

Relatório de simulação da manobra orbital *fly-by*

Gabriel Vinicius Ferreira

1 de dezembro de 2025

1 Introdução

Neste relatório, apresentado para a disciplina de Simulação Computacional em Física (CF1829, 2-2025), abordamos a simulação da manobra orbital *fly-by* — também conhecida como assistência gravitacional. Essa técnica é utilizada por sondas espaciais para alterar sua velocidade e trajetória aproveitando a gravidade de um corpo celeste [1]. Em termos simples, funciona como uma “carona gravitacional”: ao interagir com o campo gravitacional do corpo massivo, a nave consegue modificar sua velocidade sem gastar combustível. Como o transporte de propelente no espaço é limitado e custoso, manobras desse tipo são estratégicas e amplamente usadas em missões de longa duração. Entre as missões que empregaram assistência gravitacional estão as sondas *Voyager*, *Parker Solar Probe*, *Solar Orbiter*, *MESSENGER*, *Psyche* e *BepiColombo* [2].

Para este trabalho utilizamos duas simulações de *fly-by*: uma baseada no problema restrito de dois corpos e outra no problema restrito de três corpos. Em ambos os casos, consideramos apenas a força gravitacional, dada por [3]

$$\vec{F}(\vec{r}) = -\frac{GMm}{|\vec{r}|^3} \vec{r}, \quad (1)$$

onde M é a massa do corpo que gera o campo gravitacional, m é a massa do objeto que sofre a ação da gravidade, \vec{r} é o vetor distância entre eles e G é a constante gravitacional universal ($G \approx 6,6743 \times 10^{-11} \text{ m}^3 \text{kg}^{-1} \text{s}^{-2}$). Para as simulações apresentadas aqui, tomamos como corpos massivos Marte, com massa $M_M \approx 6,4171 \times 10^{23} \text{ kg}$, e o Sol, com massa $M_\odot \approx 1,9885 \times 10^{30} \text{ kg}$.

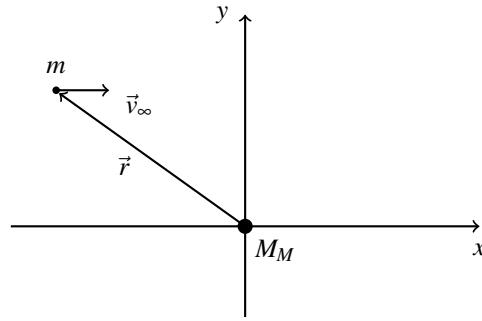


Figura 1: Representação do problema restrito de dois corpos. Marte é indicado pelo corpo de massa M_M e a sonda pelo corpo de massa m . A distância entre eles é dada pelo vetor \vec{r} . Para simplificar o cenário, a sonda se aproxima a partir de $x = -\infty$ com velocidade \vec{v}_∞ , que possui apenas componente positiva em x . A simulação descreve a interação dessa sonda com o campo gravitacional gerado por M_M .

O problema restrito de dois corpos, ilustrado na Figura 1, considera um corpo massivo de massa M fixo no centro do sistema de coordenadas e um corpo de teste de massa m aproximando-se com certa velocidade. Tratar M_M como fixo é, até certo ponto, equivalente a descrever o movimento relativo do corpo m no referencial de M_M . A interação entre eles é simples: o corpo de teste responde ao campo gravitacional descrito pela Eq. (1), mas não exerce influência apreciável sobre o corpo massivo — daí o termo “restrito” no nome do modelo.

Nesse cenário, o *fly-by* resulta em uma trajetória hiperbólica que pode ser analisada geometricamente, conforme discutido em [4]. Apesar de fisicamente limitado — já que desconsidera a presença do Sol e o movimento orbital de Marte, no contexto deste trabalho —, a própria simplicidade do modelo, aliada à sua descrição analítica, torna o *fly-by* no regime de dois corpos uma ferramenta útil no ensino de física orbital e da simulação computacional em física.

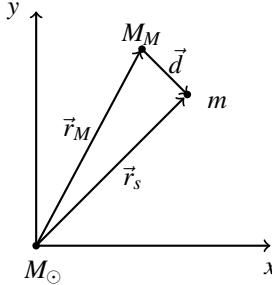


Figura 2: Representação do problema de três corpos. O objeto de massa m corresponde à sonda, M_M representa Marte e M_\odot o Sol. As distâncias entre esses corpos são dadas pelos vetores \vec{d} , \vec{r}_s e \vec{r}_M , como indicado na figura.

O problema restrito de três corpos é semelhante ao caso de dois corpos, como mostrado na Figura 2. Aqui, o Sol é colocado na origem do sistema de coordenadas e dois corpos se movem sob sua influência gravitacional sem afetar sua posição. Além disso, assumimos que, embora Marte seja afetado pelo Sol, ele não sofre influência da sonda, mantendo o caráter “restrito” do modelo.

A formulação desse problema é mais elaborada e demanda o uso do formalismo lagrangiano. Definimos a Lagrangiana do sistema como [3]

$$\mathcal{L} = T - V, \quad (2)$$

em que T é a energia cinética e V é a energia potencial. A partir dessa definição, as equações de movimento são obtidas por meio da equação de Euler–Lagrange [3]:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_k} \right) - \frac{\partial \mathcal{L}}{\partial q_k} = 0, \quad (3)$$

onde q_k e \dot{q}_k formam o k -ésimo par de coordenadas generalizadas, cuja substituição resulta nas equações de movimento do sistema. O motivo disso é que esse formalismo facilita a obtenção das equações de movimento quando o sistema envolve múltiplos potenciais e coordenadas acopladas; caso do problema de três corpos.

Alguns termos importantes para a análise de um *fly-by* são o parâmetro de impacto e o ângulo de deflexão. O parâmetro de impacto b é a distância perpendicular entre a trajetória inicial da sonda — caso ela seguisse em linha reta — e o centro do corpo responsável pela deflexão gravitacional [1]. No problema restrito de dois corpos, podemos expressá-lo de forma geométrica como [4]

$$b^2 = r_{min}^2 + \frac{2GM_{min}}{v_\infty^2}, \quad (4)$$

onde r_{min} é a distância de maior aproximação e v_∞ é a velocidade da sonda no infinito, ou seja, em uma situação onde ela não estaria sob a influência do campo gravitacional do planeta.

O ângulo de deflexão indica o quanto a trajetória da sonda se curva devido à interação com o campo gravitacional do planeta [1]. Ele é definido pelo ângulo entre a direção de entrada e a direção de saída da sonda. Para o modelo restrito de dois corpos, esse ângulo pode ser relacionado à velocidade no infinito e ao parâmetro de impacto [1, 4]:

$$\tan\left(\frac{\delta}{2}\right) = \frac{GM}{bv_\infty^2}. \quad (5)$$

O objetivo deste trabalho é analisar o *fly-by* simulado pelo modelo restrito de dois corpos, identificando suas limitações, e comparar esses resultados com a simulação baseada no problema restrito de três corpos. A intenção

é avaliar até que ponto o modelo de dois corpos pode ser considerado adequado para o ensino de física orbital e de simulação computacional. Além disso, seguindo a orientação do professor, adotamos o método de integração de Euler, conforme apresentado na apostila da disciplina [5].

2 Metodologia

Para a simulação baseada no problema restrito de dois corpos, escrevemos o sistema em coordenadas cartesianas:

$$\vec{r} = \begin{pmatrix} x \\ y \end{pmatrix}, \quad |\vec{r}| = \sqrt{x^2 + y^2}. \quad (6)$$

Com essa notação, podemos reescrever a Eq. 1 como

$$\vec{F}(x, y) = -\frac{GMm}{(x^2 + y^2)^{3/2}} \begin{pmatrix} x \\ y \end{pmatrix}. \quad (7)$$

Aplicando a segunda lei de Newton à força gravitacional, obtemos as equações diferenciais de segunda ordem

$$\ddot{x} = -\frac{GMx}{(x^2 + y^2)^{3/2}}, \quad (8)$$

$$\ddot{y} = -\frac{GMy}{(x^2 + y^2)^{3/2}}, \quad (9)$$

que podem ser convertidas em quatro equações de primeira ordem ao definirmos $\dot{x} = v_x$ e $\dot{y} = v_y$:

$$\dot{x} = v_x, \quad (10)$$

$$\dot{v}_x = -\frac{GMx}{(x^2 + y^2)^{3/2}}, \quad (11)$$

$$\dot{y} = v_y, \quad (12)$$

$$\dot{v}_y = -\frac{GMy}{(x^2 + y^2)^{3/2}}. \quad (13)$$

Para realizar a simulação numérica, aplicamos o método de Euler. Assim, cada equação diferencial passa a ser aproximada por

$$x(t + \Delta t) = x(t) + v_x(t)\Delta t, \quad (14)$$

$$v_x(t + \Delta t) = v_x(t) - \frac{GMx(t)}{(x(t)^2 + y(t)^2)^{3/2}}\Delta t, \quad (15)$$

$$y(t + \Delta t) = y(t) + v_y(t)\Delta t, \quad (16)$$

$$v_y(t + \Delta t) = v_y(t) - \frac{GMy(t)}{(x(t)^2 + y(t)^2)^{3/2}}\Delta t. \quad (17)$$

Para as condições iniciais, consideramos a sonda vindo de um “infinito” negativo e se aproximando da região de influência de Marte. Assim, tomamos $x(0) = -50R_{Marte}$, onde $R_{Marte} \approx 3,3895 \times 10^6$ m é o raio de Marte, enquanto a coordenada inicial em y é definida como $y(0) = b$, sendo b o parâmetro de impacto.

A escolha de $50R_{Marte}$ busca equilibrar dois efeitos: valores menores reduzem o domínio de integração, mas podem acentuar erros provenientes da aproximação física (a sonda não estaria distante o suficiente para que a velocidade inicial seja realmente assintótica); por outro lado, valores muito maiores diminuem esse erro matemático, mas aumentam a duração da integração e, com isso, o acúmulo de erros numéricos característico do método de Euler. O valor adotado representa um compromisso razoável entre essas duas limitações.

