad (2)$$

Onde:

G= Constante gravitacional.

M= Massa da Terra.

R=Distância do centro de massa da terra ao corpo em análise.

V= Velocidade.

### 2.3 Força de Arrasto

Na mecânica dos fluidos, Força de Arrasto é a força que faz resistência ao movimento de um objeto sólido através de um fluido, dada pela seguinte equação:

$$F_d = \frac{C_d \rho A V^2}{2} \quad (3)$$

Onde:

$C_d$ = Coeficiente de Arrasto.

$\rho$  = Densidade do Fluido.

A=Área Normal ao escoamento.

V= Velocidade Relativa entre o Objeto e o Fluido.

### 2.4 Modelo Exponencial de Variação de Densidade

Sabendo o valor da densidade em determinadas altitudes, é possível construir um modelo (**Wertz-PIBIC** (anexo 1)) que calcula a densidade em pontos desconhecidos através de uma aproximação exponencial:

$$\rho = \rho_0 e^{-k(h-h_0)} \quad (4)$$

$\rho_0$ =Valor da densidade para uma altitude de referência  $h_0$ .



O valor de k (constante) pode ser determinado sabendo duas densidades referentes a dois pontos diferentes, e isolando o valor de k.

O modelo atmosférico implementado segue o padrão CIRA-76. Foi utilizado uma tabela condensada contendo densidade e temperatura da atmosfera em função da altitude.

## 2.5- Método de Runge-Kutta

O método de Runge-kutta é uma forma numérica de Integração, que de modo geral assume a seguinte forma:

$$y_{n+1} - y_n = h\phi(x_n, y_n, h), \quad (5)$$

Onde

$$\phi(x, y, h) = \sum_{r=1}^R c_r k_r, \quad (6)$$

$$k_1 = f(x, y), \quad (7)$$

$$k_n = f\left(x + a_r h, y + h \sum_{s=1}^{r-1} b_{rs} k_s\right); r = 2, 3, \dots, R, \quad (8)$$

$$a_r = \sum_{s=1}^{r-1} b_{rs}; r = 2, 3, \dots, R. \quad (9)$$

## 2.6 Método de Aproximação Newton-Raphson

Este Método tem o objetivo de estimar raízes de uma função, a partir de uma Interação Numérica.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (10)$$

Onde  $n$  indica a  $n$ -ésima iteração do algoritmo e  $f'(x_n)$  é a derivada da função  $f$  em  $x_n$ .

Assim, a partir da anomalia média e da excentricidade (coordenadas keplerianas) é possível obter a anomalia excêntrica como veremos a seguir:

$$E_{n+1} = E_n - \frac{E_n - e \sin(E_n) - M_0}{1 - e \cos(E_n)} \quad (11)$$

### 3. METODOLOGIA

Em 01 de Agosto de 2010, na área de Dinâmica Orbital, Cálculo Numérico e Linguagens de Programação, foi iniciado esse projeto de pesquisa seguindo a Orientação do Dr. Ulisses Thadeu Vieira Guedes.

#### 3.1-Parte 1

Após um período estudando levantando informações teóricas, iniciou-se a programação em Linguagem C, visando implementar um programa que modelasse a reentrada de um satélite na atmosfera terrestre.

O Modelo de Campo gravitacional consiste na Integração numérica das equações diferenciais usando o método de integração Runge-Kutta. As condições iniciais e o resultado da integração numérica consistem de um vetor de estado  $X$  ( posição e velocidade em coordenadas retangulares) em cada instante.

A Função **derivadas()** (anexo 2) implementa o modelo dinâmico, correspondendo à integração da velocidade e aceleração atuante no satélite. Num primeiro teste considerou-se, apenas, o campo gravitacional, a aproximação de que a Terra é esférica e coeficientes do modelo GEM10. Observou-se nos testes iniciais que ocorria a reentrada do satélite sem a presença de forças dissipativas, o que o feria a condição de “bom senso”. Contudo, ao verificar as condições iniciais usadas, não satisfaziam completamente a condição orbital, ou seja, a energia cinética deve ser coerente com a energia potencial. O objetivo do teste era, até então, verificar as equações e testar o funcionamento do programa.

Com condições iniciais que satisfazem a condições orbitais, considerou-se a altitude igual a 7078160 m, e a velocidade inicial 7504 m/s (calculada através equação 2).

A partir do modelo inicial descrito acima, adicionou-se ao modelo atmosférico as componentes de arrasto, inserindo sua influencia nas equações diferenciais do sistema. Para evitar problemas de condições iniciais, implementou-se a entrada de valores iniciais em Coordenadas Keplerianas, que por definição correspondem a : semi-eixo maior ( $a$ ), excentricidade ( $e$ ), inclinação da orbita com relação ao equador ( $i$ ), ascensão reta do nodo ascendente ( $\Omega$ ), argumento do perigeu ( $\omega$ ), Anomalia Média ( $M$ ). O procedimento de

conversão de coordenadas Keplerianas para coordenadas retangulares é executado pela função **eoX()**(anexo 3). As equações utilizadas no processo de integração podem ser encontradas na referência (6).

Sabendo que se utiliza a Anomalia Excêntrica para a conversão, é necessário utilizar um método numérico de aproximação (Newton Raphson) para obter E a partir de M que é uma coordenada Kepleriana dada como condição inicial. Partindo de uma condição inicial de  $E_0 = M_0$ , a função **fnr()** (anexo 4) calcula o valor de E.

A função **derivadas** também foi modificada, agora, além da ação do campo Gravitacional, o corpo está sujeito a uma força de Arrasto, O Coeficiente de Arrasto e a área Frontal foram considerados como constantes, a velocidade relativa varia em função da posição, a densidade do meio varia em função da altura de acordo com o modelo exponencial de densidade que é utilizado para cada pequeno intervalo de altura ( $h-h_0$ ) visando diminuir as incertezas para os valores da densidade, ou seja, a cada intervalo de altura, um modelo exponencial de variação de densidade é aplicado (função **wertz-PIBIC()**).

### 3.2. Parte 2

Nesta segunda parte, acrescentou-se novos códigos ao modelo antigo para nos fornecer a Ascensão Reta e a Declinação, isso considerando nos cálculos:

- O tempo Sideral e a Data Juliana .
- Diferentes Coeficientes de Arrasto.
- Diferentes condições iniciais.
- Efeitos Sazonais e Tesserais da Terra (função **acegeo()** (anexo 5)), ou seja, a Terra não é mais considerada como uma esfera.

Consideramos também a Ascensão Reta e a declinação (figura 1) como sendo respectivamente longitude e latitude.

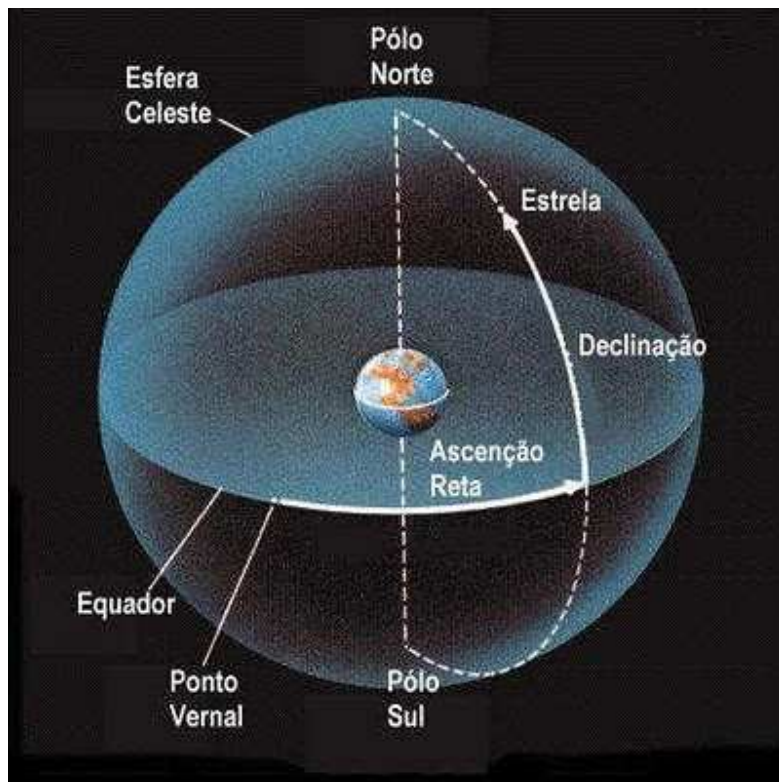


Figura 1: Ascensão reta e declinação

## 4. RESULTADOS E DISCUSSÕES

O programa implementado fornece informações sobre o comportamento do satélite durante o processo de reentrada. A partir de condições iniciais fornecidas, este fornece através de simulações dados sobre a velocidade e a localização deste (latitude, longitude e altitude).

### 4.1. Parte 1

A partir das condições iniciais fornecidas abaixo, realizou-se as simulações no programa implementado. Os dados obtidos estão ilustrados nos gráficos 1 e 2.

$a=6528160$  m,  $e= 0$ ,  $i= 60^\circ$ ,  $\Omega= 120^\circ$ ,  $\omega= 30^\circ$ ,  $M= 0^\circ$ ,  $Cd= 2$ ,  $A= 1$  m<sup>2</sup>, Massa = 250 kg.

