

PROGRAMA DE COMPUTADOR PARA SIMULAR A TRAJETÓRIA DE ESPAÇOMODELOS

Jeremy Paraná Blavier¹; Antonio Carlos Foltran²

RESUMO

O programa de computador descrito neste relatório destina-se à previsão de detalhes da trajetória de espaçomodelos durante o voo. Sua concepção permite auxiliar os membros do Grupo de Estudos em Aerodinâmica de Veículos Autopropelidos GREAVE, do curso de Engenharia Mecânica da UP, em seus estudos ou em projetos para competições entre universidades. O programa utiliza o Método de Euler para resolver a equação diferencial que descreve a trajetória de um espaçomodelo. O modelo matemático considera apenas o movimento vertical do veículo, não permite simular veículos com múltiplos estágios. A massa do espaçomodelo será considerada variável. Coeficiente de arrasto, massa específica do ar atmosférico e aceleração da gravidade podem ser considerados como função da velocidade ou altitude. A força de empuxo do motor pode ser considerada constante ou função do tempo, como em geral ocorre quando dados de experimentos são disponíveis. O programa é capaz de estimar a aceleração, velocidade e posição do espaçomodelo em função do tempo, assim como apogeu previsto (altitude máxima), velocidade máxima e tempo de voo. Em uma primeira fase o Método de Euler é implementado e validado com casos especiais para os quais a equação da trajetória possui solução analítica conhecida. Em seguida, um estudo de erros numéricos é conduzido para demonstrar a qualidade de algumas discretizações temporais selecionadas. Estas abordagens fornecerão embasamento para as análises posteriores, que envolvam casos gerais aplicáveis à voos reais e cujas soluções analíticas não são possíveis por técnicas analíticas.

Palavras-chave: Trajetória de espaçomodelos. Método de Euler. Verificação numérica.

INTRODUÇÃO

Um espaçomodelo, daqui em diante abreviado por EM, é um foguete em miniatura construído preferivelmente sem materiais metálicos e que executa todas ou quase todas as fases características de voo de um foguete de sondagem profissional, como aqueles lançados pela Força Aérea Brasileira nos centros de lançamento em Alcântara, no Maranhão ou na Barreira do Inferno, no Rio Grande do Norte. Um EM típico possui um motor de propelente sólido, em geral pólvora negra, possui uma

¹ Aluno do 4º ano do curso de Engenharia Mecânica da Universidade Positivo. E-mail: blavier.jeremy@gmail.com. Voluntário do Programa de Iniciação Científica (PIC) da UP.

² Professor do curso de Engenharia Mecânica da Universidade Positivo. Mestre em Engenharia Mecânica. E-mail: antoniocarlos.foltran@gmail.com.

estrutura externa com função aerodinâmica e um sistema de recuperação, como um paraquedas, por meio do qual o EM pode ser recuperado e usado em outros voos. É comum embarcar alguma carga útil e também um altímetro, cuja função é registrar os dados de voo.

À esquerda na FIGURA 1 abaixo veem-se dois espaçomodelos entre 20 e 30 cm de comprimento e à direita o foguete Sonda IV, com aproximadamente 11 m de comprimento e já preparado para o lançamento. O Sonda IV foi projetado no Instituto de Aeronáutica e Espaço IAE em São José dos Campos e foi lançado pela Força Aérea Brasileira entre 1984 e 1989.



FIGURA 1 – DOIS ESPAÇOMODELOS E À DIREITA O FOGUETE SONDA IV
FONTE: OS AUTORES, 2015 E INSTITUTO DE AERONÁUTICA E ESPAÇO, IAE

Antes de ser lançado, um foguete profissional precisa passar por uma avaliação de trajetória, a fim de saber se ele será capaz de cumprir a missão, se representará um perigo para as comunidades vizinhas caso haja alguma mudança de

trajetória prevista por algum problema técnico. Neste caso um mecanismo de teledestruição é acionado.

Outro exemplo da importância de conhecer a trajetória é a definição da área marítima a ser interditada no momento do lançamento, a fim de garantir a segurança de embarcações que possam transitar na área de risco desde o momento do lançamento até a recuperação da carga útil, que normalmente cai no oceano e é recuperada por meio de helicópteros.

Apesar de serem versões miniaturizadas dos foguetes profissionais, os EMs estão sujeitos às mesmas forças durante o voo, diferindo apenas em sua magnitude e quase sempre confinados ao voo em velocidades inferiores à velocidade do som (*i.e.* voo subsônico). Por esta razão, para voos de EMs também é desejável uma ferramenta rápida para estudos de trajetória.

O projeto de um espaçomodelo ou de um foguete experimental muitas vezes é um processo iterativo de predição e correção que progride desde uma estimativa inicial, definida em função da experiência do projetista, até que uma solução adequada à missão a ser alcançada.

Um bom exemplo são as missões de “apogeu fixo” do Festival de Minifoguetes de Curitiba, evento anual para o qual as equipes devem projetar um espaçomodelo sem restrições técnicas, exceto as de segurança, e uma determinada altitude deve ser alcançada. A equipe que chegar mais próximo desta altitude vence a categoria.

O uso de tal programa permite uma estimativa bastante exata das variáveis em diferentes estágios do percurso e com diferentes tipos de condições, por exemplo, verificar valores de velocidade durante sua fase propulsada, o apogeu máximo, o tempo de impacto ao solo, considerando a aceleração da gravidade variável e, com arrasto aerodinâmico.

O programa tem por objetivo reproduzir numericamente os mesmos fenômenos os quais um foguete real experimenta. Desta forma é possível simular várias condições de voo no programa que se fossem resolvidas de forma analítica (equações, papel e lápis) seriam complexas ou mesmo impossíveis.

1 REVISÃO BIBLIOGRÁFICA

Segundo Sutton e Biblarz (2001, p. 119), enquanto um foguete faz voo na atmosfera, três são as forças que agem sobre ele, portanto influenciam em sua trajetória: a força da gravidade, as forças aerodinâmicas e a força de empuxo. Quando aplicada a Lei de Newton, a resultante das forças atuantes sobre um sistema atua no sentido de variar sua quantidade de movimento, ou seja, promove uma aceleração \vec{a} sobre o sistema. Em forma matemática:

$$\vec{a} = \frac{\vec{E}_{(t)} + \vec{D}_{(\rho,v)} + \vec{P}_{(m,g)}}{m_{(t)}}, \quad \text{Eq.(1)}$$

onde m é a massa instantânea do veículo. É importante ressaltar que a massa de um foguete ou mesmo de um EM muda substancialmente durante a operação do motor, pois em geral mais da metade da massa inicial do foguete ou EM é propelente que é queimado e utilizado para impulsionar o veículo durante fase propulsada do voo. Este é o motivo de enfatizar na equação que se trata de uma variável que é função do tempo t , pelo menos até o final da queima do propelente. Após esta fase a massa do EM permanece constante, exceto se partes do veículo são separadas.

A força de empuxo produzida pelo motor do veículo é representada por \vec{E} . Esta é a única força que atua no sentido de sustentar o voo na atmosfera, impedindo que o veículo retorne à superfície do planeta. É descrita como função do tempo, pois em geral ela ocorre de acordo com o projeto da câmara de combustão do motor e da geometria do grão propelente. Muitas vezes ela é determinada por meio de ensaios em banco estático, assim seu comportamento em função do tempo é conhecido.

\vec{D} é a força de arrasto aerodinâmico provocada pelo atrito do veículo com o ar atmosférico, haja visto que há velocidade relativa entre ambos. A força aerodinâmica é, em geral, descrita por duas componentes: o arrasto \vec{D} e a força de sustentação \vec{L} , porém neste trabalho assumir-se-á que o movimento do EM ocorre apenas na direção vertical e que não há vento atuando lateralmente sobre o EM, de forma semelhante aos foguetes de sondagem profissionais. Desta forma a sustentação é nula e não será considerada. É proporcional ao quadrado da velocidade do veículo e também é dependente de sua forma geométrica.

É possível descrever a Eq. (1) na forma de equação diferencial, pois a aceleração é a derivada temporal da velocidade e a velocidade é a derivada temporal da posição. Como a aceleração da gravidade \vec{g} pode sofrer variação significativa com a altitude de voo, ela será considerada variável. A força de arrasto é proporcional à velocidade do veículo, portanto uma variável não conhecida *a priori*.

Já a força de empuxo é em geral conhecida de antemão e provém, quase sempre, de um programa de testes para caracterizar o desempenho do motor do foguete ou EM.

A Eq(1) escrita em forma escalar e não vetorial (considerando apenas o movimento vertical) assume a forma mostrada abaixo, onde os termos da força de arrasto e força gravitacional foram deslocados para o membro esquerdo da equação, a fim de ficar na forma padrão da equação diferencial:

$$m_{(t)} \frac{d^2x}{dt^2} + \frac{C_d \rho A}{2} \left(\frac{dx}{dt} \right)^2 + \frac{m_{(t)} g_0 R^2}{(R+x)^2} = E_{(t)}, \quad \text{Eq.(2)}$$

onde x é a posição instantânea do EM e é a variável dependente da equação diferencial. A variável t é o tempo, ou seja, a variável independente da equação diferencial.

A é a área da seção transversal do EM, ρ e g_0 são a massa específica da atmosfera e aceleração da gravidade ao nível do solo, respectivamente. É sabido que a massa específica da atmosfera varia com a altitude e a temperatura local. Neste trabalho haverá a opção de descrevê-la como função da altitude (lançamentos de grande altitude) ou como função da temperatura, através da equação de gás ideal (lançamentos de baixa altitude).

Quanto à aceleração gravitacional g , esta também é função da altitude, porém seu valor local, ao nível do solo g_0 , é constante. Ainda na Eq(2) $R = 6378137 \text{ m}$ é o raio médio do planeta, ou seja, considera-se que a Terra possui forma perfeitamente esférica, o que não é correto, porém é uma aproximação suficientemente razoável considerando a precisão necessária para projetos de EMs.

Observa-se que a Eq.(2) é uma equação diferencial ordinária, de segunda ordem e não linear. Mesmo que a função E seja simples, por exemplo, uma função degrau unitário que persiste por um tempo dito tempo de queima, ainda assim a solução desta equação não é simples. Provavelmente não há solução analítica para

ela, mesmo que o empuxo seja nulo (equação homogênea). Também é fato que a massa específica e o coeficiente de arrasto C_d podem ser funções complexas, assim a melhor alternativa é resolver a Eq(2) por meio de métodos numéricos.

Assumindo-se que o modelo matemático foi adequadamente implementado na forma de código computacional e que erros de programação não foram cometidos, a principal fonte de erro numérico no presente trabalho é o erro de discretização.

O erro de discretização provém do erro de truncamento de termos de ordem elevada de uma série de Taylor expandida sobre um ponto do domínio onde é tomado o valor da variável dependente para representar as derivadas da função e compor a aproximação da equação diferencial. O estudo da teoria de erros de discretização não é o foco deste trabalho, por isso, caso mais detalhes sejam necessários, recomenda-se uma consulta a Marchi (2001) ou Versteeg e Malalasekera (2007, p. 285).

Segundo Maliska (2004, p. 27), a tarefa do método numérico é resolver uma ou mais equações diferenciais, substituindo as derivadas existentes por expressões algébricas que envolvem a função incógnita. Quando não é possível a solução analítica, e se decide fazer uma aproximação numérica, se aceita a solução para um número discreto de pontos, com um determinado erro, esperando que, quanto menor o intervalo entre os pontos, mais próximo da solução analítica estará a solução aproximada. Ainda segundo este autor, os erros podem ser classificados em erros de programação, erros de discretização, erros iteração e erros de arredondamento.