Tomando a velocidade inicial no eixo y igual a zero, ou seja, $v_y(0) = 0$, podemos assumir que a velocidade inicial no eixo x seja aproximadamente a velocidade no infinito v_∞ , acrescida do ganho correspondente à conversão da energia potencial gravitacional na posição inicial. Modelamos isso pela conservação de energia do sistema:

$$|\vec{v}(t)| = +\sqrt{v_\infty^2 + \frac{2GM}{|\vec{r}(t)|}}, \quad (18)$$

onde adotamos $\vec{v}_\infty = 2,6 \text{ km/s}$, valor que produz uma trajetória hiperbólica bem definida neste modelo. Além disso, usamos a aproximação $|\vec{r}(0)| \simeq |x(0)|$, válida quando escolhemos b pequeno em comparação com $x(0)$; dessa forma, $\vec{v}(0)$ possui apenas componente em x .

Portanto, as condições iniciais utilizadas no problema são

$$x(0) = -50R_{Marte}, \quad (19)$$

$$y(0) = b, \quad (20)$$

$$v_x(0) = +\sqrt{v_\infty^2 + \frac{2GM}{|x(0)|}}, \quad (21)$$

$$v_y(0) = 0. \quad (22)$$

Para explorar tanto trajetórias desviadas quanto casos de colisão, adotamos para o parâmetro de impacto o intervalo $b \in [-10R_{Marte}; 10R_{Marte}]$. Valores positivos correspondem à sonda iniciando acima do eixo x , enquanto valores negativos representam trajetórias iniciadas abaixo do eixo. O intervalo simétrico em b permite verificar se a solução numérica preserva a simetria do problema em torno do eixo x .

Para analisar o *fly-by* no modelo de dois corpos, observamos a distância mínima entre a sonda e Marte, r_{min} , obtida de maneira numérica como o menor valor de $|\vec{r}(t)|$ ao longo da trajetória simulada. Podemos então comparar esse resultado com a previsão teórica, reescrevendo a Eq. (4) na forma

$$r_{min} = -\frac{GM}{v_\infty^2} + \sqrt{\left(\frac{GM}{v_\infty^2}\right)^2 + b^2}. \quad (23)$$

Além disso, avaliamos o erro numérico do método de Euler comparando o módulo da velocidade relativa de entrada no infinito com o módulo da velocidade de saída. Definimos

$$\Delta v = |\vec{v}_{rel,out}| - |\vec{v}_{rel,in}|. \quad (24)$$

Do ponto de vista teórico, como o sistema é conservativo – Marte está fixa e trabalhamos sempre com velocidades relativas da sonda em relação ao planeta –, as velocidades assintóticas de entrada e saída deveriam ter o mesmo módulo, pois a energia potencial tende a zero no infinito e a energia cinética deve ser preservada. Em outras palavras, espera-se $\Delta v = 0$.

Na prática, porém, isso não ocorre devido ao erro numérico acumulado pelo método de Euler. O valor de $|\vec{v}_{\infty,out}|$ é corrigido a partir da conservação de energia, conforme a Eq. (18) para minimizar esse desvio, garantindo que a comparação com a teoria não seja dominada pelo erro energético do método.

Por fim, o ângulo de deflexão pode ser obtido numericamente por meio do produto interno entre os vetores de velocidade relativa de entrada e de saída da sonda no referencial do planeta:

$$\delta = \arccos \left(\frac{\vec{v}_{in, rel} \cdot \vec{v}_{out, rel}}{|\vec{v}_{in, rel}| |\vec{v}_{out, rel}|} \right). \quad (25)$$

Com esses valores em mãos, comparamos o resultado numérico com a previsão teórica dada pela Eq. (5), que pode ser reescrita como

$$\delta = 2 \arctan \left(\frac{GM}{bv_\infty^2} \right). \quad (26)$$

Como precisamos determinar $\vec{v}_{\infty,out}$, é essencial que a sonda já tenha se afastado o suficiente de Marte. Por isso, adotamos como critério de parada da integração o instante em que $|\vec{r}(t)| > 50R_{Marte}$, seguindo o mesmo valor utilizado para a condição inicial.

Caso a trajetória indique colisão – isto é, quando $|\vec{r}(t)| < R_{Marte}$ –, a simulação é interrompida imediatamente e o valor correspondente de b é registrado como parâmetro de impacto crítico – ou seja, que gera uma colisão.

Também incluímos um critério de segurança associado ao tempo de integração, $t_{max} = 1 \times 10^{10}s$. Esse limite, entretanto, não foi atingido em nenhum dos testes realizados.

Por fim, para reduzir os erros acumulados pelo método de Euler, utilizamos um passo temporal $\Delta t = 0,001s$, registrando saídas de dados aproximadamente a cada 3 minutos de tempo simulado.

Partindo para a simulação baseada no problema restrito de três corpos, para formular o problema, utilizamos coordenadas polares, nas quais

$$\vec{r} = r\hat{e}_r + \theta\hat{e}_\theta, \quad (27)$$

com $\hat{e}_r = \cos\theta \hat{i} + \sin\theta \hat{j}$ e $\hat{e}_\theta = -\sin\theta \hat{i} + \cos\theta \hat{j}$. Com essa base, construímos a Lagrangiana que descreve a interação entre os três corpos, usando a definição da Eq. (2):

$$\mathcal{L} = \frac{M_M(r_M^2 + r_M^2\dot{\theta}_M^2)}{2} + \frac{m(r_s^2 + r_s^2\dot{\theta}_s^2)}{2} + \frac{GM_\odot M_M}{r_M} + \frac{GM_\odot m}{r_s} + \frac{GM_M m}{\sqrt{r_s^2 + r_M^2 - 2r_s r_M \cos(\theta_s - \theta_M)}}. \quad (28)$$

Aplicando a Eq. (3) à Lagrangiana obtida e realizando algumas manipulações algébricas, chegamos às equações de movimento do sistema:

$$\dot{\theta}_M = \sqrt{\frac{GM_\odot}{r_M^3}}, \quad (29)$$

$$\dot{v}_s = r_s\omega_s^2 - \frac{GM_\odot}{r_s^2} - \frac{GM_M[r_s - r_M \cos(\theta_s - \theta_M)]}{[r_s^2 + r_M^2 - 2r_s r_M \cos(\theta_s - \theta_M)]^{3/2}}, \quad (30)$$

$$\dot{r}_s = v_s, \quad (31)$$

$$\dot{\omega}_s = -\frac{GM_M r_M \sin(\theta_s - \theta_M)}{r_s[r_s^2 + r_M^2 - 2r_s r_M \cos(\theta_s - \theta_M)]^{3/2}} - \frac{2v_s\omega_s}{r_s}, \quad (32)$$

$$\dot{\theta}_s = \omega_s. \quad (33)$$

Aqui, θ_M representa a posição angular de Marte, assumida em órbita circular com raio constante $r_M \approx 2,2794 \times 10^{11} m$. As variáveis (r_s, θ_s) correspondem à posição da sonda, enquanto (v_s, ω_s) representam sua velocidade radial e angular.

Aplicando o método de Euler, obtemos as equações de evolução que utilizamos na simulação numérica:

$$\theta_M(t + \Delta t) = \theta_M(t) + \sqrt{\frac{GM_\odot}{r_M^3}} \Delta t, \quad (34)$$

$$r_s(t + \Delta t) = r_s(t) + v_s(t) \Delta t, \quad (35)$$

$$v_s(t + \Delta t) = v_s(t) + \left\{ r_s(t) \omega_s(t)^2 - \frac{GM_\odot}{r_s(t)^2} - \frac{GM_M[r_s(t) - r_M \cos(\theta_s(t) - \theta_M(t))]}{[r_s(t)^2 + r_M^2 - 2r_s(t)r_M \cos(\theta_s(t) - \theta_M(t))]^{3/2}} \right\} \Delta t, \quad (36)$$

$$\theta_s(t + \Delta t) = \theta_s(t) + \omega_s(t) \Delta t, \quad (37)$$

$$\omega_s(t + \Delta t) = \omega_s(t) + \left\{ -\frac{GM_M r_M \sin(\theta_s(t) - \theta_M(t))}{r_s(t)[r_s(t)^2 + r_M^2 - 2r_s(t)r_M \cos(\theta_s(t) - \theta_M(t))]^{3/2}} - \frac{2v_s(t)\omega_s(t)}{r_s(t)} \right\} \Delta t. \quad (38)$$

Como as próprias equações diferenciais deixam evidente quando comparadas com o modelo de dois corpos, essa simulação é significativamente mais complexa. Para tornar a comparação entre os dois modelos um pouco mais consistente, definimos o parâmetro de impacto no mesmo intervalo: $b \in [-10R_{Marte}; 10R_{Marte}]$.

Seguindo as sugestões do professor da disciplina, podemos definir o parâmetro de impacto neste problema de duas formas:

1. pela distância mínima entre a sonda e Marte ao longo da simulação;
2. ou pela distância perpendicular entre a sonda, ao entrar na região de influência de Marte – definida como a bola de raio $50R_M$ centrada no planeta –, e a trajetória que o planeta percorre naquele instante.

Como não temos como impor o valor desejado da distância mínima no início da simulação – ela só é conhecida depois que a trajetória completa é calculada –, optamos pela segunda definição. Ela permite determinar diretamente a posição e a velocidade iniciais da sonda a partir da posição e velocidade iniciais de Marte.

Para Marte, adotamos a condição inicial $\theta_M(0) = -0,01^\circ$, posicionando o planeta próximo ao eixo horizontal, mas com uma pequena margem angular. Apenas como referência, para o raio orbital de Marte, essa escolha implica uma distância inicial até o eixo de aproximadamente $3,98 \times 10^7 m$.