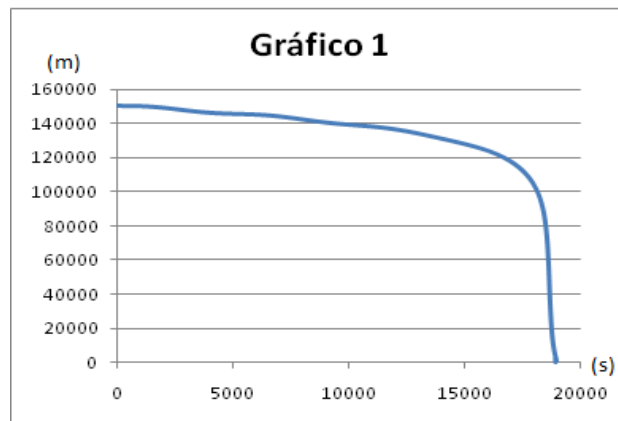


Gráfico 1: Altitude do Satélite em função do tempo

Podemos observar a partir do gráfico 1, a variação da altitude do satélite em relação ao tempo. Nota-se que inicialmente a altitude decresce ligeiramente com o tempo até o instante em que o corpo é “capturado” pela atmosfera. A partir do instante onde inicia-se a captura, a altitude decresce abruptamente, caracterizando a reentrada deste na atmosfera.

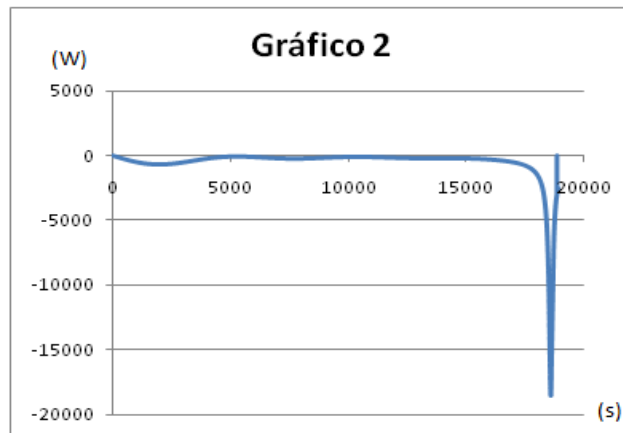


Gráfico 2: Derivada primeira da Energia com relação ao tempo

A partir do gráfico 2, nas oscilações de energia nos 5000 segundos iniciais observa-se que o corpo começa a ser monitorado próximo ao perigeu da órbita, posto que é possível visualizar uma leve variação de energia causada pela ação considerável do arrasto atmosférico devida à sua baixa altitude.

Ao se analisar o período entre 15000 e 20000 segundos, é possível notar o instante em que o corpo é “capturado” pela atmosfera.

#### 4.2. Parte

As simulações foram realizadas considerando as seguintes condições Iniciais:

**$a=6558159.94$  m,  $e= 0.004$ ,  $i= 60^\circ$ ,  $\Omega= 0^\circ$ ,  $\omega= 0^\circ$ ,  $M= 180^\circ$ ,  $A= 1$  m<sup>2</sup>, Massa = 250 kg.**

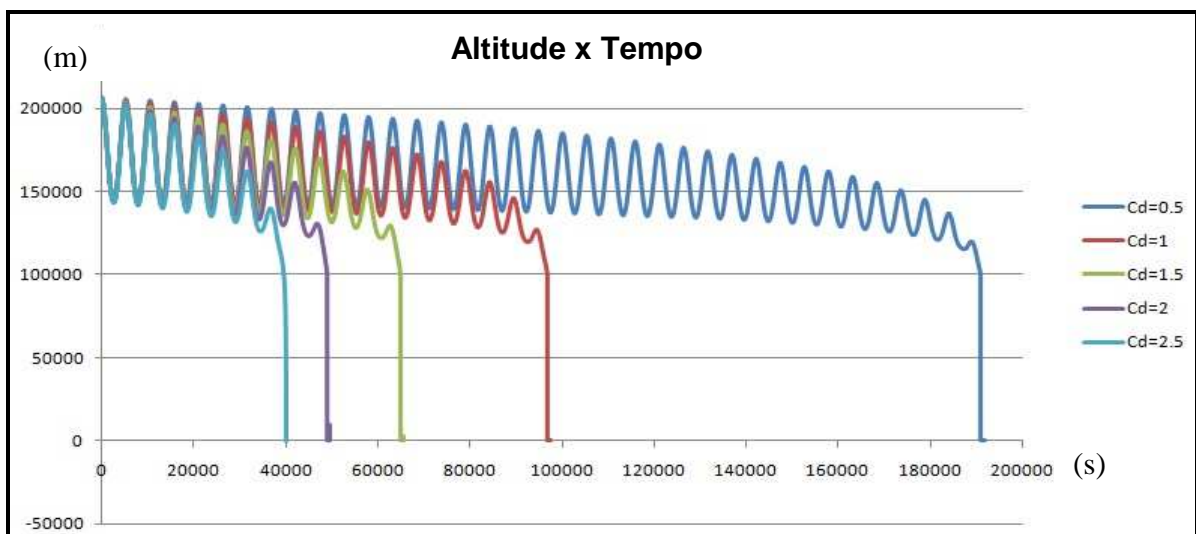
A partir das condições iniciais mencionadas acima, e através das simulações realizadas, obteve-se os resultados mencionados na Tabela 1 e nos gráficos de 3 a 20.

Na **Tabela 1** é possível observar a Influência do coeficiente de arrasto ( $C_d$ ) no tempo que o satélite leva para reentrar ( $t_s$ ), na latitude final ( $lat_f$ ) e na longitude final ( $long_f$ ):

**Tabela 1:** tempo e coordenadas finais em função do Cd

Cd	ts (dias)	latf (°)	longf (°)
2.5	0.463	29.95	289.31
2	0.575	-16.77	218.68
1.5	0.760	-3.44	159.06
1	1.127	7.73	31.33
0.5	2.220	-1.18	346.45

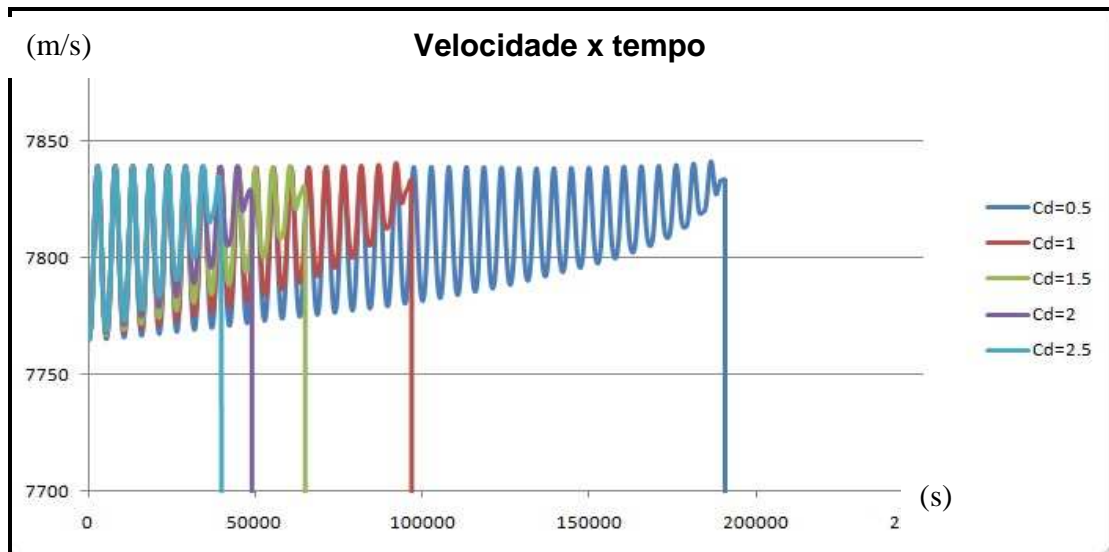
A partir do gráfico 3, é possível observar que antes do corpo ser “capturado” pela atmosfera, as oscilações de altitude se tornam cada vez menores, o que mostra a tendência de circulação de orbita devido ao efeito do arrasto.



**Gráfico 3:** Variação da altitude com relação ao tempo para diferentes Cd

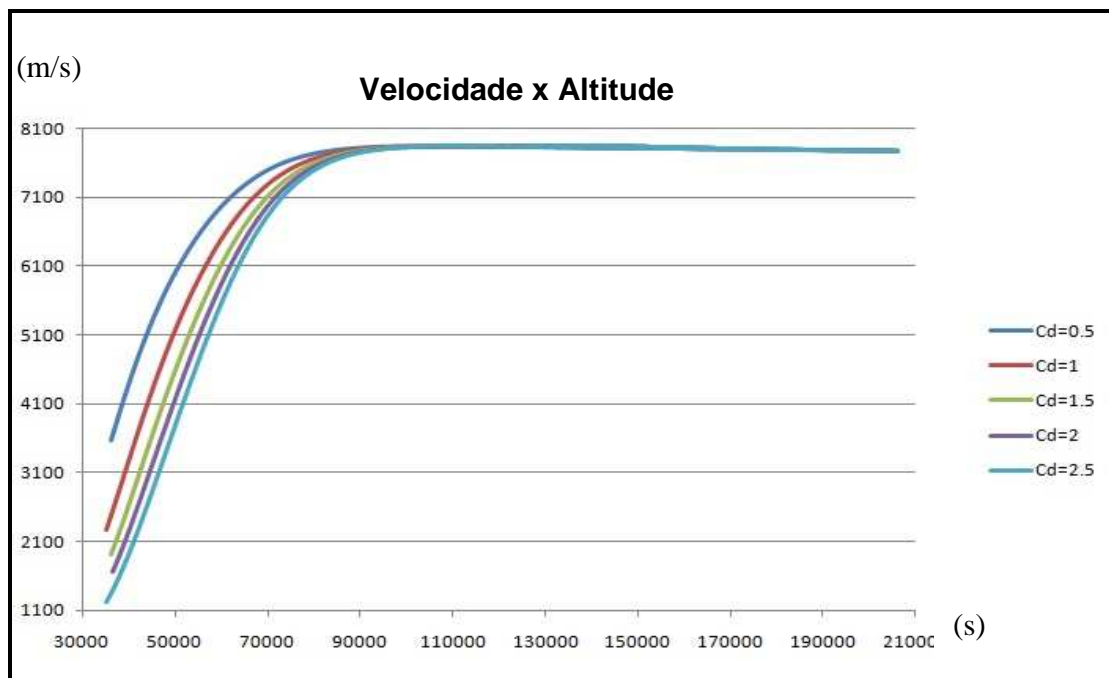
Analisando o gráfico 4 é possível comprovar que o corpo percorre uma trajetória elíptica devido as oscilações de velocidade, observar-se também a diminuição da amplitude de oscilação devido ao efeito do arrasto.