Os erros de programação são aqueles derivados da dificuldade de se traduzir a linguagem matemática, representada pelas equações, para os procedimentos escritos nas linhas do programa de computador. É importante salientar que muitas vezes as equações a serem resolvidas requerem procedimentos iterativos, com cálculos sequenciais da mesma estrutura matemática até que a convergência de termos da equação seja obtida. Nem sempre é fácil de construir um programa isento de erros de programação. Maliska (2004, p. 171) aponta o fato que programas grandes, em geral possuem erros de programação em 5% de suas linhas. Se estes erros não produzem erros apreciáveis no resultado final do código após processamento, então eles se tornam muito difíceis de serem detectados.

2 PROCEDIMENTOS METODOLÓGICOS

Considerando as informações dadas no Capítulo 1 deste trabalho decidiu-se analisar separadamente a ação de cada uma das forças atuantes sobre o veículo. Convencionou-se utilizar a seguinte metodologia:

- 1) Analiticamente elimina-se duas das três forças atuantes sobre o veículo, obtendo-se uma versão simplificada da Eq. (2);
- 2) Resolve-se a equação diferencial obtida analiticamente;
- 3) Utiliza-se a solução analítica para formular uma solução de referência;
- 4) Simplifica-se uma ou mais linhas de comando do programa a fim de simular o mesmo efeito de remover duas das três forças atuantes sobre o veículo;
- 5) Executa-se o programa e coleta-se os resultados numéricos das variáveis de interesse;
- 6) Comparam-se os resultados e determina-se o erro numérico associado.

A metodologia supracitada permitirá encontrar valores recomendados para o intervalo de tempo, ou passo de tempo a ser utilizado pelo método numérico para cada avançar a solução da Eq(2) no tempo, desde o momento do lançamento até o impacto com o solo. Os valores das variáveis serão escolhidos de acordo com casos típicos de voo de EM. Desta forma é necessário que os seis passos citados sejam repetidos para todos os problemas modelo escolhidos.

3 DESENVOLVIMENTO DAS SOLUÇÕES ANALÍTICAS

Seguindo a metodologia descrita no capítulo dois, o primeiro modelo matemático a ser descrito é o que possui apenas a força gravitacional, ou seja, apenas a força peso do EM permanece na Eq. (2).

Em Greenberg (1998, p. 53), esta equação é apresentada sem utilizar a aceleração local da gravidade, porém trata-se da mesma equação, pois $g_0 = GM/R^2$. A Eq. (2) assume então a forma de uma equação diferencial ordinária de segunda ordem cuja variável independente é o tempo t e a variável dependente a posição x :

$$m \frac{d^2x}{dt^2} = - \frac{GMm}{(x + R)^2}, \quad \text{Eq.(3)}$$

onde m é a massa do EM e M é a massa do planeta. Segundo Sutton e Biblarz (2001, p. 119), para a Terra $M = 5,976 \times 10^{24} \text{ kg}$ e $R = 6378137 \text{ m}$ é o raio médio do planeta Terra.

A Eq. (3) pode ser reformulada para variável dependente velocidade, uma vez que $dv/dt = (dv/dx)(dx/dt)$. A Eq. (3) assume então a forma:

$$mv \frac{dv}{dx} = -\frac{GMm}{(x+R)^2}. \quad \text{Eq. (4)}$$

A Eq.(4) é mais adequada e será utilizada no desenvolvimento do primeiro modelo matemático neste trabalho. Conforme mostrado em Greenberg (1998, p. 53), a partir dela é possível demonstrar a Lei da Conservação da Energia Mecânica.

Observa-se que a Eq. (4) é uma equação diferencial ordinária de primeira ordem linear. Ela possui solução analítica conhecida, necessitando apenas de uma condição inicial para que uma solução particular possa ser obtida. No caso a condição inicial pode ser a velocidade inicial do EM V_i , conhecida na plataforma de lançamento:

$$v_{(x=0)} = V_i, \quad \text{Eq. (5)}$$

Assim sua solução particular é dada pela Eq. (6):

$$v_{(x)} = \sqrt{V_i^2 - \frac{2GM}{R} \frac{x}{(x+R)}}. \quad \text{Eq. (6)}$$

Nota-se que a velocidade do EM se reduz à medida que x aumenta. Isto progride até a velocidade nula (*i.e.* apogeu). Do momento em que o apogeu é atingido em diante o EM retorna à superfície do planeta. Isto só não ocorrerá se a velocidade de escape for atingida, porém seu valor é muito elevado considerando-se EMs, portanto apenas alguns foguetes profissionais conseguem sair de órbita do planeta Terra.

A Eq.(5) pode ser utilizada para calcular a velocidade instantânea do EM desde o momento do lançamento até o impacto com o solo, mas para isso as seguintes condições precisam ser atendidas:

- 1) Apenas a força gravitacional atua sobre o EM, seja com aceleração gravitacional constante ou variável (função da altitude);
- 2) O EM parte da posição inicial com velocidade inicial V_i especificada de modo a obter um apogeu selecionado.

Na Eq.(6) a altitude instantânea pode ser isolada e a velocidade instantânea tomada como nula, desta forma a altitude máxima ou apogeu h_{max} pode ser calculado analiticamente, servindo de parâmetro de comparação para avaliar o erro de discretização. O apogeu é dado pela Eq. (7):

$$h_{max} = \frac{V_i^2 R^2}{2GM - V_i^2 R} . \quad \text{Eq. (7)}$$

Outra variável de interesse para a validação do código é a velocidade final ou velocidade de impacto V_f . Como este primeiro modelo matemático considera que há apenas a força gravitacional, então:

$$V_f = V_i . \quad \text{Eq. (8)}$$

O segundo modelo matemático considera que apenas há a força de empuxo atuando no veículo. Neste caso ela atua por um tempo chamado tempo de queima t_q . Enquanto o motor está em operação o propelente é queimado e ejetado pelo motor em elevada velocidade.

Para a maioria dos EMs comerciais e fabricados pelas equipes de foguetes no Brasil é razoável supor que a vazão dos produtos de combustão \dot{m} seja constante, pois a maioria dos motores é fabricada para manter a pressão na câmara de combustão constante. Da mesma forma o diâmetro da tubeira (*i.e.* região posterior do motor onde os produtos da combustão são acelerados) é constante, por isso é razoável supor que a vazão mássica dos gases quentes seja constante.

Conforme descrito em Fox *et al.* (2012) é possível fazer uma dedução para as equações para a aceleração, velocidade e posição do EM no momento do fim da queima do propelente $t = t_q$. As equações são obtidas combinando as equações de conservação da quantidade de movimento com a equação da conservação da massa.

Se a força de empuxo for constante desde o disparo do EM até o tempo final da queima $E = E_{cte}$, então a equação para a aceleração no fim da queima é:

$$a_{tq} = \frac{c\dot{m}}{m_o - \dot{m}t_q} - g, \quad \text{Eq.(9)}$$

onde m_o é a massa inicial do EM, ou seja, a massa do modelo carregado com a carga de propelente e c é a velocidade de exaustão dos produtos de combustão.

Observa-se que o denominador do primeiro termo do lado direito da Eq(9) representa a massa instantânea do modelo, pois é composta pela massa inicial subtraída do produto vazão mássica e tempo. Como foi escrito o tempo de queima, então se calcula a aceleração no final da queima, porém esta equação é válida para qualquer $0 \leq t \leq t_q$ sendo apenas necessário substituir o tempo de queima por um tempo t qualquer e manter a massa constante com valor da massa final do EM m_f .

Integrando a Eq.(9), obtém-se a velocidade no fim da queima V_{tq} :

$$V_{tq} = V_i \ln \left(1 - \frac{\dot{m}t_q}{m_o} \right) - gt_q. \quad \text{Eq.(10)}$$

Integrando-se novamente no tempo, obtém-se a altura do EM no final da fase propulsada h_{tq} :

$$h_{tq} = \frac{V_i m_o}{\dot{m}} \left[\left(1 - \frac{\dot{m}t_q}{m_o} \right) \left(\ln \left(1 - \frac{\dot{m}t_q}{m_o} \right) - 1 \right) + 1 \right] - \frac{gt_q^2}{2}. \quad \text{Eq.(11)}$$

Imediatamente após o final da queima do propelente ($t = t_q$) o EM entra em movimento balístico ascendente e sua velocidade e altura, ambas são função do tempo, são dadas pelas equações:

$$V_{(t)} = V_{tq} - g(t - t_q) \quad \text{Eq.(12)}$$

e

$$h_{(t)} = h_{t_q} + V_{t_q}(t - t_q) - \frac{g(t - t_q)^2}{2}, \quad \text{Eq.(13)}$$

onde, $\dot{m} = \frac{m_p}{t_q}$, é a razão entre massa de propelente m_p e tempo de queima t_q .

Para valerem, as Eqs(9 - 13) requerem as seguintes hipóteses:

- 1) Empuxo constante no tempo $0 \leq t \leq t_q$;
- 2) A velocidade inicial do EM é nula (lançamento a partir da plataforma de lançamento);
- 3) Apenas as forças de empuxo e gravitacionais são consideradas, portanto para tempos $t \geq t_q$ apenas a aceleração gravitacional é não nula e a velocidade reduzir-se-á devido à ela.

O último modelo matemático utilizado para testar o programa é considerando apenas a força de arrasto aerodinâmico. Neste caso considera-se que o foguete sai da plataforma de lançamento com velocidade inicial conhecida V_i escolhida para contemplar a gama de velocidades típicas de EMs (entre 10 e 400 km/h). Neste caso a velocidade do EM se reduzirá gradualmente. Mais intensamente no início e tendendo a um valor constante à medida que o tempo passa e a velocidade do EM se reduz.

A Eq.(2) assume a forma:

$$m_{(t)} \frac{d^2x}{dt^2} = -\frac{C_d \rho A}{2} \left(\frac{dx}{dt} \right)^2, \quad \text{Eq.(14)}$$

Sabendo que $v = dx/dt$ e supondo que a massa específica da atmosfera e o coeficiente de arrasto são constantes, então se pode combinar todas as constantes formando a variável $B = -C_d \rho A / (2m)$. Fazendo a substituição e fazendo a separação das variáveis, a integração da Eq.(14) resulta na equação:

$$v = \frac{-1}{Bt - \frac{1}{V_i}}, \quad \text{Eq.(14)}$$

onde V_i é, assim como anteriormente, a velocidade inicial do EM. Como a posição e aceleração são testados no modelo onde apenas a força de empuxo constante age

sobre o EM, então se considera que apenas a comparação da solução analítica da Eq.(14) com os resultados numéricos são suficientes para validar o modelo de arrasto aerodinâmico concebido para o programa.

4 RESULTADOS

Após a programação inicial do código, este foi denominado SBE, um acrônimo para “**S**imulador de **B**alística de **E**spaçomodelos”. Os primeiros testes realizados foram aqueles relativos às entradas e saídas de dados. O primeiro teste foi efetuar a leitura de todas as variáveis de interesse, vindas de basicamente quatro arquivos de texto.

No primeiro arquivo o usuário deve inserir as opções de modelo físico a ser adotado durante a simulação:

- 1) Comportamento da força de empuxo: valor constante ou variável (a ser lido de um arquivo contendo a curva de empuxo);
- 2) Comportamento do coeficiente de arrasto: valor constante ou variável (lido de um arquivo contendo um conjunto de valores do coeficiente de arrasto em função da velocidade);
- 3) Modelo para descrever a aceleração da gravidade: valor constante ou função da altitude;
- 4) Modelo para descrever a massa específica do ar: equação de estado ou atmosfera padrão internacional;
- 5) Modelo para simular a abertura do paraquedas, caso o modelo o possua.