A Eq. (29) indica que Marte possui apenas velocidade angular – resultado da aproximação da órbita do planeta por uma órbita circular –, que pode ser convertida para coordenadas cartesianas por

$$v_{M,x}(t) = -\sqrt{\frac{GM_\odot}{r_M}} \sin[\theta_M(t)], \quad (39)$$

$$v_{M,y}(t) = \sqrt{\frac{GM_\odot}{r_M}} \cos[\theta_M(t)]. \quad (40)$$

Também podemos escrever sua posição em coordenadas cartesianas como:

$$x_M(t) = r_M \cos[\theta_M(t)], \quad (41)$$

$$y_M(t) = r_M \sin[\theta_M(t)]. \quad (42)$$

Com isso, temos definidas as condições iniciais para Marte. Para a sonda, a construção é um pouco mais trabalhosa, mas pode ser resolvida com geometria. A ideia é considerar que Marte possui uma “circunferência de influência” de raio $50R_M$, e queremos posicionar a sonda sobre essa circunferência a uma distância perpendicular b da trajetória do planeta – isto é, da reta paralela à sua velocidade. A Figura 3 ilustra essa construção.

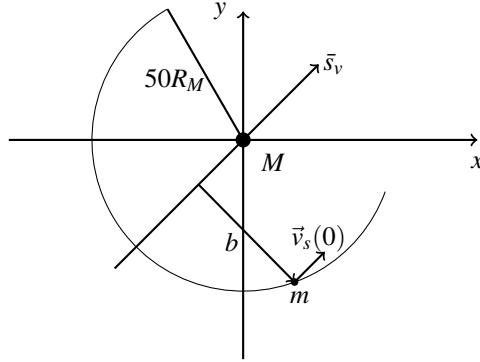


Figura 3: Representação das condições iniciais da sonda nos entornos de Marte. A circunferência indica a região que consideramos como domínio de influência do campo gravitacional de Marte, definida por um raio de $50R_M$ centrado no planeta. A partir de Marte, traçamos a reta \bar{s}_v , paralela à velocidade do planeta \vec{v}_M . Sobre essa construção, posicionamos a sonda – de massa m – em um ponto da circunferência cuja distância perpendicular até \bar{s}_v é igual ao parâmetro de impacto b . Adotamos a convenção de que $b > 0$ quando a sonda está cima de \bar{s}_v e $b < 0$ quando está abaixo (no caso do exemplo da figura, b adotaria um valor negativo). A velocidade inicial da sonda, $\vec{v}_s(0)$, é definida paralela à reta \bar{s}_v .

A direção da trajetória de Marte é dada pelo vetor unitário de sua velocidade:

$$\hat{u} = \frac{\vec{v}_M}{|\vec{v}_M|} = (-\sin[\theta_M(t)], \cos[\theta_M(t)]), \quad (43)$$

e, rotacionando esse vetor em 90° , obtemos a direção perpendicular:

$$\hat{n} = (-\cos[\theta_M(t)], -\sin[\theta_M(t)]). \quad (44)$$

Assim, a posição relativa da sonda em relação a Marte pode ser escrita na base (\hat{u}, \hat{n}) :

$$\vec{d} = \vec{r}_s - \vec{r}_M = \alpha\hat{u} + \beta\hat{n}, \quad (45)$$

onde $\beta = b$ e $\alpha = -\sqrt{(50R_M)^2 - b^2}$, garantindo que a sonda sempre comece “atrás” de Marte ao longo da direção do movimento – de forma semelhante ao que foi imposto no problema de dois corpos.

A partir dessa construção, a posição inicial da sonda em coordenadas cartesianas é

$$x_s(0) = x_M(0) + \sqrt{(50R_M)^2 - b^2} \sin[\theta_M(0)] - b \cos[\theta_M(0)], \quad (46)$$

$$y_s(0) = y_M(0) - \sqrt{(50R_M)^2 - b^2} \cos[\theta_M(0)] - b \sin[\theta_M(0)]. \quad (47)$$

Como a integração numérica é feita em coordenadas polares, convertemos esses valores para (r_s, θ_s) :

$$r_s(0) = \sqrt{[x_s(0)]^2 + [y_s(0)]^2}, \quad (48)$$

$$\theta_s(0) = \text{atan2}[x_s(0), y_s(0)] \equiv \arg[x_s(0) + i y_s(0)]. \quad (49)$$

Por conveniência, não expandimos essas expressões: os valores são calculados em coordenadas cartesianas e depois convertidos numericamente. Essa construção geométrica das condições iniciais preserva o significado do parâmetro de impacto no limite de fraca influência solar, o que cria um paralelo direto com o caso de dois corpos e facilita a comparação entre os modelos.

Definidas a posição e a velocidade iniciais de Marte, bem como a posição inicial da sonda, resta especificar a velocidade inicial da sonda. Começamos pela velocidade relativa em relação a Marte. Segundo a ideia usada no problema de dois corpos, adotamos uma velocidade de módulo igual a $|\vec{v}_{rel}| = v_x(0)$ – em que $v_x(0)$ é dado pela Eq. (21) – e paralela à trajetória de Marte, isto é, na direção de \hat{u} . Assim,

$$v_{rel, x}(0) = -|\vec{v}_{rel}| \sin[\theta_M(0)], \quad (50)$$

$$v_{rel, y}(0) = |\vec{v}_{rel}| \cos[\theta_M(0)]. \quad (51)$$

Trocando o referencial de Marte para o referencial do Sol, obtemos

$$v_{s, x}(0) = -\left(\sqrt{\frac{GM_\odot}{r_M}} + |\vec{v}_{rel}|\right) \sin[\theta_M(0)], \quad (52)$$

$$v_{s, y}(0) = \left(\sqrt{\frac{GM_\odot}{r_M}} + |\vec{v}_{rel}|\right) \cos[\theta_M(0)], \quad (53)$$

que podem ser convertidas para o sistema de coordenadas polar utilizando:

$$v_{s, r}(0) = \frac{x_s(0)v_{s, x}(0) + y_s(0)v_{s, y}(0)}{r_s(0)}, \quad (54)$$

$$v_{s, \theta}(0) = \frac{x_s(0)v_{s, y}(0) - y_s(0)v_{s, x}(0)}{[r_s(0)]^2}. \quad (55)$$

As condições de parada da simulação seguem as mesmas adotadas no problema de dois corpos: interrompemos a integração quando a distância relativa entre a sonda e Marte ultrapassa $50R_M$, saindo do domínio de influência considerado, ou quando ocorre colisão com o planeta. Também mantemos o critério de segurança associado ao tempo máximo de integração, $t_{max} = 1 \times 10^{10} s$. De forma análoga ao caso anterior, utilizamos $\Delta t = 0,001 s$, registrando saídas de dados a cada ~ 3 minutos de tempo simulado.

Para o modelo restrito de três corpos, não realizamos uma comparação analítica direta; em vez disso, a análise é qualitativa, contrastando o comportamento obtido aqui com o do modelo restrito de dois corpos. Além disso, podemos investigar características internas do sistema, como a variação da velocidade heliocêntrica da sonda:

$$\Delta v = |\vec{v}_s(t_{fim})| - |\vec{v}_s(0)|, \quad (56)$$

expressão semelhante à Eq. (24), mas sem utilizar a velocidade relativa ao planeta e sem a expectativa teórica de $\Delta v = 0$. No referencial heliocêntrico, a energia da sonda não é necessariamente conservada, pois ela pode ceder energia ao campo gravitacional do Sol ou receber energia dele ao longo do processo. Isso também reflete o fato de que, no referencial solar, a trajetória da sonda é elíptica, enquanto no referencial do planeta a órbita associada ao *fly-by* é hiperbólica.

Outra quantidade útil é a variação da velocidade relativa entre a sonda e Marte:

$$\Delta v_{rel} = |\vec{v}_s(t_{fim}) - \vec{v}_M(t_{fim})| - |\vec{v}_s(0) - \vec{v}_M(0)|. \quad (57)$$

Nesse caso, é particularmente interessante verificar se o sistema atende ou não à condição $\Delta v_{rel} = 0$, já que isso indica até que ponto o modelo reproduz ou viola a conservação de energia relativa prevista no problema de dois corpos. Esse diagnóstico também evidencia um ponto importante: embora o *fly-by* sempre altera a direção do movimento relativo, ele não necessariamente modifica o módulo da velocidade relativa.

Obviamente, a distância mínima entre a sonda e Marte é uma grandeza simples de calcular numericamente, então também a incluímos na análise. Por fim, o ângulo de deflexão pode ser obtido pela Eq. (25), sendo interessante verificar se os valores numéricos se aproximam – ou não – daqueles esperados a partir do modelo de dois corpos.

Os códigos desenvolvidos a partir da revisão teórica e da metodologia descrita até então neste trabalho estão disponíveis no GitHub: <https://github.com/gabriel-ferr/Fly-by>. No repositório, fornecemos um pequeno guia de execução e a lista das dependências necessárias. A simulação foi implementada em C, enquanto os gráficos foram gerados em Julia usando a biblioteca GLMakie. Os arquivos referentes ao problema restrito de dois corpos possuem o prefixo pr2c, enquanto os referentes ao problema restrito de três corpos utilizam o prefixo pr3c. Também foram colocados no anexo deste trabalho os trechos dos códigos das simulações em C responsáveis pelo cálculo do movimento.

3 Resultados

A seguir, apresentamos os resultados das simulações realizadas e as comparações entre os modelos propostos. Foram executados 240 testes ao longo do intervalo definido para o parâmetro de impacto, a partir dos quais analisamos os dados globais obtidos. Antes disso, porém, vale fazer uma observação rápida sobre as trajetórias calculadas – um aspecto visual interessante da simulação do *fly-by* e que ajuda a contextualizar o comportamento dinâmico observado.

A Figura 4 mostra as trajetórias da sonda em sua interação com Marte no modelo restrito de dois corpos. Os painéis (a), (b) e (c) correspondem a casos em que a sonda inicia abaixo do eixo horizontal, enquanto os painéis (d), (e) e (f) mostram situações em que ela começa acima do eixo.