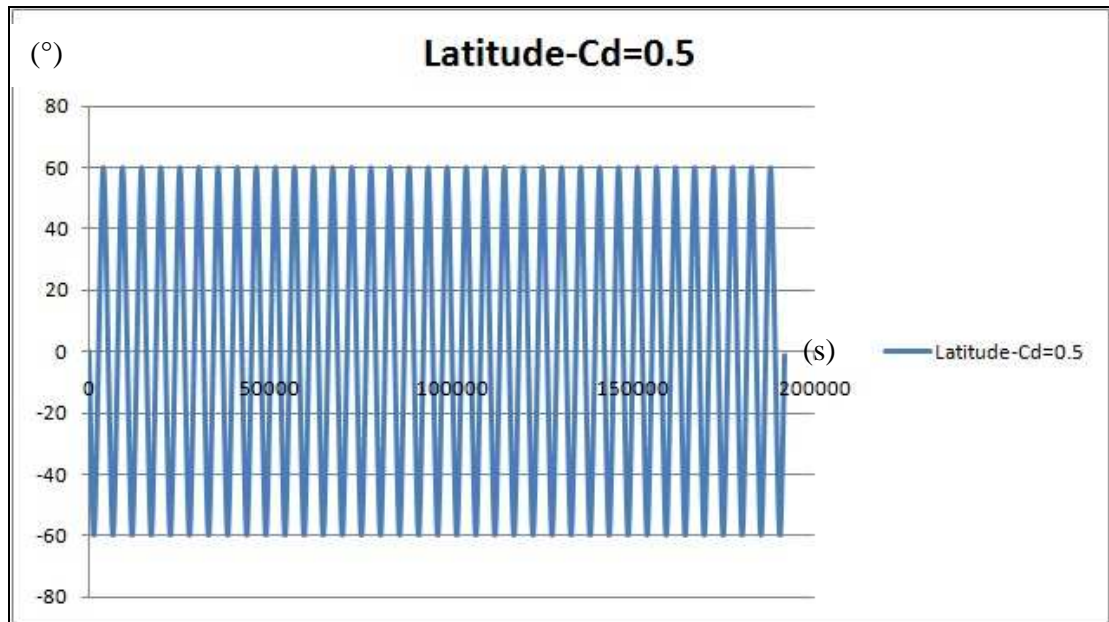
**Gráfico 4:** Variação da velocidade com relação ao tempo para diferentes  $C_d$

Pelo gráfico 5 é possível observar que, antes do corpo ser “capturado”, um pequeno aumento de velocidade devido a perda de altitude, esse aumento só não é maior, porque o arrasto dissipa parte da energia mecânica.

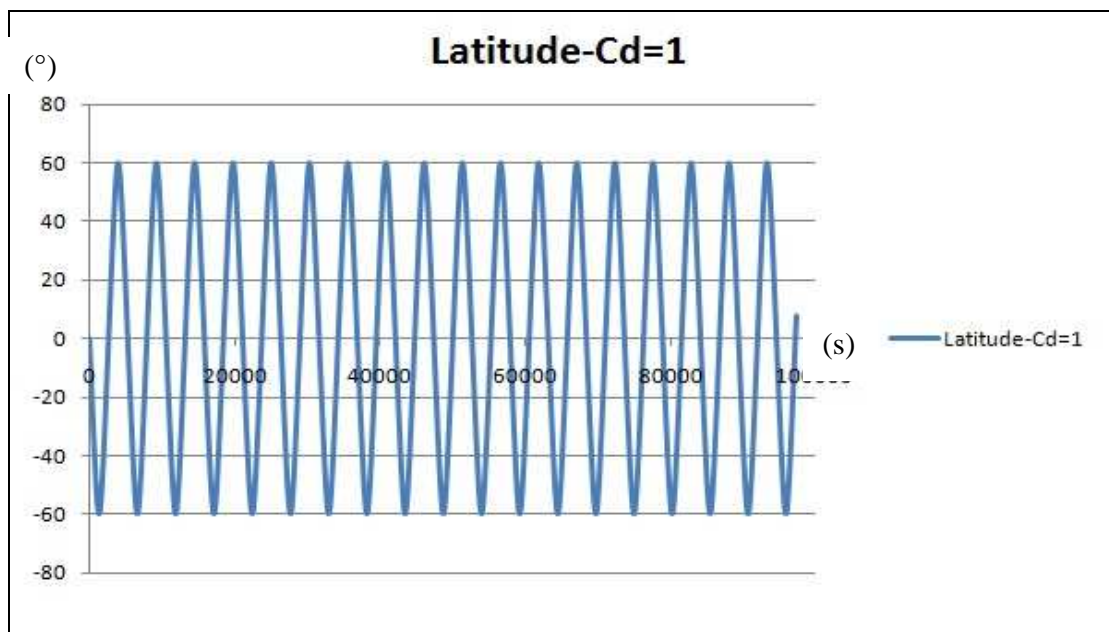


**Gráfico 5:** Variação da velocidade em função da altitude para diferentes  $C_d$

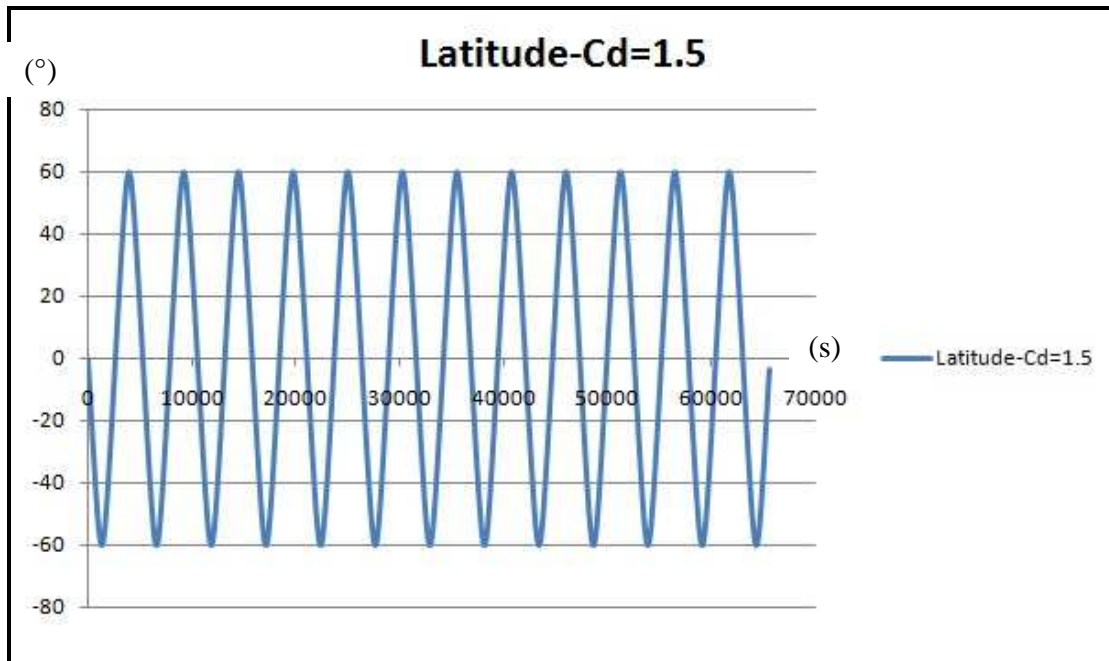
Os gráficos 6, 7, 8,9,10 descrevem a variação da latitude em função do tempo para diferentes  $C_d$ , é possível notar também que a latitude máxima corresponde a inclinação da órbita ( $60^\circ$ ).