Nos demais arquivos de texto o usuário deve informar:

- 1) Nome da simulação com até 50 caracteres (para que o programa crie um arquivo de saída com a identificação não ambígua);
- 2) Descrição de um caso específico a ser simulado e os dados do EM (máximo 50 caracteres);
- 3) Massa de propelente;
- 4) Massa final do EM (após a queima do propelente);
- 5) Tempo de queima do propelente;
- 6) Diâmetro de referência (diâmetro externo do tubo-foguete);
- 7) Valor de empuxo constante ou nome do arquivo contendo a curva de empuxo, de acordo com a seleção feita anteriormente;

- 8) Valor do coeficiente de arrasto ou nome do arquivo contendo os valores do coeficiente de arrasto em função da velocidade;
- 9) Aceleração da gravidade no nível do solo (é usado apenas no modelo de aceleração da gravidade constante);
- 10) Caso se opte por modelar a massa específica do ar pela equação de estado, a pressão atmosférica e temperatura do ar devem ser fornecidas. Do contrário pressão e temperatura atmosféricas são calculadas de acordo com o modelo de Atmosfera Internacional Padrão;
- 11) Intervalo de tempo utilizado no algoritmo para calcular as variáveis de interesse e número de iterações utilizadas para o processo de atualização das variáveis dependentes da altitude e velocidade.

A leitura dos dados de entrada e escrita dos dados de saída (mesmo antes do processamento principal do problema) foi concluída inclusive daqueles destinados à leitura dos dados das curvas de empuxo e de C_d em função da velocidade. A rotina que efetua a interpolação linear dos dados das curvas de empuxo e de C_d para serem utilizados no programa está operando adequadamente.

Depois de concluídos os testes com as entradas e saídas de dados, efetuou-se a programação do Método de Euler. Isto foi feito de forma única para as três fases do voo: fase propulsada, fase balística ascendente e fase balística descendente.

Após encontrar e corrigir alguns erros de programação iniciou-se o programa de testes do SBE. O primeiro caso analisado foi o caso simplificado de aceleração gravitacional apenas. Em seguida passou-se para o caso de arrasto aerodinâmico apenas. Por último foram testados casos gerais. Por último foi testado o voo do foguete profissional Sonda IV, mostrado à direita na FIGURA 1.

4.1 RESULTADOS PARA MODELO DE ACELERAÇÃO GRAVITACIONAL

Para testar o programa SBE, primeiramente foram removidas do cálculo da trajetória as contribuições para aceleração devidas às forças aerodinâmica e de empuxo, ficando apenas a força gravitacional. Para forçar o movimento do EM, introduziu-se um valor de velocidade inicial típico de 100 m/s na linha de comando 183 do arquivo fonte 'Fonte.f90'. Além desta alteração foi necessário manter a massa do EM constante a fim de obter os valores analíticos para as variáveis de interesse: tempo

até o apogeu, velocidade máxima de subida, velocidade máxima de descida, altitude máxima (apogeu) e tempo até o impacto.

Por esse motivo selecionou-se o modelo Sondinha II da Empresa Bandeirante Modelismo. Este modelo, quando equipado com motor da classe A fornecido pelo mesmo fabricante, possui massa final 20 gramas, diâmetro do corpo 20 mm. A massa foi mantida igual à massa final por toda a simulação.

A remoção artificial da força aerodinâmica foi feita inserindo valor nulo para o coeficiente de arrasto aerodinâmico e a remoção da força de empuxo foi feita selecionando o modelo de empuxo constante com empuxo nulo e tempo de queima nulo.

A TABELA 1 abaixo traz os resultados das variáveis estudadas para a malha mais refinada e a comparação com o valor analítico. Apesar dos valores mostrados na tabela serem fornecidos com nove algarismos significativos, seus valores foram calculados com 15 algarismos significativos.

TABELA 1 – COMPARAÇÃO ENTRE RESULTADOS NUMÉRICOS E ANALÍTICOS PARA VOO DO EM SONDINHA II CONSIDERANDO APENAS A FORÇA GRAVITACIONAL

	Tempo para apogeu [s]	Apogeu [m]	Velocidade máxima de subida [km/h]	Velocidade máxima de descida [km/h]	Tempo para impacto [s]
Analítico	10,1972526	509,862621	360	360	20,39451
Numérico	10,1972529	509,862642	359,999999	359,999999	20,3945064
Erro [%]	$-2,757 \times 10^{-6}$	$-2,351 \times 10^{-6}$	$1,004 \times 10^{-7}$	$3,433 \times 10^{-8}$	$-6,149 \times 10^{-6}$

OBS: Considerou-se velocidade inicial 100 m/s e massa do EM constante com valor 20 g.

Os resultados obtidos nas diversas malhas (diversos valores de passo de tempo ou intervalo discreto de tempo) foram utilizados para uma análise de erros de discretização.

É possível observar no GRÁFICO 1 que quanto menor o tamanho do intervalo de tempo discretizado (*i.e.* passo de tempo ou ainda tamanho da malha), mais próximo do valor analítico se torna o valor obtido para a variável numérica. Observa-se que a redução do erro é da mesma ordem de grandeza da redução do tamanho de malha, ou seja, o erro se reduz em uma ordem de magnitude para uma redução do tamanho de malha de uma ordem de magnitude. Marchi(2001) esclarece que aproximações

numéricas de primeira ordem apresentam essa característica. O Método de Euler é de primeira ordem, portanto os resultados são consistentes.

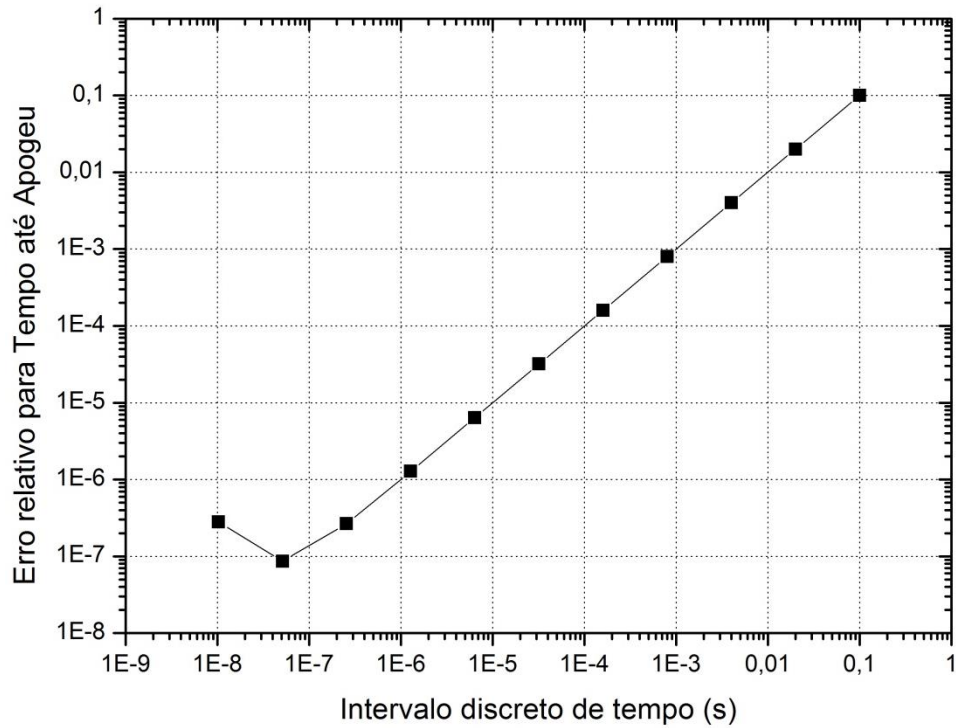


GRÁFICO 1 - ERRO RELATIVO ENTRE A SOLUÇÃO ANALÍTICA DA ALTITUDE DE APOGEU E A SOLUÇÃO NUMÉRICA OBTIDA PARA DIVERSOS TAMANHOS DE MALHA.

Na prática o tempo é tipicamente cronometrado com até quatro algarismos significativos (0,1 milisegundo), portanto o tamanho de malha pode ser da ordem de 10^{-4} . Vale lembrar que malhas mais refinadas necessitam de um maior tempo computacional, que para a análise do tempo até apogeu seria desnecessário.

Por último observa-se uma elevação do erro próximo ao tamanho de malha 10^{-8} . Isto provavelmente se deve à influência dos erros de arredondamento ou erros de máquina.

Quanto ao apogeu, utilizando o cálculo analítico considerando uma velocidade inicial de 100 m/s obteve-se o apogeu aproximadamente 510 m, o que é bastante similar à alguns EM montados com motores mais potentes, como os das classes C, D e E. Observa-se no GRÁFICO 2 que mesmo malhas bastante grosseiras, como 0,1 s já produzem resultados aceitáveis, porém deve-se ter em mente que a força de empuxo pode apresentar variações bruscas e um passo de tempo de mesma ordem

pode induzir valores muito diferentes daqueles obtidos em malhas relativamente refinadas.

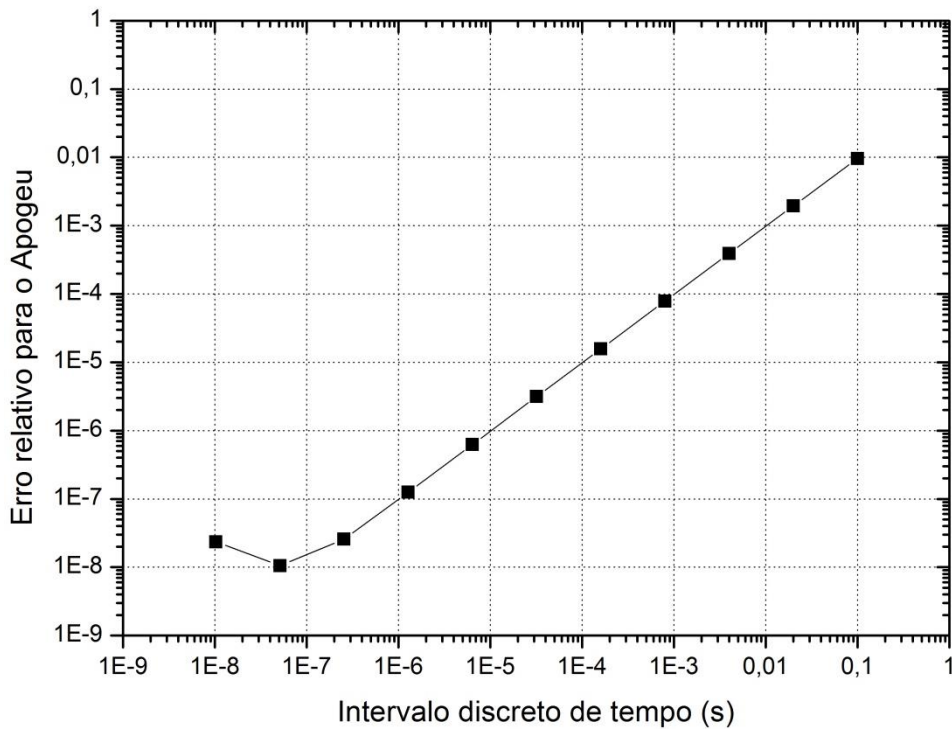


GRÁFICO 2 - ERRO RELATIVO ENTRE A SOLUÇÃO ANALÍTICA E A NUMÉRICA DA ALTITUDE DE APOGEU OBTIDA PARA DIVERSOS TAMANHOS DE MALHA.

Tanto a velocidade máxima de subida como a velocidade máxima de descida apresentaram comportamento semelhante, ambas com redução do erro relativo devido à aproximação de primeira ordem. Por este motivo apenas a velocidade máxima de subida foi apresentada no GRÁFICO 4.