Os painéis (c) e (d) ilustram cenários de colisão com o planeta. Neles, observamos pequenos “saltos” no final da trajetória, comportamento esperado devido ao uso do método de Euler com passo de tempo fixo e à natureza “dura” do sistema quando a sonda se aproxima muito do corpo massivo – região em que o campo gravitacional varia rapidamente e tende a amplificar erros numéricos.

De maneira semelhante, a Figura 5 apresenta as trajetórias obtidas para a simulação com o modelo restrito de três corpos. Nos painéis (a), (b), e (c), a sonda inicia à direita de Marte, o que é indicado pelos valores negativos do parâmetro de impacto – isto é, a sua posição inicial está abaixo da linha paralela ao vetor velocidade de Marte, como mostrado na Figura 3. De forma semelhante, os painéis (d), (e) e (f) correspondem a casos em que a sonda começa à esquerda do planeta, o que equivale a estar acima dessa linha imaginária no instante inicial. Temos ainda que, como na Figura 4, os painéis (c) e (d) também representam situações de colisão da sonda com o planeta.

Observa-se que a dinâmica torna-se mais complexa devido ao movimento mútuo dos corpos em suas órbitas elípticas; a trajetória relativa deixa de ser puramente estática, como no problema de dois corpos. Ainda assim, ao analisarmos a trajetória da sonda no referencial de Marte, percebe-se que sua curva se aproxima bastante de uma hipérbole – exatamente o comportamento esperado para uma manobra de *fly-by*.

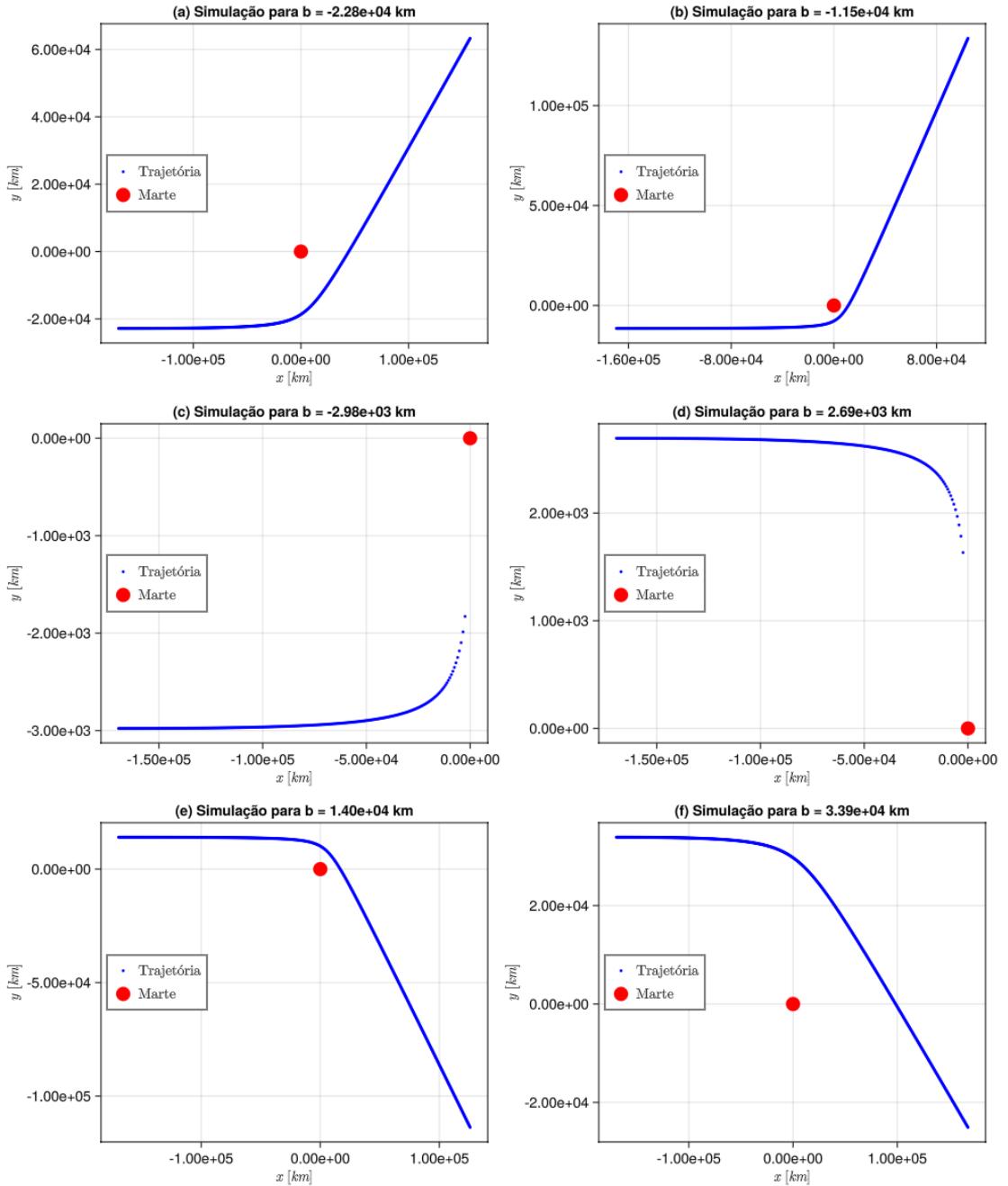


Figura 4: Exemplos de trajetórias da sonda (em azul) sob influência gravitacional de Marte, cujo centro de massa esta indicado pelo ponto vermelho. O planeta não está representado em escala; o marcador vermelho apenas serve como referência de posição. Cada painel mostra a curva descrita pela sonda para diferentes valores de parâmetro de impacto.

Para facilitar a comparação entre os dois modelos, analisaremos agora os dados globais obtidos, buscando identificar em quais situações o modelo de dois corpos é adequado e onde ele deixa de representar bem o fenômeno quando comparamos com o modelo de três corpos para as condições iniciais definidas. quais aspectos físicos do *fly-by* podem ser ensinados de forma confiável utilizando o modelo simplificado de dois corpos no contexto de simulação em física.

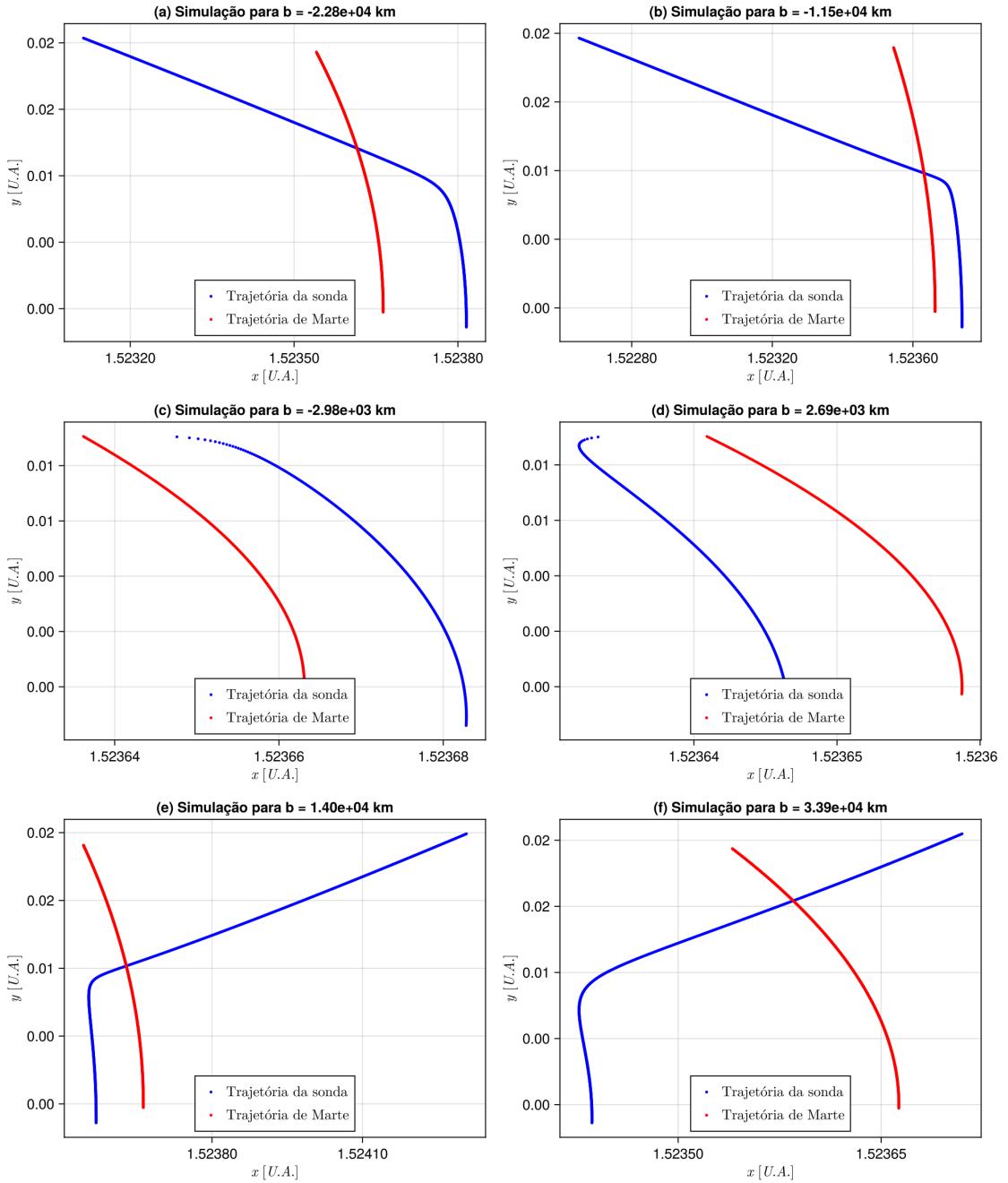


Figura 5: Exemplos de trajetórias de sonda (em azul) e de Marte (em vermelho) no modelo restrito de três corpos. As curvas representam o movimento dos centros de massa de cada corpo ao longo da interação. Cada painel mostra a deformação da trajetória da sonda para diferentes valores do parâmetro de impacto. O gráfico está em unidades astronômicas, sendo $1 \text{ U.A.} \simeq 1,496 \times 10^{11} \text{ m}$.