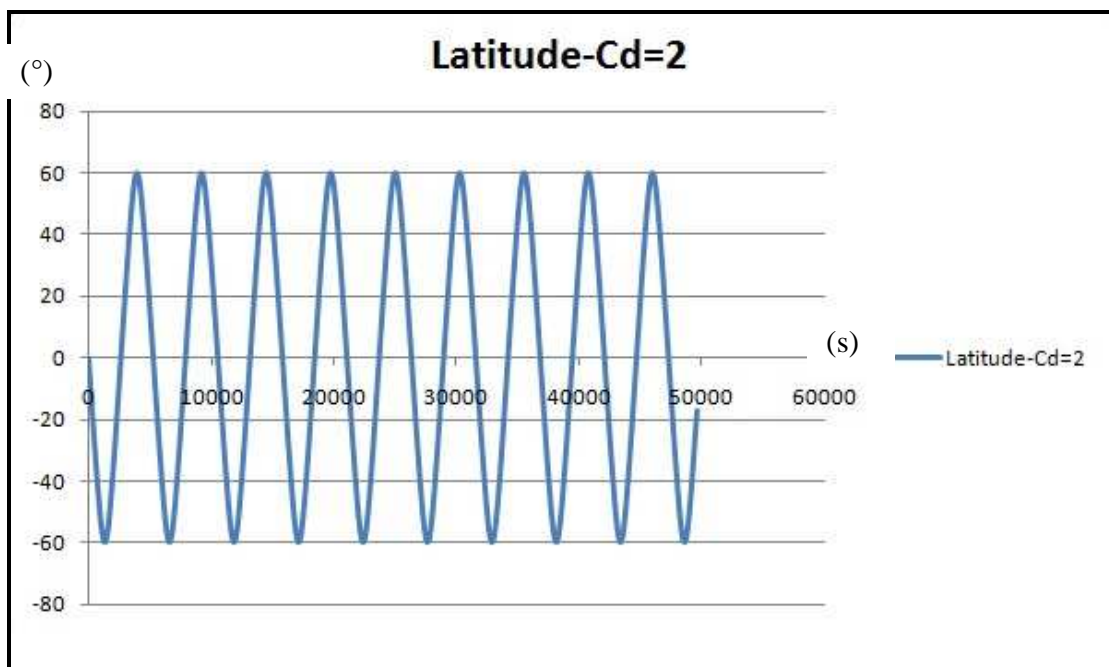
**Gráfico 6:** Variação da latitude em função do tempo, para  $C_d=0.5$



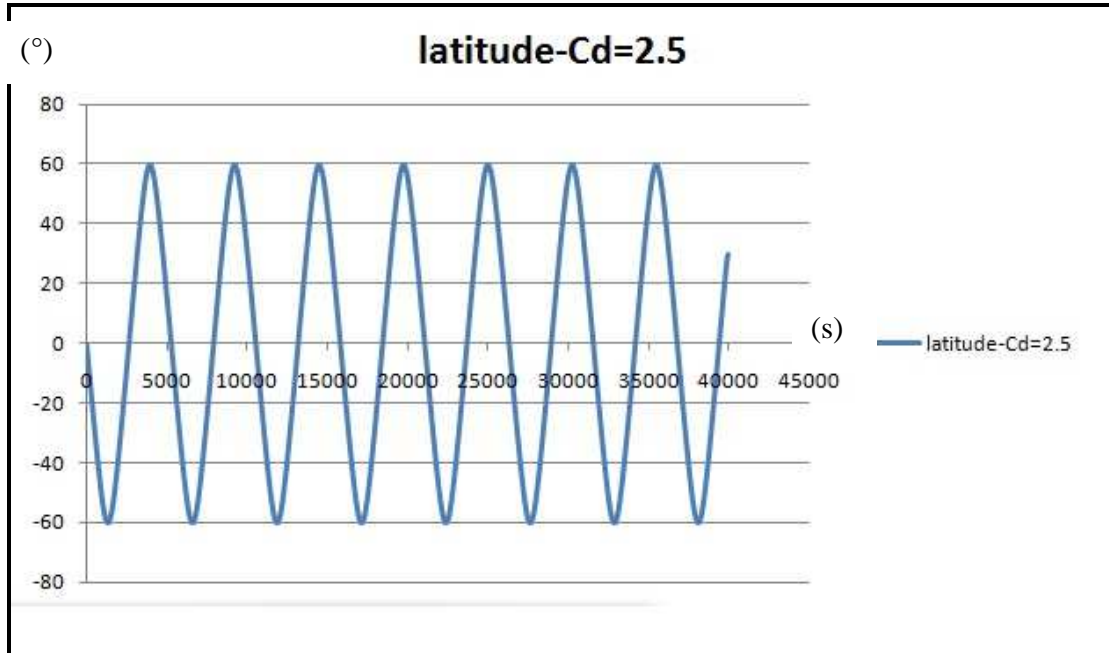
**Gráfico 7:** Variação da latitude em função do tempo, para  $C_d=1$