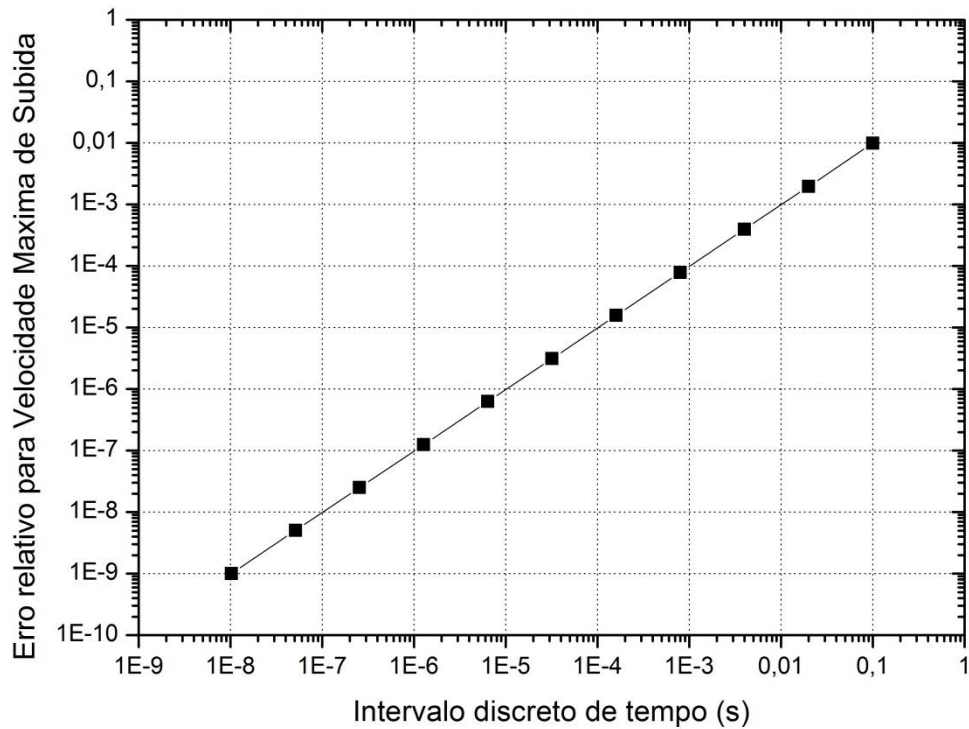


GRÁFICO 4 - ERRO RELATIVO ENTRE A SOLUÇÃO ANALÍTICA E A NUMÉRICA DA VELOCIDADE MÁXIMA DE SUBIDA OBTIDA PARA DIVERSOS TAMANHOS DE MALHA.

Uma observação importante sobre o comportamento destas duas variáveis é que se acreditava que o erro numérico se propagaria pelas iterações, pois o problema apresenta o comportamento de “marcha no tempo”, conforme descrito por Maliska (2004, p.21). Este comportamento, porém não foi observado, o que é bom, pois significa que o erro não aumenta à medida que o tempo passa e o EM está na fase balística descendente, por exemplo. Observa-se na TABELA 1 que estas duas variáveis apresentam mesmo valor.

Quanto ao comportamento da velocidade do modelo em função do tempo, é possível observar no GRÁFICO 5 abaixo, onde são mostradas as soluções numérica e analítica, que o modelo numérico implementado está fornecendo os valores previstos pela versão simplificada da equação da conservação da quantidade de movimento. Da mesma forma o GRÁFICO 6 mostra o comportamento da altitude em função do tempo para as soluções analítica e numérica (assim como no GRÁFICO 5, apenas alguns pontos da solução numérica foram representados).

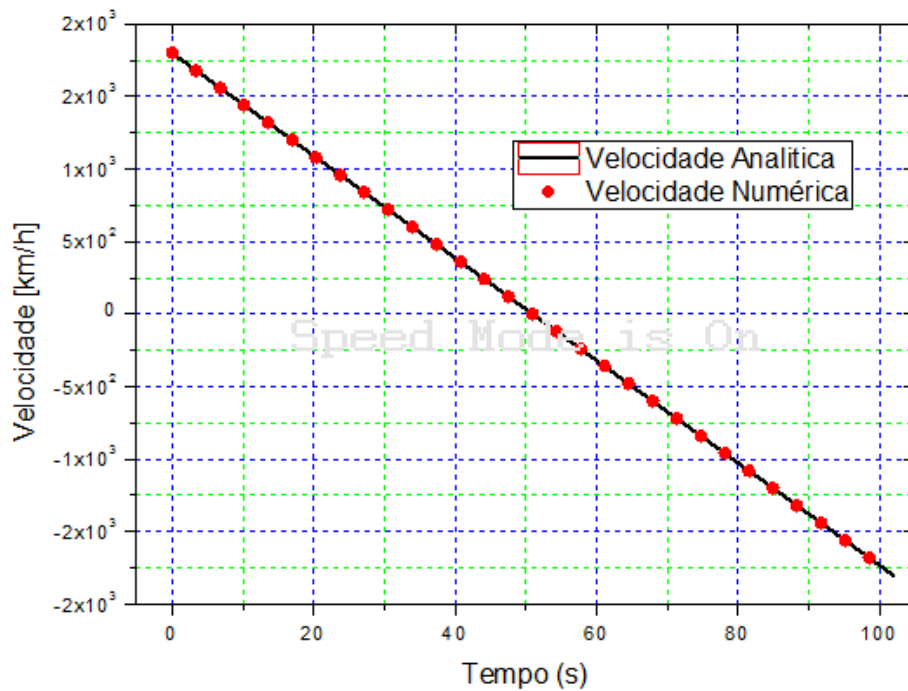


GRÁFICO 5 – SOLUÇÕES ANALÍTICA E NUMÉRICA DA VELOCIDADE EM FUNÇÃO DO TEMPO CONSIDERANDO QUE APENAS A FORÇA PESO ATUA SOBRE O EM.

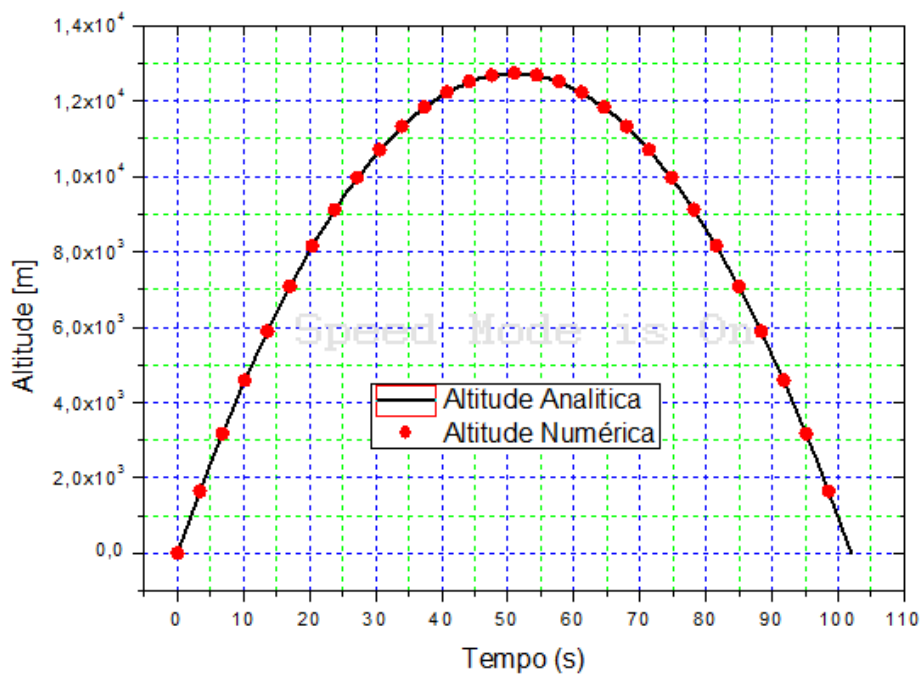


GRÁFICO 6 – SOLUÇÕES ANALÍTICA E NUMÉRICA DA ALTITUDE EM FUNÇÃO DO TEMPO CONSIDERANDO QUE APENAS A FORÇA PESO ATUA SOBRE O EM.

4.1 RESULTADOS PARA MODELO DE ARRASTO AERODINÂMICO

O arrasto aerodinâmico é uma força de retardo a qual o foguete é submetido enquanto se desloca na atmosfera terrestre e tem uma influência desprezível acima das camadas mais elevadas da atmosfera. No caso de um EM típico, seu voo sempre se dará na atmosfera terrestre, e em geral confinado à troposfera.

Desprezando a força gravitacional e sendo uma força de retardo, o único efeito que ela promove é desacelerar um EM lançado. Para que se possa obter resultados analíticos para testar o modelo foram testados casos nos quais o EM parte do solo com velocidade inicial 1 m/s, 10 m/s, 100 m/s e 1000 m/s. Os resultados são representados no GRÁFICO 7 abaixo, para o qual foi utilizado um passo de tempo de 10^{-3} s e intervalo de escrita dos dados no gráfico 10^{-2} s:

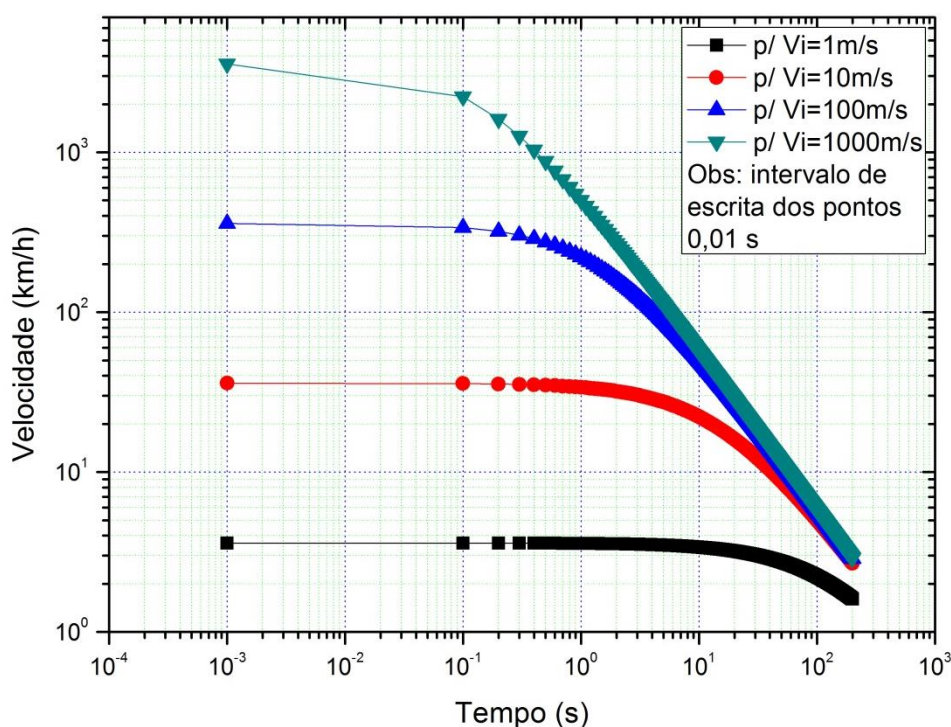


GRÁFICO 7 - VELOCIDADE EM FUNÇÃO DO TEMPO PARA O MODELO DE ARRASTO AERODINÂMICO APENAS E CONSIDERANDO DIVERSAS VELOCIDADES INICIAIS.

Conforme esperado, observa-se que quanto maior é a velocidade inicial do EM, maior é a influência da força de arrasto no sentido de retardar o EM. Também parece haver um limite além do qual, não importa a velocidade de lançamento do EM,

ele leva um mesmo tempo para ser desacelerado até a velocidade nula. Entretanto deve-se ter cuidado ao tirar conclusões sobre esse comportamento, uma vez que se trata de uma condição idealizada. Sabe-se que, quanto mais próximo da velocidade do som, mais o coeficiente de arrasto aumenta, portanto o GRÁFICO 7 não deve ser tomado como representativo dos fenômenos reais que ocorrem no voo, mas sim apenas como uma ferramenta de estudo de erros numéricos.