A Figura 6 mostra a distância mínima registrada entre a sonda e Marte para cada valor de parâmetro de impacto testado. A primeira coisa que chama atenção é o excelente acordo entre os valores simulados e a curva teórica prevista pela Eq. (23) para as trajetórias sem colisão. Esse comportamento aparece de forma clara tanto no modelo de dois

corpos quanto no de três corpos, indicando que a geometria básica do *fly-by* é bem capturada pelos dois modelos nessas regiões.

Para valores grandes de b , observa-se uma pequena divergência entre a curva teórica e os resultados numéricos. Isso ocorre porque a expressão analítica para r_{min} assume uma velocidade assintótica exatamente igual a $v_\infty = 2,6 \text{ km/s}$. Como a simulação considera uma “esfera de influência” finita de $50R_M$, a velocidade inicial não é perfeitamente assintótica, e esse efeito se torna mais perceptível quando a sonda passa mais distante do planeta. Uma esfera de influência maior reduziria essa diferença.

Outro ponto interessante é o intervalo de valores de b que leva à colisão com Marte. Ambos os modelos apresentam faixas muito semelhantes de colisão, como esperado, já que as condições iniciais foram construídas justamente para manter essa correspondência. Nessas regiões, as distâncias mínimas registradas não coincidem com os valores teóricos porque a simulação é interrompida assim que a colisão é detectada.

Por fim, nota-se um leve “serrilhado” nos pontos próximos à colisão. Esse efeito é consequência do método de Euler: quando a sonda se aproxima muito do planeta, o campo gravitacional fica mais forte e o passo de tempo fixo faz com que a integração dê “saltos” maiores. Esse comportamento já havia aparecido visualmente nas trajetórias mostradas nos painéis (c) e (d) das Figuras 4 e 5.

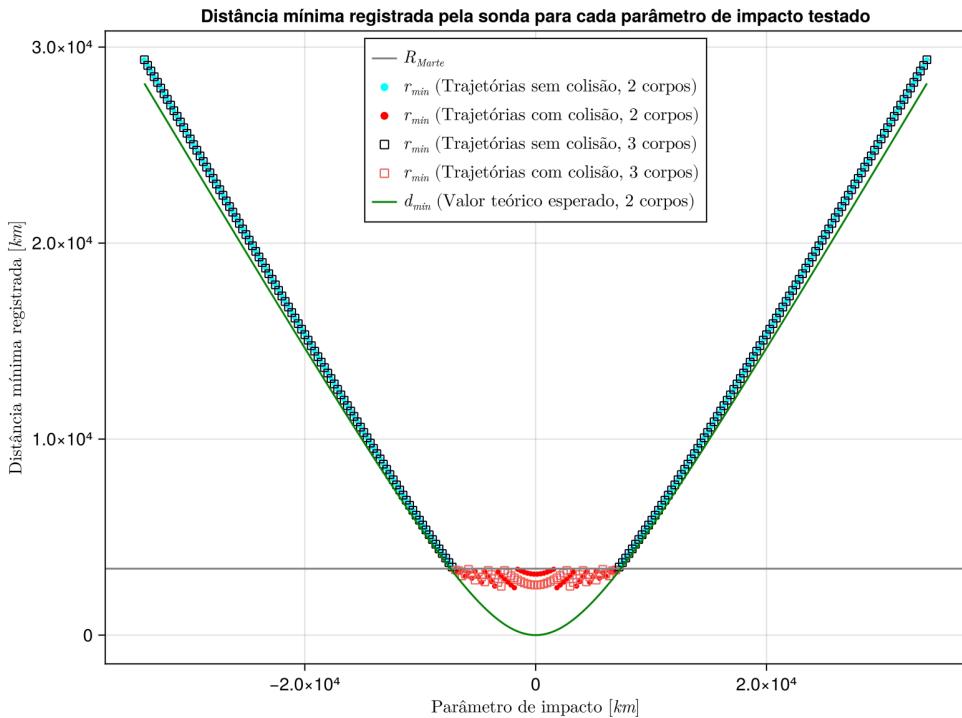


Figura 6: Distância mínima entre a sonda e Marte em função do parâmetro de impacto. A figura mostra os valores obtidos numericamente tanto para o modelo restrito de dois corpos quanto para o de três corpos, destacando casos com e sem colisão. A curva teórica esperada, dada pela Eq. (23), também é apresentada para referência, assim como a linha que representa o raio do planeta.

Partindo para a análise do ângulo de deflexão, observamos na Figura 7 que os valores obtidos nas simulações dos dois modelos concordam entre si e acompanham muito bem a curva teórica prevista pela Eq. (5). Esse resultado reforça que, fora das regiões de colisão, a dinâmica relativa entre sonda e planeta é capturada de forma bastante consistente pelos dois modelos.

Nas situações em que ocorre colisão, porém, ambas as simulações deixam de seguir a tendência teórica e “colapsam” para valores menores de deflexão. Isso acontece porque a integração é interrompida no momento em que a

colisão é identificada. Como o ângulo de deflexão depende diretamente da comparação entre as direções das velocidades relativas de entrada e de saída, a ausência dessa fase final – onde a trajetória deveria curvar-se de acordo com a teoria – impede que o valor correto seja registrado. O comportamento observado nesses pontos, portanto, é uma consequência direta do critério de parada adotado.

Mesmo assim é interessante notar que o modelo de dois corpos e o modelo de três corpos apresentam praticamente o mesmo padrão para δ , inclusive nas proximidades da região crítica. Isso indica que a influência solar, para as condições iniciais definidas, não altera de forma significativa o resultado da medida do ângulo de deflexão.

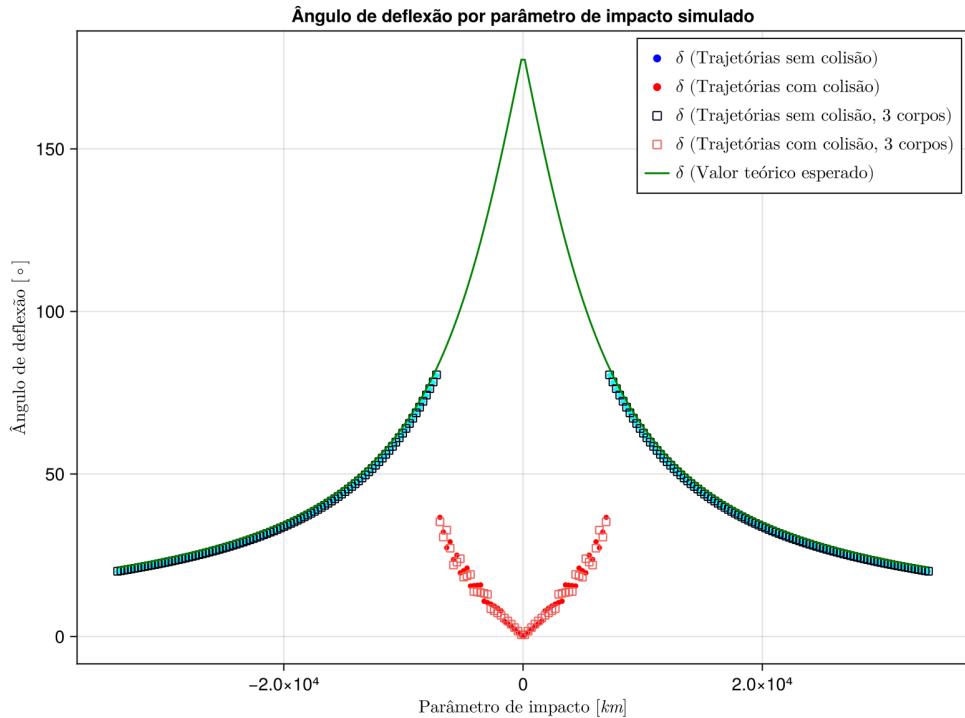


Figura 7: Ângulo de deflexão da trajetória da sonda em função do parâmetro de impacto. O ângulo é calculado a partir da direção de entrada ($\vec{v}_{in, rel}$) e da direção de saída ($\vec{v}_{out, rel}$) no referencial de Marte. A figura apresenta os valores simulados para os modelos restritos de dois e três corpos, destacando casos com e sem colisão, juntamente com o valor teórico esperado obtido pela expressão analítica do problema de dois corpos.

Seguimos agora para as medidas em que o modelo de dois corpos entra em conflito com a realidade física: a variação da velocidade. No modelo restrito de dois corpos, a sonda descreve uma órbita hiperbólica em um sistema efetivamente “fechado”, de modo que a energia mecânica total é conservada. Isso implica que não deve haver mudança no módulo da velocidade relativa no infinito, isto é, $\Delta v = 0$ segundo a Eq. (24). Já no modelo de três corpos essa condição não se aplica: a sonda se move inicialmente em uma órbita elíptica em torno do Sol, e a perturbação gravitacional do planeta quebra a conservação da energia cinética heliocêntrica, permitindo $\Delta v \neq 0$.

A Figura 8 ilustra esse comportamento. O painel (a) apresenta a variação do módulo da velocidade relativa entre entrada e saída, expressa como erro percentual em relação ao valor inicial – $|\vec{v}_\infty|$ para o problema de dois corpos e $|\vec{v}_s(0)|$ para o de três corpos. O painel (b) mostra o tempo total de integração até o acionamento de algum critério de parada.

No painel (a), vemos que o erro no modelo de dois corpos é extremamente baixo (menos que 0,1%) e tende a zero conforme diminuímos o parâmetro de impacto. Isso ocorre porque trajetórias com b menor resultam em tempos de integração menores, reduzindo o erro acumulado pelo método de Euler; assim, o modelo acaba atendendo bem a condição teórica de $\Delta v = 0$.