**Gráfico 8:** Variação da latitude em função do tempo, para  $Cd=1.5$

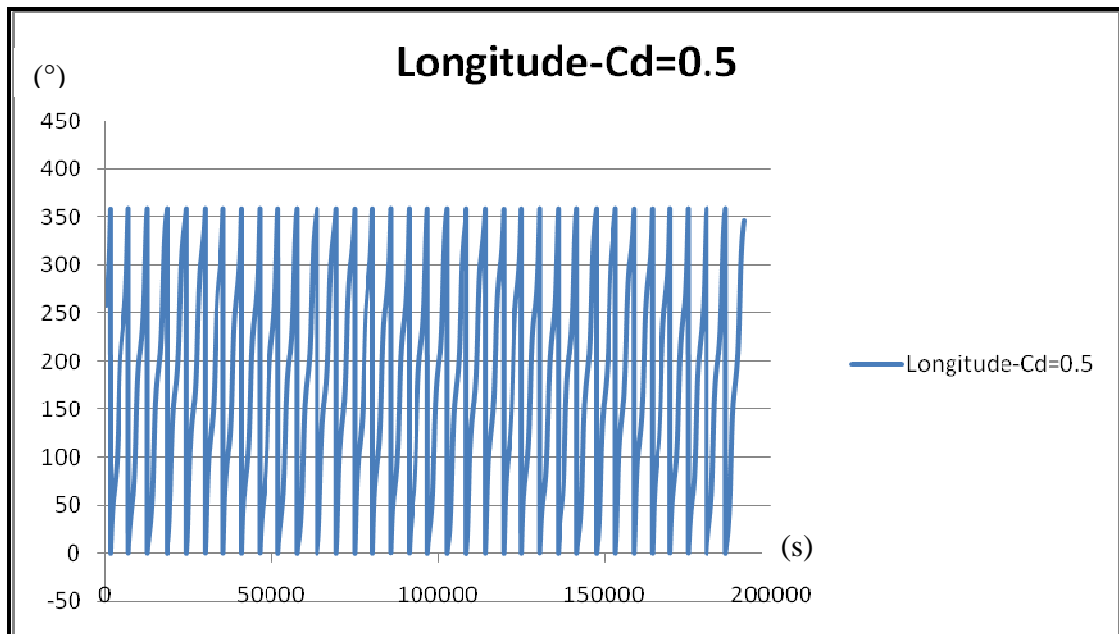


**Gráfico 9:** Variação da latitude em função do tempo, para  $Cd=2$

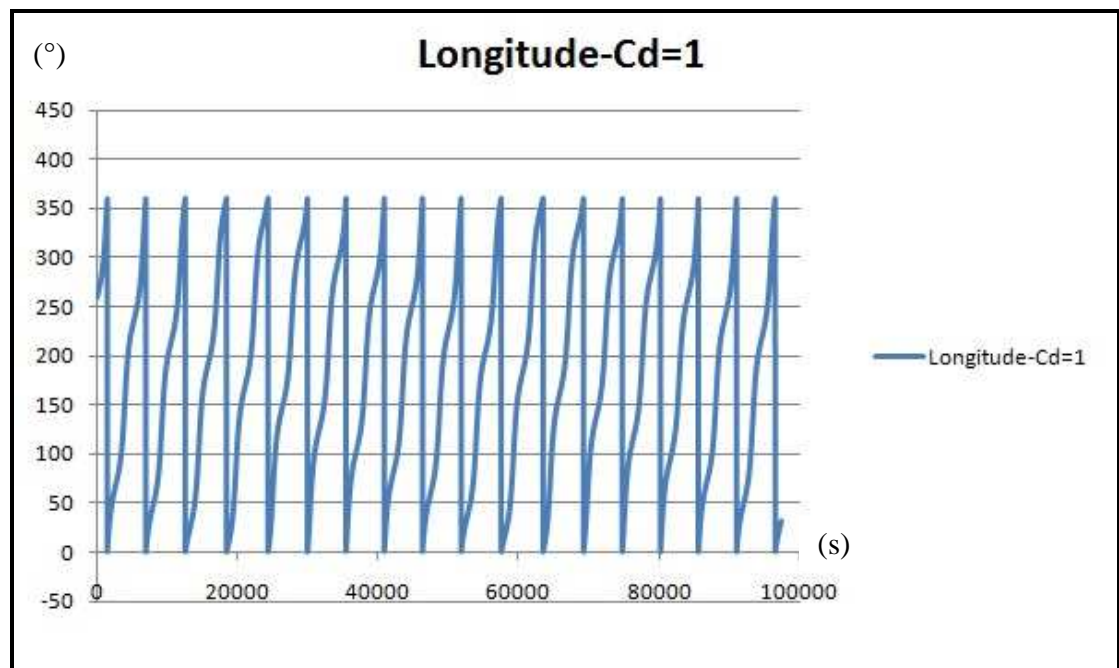


**Gráfico 10:** Variação da latitude em função do tempo, para Cd=2.5

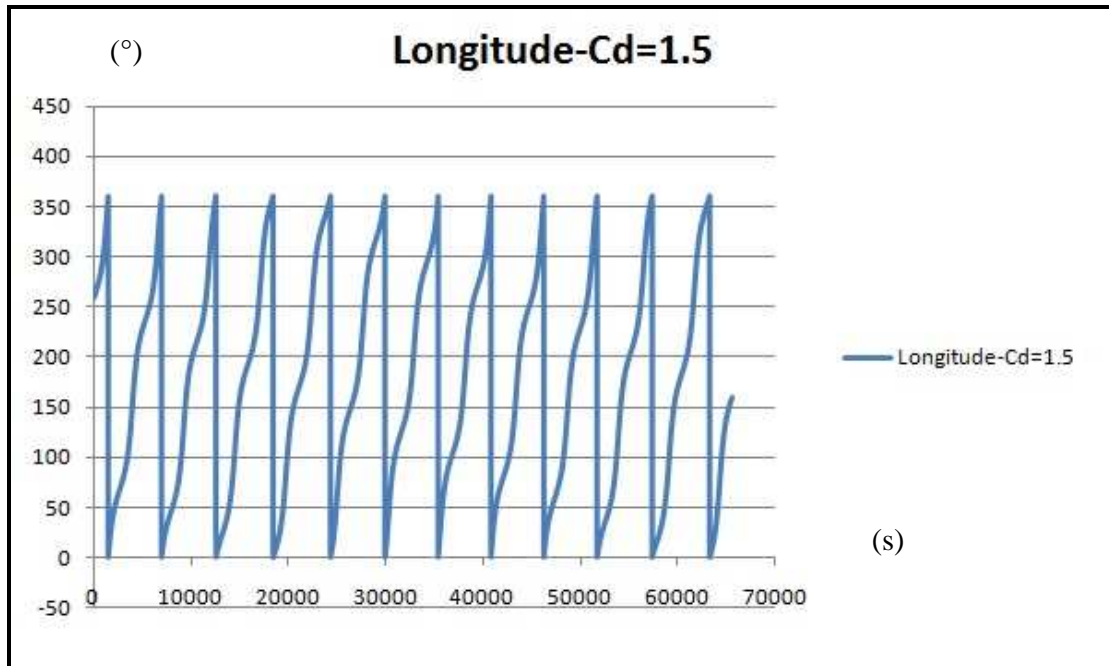
Os Gráficos 11,12,13,14,15 mostram a variação da longitude em função do tempo para diferentes valores de Cd. A latitude e a longitude são utilizadas para facilitar a visualização da posição do satélite.



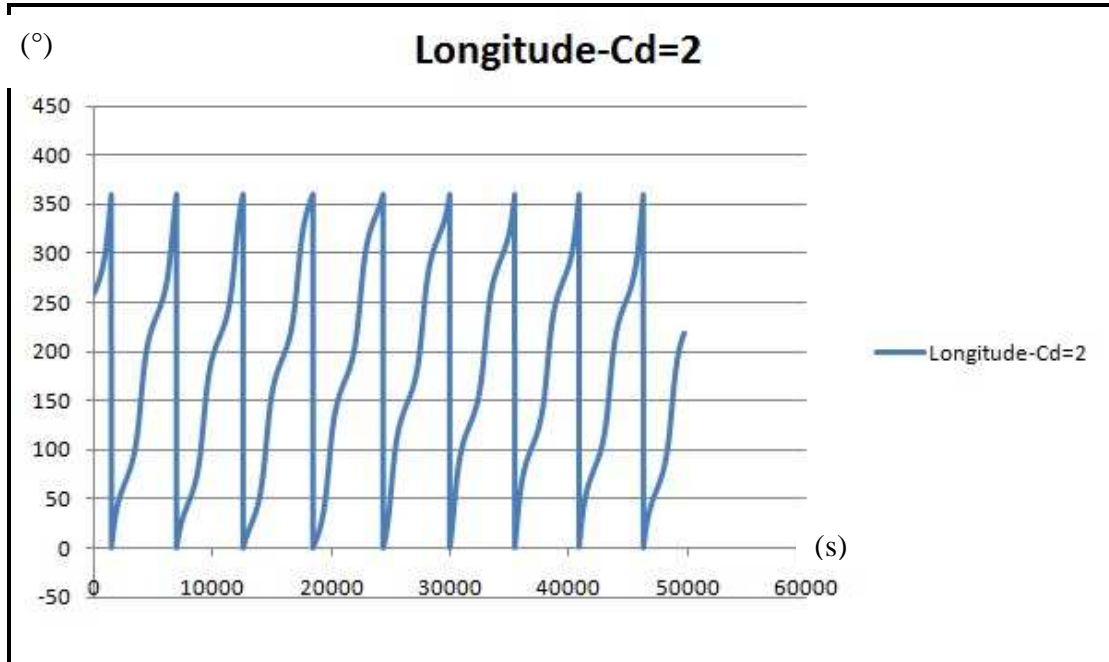
**Gráfico 11:** Variação da longitude em função do tempo, para  $C_d=0.5$



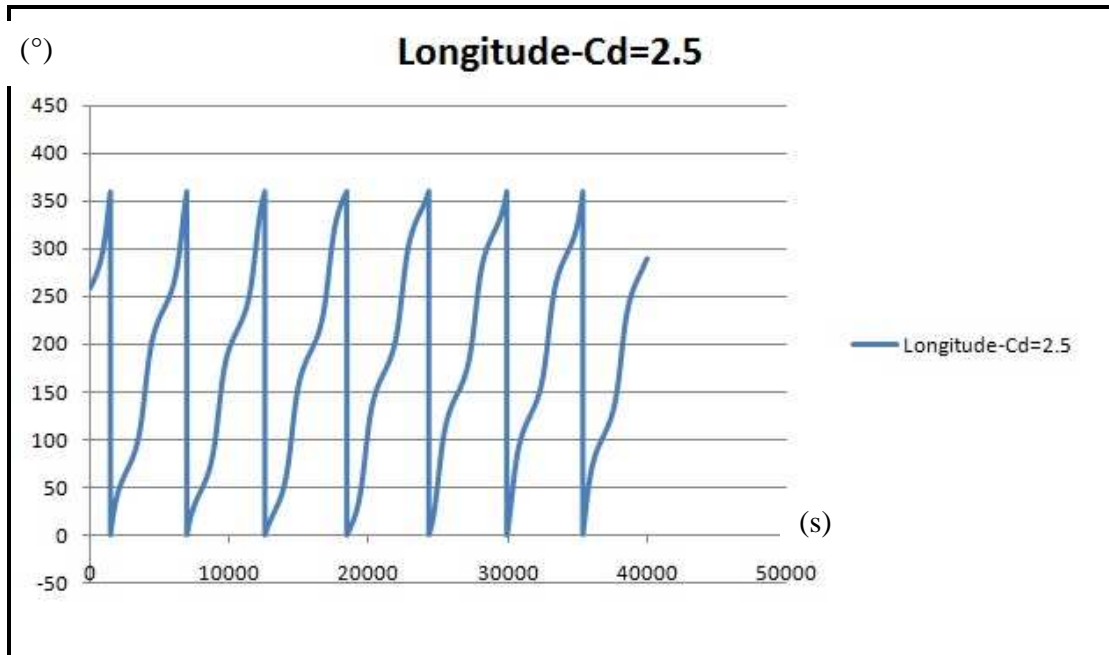
**Gráfico 12:** Variação da longitude em função do tempo, para  $C_d=1$