O erro numérico obtido para as velocidades pode ser visualizado no GRÁFICO 8 ABAIXO:

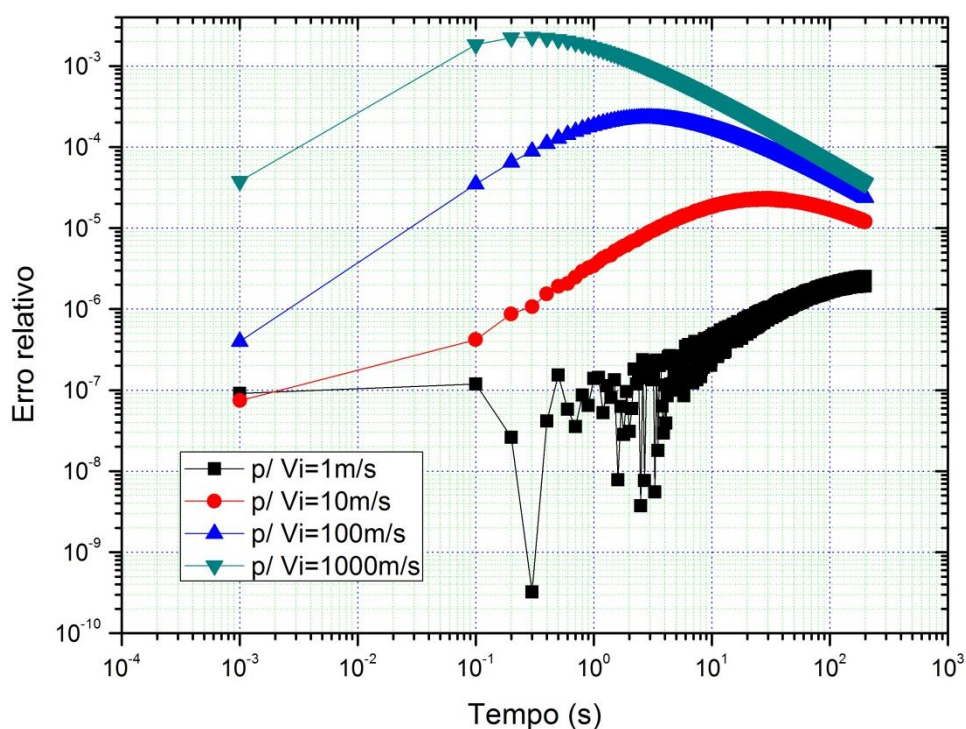


GRÁFICO 8 – ERRO NUMÉRICO PARA A VELOCIDADE INSTANTÂNEA DO EM CONSIDERANDO APENAS A FORÇA DE ARRASTO AERODINÂMICO.

Observa-se que o erro numérico é maior quanto maior a velocidade inicial do EM, indicando que EM de maior potência e que voem mais alto estarão sujeitos a maior erro nas estimativas de velocidade. Como a velocidade apresentará maior erro, estima-se que a posição do EM em função do tempo também apresentará maior erro. Ainda é possível observar dois fenômenos. O primeiro é que o erro aumenta na região do gráfico que corresponde à maior variação de velocidade. Isso indica que, quanto maior a variação de velocidade, maior é o erro numérico. O segundo é que para a velocidade inicial 1 m/s o erro de máquina ou erro de arredondamento foi atingido.

Isso é observado pelo comportamento oscilatório, meio caótico do comportamento do erro numérico para essa variável.

4.2 SIMULAÇÃO DO FOGUETE SONDA IV

A fim de mostrar o desempenho do programa SBE na simulação de um foguete real, conduziu-se uma simulação do Sonda IV, conforme mostrado à direita na FIGURA 1.

Deve-se ressaltar que é difícil de obter todos os dados necessários à simulação de um foguete real e mais ainda dados de voo que possam ser usados na validação do programa de computador. A explicação para isso é que problemas de trajetória de foguetes e trajetória de mísseis são idênticos e, por motivos óbvios, experiências e estudos são conduzidos em segredo pelos departamentos de defesa do mundo todo.

Em STRACK *et al.* (1966, p.85-89) são apresentados duas simulações de trajetória, porém não podem ser reproduzidos com o programa SBE. Tratam-se de trajetória orbital de um satélite e do lançamento a partir da Terra, de um foguete com três estágios para enviar uma sonda à Lua.

O programa N-Body Code, descrito em STRACK *et al.* (1966) é tridimensional e considera a ação gravitacional dos demais corpos celestes do Sistema Solar. O programa SBE é uma versão unidimensional (só permite a simulação do lançamento de foguetes de sondagem, ou seja, de foguetes lançados perpendicularmente à superfície do planeta ou com ângulos muito próximos).

CLBI (1995, p.13) foi a única referência que se pode usar em uma tentativa de validar o código. Esta simulação resultou uma previsão surpreendentemente acurada do apogeu alcançado no voo do Sonda IV, porém resalta-se que nem todos os dados necessários à simulação foram obtidos. Faltaram o coeficiente de arrasto em função da velocidade e a curva de empuxo dos dois motores (pois o foguete é de dois estágios). Os dados obtidos em CLBI (1995, p.13) são reproduzidos na TABELA 2 ABAIXO.

TABELA 2 – DADOS DO FOGUETE SONDA IV

	1° Estágio	2° Estágio	Veículo
Massa da estrutura [kg]	1308	403	1711
Massa de propelente [kg]	4215	879	5094
Massa de freon + SCR [kg]	112	-	112
Tempo de combustão [s]	40	22	-
Tempo de ação [s]	61	30	-
Empuxo médio no vácuo [kN]	252	95,9	-
Diâmetro externo do propulsor [mm]	1007	555	-
Massa total na decolagem [kg]	5635	1282	6917

O único resultado que pode ser comparado foi a altitude máxima ou apogeu, pois seu valor é dado em CLBI (1995, p.13) em forma gráfica, reproduzido neste trabalho no GRÁFICO 9 abaixo.

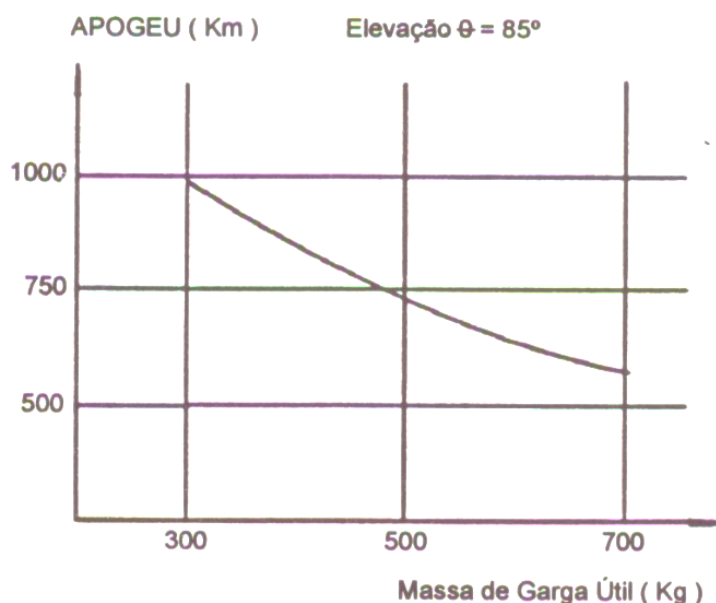


GRÁFICO 8 – APROGEU EM FUNÇÃO DA MASSA DA CARGA ÚTIL LEVADA PELO FOGUETE SONDA IV PT. 04.

Sabe-se que os dados da TABELA 2 são relativos ao quarto voo do SONDA IV, porém não se sabe a massa da carga útil, pois o voo foi tecnológico, ou seja, para testar o foguete. É provável que se tenha utilizado uma massa inerte para simular uma

carga útil. Considerou-se que a massa da carga útil está incluída na massa da estrutura do segundo estágio. O valor apresentado na TABELA 2 é consistente, pois seria 300 kg de carga útil (valor mínimo do GRÁFICO 9) e mais 103 kg de massa do envelope do motor, coifa e demais dispositivos.

Se este raciocínio está correto, então o apogeu previsto é de 867,62 km considerando o coeficiente de arrasto constante e igual a 1,0. Comparado ao apogeu real, de aproximadamente 1000 km de altitude, o resultado do SBE é 13,24% menor. Infelizmente não foi encontrada referência do coeficiente de arrasto do Sonda IV, porém se for considerado um coeficiente constante para todo o voo, com valor 0,6, então o apogeu previsto é de 992,693 km, ou seja, 0,73% menor que o valor de referência.

5 CONCLUSÃO

Com os resultados obtidos é possível concluir que o programa computacional está fornecendo resultados em acordo com o modelo matemático proposto. Conforme previsto no início do projeto, a implementação do programa de computador é mais difícil do que parece no início, principalmente a depuração dos erros de programação.

Os resultados da simulação de voo do EM Sondinha II da empresa brasileira Edge of Space, de São José dos Campos indica que o programa SBE está produzindo resultados coerentes. Como nenhuma simulação revelou um comportamento não previsto da balística do EM estudado, então o programa SBE foi considerado validado.

O estudo de erros numéricos revelou que mesmo se utilizando do Método de Euler, conhecido por ser um método de primeira ordem, ou seja, dentre as outras formulações numéricas ele é um dos que fornece dados menos exatos, ainda assim se considerados passos de tempo iguais ou menores que 10^{-3} s fornecem resultados com erros numéricos da mesma ordem de magnitude que foram considerados aceitáveis considerando a experiência prática dos autores em relação a erros de ordem experimental, conhecidos pelos membros mais experientes do Grupo de Estudos de Aerodinâmica de Veículos Autopropelidos, GREAVE, da Universidade Positivo.

Como sugestões de trabalhos futuros, recomenda-se obter dados de voo de foguetes de sondagem reais para simular lançamentos de grande altitude.

REFERÊNCIAS

FOX, Robert W.; PRITCHARD, Philip J.; MCDONALD, Alan T. **Introdução à mecânica dos fluidos**. 7.ed. Rio de Janeiro: LTC, 2011.

GREGOREK, Gerald M. **Aerodynamic Drag of Model Rockets – TR-11 Model Rocket Technical Report**. Estes Industries INC. Penrose, 1970.

GREENBERG, Michael D. **Advanced Engineering Mathematics**. 2. ed. Upper Saddle River: Prentice Hall, 1998.

MALISKA, Clovis R. **Transferência de Calor e Mecânica dos Fluidos Computacional**. 2.ed. Rio de Janeiro: LTC, 2004.

MARCHI, Carlos H. **Verificação de Soluções Numéricas Unidimensionais em Dinâmica dos Fluidos**. 305f. Tese (Doutorado em Engenharia Mecânica). Universidade Federal de Santa Catarina, Florianópolis, SC, 2001.

SUTTON, George P.; BIBLARZ, Oscar. **Rocket Propulsion Elements**. 7.ed. New York: John Wiley & Sons, 2001.

STRACK, William C; DOBSON, Wilbur F; HUFF, Vearl N. **The N-Body A General Fortran Code for Solution of Problems in Space Mechanics by Numerical Methods** NASA Technical Note TN D-1455. Cleveland: Lewis Research Center, 1963.

UNIVERSIDADE POSITIVO. **Normas para apresentação de trabalhos acadêmicos**. Curitiba: Universidade Positivo, 2011. Disponível em: <http://www.up.edu.br/CmsPositivo/uploads/imagens/galeria102/Normas_para_apresentacao_Regras_Gerais_MAR_14.pdf>. Acesso em: 31 mai. 2012.

VERSTEEG, H. K.; MALALASEKERA, W. **Computational Fluid Dynamics**, 2.ed. Harlow: Prentice Hall, 2007.