Para o modelo de três corpos, representado pelos quadrados pretos, o comportamento é diferente. O erro não diminui consideravelmente com o tempo de integração e permanece em torno de $\sim 0,2\%$, exibindo ainda um serrilhado semelhante ao observado nas situações de colisão anteriormente. Esse resultado era esperado: como o sistema heliocêntrico não conserva energia cinética, o módulo da velocidade relativa naturalmente pode variar, produzindo o deslocamento observado na figura. Essa variação é pequena porque as condições iniciais foram escolhidas justamente para que a manobra no modelo de três corpos se assemelhasse ao máximo ao caso de dois corpos, como proposto no trabalho. Além disso, como as velocidades iniciais e finais utilizadas no cálculo não são verdadeiramente assintóticas no modelo de três corpos, isso também contribui para o erro registrado. Desta forma, fica perceptível que a velocidade da sonda relativa ao planeta praticamente não muda de maneira significativa em módulo durante o *fly-by*, o que era esperado devido a órbita hiperbólica da sonda neste referencial.

Por fim, vale destacar que o serrilhado observado nos dados do modelo de três corpos é consequência do método de Euler: ao lidar com regiões em que o campo gravitacional varia rapidamente, o passo de tempo fixo introduz instabilidade numérica que aparecem como oscilações nos valores de velocidade.

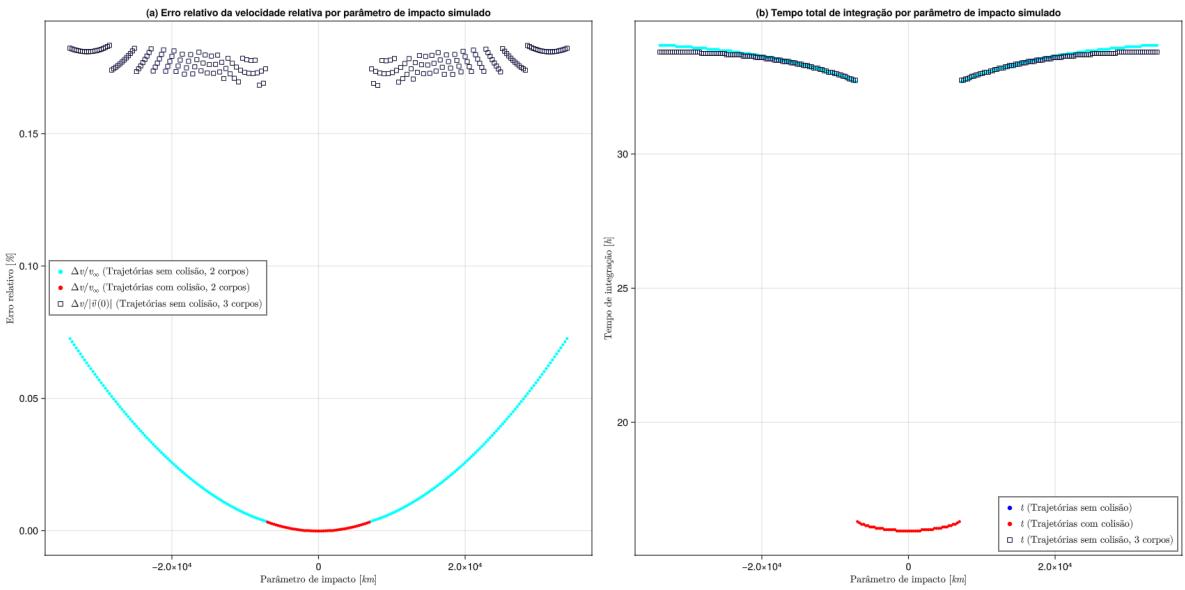


Figura 8: No painel (a) é apresentado o erro relativo no módulo da velocidade da sonda, definido como a variação de sua velocidade dividida pela velocidade inicial. No painel (b) mostramos o tempo total necessário para a integração dos modelos ao longo de cada trajetória. Os casos com colisão no modelo de três corpos foram omitidos, pois apresentaram erros superiores a 100%, decorrentes da interrupção prematura da simulação e do uso do método de Euler.

Por fim, como a velocidade da sonda não é conservada no *fly-by* descrito pelo problema restrito de três corpos, analisamos a variação de sua velocidade heliocêntrica. A Figura 9 confirma esse comportamento: para todas as trajetórias $\Delta v \neq 0$, evidenciando a influência do campo gravitacional solar durante a passagem.

Os pontos correspondentes às simulações com colisão aparecem deslocados, o que era esperado, já que essas integrações foram interrompidas antes que a trajetória pudesse se estabelecer por completo. Para as trajetórias sem colisão, no entanto, observamos uma curva bem definida: o módulo da variação da velocidade aumenta à medida que o parâmetro de impacto diminui. Isso ocorre porque impactos menores correspondem a aproximações mais intensas entre sonda e planeta, fazendo com que a interação gravitacional de Marte perturbe de forma mais significativa a órbita elíptica heliocêntrica da sonda. Essa perturbação altera a relação entre energia cinética e momento angular no referencial do Sol, resultando em uma variação maior de velocidade.

Da mesma forma, trajetórias com grandes valores de b passam longe do planeta e experimentam uma perturbação mais fraca. Nesses casos, a velocidade heliocêntrica praticamente não muda – o efeito dominante é apenas a mudança de direção, já discutida na análise do ângulo de deflexão.

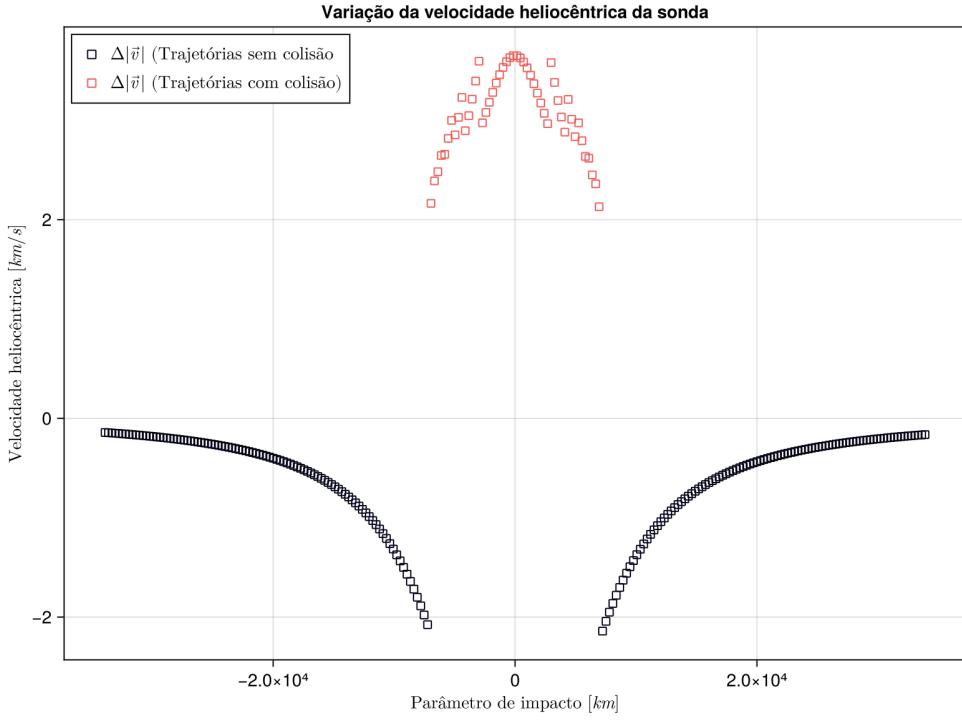


Figura 9: Variação da velocidade heliocêntrica da sonda em função do parâmetro de impacto para o modelo restrito de três corpos. Os quadrados pretos representam trajetórias sem colisão, enquanto os quadrados vermelhos correspondem aos casos em que a sonda colide com Marte.

4 Conclusão

Podemos concluir que a simulação de *fly-by* baseada no modelo restrito de dois corpos, embora não reproduza toda a complexidade física do modelo de três corpos, ainda é capaz de fornecer resultados bastante satisfatórios para a distância mínima e para o ângulo de deflexão. Além disso, quando utilizamos condições iniciais compatíveis entre os dois modelos, a diferença do erro relativa na velocidade se mantém pequeno – da ordem de $\sim 0,15\%$ – reforçando que o comportamento global das trajetórias permanece semelhante.

Dessa forma, a menos que seja necessário analisar explicitamente a variação da velocidade heliocêntrica, ou as trajetórias em si, o modelo de dois corpos se mostra perfeitamente viável para o ensino de simulação em física. Ele é simples, possui solução geométrica analítica para comparação e permite explorar conceitos centrais do *fly-by*, como parâmetro de impacto, distância mínima e ângulo de deflexão, sem comprometer de maneira significativa a fidelidade física. Além disso, o modelo facilita a visualização e discussão de erros numéricos, já que a violação da conservação da velocidade relativa aparece de modo controlado e interpretável.

Por outro lado, o modelo de três corpos se mostrou muito mais sensível ao método de integração de Euler. A presença do Sol introduz instabilidades que, combinadas com o aumento artificial de energia típico do método, produzem o serrilhado observado em diferentes resultados. Em algumas situações, torna-se difícil distinguir se a variação na velocidade heliocêntrica é de origem física ou numérica, como destacado na análise da Figura 8. Isso limita a utilidade didática do modelo quando associado a um integrador básico.

Em vista disso, o problema restrito de três corpos não parece ideal para introdução ao ensino de simulação numérica, dada sua complexidade intrínseca e sua sensibilidade a integradores simples. Uma abordagem mais adequada seria utilizá-lo apenas em disciplinas avançadas, associando-o a métodos mais robustos – como os integradores de Verlet ou, de preferência, os de Runge-Kutta – enquanto o modelo de dois corpos cumpre muito bem o papel de introduzir os principais conceitos associados ao *fly-by* de forma estável e pedagogicamente acessível.