**Gráfico 13:** Variação da longitude em função do tempo, para  $Cd=1.5$

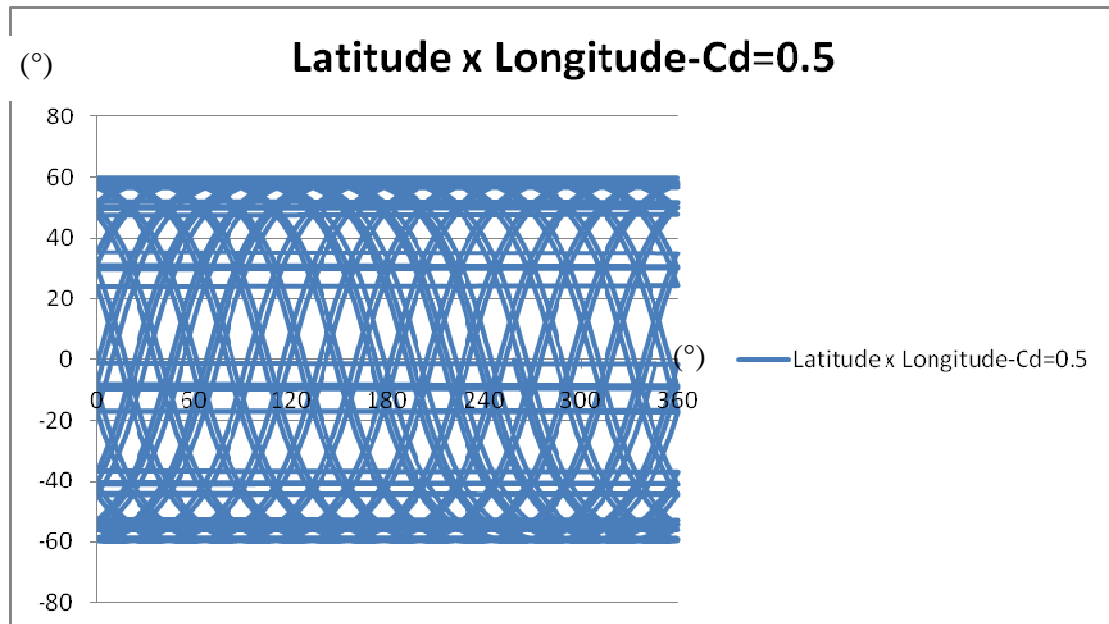


**Gráfico 14:** Variação da longitude em função do tempo, para  $Cd=2$

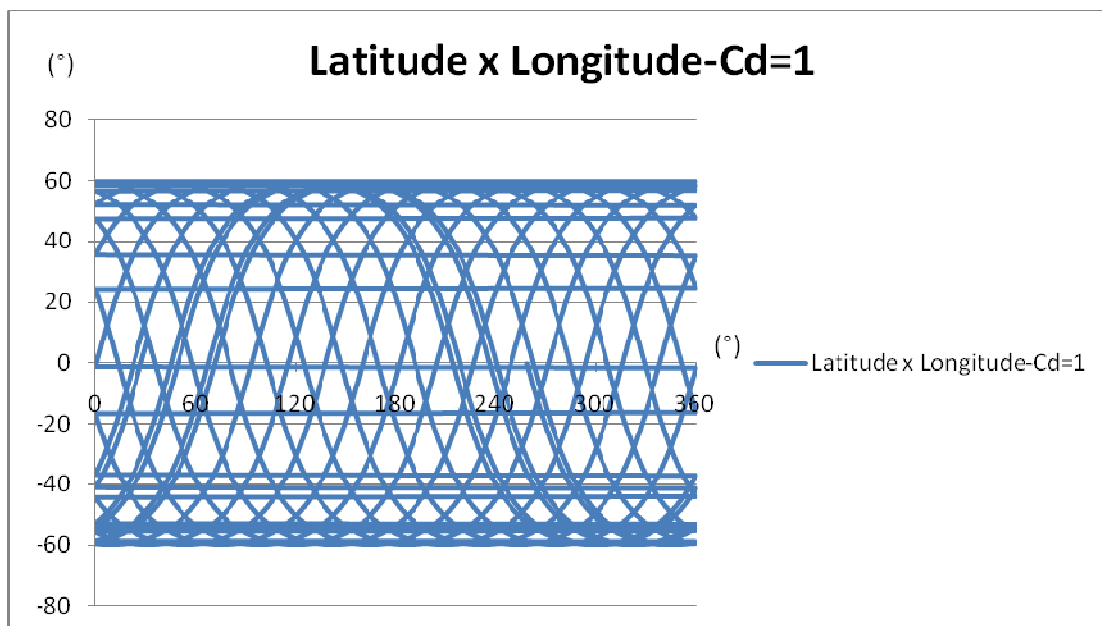


**Gráfico 15:** Variação da longitude em função do tempo, para  $C_d=2.5$

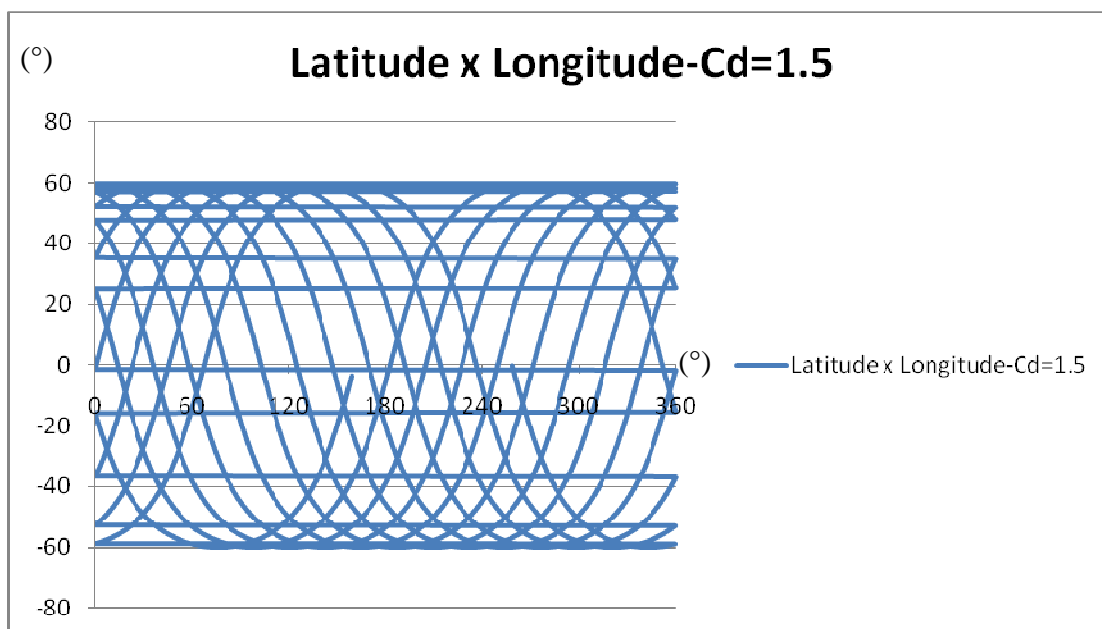
Os Gráficos 16, 17, 18, 19, 20 nos mostram a variação da latitude em função da longitude para diferentes valores de  $C_d$ .



**Gráfico 16:** Variação da latitude em função da longitude, para  $C_d=0.5$

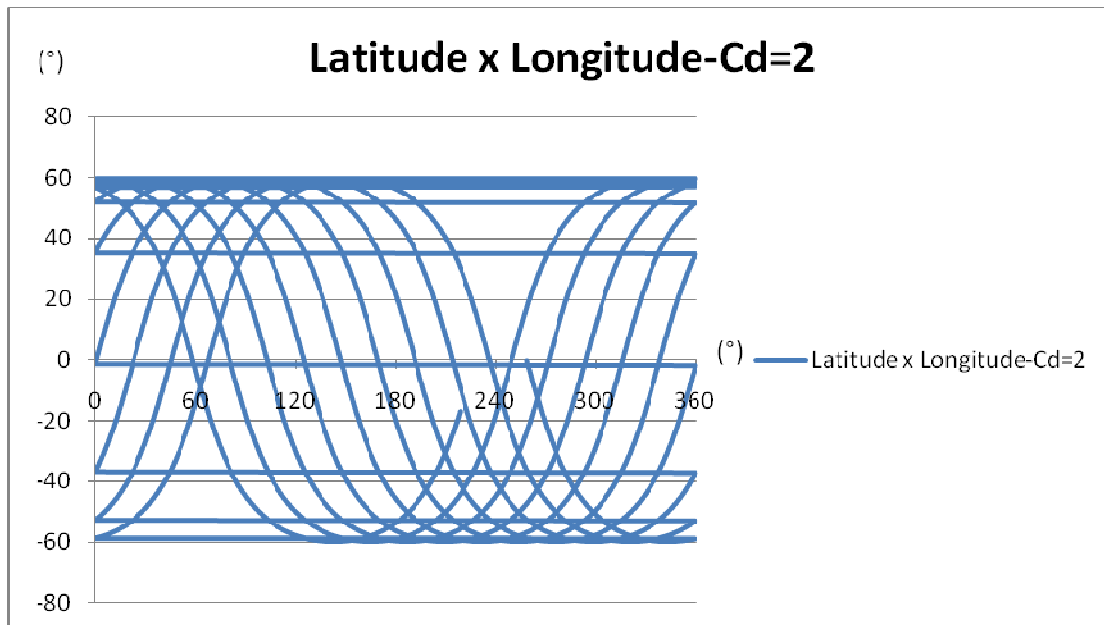


**Gráfico 17:** Variação da latitude em função da longitude, para  $Cd=1$

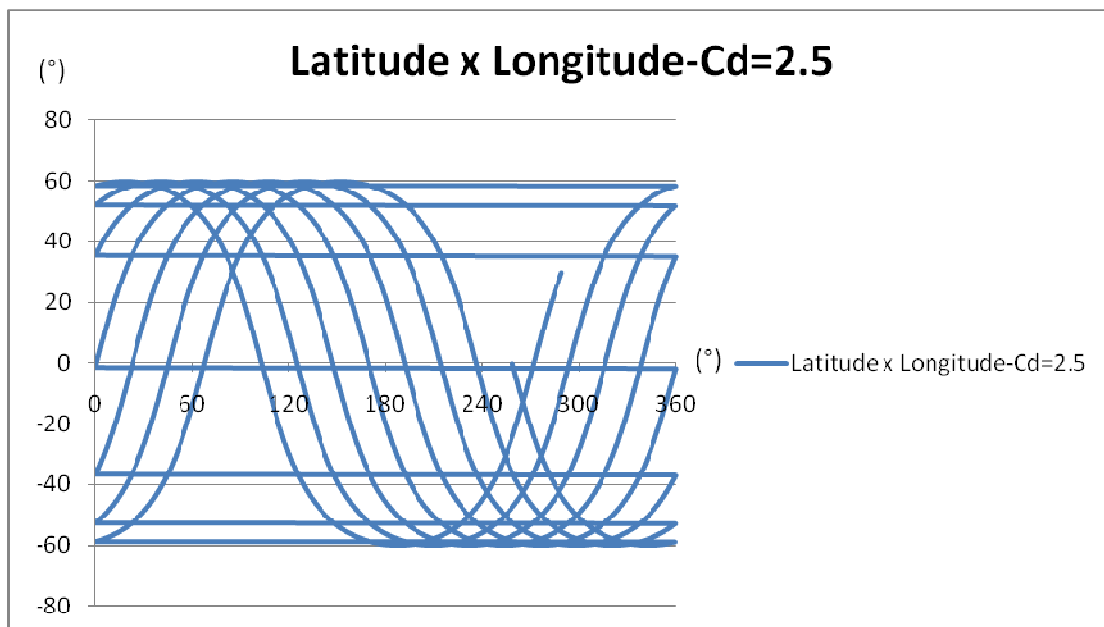


**Gráfico 18:** Variação da latitude em função da longitude, para  $Cd=1.5$





**Gráfico 19:** Variação da latitude em função da longitude, para Cd=2



**Gráfico 20:** Variação da latitude em função da longitude, para Cd=2.5

## 5. CONCLUSÃO

Neste trabalho foi projetado e implementado um programa que modela a reentrada de um satélite considerando diferentes coeficientes de arrasto ( $C_d$ ). Através de dados obtidos a partir de simulações e convertidos em gráficos, observa-se a influência direta do  $C_d$  no tempo de reentrada e nas coordenadas finais do satélite.