ANEXO A – CÓDIGO COMPUTACIONAL

program SBE

```
! SBE é o acrônimo p/ Simulador de Balística de Espaço Modelos.
! O objetivo deste programa é efetuar cálculos de balística de
! espaçomodelos e prever, teoricamente, o desempenho de projetos
! do Grupo de Aerodinâmica de Veículos Auto Propulsados - GREAVE
! do curso de Engenharia Mecânica da Universidade Positivo.

! Implementação da versão inicial (1.0)
!          versão atual: 2.0
!   Programadores: Jeremy Paraná Blavier e Antonio Carlos Foltran
!   Local:        GREAVE/Eng.Mec/UP, Curitiba, PR
!   Período:      07 a 19 de setembro de 2015
!                10 a 21 de janeiro de 2016
!   Última alteração: 13 de abril de 2018: Inclusão da abertura de
! paraquedas e novas funções p/ calcular as propriedades da atmosfera.
! Alterada lógica do Sonda IV p/ simular tempo de ação dos estágios.

! 1) O referencial utilizado neste programa para localizar a posição
! e medir a velocidade e aceleração do espaço-modelo é localizado na
! base da plataforma de lançamento e o eixo coordenado onde a altitude
! é medida aumenta na direção da atmosfera. Por isso a altitude é
! considerada nula quando o foguete está sobre a rampa de lançamento
! independentemente da altitude da rampa em relação ao nível do mar
! ou ao centro geométrico do planeta;
! 2) Modelo para descrever a massa específica do ar pode ser a equação
! de estado ou a Atmosfera Padrão Internacional, sendo que considerou-
! se nula a massa específica em altitudes maiores que 47 km;
! 3) Modelo para a aceleração gravitacional variável (ver subrotina
! "propriedades").
```

use portlib ! Biblioteca básica do FORTRAN.

implicit none

! Declaração das variáveis:

```
character*50 :: sim_ID ! Identificador da simulação;
character*50 :: caso_ID ! Identificador específico da simulação;
character*50 :: tab_E    ! Nome arquivo de dados de empuxo em função do tempo;
character*50 :: tab_Cd  ! Nome arquivo de dados de Cd em função do tempo;
integer :: ID_E      ! Seletor do modelo para descrever a força de empuxo;
integer :: ID_Cd    ! Seletor do modelo para descrever o coeficiente de arrasto;
integer :: ID_ro    ! Seletor do modelo para descrever a massa específica do ar;
```

```

integer :: ID_g ! Seletor do modelo para descrever a aceleração da gravidade;
integer :: ID_pq ! Seletor da variável de abertura do paraquedas;
integer :: i ! Índice identificador de um ponto da trajetória;
integer :: j ! Índice de um ponto qualquer da curva empuxo x tempo;
integer :: NpE ! Número total de pontos da curva empuxo x tempo;
integer :: NpCd ! Número total de pontos da curva Cd x V;
integer :: ite ! Contador;
integer :: itmax ! Número máximo de iterações para processo iterativo;
integer :: ver ! Variável de visualização dos resultados;
integer :: w ! Frequência de escrita dos resultados no arquivo de saída;
integer :: apogeu ! Assume valor zero e muda p/ 1 quando o apogeu é detectado;
integer :: fim_de_queima ! Assume valor zero e muda p/ 1 quando o final da queima é detectado;
integer :: abertura_paraquedas ! Assume valor 0 e muda p/ 1 qdo o paraquedas inicia a abertura;
real*8 :: Mp ! Massa de propelente [kg];
real*8 :: Mf ! Massa final [kg];
real*8 :: dr ! Diâmetro de referência do foguete [m];
real*8 :: dr_pq ! Diâmetro efetivo do paraquedas [m];
real*8 :: pi ! Constante matemática;
real*8 :: Ar ! Área de referência [m2];
real*8 :: Ar_f ! Área de referência do foguete[m2];
real*8 :: Ar_pq ! Área de referência do paraquedas[m2];
real*8 :: Ec ! Empuxo constante do motor [N];
real*8 :: tp ! Tempo de queima do propelente [s];
real*8 :: It ! Impulso total produzido pelo propulsor [Ns];
real*8 :: Cd ! Coeficiente de arrasto [adim.];
real*8 :: Cd_pq ! Coeficiente de arrasto do paraquedas aberto[adim.];
real*8 :: g0 ! Aceleração gravitacional local na superfície [m/s2];
real*8 :: g ! Aceleração gravitacional em função da altitude [m/s2];
real*8 :: ro ! Massa específica do ar [kg/m3];
real*8 :: p_atm ! Pressão atmosférica local [Pa];
real*8 :: T_atm ! Temperatura atmosférica local [°C];
real*8 :: R_ar ! Constante dos gases perfeitos do ar [J/kg.K];
real*8 :: dt ! Intervalo de tempo utilizado na discretização [s];
real*8 :: Vms ! Velocidade máxima na subida [m/s];
real*8 :: Vmd ! Velocidade máxima na descida [m/s];
real*8 :: Vi ! Velocidade do impacto [m/s];
real*8 :: t ! Instante de tempo [s];
real*8 :: th ! Instante de tempo em que o apogeu é atingido [s];
real*8 :: t_pq ! Tempo após o lançamento no qual o paraquedas é aberto [s];
real*8 :: ta_pq ! Tempo que o paraquedas leva para abrir [s];
real*8 :: ti ! Instante de tempo em que ocorre o impacto [s];
real*8 :: h ! Altitude do espaçomodelo [m];
real*8 :: hmax ! Apogeu alcançado pelo espaçomodelo [m];
real*8 :: h_pq ! Altitude após o apogeu na qual o paraquedas abre [m];
real*8 :: V ! Velocidade do espaçomodelo [m/s];
real*8 :: a ! Aceleração do espaçomodelo [m/s2];

```

```

real*8      :: M      ! Massa instantânea do espaçomodelo [kg];
real*8      :: P      ! Força peso do espaçomodelo [N];
real*8      :: E      ! Força de empuxo do propulsor [N];
real*8      :: D      ! Força de arrasto [N];
real*8      :: Cv     ! Cd [adim.];
real*8      :: t0     ! Instante de tempo na iteração anterior [s];
real*8      :: t00    ! Instante de tempo duas iterações anteriores [s];
real*8      :: a0     ! Aceleração do espaçomodelo na iteração anterior [m/s2];
real*8      :: V0     ! Velocidade do espaçomodelo na iteração anterior [m/s];
real*8      :: h0     ! Altitude do espaçomodelo na iteração anterior [m];
real*8      :: h00    ! Altitude do espaçomodelo duas iterações anteriores [m];
real*8      :: L1_a, L1_v, L1_h ! Norma L1 da aceleração, velocidade e altitude;
real*8      :: a_aux, V_aux, h_aux ! Variáveis auxiliares p/ cálculo da norma;
real*8      :: tol    ! Tolerância para encerrar o processo iterativo;
real*8,dimension(:,allocatable) :: Et ! empuxo da tabela empuxo x tempo [N];
real*8,dimension(:,allocatable) :: tt ! Instante de tempo da tabela empuxo x tempo [s];
real*8,dimension(:,allocatable) :: Ct ! Cd da tabela Cd x V [adim.];
real*8,dimension(:,allocatable) :: Vt ! V da tabela Cd x V [adim.];
! Variáveis auxiliares
integer :: bandeira, bandeira1, io
real*8  :: t1, e1, C1, V1
real*8  :: tcpu ! tempo de processamento

! Leitura do arquivo inicial do programa SBE:
call ler_dados_de_entrada

! Inicialização das variáveis:
pi = dacos(-1.0d0)
tol = 1.0d-15
apogeu = 0
fim_de_queima = 0
abertura_paraquedas = 0
Ar_f = 2.5d-1*pi*dr*dr
Ar_pq = 1.25d-1*pi*dr_pq*dr_pq
if (ID_E == 0) It=Ec*tp
R_ar = 286.9d0 ! J/(kgK)
if (ID_ro == 0) ro = p_atm / (R_ar*(T_atm+273.15d0))
if (ID_pq == 0) then
    t_pq = 0.0d0
    h_pq = 0.0d0
    ta_pq = 0.0d0
end if
if (ID_E == 1) then
    write(*,*) 'O arquivo de texto da CURVA DE EMPUXO foi aberto.'
    write(*,*) 'Se a curva de empuxo mostrada precisa ser alterada, efetue as alteracoes e salve o arquivo'
    write(*,*) 'Primeira coluna: Instante de tempo [s]. Segunda coluna: Empuxo [N]'

```

```

write(*,*)
ver = system('notepad '// tab_E)
open(1,file = tab_E)
bandeira = 0
NpE = 0
do while (bandeira == 0)
    read(1,*,iostat=io) t1, E1
    if (io < 0) then
        bandeira = 1
    else
        NpE = NpE + 1
    end if
end do
close (1)
open(1,file = tab_E)
allocate(tt(NpE), Et(NpE))
It = 0.0d0
do j = 1, NpE
    read(1,*) tt(j), Et(j)
    if (j /= 1) then
        It = It + 0.5d0*(Et(j)+Et(j-1))*(tt(j)-tt(j-1))
    end if
end do
close(1)
tp = tt(NpE)
end if

if (ID_Cd == 1) then
write(*,*) 'O arquivo de texto da CURVA DE Cd foi aberto.'
write(*,*) 'Se a curva de Cd mostrada precisa ser alterada, efetue as alteracoes e salve o arquivo'
write(*,*) 'Primeira coluna: Velocidade do foguete [m/s]. Segunda coluna: Cd [adm]'
write(*,*)
ver = system('notepad '// tab_Cd)
open(1,file = tab_Cd)
bandeira = 0
NpCd = 0
do while (bandeira == 0)
    read(1,*,iostat=io) V1, C1
    if (io < 0) then
        bandeira = 1
    else
        NpCd = NpCd + 1
    end if
end do
close (1)
open(1,file = tab_Cd)

```

```

allocate(Vt(NpCd), Ct(NpCd))
do j = 1, NpCd
    read(1,*) Vt(j), Ct(j)
end do
close(1)
end if

```

! Abertura do arquivo de saída de dados e registro de alguns dados de saída e da simulação:

```

open(2,file='Saida_de_dados_SBE.txt')
call gravar_dados_de_saida_1

```

! Desenvolvimento dos cálculos principais

```

tcpu = timef()
bandeira = 0
bandeir1 = 0
VMS = 0.0d0
VMD = 0.0d0
itmax = 100000000
t0 = 0.0d0
a0 = 0.0d0
V0 = 0.0d0
h0 = 0.0d0

```

```

write(2,18)

```

```

18 format(' >>> Fase Propulsada <<<', /, &
5x, ' i', t13, 't [s]', t29, 'h [m]', t45, 'V [km/h]', t61, 'a [g]', &
t77, 'M [g]', t93, 'P [N]', t109, 'D [N]', t125, 'E [N]', t141, 'Cd [adim.]')

```

```

t = t0
V = V0
do i = 1, itmax

```

```

t = t0 + dt

```

```

if (t < tp) then

```

```

M = Mf + Mp*(1.0d0 - t/tp)

```

!if (t < 40.0d0) then !Executa, de forma bastante improvisada, o cálculo para foguetes de dois estágios. No caso, o Sonda IV.

```

! M = Mf + Mp * ( 1 - t/40.0d0 )

```

```

!else

```

```

! if (t < 61.0d0) then

```

```

! M = 2702.0d0

```

```

! dr = 1.007d+0

```

```

! Ec = 0.0d0

```

```

! else

```

```

! if (t < 83.0d0) then

```

```

! Mf = 403.0d0

```

```

!           Mp = 879.0d0
!           M = Mf + Mp * ( 1 - (t-61.0d0)/22.0d0 )
!           Ec = 95.9d3
!           dr = 555.0d-3
!           else
!           M = 403.0d0
!           Ec = 0.0d0
!           dr = 555.0d-3
!           end if
!       end if
!end if
else
  if (fim_de_queima == 1) then
  else
    fim_de_queima = 1
    bandeira = 1 ! usada aqui p/ escrever linha de resultados no arquivo de saída dados do fim da queima.
    t = tp
  end if

  M = Mf
end if

if ( ID_E == 0 ) then
  if ( t > tp) then
    E = 0.0d0
  else
    E = Ec
  end if
else
  if ( t > tt(NpE)) then
    E = 0.0d0
  else
    do j = 1, NpE-1
      if ( t >= tt(j) .and. t < tt(j+1) ) then
        E = Et(j) + (Et(j+1)-Et(j))*(t-tt(j))/(tt(j+1)-tt(j))
        exit
      end if
    end do
  end if
end if

if (ID_pq == 1) then
  if (t <= t_pq) then
    Ar = Ar_f
  else
    if (t > (t_pq + ta_pq)) then