Referências

- [1] P Berg. «The fundamental concepts of the gravity-assist manoeuvre». Em: *European Journal of Physics* 44.2 (2023), p. 025002. DOI: 10.1088/1361-6404/acaad8.
- [2] NASA. *Lista de missões da NASA*. Acessado em 2025. URL: <https://www.nasa.gov/a-to-z-of-nasa-missions/>.
- [3] Jerry B. Marion e Stephen T. Thornton. *Classical Dynamics of Particles & Systems*. 5^a ed. Belmont, CA: Brooks/Cole, 2004. ISBN: 9780534408961.
- [4] Howard D. Curtis. *Orbital Mechanics for Engineering Students*. 3^a ed. Oxford: Butterworth-Heinemann, 2014.
- [5] Dietmar William Foryta. *Apostila de simulação computacional*. Acessado durante o curso, em 2025. URL: <https://fisica.ufpr.br/foryta/Manual-CF1829.pdf>.

5 Anexos

5.1 Códigos de simulação

Neste anexo apresentamos os trechos de código responsáveis pelas rotinas centrais das simulações numéricas. Vale destacar que eles não representam o projeto completo, mas apenas as partes essenciais para a integração das equações de movimento. O conjunto completo de arquivos, junto com instruções de uso e dependências, está disponível em: <https://github.com/gabriel-ferr/Fly-by>. Os comentários foram parcialmente removidos do código para reduzir o volume de texto.

5.1.1 Problema restrito de dois corpos

```
1 // const int test                         - Numero de identificacao do teste.
2 // const double b                          - Parametro de impacto.
3 // double* d_min_value                   - Referencia: altera a distancia minima
4 // encontrada entre a sonda e Marte.
5 // double* delta_v_value                 - Referencia: altera a variacao de velocidade
6 // entre a sonda e Marte na entrada e saida.
7 // double* deflection_angle_value       - Referencia: altera o angulo de deflexao
8 // encontrado.
9 // int* collision                       - Referencia: altera o indicador de colisao.
10 // double* time_end                     - Referencia: altera o tempo em segundo que
11 // levou para finalizar a simulacao.
12 void simulate(const int test, const double b, double* d_min_value, double* delta_v_value,
13               double* deflection_angle_value, int* collision, double* time_end) {
14     // - Declaracao das variaveis locais.
15     int f;
16
17     // Variaveis de estado.
18     double r[N_DIMS + 1];
19     double v[N_DIMS + 1];
20
21     // Variaveis de estado temp.
22     double r_temp[N_DIMS + 1];
23     double v_temp[N_DIMS + 1];
24
25     // Vetores de velocidade de entrada e saida.
26     double velocity_in[N_DIMS + 1];
27     double velocity_out[N_DIMS + 1];
28     double velocity_out_direction[N_DIMS + 1];
29
30     double div;
31     double time;
32     double distance;
33     char filename[200];
34     FILE *fo;
35
36     // - Cond.iciais, aplicadas...
37     r[1] = x_init;
38     r[2] = b;
39     v[1] = v_x_init;
40     v[2] = 0.0;
41
42     velocity_in[1] = v_infinite_in;
43     velocity_in[2] = 0.0;
44
45     distance = sqrt(r[1] * r[1] + r[2] * r[2]);
46
47     *collision = 0;
48     *d_min_value = distance;
49
50     // - Prepara para salvar os dados.
```

```

46     f = 0;
47     sprintf(filename, "%s/pr2c/data_%03d.csv", test_name, test + 1);
48     fo = fopen(filename, "w");
49
50     fprintf(fo, "t,x,y,v_x,v_y,d\n");
51
52     // - Processo de simul.
53     for (time = 0; time < max_int_time; time += dt) { // NOLINT(*-flp30-c)
54
55         // Adiciona os dados ao arquivo de saida.
56         if (f <= 0) {
57             f = steps_to_output;
58             fprintf(fo, "%.8e,%.12e,%.12e,%.12e,%.12e,%.12e\n",
59                     time, r[1], r[2], v[1], v[2], distance);
60
61             // Parada.
62             // 1. Verifica se a sonda colidiu com Marte.
63             if (distance < RAIO_MARTE) {
64                 *d_min_value = distance;
65                 *collision = 1;
66                 break;
67             }
68
69             // 2. Verifica se a sonda esta suficientemente longe de Marte.
70             // O tempo aqui e apenas para impedir que ele pare no começo da simul.
71             if (distance >= stop_value && time > 10 * STEPS_PARA_OUTPUT) {
72                 break;
73             }
74         }
75
76         f--;
77         // Integracao, conforme Eqs (14-17).
78         div = r[1] * r[1] + r[2] * r[2];
79         r_temp[1] = r[1] + v[1] * dt;
80         r_temp[2] = r[2] + v[2] * dt;
81         v_temp[1] = v[1] - (CONSTANTE_GRAVITACIONAL * MASSA_MARTE * r[1] / (div * sqrt(div)))
82             * dt;
83         v_temp[2] = v[2] - (CONSTANTE_GRAVITACIONAL * MASSA_MARTE * r[2] / (div * sqrt(div)))
84             * dt;
85
86         // Atualiza os estados.
87         r[1] = r_temp[1];
88         r[2] = r_temp[2];
89         v[1] = v_temp[1];
90         v[2] = v_temp[2];
91
92         // Calcula a dist. entre Marte e a sonda.
93         distance = sqrt(div);
94         if (distance < *d_min_value) *d_min_value = distance;
95
96         // Fecha o arquivo de dados.
97         fclose(fo);
98
99         // Calcula o angulo de deflexao e a variação da velocidade relativa.
100        div = sqrt(v[1] * v[1] + v[2] * v[2]);
101        velocity_out_direction[1] = v[1] / div;
102        velocity_out_direction[2] = v[2] / div;
103
103        div = sqrt(div * div - 2 * CONSTANTE_GRAVITACIONAL * MASSA_MARTE / sqrt(r[1] * r[1] + r
104            [2] * r[2]));
105
105        velocity_out[1] = div * velocity_out_direction[1];
106        velocity_out[2] = div * velocity_out_direction[2];
107

```

```

108 *delta_v_value = sqrt(velocity_out[1] * velocity_out[1] + velocity_out[2] * velocity_out
109 [2]) - sqrt(velocity_in[1] * velocity_in[1] + velocity_in[2] * velocity_in[2]);
110
111 *deflection_angle_value = (velocity_in[1] * velocity_out[1] + velocity_in[2] *
112 velocity_out[2]) /
113 (sqrt(velocity_in[1] * velocity_in[1] + velocity_in[2] * velocity_in[2]) *
114 sqrt(velocity_out[1] * velocity_out[1] + velocity_out[2] * velocity_out[2]));
115
116 if (*deflection_angle_value > 1) *deflection_angle_value = 1.0;
117 if (*deflection_angle_value < -1) *deflection_angle_value = -1.0;
118
119 *deflection_angle_value = acos(*deflection_angle_value);
120 *time_end = time;
121 }
```

Listing 1: Rotina principal do integrador para o problema de dois corpos.

5.1.2 Problema restrito de três corpos

```

1 // const int test                                - N. de identificacao do teste.
2 // const double b                               - Parametro de impacto.
3 // double* d_min_value                         - Referencia: altera a dist. min. encontrada
4 // entre a sonda e Marte.
5 // double* delta_v_value                       - Referencia: altera a var. de velocidade helio.
6 // da sonda.
7 // double* delta_v_value_rel                  - Referencia: altera a var. de velocidade
8 // relativa da sonda.
9 // double* deflection_angle_value           - Referencia: altera o angulo de deflexao
10 // encontrado.
11 // int* collision                            - Referencia: altera o indicador de colisao.
12 // double* time_end                           - Referencia: altera o tempo em segundo que
13 // levou para finalizar a simul.
14 void simulate(int test, double b, double* d_min_value, double* delta_v_value, double*
15 delta_v_value_rel, double* deflection_angle_value, int* collision, double* time_end) {
16
17 //          - Vars. local.
18 int f;
19
20 //      Estado (coordenadas polares)
21 double mars_coord_polar[N_DIMS + 1];
22 double mars_velocity_polar[N_DIMS + 1];
23 double ship_coord_polar[N_DIMS + 1];
24 double ship_velocity_polar[N_DIMS + 1];
25
26 //      Estado (coordenadas cartesianas)
27 double mars_coord_cartesian[N_DIMS + 1];
28 double mars_velocity_cartesian[N_DIMS + 1];
29 double ship_coord_cartesian[N_DIMS + 1];
30 double ship_velocity_cartesian[N_DIMS + 1];
31
32 //      Estado (temp)
33 double mars_coord_polar_updated[N_DIMS + 1];
34 double ship_coord_polar_updated[N_DIMS + 1];
35 double ship_velocity_polar_updated[N_DIMS + 1];
36
37 //      Vetores de velocidade.
38 double velocity_in[N_DIMS + 1];
39 double velocity_out[N_DIMS + 1];
40 double velocity_in_rel[N_DIMS + 1];
41 double velocity_out_rel[N_DIMS + 1];
42 double velocity_out_direction[N_DIMS + 1];
43
44 double div;