Considerando os coeficientes sazonais e tesserais da Terra iguais a 6 foi possível observar também que um corpo com 250 kg de massa,  $1 \text{ m}^2$  de área em uma órbita elíptica de excentricidade e inclinação respectivamente iguais a 0.004 e  $60^\circ$ , partindo de uma altura de 206232.6 m, pode levar de 12 a 53 horas para reentrar quando o  $C_d$  varia de 0.5 a 2.5. Essa grande incerteza no tempo nos mostra uma grande incerteza de posição, demonstrando assim importância do estudo do  $C_d$  na reentrada de um satélite. Posto que, ao efetuar a reentrada de um satélite ou de estruturas espaciais é necessário assegurar a posição que este vai atingir o solo com precisão para poder efetuar o resgate destes, garantindo que não atinjam pessoas, causem destruição ou acabem sendo perdidos, como por exemplo se reentrarem na atmosfera e “estacionarem” num oceano.

O algoritmo desenvolvido possibilita observarmos o comportamento do satélite, fornecendo a velocidade, a latitude e a longitude deste a partir de condições iniciais explicitadas nos resultados e discussões. Os dados de monitoramento fornecidos pelas simulações são imprescindíveis para a atividade de reentrada, sendo um fator decisivo para a escolha dos melhores parâmetros e condições para esta ser realizada.

## 6 REFERÊNCIAS BIBLIOGRÁFICAS

[1] **INTRODUCTION TO CELESTIAL MECHANICS** . Londres: PUBLISHING COMPANY, INC, 1963. S.W.McCUSKEY, ADDISON-WESLEY .

[2] **CÁLCULO NUMÉRICO**. Rio de Janeiro: Editora LTC, 2007. Franco, Neide Maria Bertoldi;

[3] **FUNDAMENTOS DE FÍSICA I**. 6º Edição: Editora LTC 2002. HALLIDAY RESNICK WALKER

[4] Figura 1- disponível em :

[http://www.google.com.br/imgres?imgurl=http://astro.if.ufrgs.br/ardec.jpg&imgrefurl=http://www.cempem.fae.unicamp.br/lapemmec/cursos/el654/2001/juliana\\_e\\_vera/EL654/pagesistemascordenadas.htm&usg=\\_\\_An9jsJTrJi0snM9iysLxOzLhdM=&h=420&w=420&sz=24&hl=ptBR&start=93&sig2=WzsWFwRhj7XcUEJSWzIXYA&zoom=1&tbnid=ToqwS5Lm69WG8M:&tbnh=133&tbnw=145&ei=6MsdTpmVBISNsALR3rijCA&prev=/search%3Fq%3Dascens%25C3%25A3o%2Breta%2Be%2Bdeclina%25C3%25A7%25C3%25A3o%6hl%3DptBR%26sa%3DX%26pwst%3D1%26biw%3D1280%26bih%3D602%26tbn%3Disch&itbs=1&iact=hc&vpx=180&vpy=270&dur=2995&hovh=225&hovw=225&tx=128&ty=83&page=6&ndsp=18&ved=1t:429,r:6,s:93](http://www.google.com.br/imgres?imgurl=http://astro.if.ufrgs.br/ardec.jpg&imgrefurl=http://www.cempem.fae.unicamp.br/lapemmec/cursos/el654/2001/juliana_e_vera/EL654/pagesistemascordenadas.htm&usg=__An9jsJTrJi0snM9iysLxOzLhdM=&h=420&w=420&sz=24&hl=ptBR&start=93&sig2=WzsWFwRhj7XcUEJSWzIXYA&zoom=1&tbnid=ToqwS5Lm69WG8M:&tbnh=133&tbnw=145&ei=6MsdTpmVBISNsALR3rijCA&prev=/search%3Fq%3Dascens%25C3%25A3o%2Breta%2Be%2Bdeclina%25C3%25A7%25C3%25A3o%6hl%3DptBR%26sa%3DX%26pwst%3D1%26biw%3D1280%26bih%3D602%26tbn%3Disch&itbs=1&iact=hc&vpx=180&vpy=270&dur=2995&hovh=225&hovw=225&tx=128&ty=83&page=6&ndsp=18&ved=1t:429,r:6,s:93)

[5] [http://www.space.com/sciencefiction/phenomena/south\\_african\\_spheres\\_000502.html](http://www.space.com/sciencefiction/phenomena/south_african_spheres_000502.html)

[6] [[www.2dem.inpe.br/val/cursosinpe/apostilas/ada032](http://www.2dem.inpe.br/val/cursosinpe/apostilas/ada032)].

## ANEXOS

### ANEXO 1: Função Wertz-PIBIC ().

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define LINHAS 50
struct _WERTZDAT_ {
    int wertzinit;
    double hgeo[LINHAS];
    double rho[LINHAS];
} wertzdat={0,{LINHAS*0},{LINHAS*0}};

/* Entrada: h - altitude em metros */
/* Saida: densidade em kg/m^3 */
double wertz(double h)
{

#define WERTZFILE "wertz.dat"
    double hgeof, tempf, rhof, dif;
    double difin,difhn;
    FILE *fpwertz;
    int i,n;

    if (!wertzdat.wertzinit) {
        /* Abrir o arquivo com a tabela de altitude e densidade*/
        fpwertz=fopen(WERTZFILE,"r");
        if (fpwertz == NULL) {
            printf("Arquivo %s não encontrado\n",WERTZFILE);

            exit(0);
        }
        i=0;
        /* Leitura, armazenamento e contagem do numero de registros da
        tabela */
        while (fscanf(fpwertz,"%lf %lf %lf\n",&hgeof,&tempf,&rhof) == 3)
        {
            wertzdat.hgeo[i]=hgeof;
            wertzdat.rho[i]=rhof;
            i++;
        }
        /* Fechamento do arquivo e armazenamento do numero de registros
```

```

        em variável global */
        fclose(fpwertz);
        wertzdat.wertzinit=i;
    }
    /* Deveria ser uma altitude geodésica, mas consideraremos uma
    altitude inercial */
    hgeof=h/1000;
    /* Busca a faixa onde se encontra a altitude informada */
    i=0;
    while ( (hgeof > wertzdat.hgeo[i]) &
            (i<wertzdat.wertzinit) ) {
        i++;
    }
    n=i-1; /* reduz para o anterior; */
    if (n<0) n=0; /* ajusta para zero se negativo */
    /* Verificando a altitude Z */
    if ( hgeof == wertzdat.hgeo[n] ) {
        return (wertzdat.rho[n]);
    }
    /* Interpolando exponencialmente */
    difin = wertzdat.hgeo[i]-wertzdat.hgeo[n];
    difhn = hgeof-wertzdat.hgeo[n];
    /* Interpolando a Densidade */
    tempf=log(wertzdat.rho[i]/wertzdat.rho[n])/difin;
    /* Retornando o valor da densidade */
    return (wertzdat.rho[n]*exp(tempf*difhn));
}
#ifdef PROGTEST
int main(int argc, char *argv[])
{
    double alt,alt0,dens;

    do {
        printf("Entre com altitude [m]: ");
        scanf("%lf",&alt);
        if (alt < 0) break;
        alt0=alt;
        dens=wertz(alt0);
        printf("Em %lf [m] a densidade é %lg [kg/m³]\n",alt0,dens);
    } while(1);
    return 0;
}
#endif

```

## ANEXO 2: Função derivadas

```
#include <math.h>
#include <stdio.h>
#include <stdlib.h>

int djm(int dia, int mes, int ano);

double gst(int dj, double t);

void acegeo(int nz, int nt, double tsg, double *xi, double *ac);

extern double wertz(double h);

extern void MvoDensi(int modelo, int jdm, double ts, double tsid, double *pos, double
                    *dens, double *tlocal, double *wind);

typedef struct _PARAMREENTRA_ {
    int hz;
    int ht;
    double Cd;
} parstru;

void derivadas(double *t, double *x, double *xp, void *p)
{
    /* vetor X
    0 - posicao em X
    1          posicao em Y
    2          posicao em Z
    3          velocidade em X
    4          velocidade em Y
    5          velocidade em Z

    vetor XP
    0          velocidade em X
    1          velocidade em Y
    2          velocidade em Z
    3          aceleracao em X
    4          aceleracao em Y
    5          acele... Z
```