```

```

        Ar = Ar_pq
    else
        Ar = Ar_f + ((t-t_pq)/ta_pq) * (Ar_pq - Ar_f)
        if (abertura_paraquedas == 0) then
            h_pq = h
            abertura_paraquedas = 1
        end if
    end if
end if
else
    Ar = Ar_f
end if

a_aux = 0.0d0
V_aux = 0.0d0
h_aux = 0.0d0
do ite = 1, 100

    h = h0 + V*dt

    call propriedades

    P = M*g

    if (abertura_paraquedas == 0) then
        D = Cv*0.5d0*ro*V*V*Ar
    else
        D = Cd_pq*0.5d0*ro*V*V*Ar
    end if
    a = -g + (E - sign(D,V))/M

    if (a<0.0d0.and.h<=0.0d0.and.t<tp) a=0.0d0 ! Condição de contato com a plataforma de lançamento.

    V = V0 + dt * a

    ! calcula a norma L1 adimensionalizada de a, V e h.
    L1_a = dabs(a - a_aux)
    L1_V = dabs(V - V_aux)
    L1_h = dabs(h - h_aux)

    if (L1_a<tol .and. L1_V<tol .and. L1_h<tol) exit

    a_aux = a
    V_aux = V
    h_aux = h

```

```

end do

h = h0 + V*dt

if (V > VMS) VMS = V
if (V < VMD .and. V < -1.0d-15) VMD = V

if (h < h0 .and. apogeu == 0) then ! Apogeu detectado.
  apogeu = 1
  t = t0 - (t-t0)*V0/(V-V0)
  th = t
  h = h00 + (h0-h00)*(t-t00)/(t0-t00)
  hmax = h
  a = -g
  D = 0.0d0
  V = 0.0d0
  if ( ID_Cd == 1 ) then
    Cv = Ct(1)
  else
    Cv = Cd
  end if
  write(2,19)
  19 format(' >>> Fase Balística Descendente <<<', /, &
    5x,' i', t13,'t [s]', t29,'h [m]', t45,'V [km/h]', t61,'a [g]', &
    t77,'M [g]', t93,'P [N]', t109,'D [N]', t125,'E [N]', t141,'Cd [adim.]')
  write(2,12) i, t, h, V*3.6d+0, a/g0, M*1000, P, D, E, Cv
end if

if(ID_pq == 2) then
  if (apogeu == 1 .and. h <= h_pq) then
    if (abertura_paraquedas == 0) then
      t_pq = t
      abertura_paraquedas = 1
    end if
    if (t >= t_pq + ta_pq) then
      Ar = Ar_pq
    else
      Ar = ((t-t_pq)/ta_pq) * (Ar_f + Ar_pq)
    end if
  else
    Ar = Ar_f
  end if
end if

if (h<0.0d0) exit ! Impacto com o solo detectado.

```



```

t00 = t0
t0 = t
a0 = a
V0 = V
h00 = h0
h0 = h

if (mod(i,w)==0 .or. i==1 .or. bandeira == 1) then ! Escreve resultados no arquivo de saída.
    write(2,12) i, t, h, V*3.6d+0, a/g0, M*1000, P, D, E, Cv
    if (bandeira == 1) then
        write(2,21)
        21 format(' >>> Fase Balística Ascendente <<<', /, &
            5x, ' i', t13, 't [s]', t29, 'h [m]', t45, 'V [km/h]', t61, 'a [g]', &
            t77, 'M [g]', t93, 'P [N]', t109, 'D [N]', t125, 'E [N]', t141, 'Cd [adim.]')
        bandeira = 0
    end if
end if

end do
if (i > itmax) then
    write(*,*) 'Numero maximo de intervalos de tempo discretizados.'
    write(*,*) 'Aumentar valor da variavel dt e simular novamente.'
end if

12 format (i8, 9 (1x, 1pe15.6))
17 format (i8, 3 (1x, 1pe12.6))

t = t0 - h0*(t-t0)/(h-h0) ! Encontra o momento do impacto por interpolação.
ti = t
a = a0 - h0*(a-a0)/(h-h0)
V = V0 - h0*(V-V0)/(h-h0)
h = h0 - h0*(h-h0)/(h-h0)
Vi = V
if (V < VMD) VMD = V

write(2,12) i, t, h, V*3.6d+0, a/g0, M*1000, P, D, E, Cv

tcpu = timef() - tcpu

call gravar_dados_de_saida_2

close(2)

ver = system('Saida_de_dados_SBE.txt')

if (ID_E == 1) then

```

```

        deallocate(tt, Et)
end if
if (ID_Cd == 1) then
    deallocate(Vt, Ct)
end if
!=====

contains

!-----

subroutine ler_dados_de_entrada
ver = system('Modelo_matematico.txt')
open(1,file='Modelo_matematico.txt')
read(1,*)
read(1,*)
    read(1,*) ID_E ! Identificador do modelo para a força de empuxo (0 -> E constante; 1 -> E tabelado);
    read(1,*) ID_Cd ! Identificador do modelo para o coeficiente de arrasto (0 -> Cd constante; 1 -> Cd tabelado);
    read(1,*) ID_g ! Modelo para a aceleração da gravidade (0 -> g cte; 1-> função da altitude);
    read(1,*) ID_ro ! Modelo para a massa específica do ar (0 -> gás perfeito; 1 -> atmosfera padrão internacional);
    read(1,*) ID_pq ! Variável de comando de acionamento do paraquedas;
close(1)

open(2,file="Entrada_de_dados_SBE.txt")
write(2,*)
write(2,*) ' ARQUIVO DE ENTRADA DE DADOS DO PROGRAMA SBE, versao 2.0'
write(2,*) ' Entre com os dados no início de cada linha abaixo:'
write(2,*) ' SondinhaII    ....: sim_ID = Identificador da simulação'
write(2,*) ' voo_geral    ....: caso_ID = Descrição específica da simulação (estudo de caso)'
write(2,8) 10.0d-3, ' ....: Mp      = Massa de propelente [kg]'
write(2,8) 20.0E-3, ' ....: Mf      = Massa final do espaçomodelo [kg]'
write(2,8) 20.0E-3, ' ....: dr      = Diâmetro de referência [m]'
if (ID_E == 0) then
    write(2,8) 3.0E+0, ' ....: Ec      = Empuxo constante do motor [N]'
    write(2,8) 0.7E+0, ' ....: tp      = Tempo de queima do propelente [s]'
else
    write(2,12) 'Curva_empuxo.txt', ' ....: tab_E    = Arquivo de dados de empuxo em função do tempo'
end if
if (ID_Cd == 0) then
    write(2,8) 0.9E+0, ' ....: Cd      = Coeficiente de arrasto constante [adim.]'
else
    write(2,12) 'Curva_Cd.txt', ' ....: tab_Cd   = Arquivo de dados de Cd em função do tempo'
end if
if (ID_g == 0) then
    write(2,9) 9.806563E+0, ' .....: g0      = Aceleração gravitacional local no nível do solo [m/s2]'
end if

```

```

if(ID_ro == 0) then
    write(2,10) 101.325E+3, ' .....:p_atm    = Pressão atmosférica local [Pa]'
    write(2,8)  25.0E+0, ' ...:T_atm     = Temperatura atmosférica local [°C]'
else
end if
if (ID_pq == 1) then
    write(2,8) 5.0E+0, ' ...:t_pq      = Tempo após o lançamento no qual o paraquedas se abre [s]'
    write(2,8) 1.0E+0, ' ...:ta_pq     = Tempo de abertura do paraquedas [s]'
    write(2,8) 2.0E-1, ' ...:dr_pq     = Diâmetro efetivo do paraquedas quando aberto [m]'
    write(2,8) 1.0E+0, ' ...:Cd_pq     = Cd_pq= Coeficiente de arrasto do paraquedas [adim]'
end if
if (ID_pq == 2) then
    write(2,8) 1.0E+2, ' ...:h_pq      = Altitude após o apogeu na qual o paraquedas se abre [s]'
    write(2,8) 1.0E+0, ' ...:ta_pq     = Tempo de abertura do paraquedas [s]'
    write(2,8) 2.0E-1, ' ...:dr_pq     = Diâmetro efetivo do paraquedas quando aberto [m]'
    write(2,8) 1.0E+0, ' ...:Cd_pq     = Cd_pq= Coeficiente de arrasto do paraquedas [adim]'
end if
write(2,8) 1.0E-4, ' ...: dt       = Intervalo de tempo utilizado na discretização [s]'
write(2,11) 100, ' ...: w       = Frequência de escrita dos resultados no arquivo de saída'
write(2,*)
write(2,16) '>>> Simulador de Balística de Espaçomodelos - Versão: 2.0, Data: 13/Abr/2018 <<<'
close(2)
8 format ((1pe13.6),A)
9 format (F9.6,A)
10 format (F10.2,A)
11 format (i13,A)
12 format (A16,A)
16 format (A)

ver = system('Entrada_de_dados_SBE.txt')
open(3,file="Entrada_de_dados_SBE.txt")
read(3,*)          ! Linha em branco no arquivo de entrada de dados;
read(3,*)          ! Linha em branco no arquivo de entrada de dados;
read(3,*)          ! Linha em branco no arquivo de entrada de dados;
read(3,*) sim_ID   ! Identificador da simulação;
read(3,*) caso_ID  ! Identificador específico da simulação (estudo de caso);
read(3,*) Mp       ! Massa de propelente [kg];
read(3,*) Mf       ! Massa final do espaçomodelo [kg];
read(3,*) dr       ! Diâmetro de referência [m];
if (ID_E == 0) then ! Empuxo constante;
    read(3,*) Ec    ! Empuxo constante do motor [N];
    read(3,*) tp    ! Tempo de queima do propelente [s];
else
    ! Empuxo dado pela tabela de empuxo;
    read(3,*) tab_E ! Nome da tabela onde estão os dados do empuxo em função do tempo;
end if
if (ID_Cd == 0) then ! Cd constante;

```

```

    read(3,*)      Cd ! Coeficiente de arrasto [adim.];
else
    ! Cd variável dado pela tabela de Cd função da velocidade;
    read(3,*) tab_Cd ! Nome da tabela onde estão os dados do Cd em função da velocidade;
end if
if (ID_g == 0) then ! Aceleração da gravidade constante;
    read(3,*)      g0 ! Aceleração gravitacional local no nível do solo [m/s2];
else
    ! Aceleração da gravidade variável, função da altitude;
    g0 = 9.806563d0 ! Aceleração gravitacional local [m/s2];
end if
if (ID_ro == 0) then ! Massa específica do ar dado pelo modelo de gás perfeito;
    read(3,*)      p_atm ! Pressão atmosférica local [Pa];
    read(3,*)      T_atm ! Temperatura atmosférica local [°C];
else
    ! Massa específica do ar dado pela atmosfera padrão internacional;
    ! ver subrotina "propriedades";
end if
if (ID_pq == 1) then
    read(3,*) t_pq ! Tempo após o lançamento no qual o paraquedas se abre [s]'
    read(3,*) ta_pq ! Tempo de abertura do paraquedas [s]'
    read(3,*) dr_pq ! Diâmetro efetivo do paraquedas quando aberto [m]'
    read(3,*) Cd_pq ! Coeficiente de arrasto do paraquedas [adim]'
```

```

end if
if (ID_pq == 2) then
    read(3,*) h_pq ! Altitude após o apogeu na qual o paraquedas se abre [s]'
    read(3,*) ta_pq ! Tempo de abertura do paraquedas [s]'
    read(3,*) dr_pq ! Diâmetro efetivo do paraquedas quando aberto [m]'
    read(3,*) Cd_pq ! Coeficiente de arrasto do paraquedas [adim]'
```

```

end if
    read(3,*)      dt ! Intervalo de tempo utilizado na discretização [s];
    read(3,*)      w ! Frequência de escrita dos resultados no arquivo de saída.
    close(3)
end subroutine ler_dados_de_entrada

!-----

subroutine gravar_dados_de_saida_1
    write(2,*) '=====
    write(2,*) ' >>> SBE versão 2.0 <<<'
    write(2,*) 'Arquivo de saída de dados da simulação'
    write(2,*) '=====
    write(2,*)
    write(2,*) trim(adjustl(sim_ID)) , ' = Identificador da simulação'
    write(2,*) trim(adjustl(caso_ID)) , ' = Estudo de caso (identificador específico da simulação)'
    write(2,1) Mf, Mp, Mp+Mf, dr, tp
    1 format(1pe13.6, ' = Mf = Massa final [kg]',/, &
        1pe13.6, ' = Mp = Massa de propelente [kg]',/, &
        1pe13.6, ' = Mo = Massa inicial [kg] = Mf + Mp',/, &