```

```

39
40     double time;
41     double distance;
42     char filename[200];
43     FILE *fo;
44
45     // - Cond. iniciais...
46     mars_coord_polar[1] = DISTANCIA_MARTE_SOL;
47     mars_coord_polar[2] = mars_angle_init;
48     mars_velocity_polar[1] = 0.0;
49     mars_velocity_polar[2] = sqrt(CONSTANTE_GRAVITACIONAL * MASSA_SOL / (DISTANCIA_MARTE_SOL
50         * DISTANCIA_MARTE_SOL * DISTANCIA_MARTE_SOL));
51
52     // 2. Converte as coordenadas de marte para os valores cartesianos.
53     mars_coord_cartesian[1] = mars_coord_polar[1] * cos(mars_coord_polar[2]);
54     mars_coord_cartesian[2] = mars_coord_polar[1] * sin(mars_coord_polar[2]);
55     mars_velocity_cartesian[1] = - mars_coord_polar[1] * mars_velocity_polar[2] * sin(
56         mars_coord_polar[2]);
57     mars_velocity_cartesian[2] = mars_coord_polar[1] * mars_velocity_polar[2] * cos(
58         mars_coord_polar[2]);
59
60     // 3. Calcula a pos. cartesiana inicial da sonda.
61     ship_coord_cartesian[1] = mars_coord_cartesian[1] + sqrt(r_factor * r_factor - b * b) *
62         sin(mars_coord_polar[2]) - b * cos(mars_coord_polar[2]);
63     ship_coord_cartesian[2] = mars_coord_cartesian[2] - sqrt(r_factor * r_factor - b * b) *
64         cos(mars_coord_polar[2]) - b * sin(mars_coord_polar[2]);
65
66     // 4. Converte os valores de pos. para coordendas polares.
67     ship_coord_polar[1] = sqrt(ship_coord_cartesian[1] * ship_coord_cartesian[1] +
68         ship_coord_cartesian[2] * ship_coord_cartesian[2]);
69     ship_coord_polar[2] = atan2(ship_coord_cartesian[2], ship_coord_cartesian[1]);
70
71     // 5. Calcula a velocidade relativa de entrada; e aproveita para calcular o valor da
72     // velocidade helo..
73     velocity_in_rel[1] = - v_sonda_init * sin(mars_coord_polar[2]);
74     velocity_in_rel[2] = v_sonda_init * cos(mars_coord_polar[2]);
75
76     ship_velocity_cartesian[1] = velocity_in_rel[1] + mars_velocity_cartesian[1];
77     ship_velocity_cartesian[2] = velocity_in_rel[2] + mars_velocity_cartesian[2];
78
79     velocity_in[1] = ship_velocity_cartesian[1];
80     velocity_in[2] = ship_velocity_cartesian[2];
81
82     // 6. Converte as velocidades cartesianas para polar.
83     ship_velocity_polar[1] = (ship_coord_cartesian[1] * ship_velocity_cartesian[1] +
84         ship_coord_cartesian[2] * ship_velocity_cartesian[2]) / ship_coord_polar[1];
85     ship_velocity_polar[2] = (ship_coord_cartesian[1] * ship_velocity_cartesian[2] -
86         ship_coord_cartesian[1] * ship_velocity_cartesian[1]) / (ship_coord_polar[1] *
87         ship_coord_polar[1]);
88
89     // 7. Dist. entre a sonda e Marte.
90     distance = sqrt((ship_coord_cartesian[1] - mars_coord_cartesian[1]) * (
91         ship_coord_cartesian[1] - mars_coord_cartesian[1]) + (ship_coord_cartesian[2] -
92         mars_coord_cartesian[2]) * (ship_coord_cartesian[2] - mars_coord_cartesian[2]));
93
94     *collision = 0;
95     *d_min_value = distance;
96     // Prepara para salvar os dados.
97     f = 0;
98     sprintf(filename, "%s/pr3c/data_%03d.csv", test_name, test + 1);
99     fo = fopen(filename, "w");
100
101    fprintf(fo, "t,x_mars,y_mars,x_ship,y_ship,v_x_mars,v_y_mars,v_x_ship,v_y_ship,d\n");
102
103    // - Processo de simul.
104    for (time = 0; time < max_int_time; time += dt) { // NOLINT(*-f1p30-c)

```

```

92         // Adiciona os dados ao arquivo de saida.
93     if (f <= 0) {
94         f = steps_to_output;
95         fprintf(fo, "% .8e,%.15e,%.15e,%.15e,%.15e,%.15e,%.15e,%.15e,%.15e\n",
96                 time, mars_coord_cartesian[1], mars_coord_cartesian[2], ship_coord_cartesian
97                         [1], ship_coord_cartesian[2], mars_velocity_cartesian[1],
98                         mars_velocity_cartesian[2], ship_velocity_cartesian[1],
99                         ship_velocity_cartesian[2], distance);
100
101        // Parada.
102        // 1. Verifica se a sonda colidiu com Marte.
103        if (distance < RAIO_MARTE) {
104            *d_min_value = distance;
105            *collision = 1;
106            break;
107        }
108
109        // 2. Verifica se a sonda esta suficientemente longe de Marte.
110        // O tempo aqui e apenas para impedir que ele pare no começo da simulação.
111        if (distance >= stop_value && time > 10 * STEPS_PARA_OUTPUT) {
112            break;
113        }
114    }
115
116    f--;
117
118    // Integracao, conforme Eqs (34-38).
119    div = ship_coord_polar[1] * ship_coord_polar[1] + mars_coord_polar[1] *
120          mars_coord_polar[1] - 2 * ship_coord_polar[1] * mars_coord_polar[1] * cos(
121              ship_coord_polar[2] - mars_coord_polar[2]);
122
123    mars_coord_polar_updated[2] = mars_coord_polar[2] + mars_velocity_polar[2] * dt;
124    ship_coord_polar_updated[1] = ship_coord_polar[1] + ship_velocity_polar[1] * dt;
125    ship_coord_polar_updated[2] = ship_coord_polar[2] + ship_velocity_polar[2] * dt;
126    ship_velocity_polar_updated[1] = ship_velocity_polar[1] + (ship_coord_polar[1] *
127                  ship_velocity_polar[2] * ship_velocity_polar[2] -
128                  CONSTANTE_GRAVITACIONAL * MASSA_SOL / (ship_coord_polar[1] * ship_coord_polar
129                      [1]) -
130                  CONSTANTE_GRAVITACIONAL * MASSA_MARTE * (ship_coord_polar[1] - mars_coord_polar
131                      [1] * cos(ship_coord_polar[2] - mars_coord_polar[2])) /
132                  (div * sqrt(div))) * dt;
133    ship_velocity_polar_updated[2] = ship_velocity_polar[2] +
134        (- CONSTANTE_GRAVITACIONAL * MASSA_MARTE * mars_coord_polar[1] * sin(
135            ship_coord_polar[2] - mars_coord_polar[2]) /
136            (ship_coord_polar[1] * div * sqrt(div)) - 2 * ship_velocity_polar[1] *
137            ship_velocity_polar[2] / ship_coord_polar[1]) * dt;
138
139    // Atualiza os estados
140    mars_coord_polar[2] = mars_coord_polar_updated[2];
141    ship_coord_polar[1] = ship_coord_polar_updated[1];
142    ship_coord_polar[2] = ship_coord_polar_updated[2];
143    ship_velocity_polar[1] = ship_velocity_polar_updated[1];
144    ship_velocity_polar[2] = ship_velocity_polar_updated[2];
145
146    mars_coord_cartesian[1] = mars_coord_polar[1] * cos(mars_coord_polar[2]);
147    mars_coord_cartesian[2] = mars_coord_polar[1] * sin(mars_coord_polar[2]);
148
149    ship_coord_cartesian[1] = ship_coord_polar[1] * cos(ship_coord_polar[2]);
150    ship_coord_cartesian[2] = ship_coord_polar[1] * sin(ship_coord_polar[2]);
151
152    mars_velocity_cartesian[1] = mars_velocity_polar[1] * cos(mars_coord_polar[2]) -
153        mars_coord_polar[1] * mars_velocity_polar[2] * sin(mars_coord_polar[2]);
154    mars_velocity_cartesian[2] = mars_velocity_polar[1] * sin(mars_coord_polar[2]) +
155        mars_coord_polar[1] * mars_velocity_polar[2] * cos(mars_coord_polar[2]);

```

```

145     ship_velocity_cartesian[1] = ship_velocity_polar[1] * cos(ship_coord_polar[2]) -
146         ship_coord_polar[1] * ship_velocity_polar[2] * sin(ship_coord_polar[2]);
147     ship_velocity_cartesian[2] = ship_velocity_polar[1] * sin(ship_coord_polar[2]) +
148         ship_coord_polar[1] * ship_velocity_polar[2] * cos(ship_coord_polar[2]);
149
150     // Dist. entre Marte e a sonda.
151     distance = sqrt(div);
152     if (distance < *d_min_value) *d_min_value = distance;
153 }
154
155     // Fecha o arquivo de dados.
156     fclose(fo);
157
158     // Angulo de deflexao e a var. da velocidade.
159     velocity_out[1] = ship_velocity_cartesian[1];
160     velocity_out[2] = ship_velocity_cartesian[2];
161
162     *delta_v_value = sqrt(velocity_out[1] * velocity_out[1] + velocity_out[2] * velocity_out[2]) -
163         sqrt(velocity_in[1] * velocity_in[1] + velocity_in[2] * velocity_in[2]);
164
165     velocity_out_rel[1] = ship_velocity_cartesian[1] - mars_velocity_cartesian[1];
166     velocity_out_rel[2] = ship_velocity_cartesian[2] - mars_velocity_cartesian[2];
167
168     *delta_v_value_rel = sqrt(velocity_out_rel[1] * velocity_out_rel[1] + velocity_out_rel[2] *
169         velocity_out_rel[2]) - sqrt(velocity_in_rel[1] * velocity_in_rel[1] +
170         velocity_in_rel[2] * velocity_in_rel[2]);
171
172     div = sqrt(velocity_out_rel[1] * velocity_out_rel[1] + velocity_out_rel[2] *
173         velocity_out_rel[2]);
174     velocity_out_direction[1] = velocity_out[1] / div;
175     velocity_out_direction[2] = velocity_out[2] / div;
176
177     *deflection_angle_value = (velocity_in_rel[1] * velocity_out_rel[1] + velocity_in_rel[2] *
178         velocity_out_rel[2]) /
179         (sqrt(velocity_in_rel[1] * velocity_in_rel[1] + velocity_in_rel[2] * velocity_in_rel[2]) *
180             sqrt(velocity_out_rel[1] * velocity_out_rel[1] + velocity_out_rel[2] *
181                 velocity_out_rel[2])));
182
183     if (*deflection_angle_value > 1) *deflection_angle_value = 1.0;
184     if (*deflection_angle_value < -1) *deflection_angle_value = -1.0;
185
186     *deflection_angle_value = acos(*deflection_angle_value);
187     *time_end = time;
188 }

```

Listing 2: Rotina principal do integrador para o problema de três corpos.