```
*/
```

```
double d2,d,d3;

double A= 1; /* area de um disco*/
double m=250;
double K=3.9860064e+14L;
double densid0= 1.225;
double v;
double posi;
double Ca;
double dens;
double teta=7.29211585468195e-5L;
double m2=m+m;
double Vat[3],Vr[3];
int dj1;
double tsid;
double ac[3];
double ts;
double tlocal;
double wind[3];
int modelo=6;

parstru *p1;
p1=(parstru *) p;

xp[0]=x[3];
xp[1]=x[4];
xp[2]=x[5];
Vat[0]=-teta*x[1];
Vat[1]=teta*x[0];
Vat[2]=0;
/* A velocidade relativa ? a velocidade do corpo - a velocidade
da atmosfera */
Vr[0]=x[3]-Vat[0];
Vr[1]=x[4]-Vat[1];
Vr[2]=x[5];
v=sqrt(Vr[0]*Vr[0]+Vr[1]*Vr[1]+Vr[2]*Vr[2]);

d2=x[0]*x[0]+x[1]*x[1]+x[2]*x[2];
d=sqrt(d2);
d3=d*d2;
/* posi=(sqrt(x[0]*x[0]+x[1]*x[1]+x[2]*x[2])-6378160)/1000;
```

```

    dens=densid0*exp(-Kn*posi);
    dens=490326869239.88*pow(posi,-8.92);
*/

    dj1=(int) (*t/86400); /* parte inteira do tempo*/
    ts = (*t/86400 - dj1)*86400;
    tsid= gst(dj1, ts);
    acegeo(p1->hz, p1->ht, tsid, x, ac);

#ifdef WERTZPIBIC
    posi=sqrt(x[0]*x[0]+x[1]*x[1]+x[2]*x[2])-6378160;
    dens= wertz(posi);
#else
    MvoDensi( modelo, dj1, ts, tsid, x, &dens, &tlocal, wind);
#endif

    Ca= A*p1->Cd*v*dens/m2;

    xp[3]= ac[0] /*- K * x[0]/ d3 */ -Ca*Vr[0] ; /*acelera??o em x */
    xp[4]= ac[1] /*- K * x[1]/ d3 */ -Ca*Vr[1] ; /* acelera??o em Y */
    xp[5]= ac[2]/*- K * x[2]/ d3*/ -Ca*Vr[2] ; /* acelera??o em Z */

    return;
}

```



### ANEXO 3: Função eox().

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
extern double NewIKepler(double d1, /* a variável não precisa possuir valor, é um
protótipo que corresponderá a excentricidade */
double d2);/* protótipo da anomalia média*/

void eox(double cord[6], double saida[6]){/* alterando o valor da saida, altera o valor de X
= a, e, i, ascensão reta , Argumento do perigeu, M*/
double d3;
double coordorbit[6];
double cteorb;

d3= NewIKepler(cord[1],cord[5]); /* nos fornece a Anomalia excentrica, cord[1]=
excentricidade, cord [5]= anomalia média*/
printf("Sai do kepler EM EOX: %lf",d3);
system("pause");
cteorb= sqrt(
3.9860064e+14L/
(cord[0]*(1-cord[1]*cos(d3))*(1-cord[1]*cos(d3))) );

coordorbit[0]= cord[0]*(cos(d3)-cord[1]); /* Equação para transformar de coordenadas
Keplerianas para coordenadas no Plano Orbital */
coordorbit[1]= cord[0]*sin(d3)* sqrt(1-cord[1]*cord[1]);
coordorbit[2]=0;
coordorbit[3]= -cteorb*sin(d3);
coordorbit[4]= cteorb*cos(d3)*sqrt(1-(cord[1]*cord[1]));
coordorbit[5]=0;

saida[0]= coordorbit[0]*cos(cord[3])*cos(cord[4])-
coordorbit[0]*cos(cord[2])*sin(cord[4])*sin(cord[3])-
coordorbit[1]*cos(cord[3])*sin(cord[4])-
coordorbit[1]*cos(cord[2])*cos(cord[4])*sin(cord[3])+
coordorbit[2]*sin(cord[2])*sin(cord[3]);
saida[1]= coordorbit[0]*cos(cord[4])*sin(cord[3])+
coordorbit[0]*cos(cord[2])*sin(cord[4])*cos(cord[3])+
coordorbit[1]*cos(cord[2])*cos(cord[4])*cos(cord[3])-
coordorbit[1]*sin(cord[4])*sin(cord[3])-
coordorbit[2]*sin(cord[2])*cos(cord[3]);
saida[2]= coordorbit[0]*sin(cord[2])*sin(cord[4])+
coordorbit[1]*sin(cord[2])*cos(cord[4])+ coordorbit[2]*cos(cord[2]);
```

```

saida[3]=
    coordorbit[3]*cos(cord[3])*cos(cord[4])-
    coordorbit[3]*cos(cord[2])*sin(cord[4])*sin(cord[3])-
    coordorbit[4]*cos(cord[3])*sin(cord[4])-
    coordorbit[4]*cos(cord[2])*cos(cord[4])*sin(cord[3])+
    coordorbit[5]*sin(cord[2])*sin(cord[3]);
saida[4]=
    coordorbit[3]*cos(cord[4])*sin(cord[3])+
    coordorbit[3]*cos(cord[2])*sin(cord[4])*cos(cord[3])+
    coordorbit[4]*cos(cord[2])*cos(cord[4])*cos(cord[3])-
    coordorbit[4]*sin(cord[4])*sin(cord[3])-
    coordorbit[5]*sin(cord[2])*cos(cord[3]);
saida[5]=
    coordorbit[3]*sin(cord[2])*sin(cord[4])+
    coordorbit[4]*sin(cord[2])*cos(cord[4])+ coordorbit[5]*cos(cord[2]);
/* equações de rotação, para transformar coordenadas orbitais em coordenadas cartesianas
*/
printf("Saindo do eox\n %lf \n %lf \n %lf \n %lf \n %lf \n %lf \n %lf
\n",saida[0],saida[1],saida[2],saida[3],saida[4],saida[5]);
system("pause");
return;
}

```

#### ANEXO 4: Função fnr() .

```
#include <stdio.h>
#include <math.h>

double NewIKepler(double exc, /* Anomalia Media em Radianos */
                  double AM) /* Excentricidade */
/* A função retorna a Anom. Exc em radianos */
{
    double E, En;
    double d3,M,ex;
    int n;
#define PIV2 M_PI+M_PI

    M=AM;
    ex=exc;
    n=10;
    E=M;

    do {
        d3=(E-ex*sin(E)-M)/(1-ex*cos(E));
        En= E - d3;
        if (fabs(d3) < 5.e-10) break;
        E=En;
        n--;
        printf("numero: %d\n",n);
    } while (n); /* qual o significado do 1 ?*/
    printf("Saindo do kepler: anom. Exc: %lf\n",En);
    return En;
}
```

## ANEXO 5: Função acegeo()

```
void acegeo(int nz, int nt, double tsg, double *xi, double *ac)
{
/*
c-----
c
c purpose : this routine manages the "acezon" and "acetes"
c           routines calling][ and computes the inertial
c           acceleration due to geopotential effects
c
c input  :
c         nz][nt....order and degree of the zonal and
c           tesseral spherical harmonic coefficients][
c           up to 30.
c
c         tsg....greenwich sidereal time][ in radians.
c
c         xi....3 dimensional satellite position vector
c           in inertial coordinate system][ in meters.
c
c output :
c         act....vector of satellite's accelerations in
c           inertial system (m/s**2)][ dimension 3.
c
c helio/valder/valdemir ** inpe -dmc/ddo 10-82
c
c
Esta rotina controla "acezon" e "Acetes"
rotinas de chamar [] e calcula a aceleração inercial devido à entrada de efeitos geopotencial:
    nz] NT [... a ordem e grau do zonal e
           tesseral coeficientes harmônicos esféricos []
           até 30.

           greenwich tsg .... sidereal tempo [] em radianos.

           xi .... 3 vetor posição tridimensional de satélite
           em coordenadas inercial [] sistema em metros.

saída:
vetor de acelerações agir .... satélite em

           sistema inercial (m / s ** 2) A dimensão [3].
c-----
c*/
```

```

double acz[3],act[3];
double ct,st,x,y,z;
double xg,yg,zg,xyq,rgq;
double acdc,ac1,ac2,ac3;
double cssc,ccss;

ct = cos(tsg);
st = sin(tsg);
x = xi[0];
y = xi[1];
z = xi[2];
xg = ct*x + st*y;
yg = -st*x + ct*y;
zg = z;
xyq = xg*xg + yg*yg;
rgq = xyq + zg*zg;
Ctegem.rg = sqrt(rgq);
Ctegem.sf = zg/Ctegem.rg;
Ctegem.cf = sqrt(xyq/rgq);
xyq = sqrt(xyq);
Ctegem.sl = yg/xyq;
Ctegem.cl = xg/xyq;
acdc = Earth.gm/rgq;

if (nz == 0) {
    acz[0] = 0;
    acz[1] = 0;
    acz[2] = 0;
} else {
    acezon(nz,acz);
}

if (nt == 0) {
    act[0] = 0;
    act[1] = 0;
    act[2] = 0;
} else {
    acetes(nt,act);
}

ac1 = acdc*( 1.e-6*(acz[0] + act[0]) - 1 );
acdc=acdc*1.e-6;
ac2 = acdc*( acz[1] + act[1] );
ac3 = acdc*( acz[2] + act[2] );

ccss = ct*Ctegem.cl - st*Ctegem.sl;

```

```
cssc = ct*Ctegem.sl + st*Ctegem.cl;
```

```
ac[0] = csss*Ctegem.cf*ac1 - cssc*ac2 - csss*Ctegem.sf*ac3;
```

```
ac[1] = cssc*Ctegem.cf*ac1 + csss*ac2 - cssc*Ctegem.sf*ac3;
```

```
ac[2] = Ctegem.sf * ac1 + Ctegem.cf * ac3;
```

```
return;
```

```
}
```