```

```

        1pe13.6, ' = dr = Diâmetro de referência [m]',/, &
        1pe13.6, ' = tp = Tempo de queima do propelente [s]')
if (ID_E == 0) then
    write(2,*) 'Modelo p/ a força de empuxo: empuxo constante'
    write(2,5) Ec, It
    5 format(1pe13.6, ' = Ec = Empuxo constante do motor [N]',/, &
            1pe13.6, ' = It = Impulso total [Ns]')
else
    write(2,*) 'Modelo p/ a força de empuxo: empuxo variável conforme tabela abaixo'
end if
if (ID_Cd == 0) then
    write(2,*) 'Modelo p/ o coeficiente de arrasto: Cd constante'
    write(2,6) Cd
    6 format(1pe13.6, ' = Cd = Coeficiente de arrasto [adim.]')
else
    write(2,*) 'Modelo p/ Cd: Cd função da velocidade conforme tabela abaixo'
end if
if (ID_g == 0) then
    write(2,*) 'Modelo p/ a aceleração gravitacional: g constante'
    write(2,2) g0
    2 format(1pe13.6, ' = g0 = Aceleração gravitacional local [m/s2]')
else
    write(2,3)
    3 format(' Modelo p/ a aceleração gravitacional: g função da altitude')
    write(2,2) g0
end if
if (ID_ro == 0) then
    write(2,*) 'Modelo p/ a massa específica do ar: equação de gás perfeito'
    write(2,4) p_atm, T_atm, R_ar, ro
    4 format(1pe13.6, ' = p_atm= Pressão atmosférica local [Pa]',/, &
            1pe13.6, ' = T_atm= Temperatura atmosférica local [°C]',/, &
            1pe13.6, ' = R_ar = Constante dos gases perfeitos do ar [J/kg.K]',/, &
            1pe13.6, ' = rō = Massa específica do ar [kg/m3]')
else
    write(2,*) 'Modelo p/ a massa específica do ar: atmosfera internacional padrão'
end if
if (ID_pq == 0) write(2,*) 'Modelo p/ comando de abertura do paraquedas: sem paraquedas.'
if (ID_pq == 1) then
    write(2,*) 'Modelo p/ comando de abertura do paraquedas: tempo após lançamento'
    write(2,22) t_pq, ' = t_pq = Tempo após o lançamento no qual o paraquedas se abre [s]'
    write(2,22) ta_pq, ' = ta_pq= Tempo de abertura do paraquedas [s]'
    write(2,22) dr_pq, ' = dr_pq= Diâmetro efetivo do paraquedas quando aberto [m]'
    write(2,22) Cd_pq, ' = Cd_pq= Coeficiente de arrasto do paraquedas [adim]'
end if
if (ID_pq == 2) then
    write(2,*) 'Modelo p/ comando de abertura do paraquedas: altitude após o apogeu'

```

```

    write(2,22) h_pq,      ' = h_pq = Altitude após o apogeu na qual o paraquedas se abre [s]'
    write(2,22) ta_pq,    ' = ta_pq= Tempo de abertura do paraquedas [s]'
    write(2,22) dr_pq,    ' = dr_pq= Diâmetro efetivo do paraquedas quando aberto [m]'
    write(2,22) Cd_pq,    ' = Cd_pq= Coeficiente de arrasto do paraquedas [adim]'
end if
22 format (1pe13.6,A)
write(2,7) dt, w
7 format(1pe13.6, ' = dt = Intervalo de tempo utilizado na discretização [s]',/, &
    i13,1x, '= w = Frequência de escrita dos resultados de campo',/)
if (ID_E == 1) then
    write(2,*) '====='
    write(2,*) 'Tabela de empuxo utilizada na simulação:'
    write(2,13) 't [s]','E [N]'
    do j = 1, NpE
        write(2,14) tt(j), Et(j)
    end do
    write(2,*) 'Impulso total (regra retangulo):', It, '[Ns]'
    write(2,*)
end if
13 format (4x, A5, 20x, A5)
14 format (2(1pe25.15))
if (ID_Cd == 1) then
    write(2,*) '====='
    write(2,*) 'Tabela de coeficiente de arrasto utilizada na simulação:'
    write(2,15) 'V [m/s]','Cd [adm.]'
    do j = 1, NpCd
        write(2,14) Vt(j), Ct(j)
    end do
    write(2,*)
end if
15 format (4x, A7, 18x, A9)
write(2,*) '====='

end subroutine gravar_dados_de_saida_1

!-----
subroutine gravar_dados_de_saida_2

    write(2,*) '====='
    write(2,*)
    write(2,*) 'Dados globais do voo:'
    write(2,*)
    write(2,20) 'Tempo de voo até o apogeu:', th, ' s '
    if (ID_pq /= 0) then
        write(2,20) 'Início da abertura do paraquedas:', t_pq, ' s '
        write(2,20) 'Altitude na abertura do paraquedas:', h_pq, ' m '
    end if
end subroutine

```

```

end if
write(2,20) 'Altitude no apogeu:', hmax, ' m '
write(2,20) 'Velocidade máxima na subida:', VMS*3.6d0, ' km/h'
write(2,20) 'Velocidade máxima na descida:', VMD*3.6d0, ' km/h'
write(2,20) 'Velocidade de impacto:', Vi*3.6d0, ' km/h'
write(2,20) 'Tempo de voo até o impacto:', ti, ' s '
write(2,20) 'Tempo de processamento:', tcpu, ' s '
write(2,*)
write(2,*) '=====
write(2,*) ' >>> SBE versão 2.0 <<<'
write(2,*) ' Suportado pela Coordenação de Pesquisa e'
write(2,*) ' pela Coordenação de Engenharia Mecânica da'
write(2,*) ' Universidade Positivo - UP, Curitiba/PR'
write(2,*) ' Programa de Iniciação Científica - PIC1434/15'
write(2,*) ' Programadores: Jeremy Blavier Paraná'
write(2,*) ' Antonio Carlos Foltran'
write(2,*) '=====
20 format(A35, lpe25.15, A5)

```

```
end subroutine gravar_dados_de_saida_2
```

```
!-----
```

```
subroutine propriedades
```

```
real*8 :: Temp ! temperatura [K];
```

```
real*8 :: pres ! pressão [Pa];
```

```
real*8 :: aux
```

```
if ( ID_Cd == 1 ) then
```

```
do j = 1, NpCd-1
```

```
if ( dabs(V) >= Vt(j) .and. dabs(V) < Vt(j+1) ) then
```

```
Cv = Ct(j) + (Ct(j+1)-Ct(j))*(dabs(V)-Vt(j))/(Vt(j+1)-Vt(j))
```

```
exit
```

```
end if
```

```
end do
```

```
if ( dabs(V) > Vt(NpCd) .and. bandeiral == 0) then
```

```
write(*,*) '*****'
```

```
write(*,*) 'ATENCAO: Cd requerido alem dos valores da tabela'
```

```
write(*,*) '*****'
```

```
bandeiral = 1
```

```
end if
```

```
else
```

```
Cv = Cd
```

```
end if
```

```
if (ID_g == 1) then
```

```
g = 9.806563d0*(6378137.0d0 / (6378137.0d0 + h))**2.0d0
```

```
!g = 3.989371d14 / ((6378137d0 + h)**2.0d0)
```

```

else
    g = g0
end if

if (ID_ro == 1) then
!       if (h < 11.0d3) then ! troposfera    ! Fórmulas extraídas de CAVCAR, Mustafa. The International Standard Atmosphere
!       (ISA). Eskisehir, TR: Anadolu University. Documento não publicado.
!       Temp = 288.15d0 - 6.5d0*h/1.0d3
!       pres = 101325.0d0*(1.0d0 - 6.5d-3*h/288.15d0)**5.2561d0
!       ro = pres/(287.04d0*Temp)
!
!       else
!       if (h >= 11.0d3 .and. h < 20.0d3) then ! tropopausa
!       Temp = 216.65d0
!       pres = 22630.61892d0*dexp(-9.80665d0*(h - 11.0d3)/(287.04d0*216.65d0))
!       ro = pres/(287.04d0*Temp)
!       else ! acima da tropopausa
!       ro = 0.0d0
!       end if
!       end if
!       if (h < 47.0d+3) then ! Fórmulas extraídas do capítulo 1 de DRELA, Mark. Flight Vehicle Aerodynamics, Cambridge: The
MIT Press, 2014.
aux = 1.0d-3*h
Temp = 216.65d0 + 2.0d0*dlog(1.0d0 + dexp(35.75d0 - 3.25d0*aux) + dexp(-3.0d0 + 3.0d-4*aux**3.0d0))
pres = 1.0132d5*dexp(-0.118d0*aux - 0.0015d0*aux**2.0d0/(1.0d0-0.018d0*aux+0.0011d0*aux**2.0d0))
ro = pres/(287.04d0*Temp)
else
    ro = 0.0d0
end if
end if

end subroutine propriedades
end program SBE

```


ANEXO B – ARQUIVOS .TXT UTILIZADOS PELO PROGRAMA

ARQUIVO “Modelo_matematico.txt”:

SBE

Entre com as opções de modelos matemáticos para a simulação:

- 0 = Identificador do modelo para a força de empuxo (0 -> E constante; 1 -> E tabelado);
- 0 = Identificador do modelo para o coeficiente de arrasto (0 -> Cd constante; 1 -> Cd tabelado);
- 1 = Modelo para a aceleração da gravidade (0 -> g cte; 1-> função da altitude);
- 0 = Modelo para a massa específica do ar (0 -> gás perfeito; 1 -> US Standard Atmosphere 1976);
- 0 = Abertura do paraquedas comandada por (1 -> tempo após lançamento; 2 -> altitude após o apogeu; 0 -> sem paraquedas).

ARQUIVO “Entrada_de_dados_SBE.txt”:

ARQUIVO DE ENTRADA DE DADOS DO PROGRAMA SBE, versao 2.0

Entre com os dados no início de cada linha abaixo:

SondinhaII ...: sim_ID = Identificador da simulação
voo_geral ...: caso_ID = Descrição específica da simulação (estudo de caso)
1.000000E-02 ...: Mp = Massa de propelente [kg]
2.000000E-02 ...: Mf = Massa final do espaçomodelo [kg]
2.000000E-02 ...: dr = Diâmetro de referência [m]
3.000000E+00 ...: Ec = Empuxo constante do motor [N]
7.000000E-01 ...: tp = Tempo de queima do propelente [s]
9.000000E-01 ...: Cd = Coeficiente de arrasto constante [adim.]
101325.00:p_atm = Pressão atmosférica local [Pa]
2.500000E+01 ...:T_atm = Temperatura atmosférica local [°C]
1.000000E-04 ...: dt = Intervalo de tempo utilizado na discretização [s]
100 ...: w = Frequência de escrita dos resultados no arquivo de saída

>>> Simulador de Balística de Espaçomodelos - Versão: 2.0, Data: 13/Abr/2018 <<<

ARQUIVO “Curva_empuxo.txt”:

Desde a primeira linha do arquivo devem-se escrever duas colunas, a primeira com os dados do tempo [s] e a segunda com dados da força de empuxo produzida pelo motor [N]. O número de linhas deve ser o número de medições experimentais de força de empuxo obtidas durante experimentos em banco estático.

ARQUIVO "Curva_Cd":

Idem ao arquivo descrito anteriormente porém agora a primeira coluna deverá conter a velocidade do EM em m/s e na segunda o coeficiente de arrasto, que é uma grandeza adimensional.